THE UNIVERSITY OF
SYDNEY

James Underwood
Senior Research Fellow, Australian Centre for Field Robotics

14th June 2018

ERRATA NOTICE
Reliable and Safe Autonomy for Ground Vehicles in Unstructured Environments
Author of Notice: Dr James Patrick Underwood
Date of Notice: 14 June 2018

I modified thesis document from the original version by correcting equation 3.7 on page 37. The equation was edited by adding a “transpose operator”.

The change was originally made in 2009.

Sincerely,

James Underwood

Reliable and Safe Autonomy for Ground Vehicles in Unstructured Environments

James Patrick Underwood

A thesis submitted in fulfillment
of the requirements for the degree of
Doctor of Philosophy

ARC Centre of Excellence in Autonomous Systems
Australian Centre for Field Robotics
School of Aerospace, Mechanical and Mechatronic Engineering
The University of Sydney

December 2008

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the University or other institute of higher learning, except where due acknowledgement has been made in the text.

James Patrick Underwood

December 19, 2008

Abstract

James Patrick Underwood
The University of Sydney

Doctor of Philosophy
December 2008

Reliable and Safe Autonomy for Ground Vehicles in Unstructured Environments

This thesis is concerned with the algorithms and systems that are required to enable safe autonomous operation of an unmanned ground vehicle (UGV) in an unstructured and unknown environment; one in which there is no specific infrastructure to assist the vehicle autonomy and complete *a priori* information is not available.

Under these conditions it is necessary for an autonomous system to perceive the surrounding environment, in order to perform safe and reliable control actions with respect to the context of the vehicle, its task and the world. Specifically, exteroceptive sensors measure physical properties of the world. This information is interpreted to extract a higher level perception, then mapped to provide a consistent spatial context. This map of perceived information forms an integral part of the autonomous UGV (AUGV) control system architecture, therefore any perception or mapping errors reduce the reliability and safety of the system.

Currently, commercially viable autonomous systems achieve the requisite level of reliability and safety by using strong structure within their operational environment. This permits the use of powerful assumptions about the world, which greatly simplify the perception requirements. For example, in an urban context, things that look approximately like roads are roads. In an indoor environment, vertical structure must be avoided and everything else is traversable. By contrast, when this structure is not available, little can be assumed and the burden on perception is very large. In these cases, reliability and safety must currently be provided by a tightly integrated human supervisor.

The major contribution of this thesis is to provide a holistic approach to identify and mitigate the primary sources of error in typical AUGV sensor feedback systems (comprising perception and mapping), to promote reliability and safety. This includes an analysis of the geometric and temporal errors that occur in the coordinate transformations that are required for mapping and methods to minimise these errors in real systems. Interpretive errors are also studied and methods to mitigate them are presented. These methods combine information theoretic measures with multiple sensor modalities, to improve perceptive classification and provide sensor redundancy.

The work in this thesis is implemented and tested on a real AUGV system, but the methods do not rely on any particular aspects of this vehicle. They are all generally and widely applicable. This thesis provides a firm base at a low level, from which continued research in autonomous reliability and safety at ever higher levels can be performed.

Acknowledgements

I would like to express my sincerest gratitude to my family, friends and colleagues who have made this thesis possible. Firstly, I wish to thank my supervisor, Steve Scheding for giving me the perfect balance of freedom and guidance that has enabled this endeavour to be both worthwhile and fun. I have learnt so much. Thank you for teaching me how to write, by a constant process of feedback (high bandwidth due to the 'frankness' and with an appropriate block diagram of course). Thanks also for creating such a tight group of people and instilling in us a culture to strive for rigorous engineering.

To Hugh Durrant-Whyte, thank you for creating the opportunity to work in an environment full of enthusiastic, talented and friendly people. The ACFR is a truly remarkable phenomenon.

To Tim Bailey, for knowing the answers to painfully specific questions and sometimes for knowing which questions I ought to be asking instead. And thanks for the debates.

To Ross Hennessey for teaching me how to program for the third time and then continuing to learn with me afterwards. Working with you has been immensely educational, productive and enjoyable.

Thanks to the entire Argo and Q teams (and honorary members), for making things happen: Steve Scheding, Craig Rodgers, Andrew Hill, Tom Allen, Laura Connolly, Ben Soon, Thierry Peynot, Sami Terho, Richard Grover, Ross Hennessey, Florian Oppolzer, Sisir Karumanchi, Mark Calleija, Dai Bang Nguyen, Iain (Buzz) Brown, Bruce Crundwell, Andrew Maclean, Jeremy Randle and the late Al Trinder. Also to Bertrand (Bart) Douillard, Fabio Ramos, Alen Alempijevic, Suresh Kumar, Dave Wood and Paul Thompson, at various times for discussing, encouraging, motivating and collaborating.

Craig, without your tireless efforts and awesome skill, we would all be smashing rocks together. There would be no ground vehicle. We also wouldn't know what happens to a phone when placed in a beer. To Andrew, Tom and Laura, the three with whom I've worked most closely (most often in the back of a land cruiser, on the farm at 4am), thanks for being unfailingly great to work with (and to drink with).

Thanks to my family, for your endless advice, support (cooking, tolerating, proof reading) and love. Thanks also to Susannah for these things and more. Every day is wonderful because of you.

FIRE & ICE

Some say the world will end in fire,
Some say in ice.
From what I've tasted of desire
I hold with those who favor fire.
But if it had to perish twice,
I think I know enough of hate
To say that for destruction ice
Is also great
And would suffice.

Robert Frost

AND STEEL

…Others say that robots will
Rise up and all the humans kill.
From what I've tasted of cold steel,
To all researchers I appeal:
When you build robots, roll no dice.
You'd better bloody make them nice,
Or Robert Frost
Will have to perish thrice.

James Patrick Underwood

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis is concerned with the algorithms and systems required to enable safe autonomous operation of an unmanned ground vehicle (UGV) in an unstructured and unknown environment; one in which there is no specific infrastructure to assist the vehicle autonomy and complete *a priori* information is not available. This is achieved by identifying and addressing the primary sources of error in the system. Autonomous operation is defined in this thesis as the process of navigating from one location to another with no intervention from an operator, other than to specify the goal location that the vehicle must reach. Automation is achieved with a feedback control system that makes use of onboard exteroceptive and proprioceptive sensors to guide the vehicle through the environment to the destination. The principle contribution of this thesis is the improvement of exteroceptive sensor feedback processes for AUGVs by addressing three primary sources of error:

- Systematic errors due to sensor misalignment and synchronisation,
- Stochastic uncertainty due to sensor measurement errors,
- Sensor interpretation errors due to unmodeled environmental conditions.

By holistically addressing the primary sources of error in the exteroceptive sensor feedback processes, this thesis achieves an increase in overall reliability and safety of AUGV systems.

1.1 Background

Any ground based mobile autonomous system that is operating in an unknown area must use exteroceptive sensors to perceive the world, thus forming and maintaining a map of the environment, in order to safely navigate through it. There are many different approaches to perception and mapping and the type of map employed depends on the nature of the vehicle and the environment.

For some classes of environment, structure simplifies the perception and mapping requirements. In indoor spaces for example, perception can be simplified to a binary classification of space as either obstacles or free space, because obstacles are salient vertical structures and everything else is flat. A two dimensional world can also be assumed, which simplifies mapping. Similarly in urban environments, tarmac roads can be assumed to be flat and deviations from this assumption are treated as specific obstacles. Such assumptions are commonly used because they reduce the complexity of many of the components of the autonomous system. Systems that leverage the specific structure of the environment are successful when the true environment conforms, although they can fail completely when nuances in the environment violate the basic structural assumptions. Autonomous reliability and safety is often achieved by employing appropriate assumptions about the available structure.

For outdoor scenarios where these assumptions are less valid than in other environments, a more general model of the interaction between the vehicle and the ground support surface is useful. This is usually done by constructing a three dimensional model of the geometry of the environment using exteroceptive sensors, then transforming this to a traversability space by considering the ability of the vehicle to drive over a particular geometry. Modeling the geometry of the environment alone is often insufficient to accurately determine the likely vehicle interaction, thus higher levels of reasoning are also required. For example, rocks and shrubs may have similar geometry, but the contact between the them and a vehicle will differ substantially. Perception algorithms that interpret sensor data for high level reasoning, are prone to reliability problems as it can be difficult to estimate when they are likely to succeed or fail. Tracking their performance in real-time is a challenging unsolved problem.

When building large maps that cover the entire area of interest, it is rarely possible to sense the whole region from one vantage point. Sensors provide information in a sub-region only, so ego-motion must be used to form a more complete picture. The mobile platform on which the sensors are mounted is effectively used as a scanner and the sensory information is fused over time to form the larger map. To achieve autonomy in unstructured environments, the map of the surrounding environment is used in a control system to allow the vehicle to navigate. The map is therefore part of a control feedback loop, so mapping errors (geometric or from sensor interpretation) translate directly to potential safety breaches in the system, where the severity depends on the particular application.

1.2 Problem Statement

All mapping algorithms use at least one sensor, which interacts actively or passively with the environment. The first potential source of error arises in the *interpretation* of this sensory information. A model must be used to relate the sensor information to the underlying state of the environment and errors or omissions in this model will cause an error in the map. Secondly, the information must be interpreted in a *geometric context*. From precisely where in the world was the measurement made? For a mobile robot, this information is typically provided by the combination of the non-constant vehicle localisation and the constant offset from this location to the position of the sensor. Stochastic measurement errors are always present in the former and constant biases arise from imperfect measurement or calibration of the latter.

Regardless of the choice of sensor and the choice of mapping algorithm, any unaccounted source of error in the sensing and mapping process will result in a difference between the internal map of the environment and the ground truth. If this information is then used for the task of navigation, the system will be unsafe.

In this thesis, a holistic approach is taken to identify and address the primary sources of error in the perception and mapping process for AUGV feedback control systems.

1.3 Principal Contributions

This thesis identifies common sources of error in the exteroceptive sensor perception and mapping process, that is used for AUGV feedback control. The error identification and analysis provide a method for quantifying geometric mapping errors, so that they can be calculated in the system design phase and importantly, so that these errors can be estimated at run-time. This also provides an intuition about the nature of errors that will occur and how they will affect the map. The error identification process serves as a mathematical framework for solutions in this thesis to minimise sources of error and to mitigate against their effect. An emphasis is placed on providing techniques that do not rely on the specific structure of one particular AUGV. Many of the solutions are generally applicable to robotic systems with exteroceptive sensors, including other non ground based systems, although the scope of the thesis permits testing only on an AUGV.

Specifically, this thesis contributes:

- An analysis of the sensitivity of mapped data to the primary sources of measurable error, both *stochastic* in the form of localisation errors and sensor noise and *systematic* in the form of spatial sensor misalignment and sensor timing synchronisation errors.

- An analysis and development of an error model to calculate the combined geometric map uncertainty, given the joint effect of localisation, noise, misalignment, and time synchronisation.

- A calibration method to learn the optimal sensor alignment parameters with a full six degrees of freedom and a method to calculate the precision of the learnt parameters.

- An extension of this method to learn the joint alignment parameters of multiple sensors, so that their information can reliably be fused into one consistent map.
- The rigorous testing of the calibration procedure for a single sensor, multiple homogenous sensors and multiple heterogenous sensors. This demonstrates the minimisation of systematic errors, the minimisation of systematic contradictions between sensors to allow data fusion and the leveraging of a high precision sensor to assist the calibration of a lower precision sensor.
- Examples of calculating and meeting system design requirements that are driven by a desired mapping accuracy.
- An analysis and discussion of multi-modal sensor fusion to reduce interpretive errors in perception algorithms and an analysis of multi-modal hardware redundancy as a model-free method of fault detection for arbitrary exteroceptive sensors.
- A specific multi-modal sensor fusion algorithm that uses a laser and a radar to detect when airborne dust particles are impeding the use of the laser.
- An evaluation of a navigation system architecture for mapping, that is designed to reduce the effect of stochastic global positioning errors.

1.4 Thesis Structure

Chapter 2 presents an overview of typical AUGV control system architectures and describes how exteroceptive feedback is used by them for navigation. Two specific AUGV systems from the literature are discussed with particular reference to how they achieve robustness and safety. The discussion in this chapter provides the overall motivation for the rest of this thesis.

Chapter 3 presents an analysis of the effect of the three main sources of error in exteroceptive feedback systems. Experimental validation of the analysis is provided.

Reliability is promoted by the highly accurate error model that is developed in this chapter, as it provides a tool to estimate mapping uncertainty both at design-time and run-time. This analysis also provides the background and mathematical framework behind the solutions and techniques that are presented in Chapters 4 and 5.

Chapter 4 addresses the geometric and temporal sources of error that were identified in Chapter 3. This is achieved by providing methods to minimise the errors, to detect when such errors will have an effect on the integrity of the system and to manage them. These techniques are validated experimentally on a real AUGV system.

Chapter 5 addresses the interpretive errors that were described at the end of Chapter 3 by providing improved perception and redundancy models. Generally applicable techniques are described, with experimental validation for specific examples on a real AUGV.

Chapter 6 presents conclusions and discusses the future work in this field.

Chapter 2

Background

The purpose of this chapter is to provide general background information about the methods that are used to achieve ground vehicle autonomy. It also provides the motivation for the discussion and analysis of and solutions to common problems that are presented in the subsequent chapters. More detailed background information and references to the literature are presented *in-situ* in Chapters 3, 4 and 5. This is done to focus the discussion of directly related prior work to the relevant sections within this thesis.

Any ground based mobile autonomous system that is operating in an unknown environment must use exteroceptive sensors to form and maintain a map of the surroundings, in order to safely navigate through the environment. This chapter describes the use of exteroceptive sensor information in typical AUGV systems and some specific examples from the literature are discussed. These approaches are all subject to the systematic, stochastic and modeling errors that arise when using exteroceptive sensors on a mobile platform. In discussing these systems, emphasis is placed on how they achieve reliability and safety, in order to motivate the analyses and solutions that are presented in this thesis. The effects of these errors are analysed in Chapter 3, in order to motivate the solutions in Chapters 4 and 5.

2.1 General AUGV Control Systems

This section describes the general form of control system that is used to achieve AUGV autonomy. The basic form of the control system that enables the autonomous following of a trajectory is presented in Section 2.1.1 and in Section 2.1.2 this is extended to describe perception and path planning algorithms as a predictive component of the controller. Although there are many different perception and path planning algorithms in the literature, they can be seen as functionally equivalent in this control theoretic framework. Together, perception and planning systems either implicitly or explicitly make use of a vehicle model to predict future interactions between the vehicle and the environment, in order to choose an optimal control strategy. The perception system transforms the raw sensor information into a context that is relevant to the vehicle model. For example, laser range measurements are converted to the 'roughness' of the terrain. This information is *mapped* to provide a consistent spatial context that allows the association of the vehicle's current or future location with the representation of the environment. The planning system can then apply the vehicle model to that information, to achieve an optimal control strategy.

By considering the different ground vehicle automation methods from a simple control theoretic perspective, the wide variety of techniques can be unified for the purpose of this thesis. The aim is not to describe any one specific control strategy, but rather to argue that perception, mapping and planning systems provide the *same* function in most if not all AUGV systems. This thesis addresses the most common types of errors that occur in general perception and mapping systems, so this should have wide application regardless of the specific algorithms that are chosen.

2.1.1 Feed-Back Control

The basic form of all feedback control systems is shown in Figure 2.1. A controller is designed to minimise the error between the desired output and the *measured* output, by applying an appropriate control input to the plant. Disturbances that are

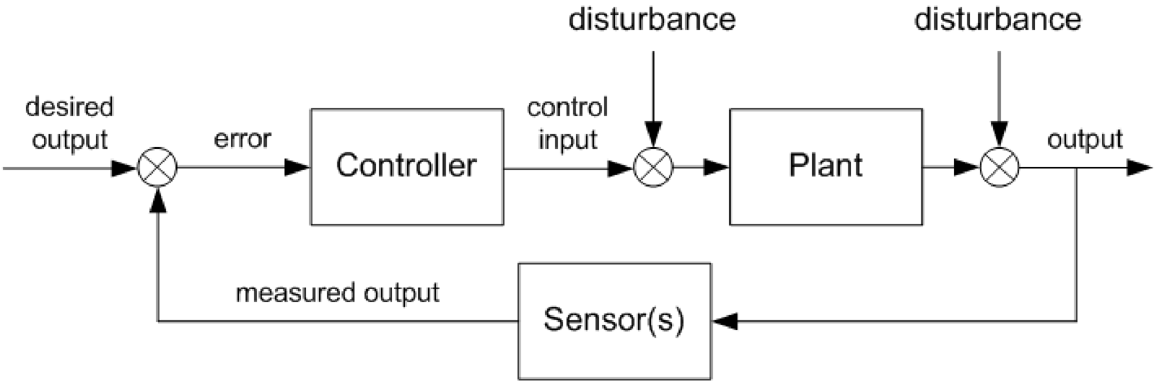


Figure 2.1: The basic feedback control loop [1, p12].

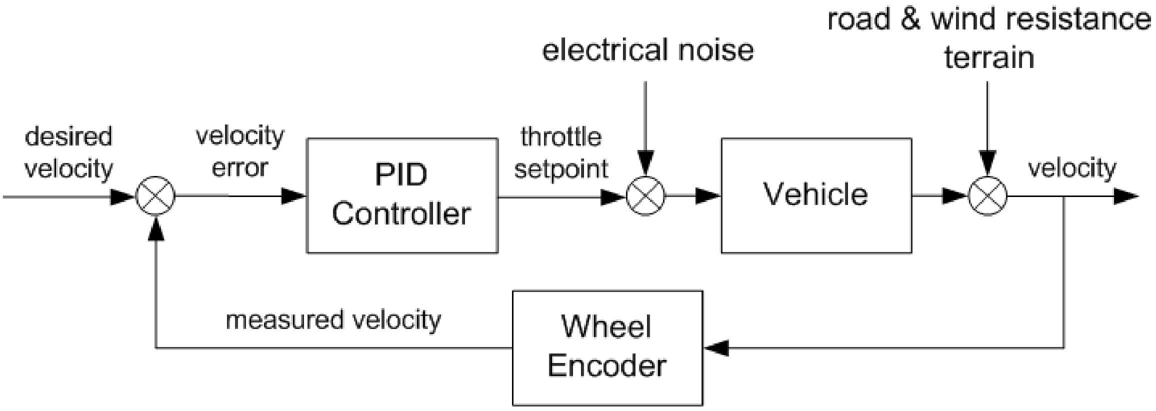


Figure 2.2: A basic feedback control loop applied to the velocity control of a vehicle. In this case, a PID controller has been chosen for the control block and a wheel encoder is selected for sensing. This is just one of many possible combinations that can be used to achieve basic velocity control for a vehicle.

often difficult or impossible to model are present at multiple locations within the control loop. The feedback component allows the controller to compensate for this by measuring the instantaneous error between the desired and current output of the system. A successful control system must meet a variety of constraints, the main ones including stability, bandwidth or responsiveness and the steady state response (how small can the error eventually be made). Another objective is the robustness of the controller to unmodeled variations of the system parameters, which requires the controller to be as insensitive to these as possible [1].

The basic control system shown in Figure 2.1 can be directly applied to some aspects of autonomy for unmanned ground vehicles (UGVs). For example, Figure 2.2 illus-

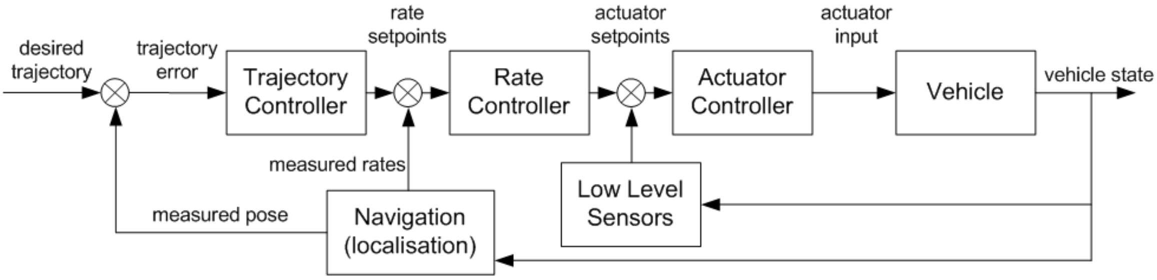


Figure 2.3: An example of a vehicle trajectory controller, using multiple control layers to provide modularity. The trajectory control strategy outputs the desired turn rate and velocity, yet leaves the specific method of realisation to the rate controller. Similarly, the rate controller makes requests to the actuators, but defers the lowest level control to the actuator controllers. Despite the additional complexity, the system is a closed loop feedback control strategy, similar to Figure 2.1. Note that this is a 'blind' trajectory controller; it does not have any perception of the surroundings and no concept of obstacle avoidance.

trates a typical velocity controller of exactly this form. However, to achieve higher levels of autonomy such as the ability to follow a particular trajectory, more complicated control strategies are usually required. The trajectory control system for the Centre for Autonomous Systems (CAS) Outdoor Research Demonstrator (CORD) AUGV is shown in Figure 2.3. The system comprises three cascading control layers made up of separable trajectory, rate and actuation controllers. Although the system could potentially have one controller with trajectory errors as the input and actuation commands as the output, splitting the system into three separable components improves the modularity. Were the controller to be applied to a completely different vehicle, the actuation and rate controllers would need to be modified, but the higher level trajectory control component could likely remain unchanged. Although the system is more complicated than the simple velocity controller in Figure 2.2, by grouping functionally similar components together, it can be considered to be in exactly the same feedback control form as shown in Figure 2.1. This is illustrated by grouping together the control and sensing components, as shown in Figure 2.4.

By designing a control system architecture that employs standard control theoretic methods, many proven guarantees of robustness, bandwidth, stability and steady state response can be applied to the autonomous system. Generic feedback control methods (PID control for example) are a valuable tool for AUGV control systems

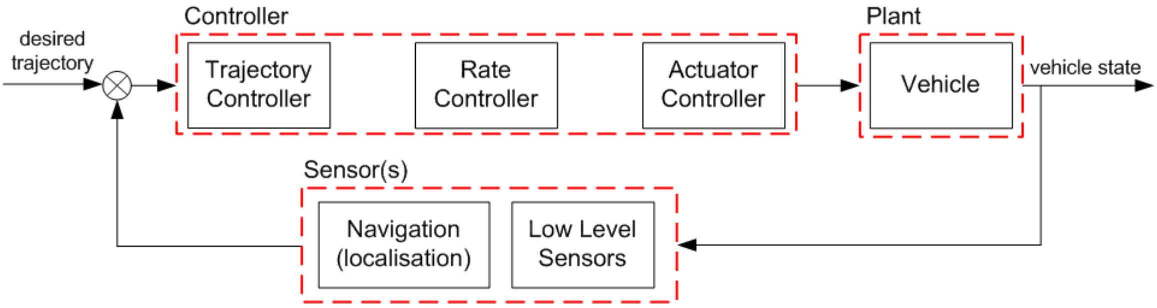


Figure 2.4: The trajectory controller shown in Figure 2.3 is grouped into functionally similar units, to illustrate that it is a basic feedback controller as in Figure 2.1.

where they can be applied. As typified by the CORD trajectory controller in Figure 2.3, AUGV control architectures often have modules ranging from the very low level such as actuator controllers, to higher levels such as trajectory control. Even higher levels of control are then required to achieve mission level planning, such as *what path* the vehicle should take, rather than how to achieve the path. The lower levels can often be achieved with simple control theoretic methods, complete with guarantees on system performance. However, the complexity of the highest level mission control and planning systems often requires a more *ad hoc* approach, which cannot leverage any proven performance guarantees.

2.1.2 Predictive Control

The type of trajectory controller that was considered in Section 2.1.1 is 'blind' in the sense that it minimises the error between the estimated vehicle location and the desired trajectory, without any consideration of the interaction between the vehicle and the environment. The input to such systems is the fully specified desired trajectory, so this form of controller is only sufficient for safe vehicle autonomy when safe trajectories (within the operational constraints of the particular vehicle, including environmental interactions) are provided. For AUGV systems where an operator is able to provide this input directly, no additional control framework is required. When this is not possible, the system must also include the ability to represent the environment around the vehicle and to convolve this with the vehicle model to calculate safe con-

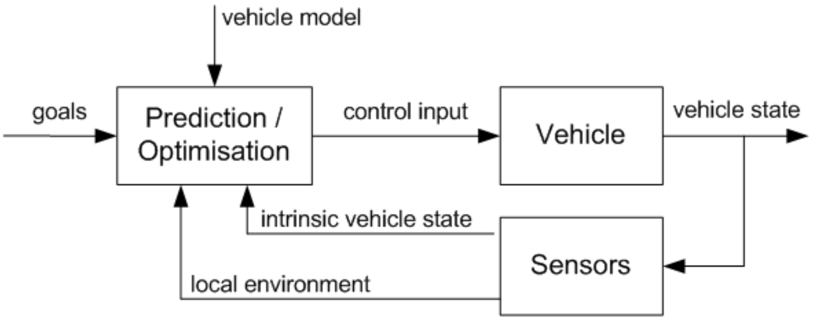


Figure 2.5: A predictive controller makes use of a vehicle model and a representation of the environment to plan appropriate and safe control actions to achieve a specified goal. The predictive block incorporates the processes that are typically referred to as *perception*, *mapping* and *planning*.

trol actions. This component encompasses the processes that are typically referred to as *perception*, *mapping* and *planning*, in which the future interaction between the vehicle and the ground is simulated ahead of time with a model. This is shown for the most general case in Figure 2.5. This general structure broadly defines the inputs and outputs of this component in general, but does not specify any particular methodology (such as control theory) for the implementation. Thus some of the guarantees provided by control theory for the lower level controllers could be lost. The complexity of this block varies significantly and it depends strongly on the application. For well structured environments, assumptions can often be made to simplify or remove this block entirely. For complex, unstructured environments, this block is usually the most difficult to implement and it often forms a bottleneck on overall system performance. A combination of feedback control and predictive control can be used to promote modularity and to maintain a rigorous control theoretic approach for the design of the lower level components. For example, the ‘blind’ trajectory controller from Figure 2.4 can be extended with a predictive component, as shown in Figure 2.6. In [2], the long term benefit of using simple “proven” control strategies over potentially better short term performance is cited as a worthy trade-off.

The predictive control system has a similar appearance to the feedback controller in Figure 2.1. The perception of the environment is continuously fed back to the controller, so that adjustments can be made to the control action that is sent to the

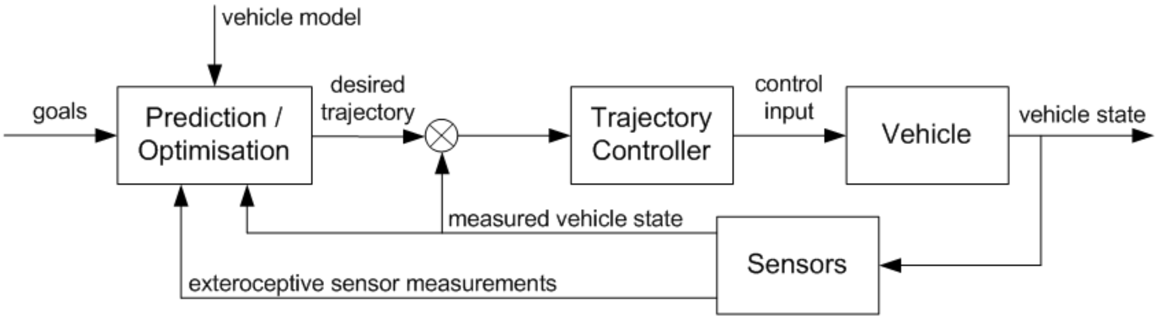


Figure 2.6: The trajectory controller from Figure 2.4 is extended to including a predictive block, to optimise the vehicle trajectory given the perception of the environment and the vehicle model.

vehicle. However, the framework differs from the basic feedback controller because instead of making instantaneous decisions based on the current local information, the predictive controller calculates an optimal set of actions by considering the effect of those actions into the future, similar to the model predictive controller (MPC) architecture [3], but without any restriction on the mathematical structure.

Although there are many different approaches to the implementation of the predictive control block in the literature, all systems that employ some form of perception, mapping and path planning for navigation purposes can be thought of as predictive controllers of this form. As the predictive component does not strictly conform to any one particular design theory, the potential for design errors is unbounded. This thesis analyses the most common types of systematic, stochastic and modeling errors that occur in the perception and mapping systems that form the feedback path and predictive control in typical AUGV system architectures. Although the flexibility of the highest level control systems makes it difficult to bound the type of errors that can be incurred, this thesis identifies common mapping and perception errors in Chapter 3, and minimises or mitigates their effects in Chapters 4 and 5.

2.1.3 Approaches to Perception, Mapping and Planning

Given infinite sensing and model fidelity and infinite computational resources, an optimal implementation of the predictive controller shown in Section 2.1.2 would consist of building a precise representation of the world, over which a dynamic vehicle simulation could be run. The simulated control inputs that yield the best simulated result would then be passed to the vehicle for faithful execution (in a world with perfect models and perfect sensing, no feedback would be required because the true output would be equal to the simulated output). In reality this is not possible and tractable suboptimal implementations always require a compromise. The particular nature of the compromise depends on the specific application, which has led to a large variety of mapping and planning solutions.

The different algorithms exist to leverage the available structure in the environment, in order to achieve the best compromise between high fidelity sensing and modeling, and computational feasibility. In a completely unstructured setting, sensing and representing the world accurately is difficult due to the high complexity and variability of the environment. This complexity also results in highly non-linear dynamic vehicle interactions. Furthermore, even if sensing and modeling were perfect, the complexity requires too much computation to simulate all of the interactions in real time. When structure is added to the operating environment of the robot, the complexity of the sensing, representation, modeling and simulation tasks is reduced, often resulting in an increase in reliability and safety. For example, consider a wheeled robot operating in an unstructured natural environment. To accurately forward predict its motion requires a model of the interaction between the tyres and ground, including the soil mechanics, which is prone to error. On the other hand, if the robot was in a structured urban environment with tarmac roads, then this interaction could possibly be assumed constant. An overview of the development of ground vehicle robotics is presented in [4], which notes the relationship between increases in achievable autonomous vehicle speeds and the assumptions of structure within the environment. In the following section, two different AUGV control systems from the literature are presented. Each relies on different assumptions about the structure of the environment and therefore makes different compromises to achieve computational tractability. The different amounts of environmental structure in these two examples relates directly to the reliability that the systems achieve.

Arguably the most structured environment in which AUGVs operate is indoors in an office-like space. Here it is reasonable to assume that the ground surface is uniformly flat, that the interaction between the vehicle and this uniform surface is constant everywhere and that the only other category of space is non-traversable regions categorised by salient non flat objects. Representations such as Occupancy Grids [5] perform a binary classification of the environment into traversable and non-traversable regions. In most cases, these systems are using an implicit (yet reasonably high fidelity) vehicle/world interaction model. For example, occupancy grid representations of laser range sensor data imply a model where tall things that reflect laser energy are non-traversable and things that permit laser energy are traversable. The traversable regions are assumed uniformly flat with no direct sensor information required. In most indoor environments this is reasonable, so the simple binary classification approach used by occupancy grids is made possible by the well defined structure of the environment. It is worth noting that the implicit model is easily violated by transparent hard surfaces such as glass and this type of perception problem is discussed in detail in Chapters 3 and 5.

By contrast, outdoor unstructured environments cannot be easily represented by a binary classification of traversability. Many different facets of the environment must be considered, including the geometry and nature of the ground (flat, rough, grass, tarmac) and the presence of objects on that surface that may or may not be traversable by the UGV (rocks or shrubs). To accurately simulate the effect of driving over such terrain would require not only the ability to sense the specific properties of the ground and the objects that lay upon it, but also to simulate the dynamic effects of the vehicle under different control actions. Solutions to forward prediction in unstructured environments typically fall into two categories; (i) simplified dynamic simulation over a purely geometric terrain representation and (ii) cost mapping, which performs a spatial classification/transformation of the world into a continuous traversability space. The former is suboptimal because simplifications must be made in the vehicle and environment models to ensure computational tractability. The latter is suboptimal because the spatial classification neglects the additional dimensionality provided by the control space; an area that is non-traversable under one control action may be traversable under another.

There also exist many environments between the two extremes above. An outdoor urban environment contains less structure than an indoor office, yet the presence of relatively flat tarmac roads provides some valuable assumptive power. Soil mechanics can now be disregarded and the purely geometric world model is now likely to be accurate.

2.2 Real World AUGV Systems

In Section 2.1, the control system architectures that are frequently used for AUGVs were discussed in general terms. There are too many different implementations of AUGV autonomy in the literature to list them all, particularly in the areas of perception, mapping and planning, and the main reasons for this were given in Section 2.1.3. In this section, two specific AUGV systems are discussed, with emphasis on how they achieve robustness and reliability. This serves to instantiate the abstract examples in Section 2.1 and to motivate the work in this thesis.

2.2.1 An Autonomous Vehicle for Cargo Handling

Some of the most significant AUGV developments of the past decade have been in the area of port cargo handling. In [6], an AUGV system for carrying shipping containers at a port is described in detail. The work is of particular significance because the system is commercially viable and has been successfully integrated into port operations. This is of interest because of the high safety requirements imposed by any commercial operation, and the high reliability and robustness requirements that are imposed by commercial viability. In this section, the methods of [6] that were used to achieve the high reliability requirements are discussed.

The commercial viability of the system comes from three main advantages of autonomy. Firstly, the ability to control the traffic flow within the port allows for a higher level logistical control element that increases efficiency. Similarly, the logistics of inventory management can be streamlined. Thirdly the running costs of the vehicles are reduced by the consistency of autonomous control. Due to the need to operate within the existing port infrastructure, many constraints are imposed on the system from the beginning. These include factors such as size, operation time, cost, safety and reliability. Of all of these constraints, reliability is cited as the most difficult issue to resolve. Operator interventions may only occur at a rate of fewer than one per one hundred hours of autonomous operation. "System faults particularly in sensing, navigation, and control systems must be minimised, and those that do occur must be identified" [6]. It is this requirement in particular that differentiates this system from many experimental (non-commercial) AUGVs, which often struggle with reliability.

The highly structured operating environment of this vehicle is the key factor in permitting the requisite level of reliability. Considering the diagram in Figure 2.5, in many systems (including the PerceptOR program discussed in Section 2.2.2) the optimisation strategy, which includes perception, mapping and planning is the most difficult component to implement, and can be a bottleneck for the reliability of the whole system. However, the structure in the port environment allows some safe assumptions to be made that reduce the demand on the perception, mapping and planning systems. In particular, the structure of the environment is two dimensional, static and known *a-priori*. This avoids the need to perceive, map and represent complex unstructured terrains. The complete prior knowledge also bounds and simplifies the planning task considerably. The interaction between the ground and the vehicle can be assumed to be constant everywhere, meaning that the vehicle model is independent of feedback about the perceived nature of the ground at a given location. Largely due to these assumptions, the final software implementation was done in 12 'man-months' and consisted of 4000 lines of code. By contrast, the PerceptOR program [4] (which is discussed further in Section 2.2.2) developed an AUGV to operate in unstructured environments and was not able to rely upon any of these assumptions. This program took 500,000 lines of code and 30 'man-years' to develop. Both systems employed a rigorous program of testing and evaluation, yet the reliability of the cargo vehicle is superior.

Although the structure made reliability possible in [6], much effort was still required to achieve it, particularly for the navigation (localisation) and control of the vehicle. Differential Global Positioning System (DGPS) based navigation systems were too unreliable and inaccurate, due to occlusions from overhanging structures in the port such as cranes. Instead, further structure was added to the environment in the form of precisely surveyed, static radar beacons, which could be detected by a custom developed scanning radar sensor. The range and bearing information provided by the sensor is used together with a model of the vehicle motion to provide pose estimates of the vehicle. This is done with a Kalman filter [7] based method, which is equal to the localisation half of the popular Simultaneous Localisation and Mapping (SLAM) technique, (an overview of which can be found in [8, §37]). By fixing the beacons in known locations, the process of associating radar observations with particular beacons is constrained, thus increasing the robustness.

To minimise systematic errors, a calibration procedure was performed to estimate the parameters of the radar sensor. The radar calibration procedure was also leveraged from environmental structure; a straight section of rail was available, so the sensor was mounted on a buggy to constrain the motion to lie along the track. Beacon observations could then be corrected by refining the estimates of the radar parameters. In Chapter 3, the effect of sensor misalignment is analysed in detail and in Chapter 4, a generic calibration procedure is given to provide optimal estimates (including the uncertainty) of the geometric parameters for range based sensors.

The system calculates the uncertainty of the 2D vehicle pose, which is then propagated through the coordinate transformations to provide consistent estimates of overall uncertainty for the data association. These types of coordinate transformations are common in vehicles with exteroceptive sensors. Similar calculations are analysed and extended in Chapters 3 and 4, to a full three dimensions, and to include additional sources of uncertainty such as timing offset errors between sensors.

More recently, Global Positioning System (GPS) technology has developed further, but the problem of reliability and accuracy is still present, particularly in environments which partially occlude the satellite signal reception, or where the environment reflects the signal, causing multipath [9]. The automation of another port cargo handling vehicle is described in [10], where reliability is achieved by employing a 'high integrity' navigation system, that is guaranteed to know when it is unable to provide a reasonable pose estimate. The reasoning behind such navigation systems is that complete reliability of the estimated vehicle pose cannot be achieved all the time. By explicitly allowing for the possibility of failure and focusing effort to detect when this occurs, the system as a whole can be relied upon, although sometimes pose estimates will not be available [11]. In [10], high integrity navigation is achieved by using dual pose estimators, one using a similar radar/beacon method as described by [6], and the other based on modern commercial DGPS technology. The pose estimates from both are compared to check for possible errors in either, as a form of multi-modal hardware redundancy. An undetected failure can therefore only occur if both systems fail simultaneously, but this is unlikely due to the different physical processes involved [10]. Notably, [10] states that "After many years of operation, we have concluded that the physics of GPS precludes it ever being made sufficiently reliable to be used as a stand-alone autonomous navigation system. This is clearly contrary to the received wisdom in the autonomous vehicle area". Indeed the use of GPS based systems has been one of the most commonly cited reliability problems for modern experimental AUGV systems, such as several of the finalists in the 2007 DARPA Urban Challenge [12, 13, 14, 2, 15] and the PerceptOR program [4], which is discussed further in Section 2.2.2. An open challenge faced by the navigation community is to provide high reliability, high integrity systems that do not rely on strong structure within the environment.

2.2.2 The DARPA PerceptOR Program

The DARPA PerceptOR program was undertaken by Carnegie Mellon University (CMU) over a three year period, with the aim of developing perception systems for AUGVs in completely unstructured and partially structured environments. A description of the program and the system architecture that was developed for it can be found in [4]. In this section the system is summarised and discussed with particular emphasis on the main sources of error that impact on the overall robustness and safety.

The system consisted of an AUGV that was developed for the program and an unmanned helicopter, that was previously developed at CMU. The scope of the program was large, including developments in the areas of planning, perception, navigation (localisation) and control. Perception was identified as the primary bottleneck of the whole system, so the focus was placed in that area. An outstanding element of the program was the testing procedure, in which the system was run in many different environments across several states of North America throughout the three year period. The goal during testing was for the vehicle to autonomously reach sparse globally positioned waypoints, under the supervision of an operator. The operator was given information about the test site immediately before the test began, and had no direct interaction with the site other than through the system, by visualising *a-priori* data, telemetry and sensor data from the AUGV and the unmanned aerial vehicle (UAV). A separate team followed the AUGV in the field, to directly monitor and record its performance and to issue emergency stop commands if necessary. This team did not communicate directly with the operator.

To build maps of the environment, the AUGV employed ranging laser scanners, colour cameras and stereo camera pairs. The UAV had ranging lasers that were synchronised with colour sensors to provide 'coloured' three dimensional data. The UAV system was effectively used by the AUGV as a remote sensor, providing an advantageous viewpoint from above, that was capable of perceiving features of the terrain that would not be possible from the perspective of the AUGV, such as negative obstacles (ditches or other features of the terrain that are lower than the surrounding ground). In addition to the onboard sensors, *a-priori* information in the form of high resolution digital elevation maps was used to provide the global planning layers with initial information to guide the vehicle towards the goals.

One of the most interesting aspects of this work is the detailed discussion of the evolution of the system. Testing was performed throughout the three year program and [4] summarised the flaws in the system at the time of testing. The solutions that were engineered to overcome these challenges were then presented including their degree of success in the subsequent tests. Of the issues that resulted in poor performance, particularly in the earlier tests, the vast majority could be described as sensor interpretation errors due to unmodeled environmental conditions. This occurs when there are errors in the explicit translation between sensor information and the perception of the environment, or due to violated assumptions about the environment that were relied upon for this translation.

Navigation (Localisation) Difficulties

One of the largest problems cited by the PerceptOR program was the inability of GPS based navigation equipment to provide reasonable estimates of the vehicle pose when under tree canopies, due to occlusions and multipath of the requisite satellite signals. This is the same problem that was cited in [6], due to overhead cranes. As mentioned at the beginning of this chapter, exteroceptive information must be mapped to provide the vehicle with a perception of the environment through which it must navigate. The importance of precise localisation for this task is paramount, as recognised by [4] and discussed in detail in Chapter 3. Despite using very high quality commercial navigation equipment, the PerceptOR team reported that under certain environmental conditions, the navigation equipment was not only providing imprecise, inaccurate information, but that it was not able to reliably estimate the quality of its own output; it was a low integrity system. A significant requirement for robustness is the ability to recognise the limitations of the system. This includes constant limitations that are imposed by static design constraints, but also dynamic limitations imposed by fluctuations in the quality of sensor information and the processing algorithms that use this. In Chapters 3 and 4, algorithms are developed to calculate both design-time and run-time constraints of exteroceptive sensor mapping algorithms.

To overcome this problem, the PerceptOR system maintained two separate estimates of the vehicle pose. A global estimate was provided by the standard fusion of GPS and an inertial navigation system (INS) and a local estimate was maintained without the use of any global information, such that it would be immune to GPS problems under canopies or in other problematic environments. Since then, the use of dual global and local frames has become more popular for AUGVs. At least three of the six teams [13, 14, 2] that completed the 2007 DARPA Urban Challenge [12] made explicit use of separate navigation frames in this way, due to the inadequacies of commercial GPS/INS. The latter team provide a detailed description of a method of managing the uncertainty of the two frames in [15]. They conclude with the hope that INS manufacturers will develop products that provide better functionality to assist the implementation of a local and global frame of reference. The "All Source Navigation Filter" (ASNF) project [16, 17] being undertaken at the Australian Centre for Field Robotics (ACFR) at the University of Sydney has this as one of it's primary aims. The project has the ambitious goal of fusing all available relevant sources of information into one consistent, tightly coupled structure, while providing the flexibility to easily add new information sources or to change which sources are in use. The core filter is designed to be a black box solution, by providing pose estimates in local and global frames, without requiring a vehicle model. Currently the system fuses GPS, INS, range sensor data in a simultaneous localisation and mapping (SLAM) framework [18] and time-differenced carrier phase (TDCP) information [19]. This navigation system is ideal for robotic applications in general, but particularly for vehicles operating in complex unstructured environments such as PerceptOR. The problems caused by overhanging structures such as canopies that occlude satellites, or other geometric structures that can cause GPS multipath are lessened or avoided by the reduced requirement for GPS information. Highly accurate absolute velocity measurements can be derived using TDCP with observations from only one satellite. This has the additional benefit of bounding absolute angular pose errors. Also, the complexity of the interactions between the vehicle and the terrain makes it difficult to specify an accurate vehicle model, so 'black-box' navigation systems that do not require such models are ideal. Furthermore, if a particular aspect of the environment can be leveraged by a well matched sensor, this can be easily added to the filter. The ASNF is evaluated for an AUGV in Chapter 4, in similar environmental conditions to the PerceptOR program, to test its ability to mitigate against the combined localisation-perception problems encountered by the current state-of-the-art AUGVs.

Nuances of the Environment

A secondary problem encountered by the PerceptOR program is one relating to the qualitative interpretation of sensor information, given the presence of complicated nuances in the environment. Initially the system mapped exteroceptive sensor information from laser range scanners and stereo camera pairs to a purely geometric representation of the environment. This implicit perceptive layer assumes that the opacity of objects to visible light (in the case of stereo cameras) and near infra-red light (in the case of laser scanners) has a direct relationship to the way the geometry will support the mass of the vehicle. In other words, concrete objects are perceived as identical to foliage, despite the fact that the vehicle can safely traverse the latter, but must avoid the former. This causes the system to be overly cautious at best, and prevents it from reaching destinations at worst. The perceptive model was improved to include the nuance of density. A three dimensional density model was employed to record the differences between completely opaque surfaces and ones that are partially penetrable to light. The model still relies directly on the opacity of objects, but now considers that some objects are more permissible to light than others, which in many cases (although not all) is related to the object's hardness. The improved model was successful at allowing the vehicle to drive through soft structures while avoiding most hard obstacles, yet it had the unwanted effect of also permitting the vehicle to collide with thin hard structures such as fence posts, wires and cables. These objects are problematic because they have a similar appearance to foliage according to the chosen density based classifier. When perceptive models require adaptation to allow for discrimination of ever more specific objects or nuances of the environment, a classification of sensory information is needed. Classification metrics (such as false negative and false positive rates) are insufficient for detecting the robustness of the classification output, because the system may only fail when a particular aspect of the environment is observed. The error rates in the results may reflect the frequency of the phenomena in the environment, rather than the actual success rate of the classifier. Specifying robust classification methods remains an open and challenging problem and this is discussed in more detail in Chapters 3 and 5. In the PerceptOR program, very good results were achieved by their classification method, but at the highest level the robustness was provided by operator supervision. Although many hours of continuous autonomy were achieved, this was surely due to a lack of fence posts, wires and cables in the operating environment, rather than an inherent robustness of the automation.

Other environmental conditions such as airborne dust and rain also had an adverse effect on the perception of terrain and obstacles surrounding the vehicle, because these phenomena violated some of the assumptions of the initial environment model. It was initially assumed that the range information from sensors such as laser scanners or stereo cameras would relate directly to a geometric model of the terrain surrounding the vehicle. The presence of airborne particles due to dust or rain strongly violates this assumption, because the particles are often detected in mid-air by the sensor. This was mitigated by the PerceptOR system by using the same density metric that was constructed to detect soft vegetative structures. Airborne dust has been noted as an unsolved problem for other systems in the literature [2, 20, 21, 22], which is especially noteworthy because they were nominally fielded in paved environments during the 2007 Urban Challenge [12], where significant volumes of dust might not be expected. The detection of airborne dust particles is used as a recurring theme in Chapter 5 to illustrate the potential for multi-modal sensor fusion to increase classification robustness. A specific solution that fuses millimetre wave radar and laser sensors is also developed and implemented on an AUGV.

Robustness

The PerceptOR system is arguably one of the most complete, most robust implementations of outdoor ground vehicle autonomy for operations in unstructured environments. By performing a rigorous iterative testing regime, followed by a development cycle to address problems with reliability and missing functionality, the final system is capable of completely autonomous driving for long periods of time (multiple hours) and over large distances (multiple kilometres). However, it is notable that the operator provides much of the overall reliability and safety of the system. Although the system worked most of the time, the operator was solely responsible for identifying malfunctions at the highest level. Initially the operator did not have enough feedback to perform this function, so additional sensors (video cameras) were added specifically for this purpose.

In the near future of ground vehicle robotics, commercial viability will require the development of safe, robust systems that employ ever fewer operators controlling ever more autonomous vehicles. The current state-of-the-art AUGV systems operating in completely unstructured environments *require* operator supervision and intervention at some level. This is unlikely to change in the near future, due to the complexity of the environment, and the current inadequacy of perception systems to deal with this. It is the ratio of time invested by an operator to monitor or manually control one robot, compared to the amount of time that the robot can spend *completely free from supervision* that dictates how many robots a single operator could feasibly control. If multiple robots are to be controlled by a single operator, then an absolute requirement will be the reduction in operator supervision. This does not necessarily require completely *persistent* autonomy, where sensors and algorithms never fail, rather it requires a more broad definition of persistent autonomy, where the failure state of a whole system is persistently known. This is identical to the concept of 'high integrity' in the navigation community [11].

Provided that autonomous systems can calculate when perception algorithms are not adequately spanning the tasks (for whatever reason, including sensor failure), operator intervention is required, but constant supervision is not. This increases the robots per operator to a number potentially greater than one. Autonomous systems can safely and reliably be deployed as soon as this is achieved, even if intervention is required quite frequently in the short term. The commercial viability of these systems will depend on the specific application and the costs involved. Ultimately the autonomous durations will increase as the model fidelity is improved incrementally over time. In Chapters 3, 4 and 5, algorithms and methods are developed to promote robustness and safety by providing design-time and run-time estimations of uncertainty due to some of the most common sources of error that are found in typical perception algorithms.

Chapter 3

Identification of Mapping Errors

In Chapter 2 the three primary sources of error for mapping were identified as:

- Systematic errors due to sensor misalignment and synchronisation,
- Stochastic uncertainty due to sensor measurement errors,
- Sensor interpretation errors due to unmodeled environmental conditions.

In this chapter, the effect of these errors on the mapping process will be analysed in detail. When mapping is used as part of the exteroceptive sensor feedback process in an autonomous unmanned ground vehicle (AUGV) control system, the safety of the chosen actions relates directly to the accuracy of the map. Therefore an understanding of the sources of error and the magnitude of their effect on mapping is critical for robust and safe implementations of AUGV systems.

This chapter is divided into four sections. Firstly, Section 3.1 introduces the coordinate frames and transformations that are required regardless of the choice of mapping algorithm or representation. In Section 3.2 a sensitivity analysis is presented to quantify the magnitude of error in the map, due to varying sensor and platform configurations, localisation systems and timing considerations. This allows mapping errors to be considered at design-time and monitored actively at run-time. Section

3.3 presents experimental data that validate the mathematics and provide greater understanding of the empirical nature of the mapping errors. Finally, Section 3.4 discusses the potential for an interpretive perception component to add unobservable errors to the system. The analysis in this chapter provides a basis for Chapter 4, in which solutions are presented to mitigate and manage mapping errors, and Chapter 5, in which solutions to reduce perceptual misinterpretation are given. The approach taken in Section 3.1 and 3.2 is similar to the analysis done in the first half of [23] and is mentioned in [24]. This thesis provides a complete model of the errors and a rigorous evaluation of their effect.

3.1 Coordinate Frames and Transformations

In almost all non-trivial mapping scenarios, the sensors can only sample from a small region of the larger area to be mapped. The complete map is built by physically moving the sensors (by moving the entire platform, or by moving the sensors independently from the platform, or both) and fusing the information into a single representation. Regardless of the choice of mapping algorithm, the use of a single representation usually necessitates the use of several coordinate frames. In this section, the three coordinate frames that will be used are described, as are the transformations between them. In Section 3.2, a sensitivity analysis is done to determine the error in the final map due to these transformations.

3.1.1 Sensor, Body and Navigation Coordinate Frames

The three main coordinate frames used in this chapter are the sensor, body and navigation frames, shown in Figure 3.1 for a typical UGV platform configuration, although they apply equally well to any type of mobile platform with one or more exteroceptive sensors. The choice of these three frames arises from the utility of having a frame attached to each of the components of the system: the sensor, the vehicle and the operating environment, each of which is discussed in detail below. In

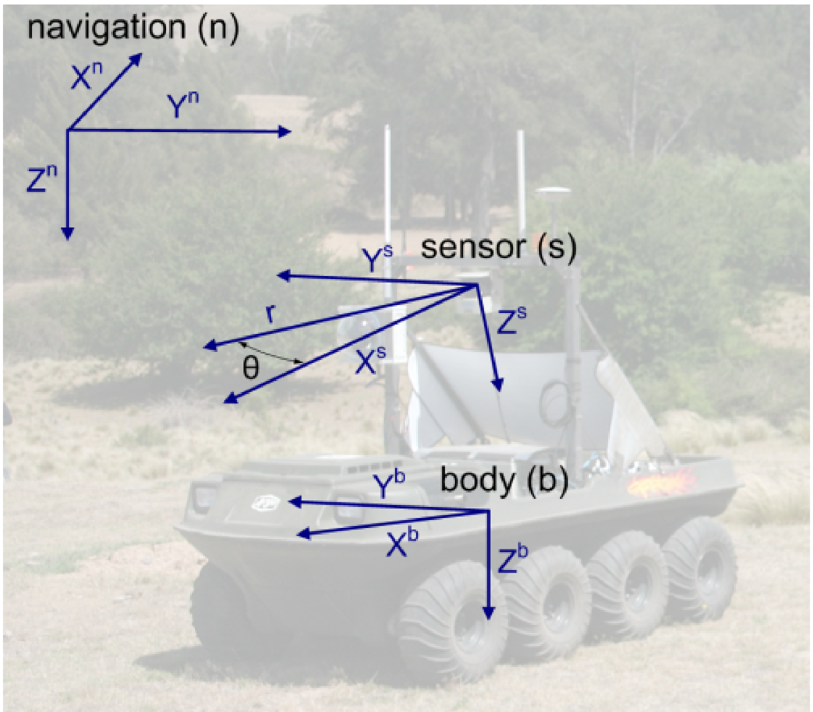


Figure 3.1: An example configuration of the sensor, body and navigation frames superimposed on the CAS Outdoor Research Demonstrator platform. The sensor frame, denoted s, is fixed to the scanning axis of a 2D laser range scanning sensor. The body frame b is fixed to the centre of gravity of the vehicle (although it can be fixed to any point on the vehicle that is rigidly connected). The navigation frame n is fixed to some stationary point in the world.

this thesis, all of the mathematics used to describe the frames are explicitly formed in three dimensions, even though some UGV implementations operate with a strictly two dimensional representation (such as building a two dimensional map of the walls of an indoor office space). All of the three dimensional equations can be used to solve two dimensional problems by setting one of the axes to a constant (in this case $Z = 0$ is the most natural choice). Although the two dimensional equations are simpler to analyse, the physical implementations of *all* UGV systems are intrinsically three dimensional anyway, so the full three dimensional analyses allow the assumptions of a two dimensional world to be tested.

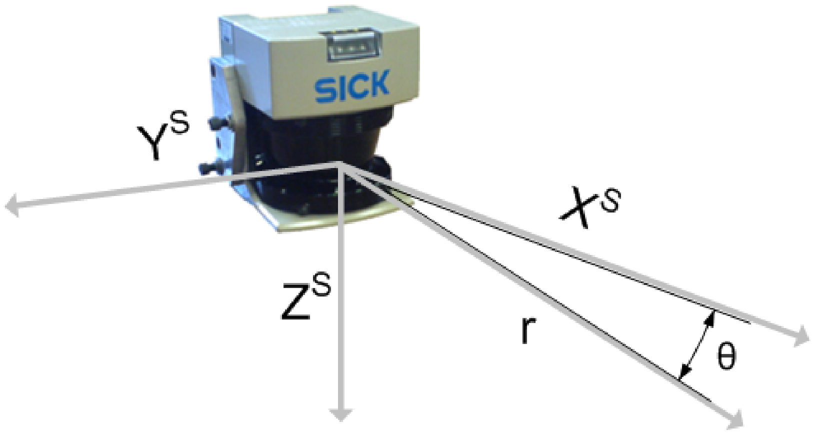


Figure 3.2: The sensor frame for a 2D scanning laser (SICK-LMS291). The vectors X^s, Y^s and r are coplanar. θ is the positive angle around the Z^s axis, with an angle of 0 corresponding to a point on the X^s axis.

Sensor Frame

The sensor frame s_i is defined by its alignment to the *ith* sensor, so if multiple sensors are available, then each has a unique sensor frame. If only one sensor is present, then its frame is denoted simply by s. In this chapter, no assumption is made about the *type* of sensor information, but it is assumed that the sensor is capable of providing a three dimensional spatial location $\mathbf{p}^s \equiv [p^s_x, p^s_y, p^s_z]^T$ in s, corresponding to the measurement. For example, a laser range scanner, a radar range scanner and a stereo camera pair provide different types of information (near infra-red reflectivity, radar reflectivity and colour respectively), but they all provide this information with associated three dimensional coordinates in the sensor frame. A particular sensor may natively provide information in polar coordinates, but the three Cartesian coordinates are directly observable from this. This chapter does not apply to sensors where the full three dimensions are not directly observable, such as a single camera.

As an example, Figure 3.2 illustrates the sensor frame for a SICK-LMS291 2D range scanning laser. The laser returns the range r and the bearing θ to an area of the environment that reflects enough of the transmitted laser energy to exceed an internal threshold. This sensor provides the full three dimensional spatial information as given by the polar to Cartesian conversion:

$$\mathbf{p}^s = \begin{bmatrix} r\cos(\theta) \\ r\sin(\theta) \\ 0 \end{bmatrix} \tag{3.1}$$

Body Frame

The body (or platform) coordinate frame b is fixed at some point on the mobile platform. For this thesis, the frame is positioned so that the X^b axis points forward, the Y^b axis points to the right and the Z^b axis points down, as illustrated in Figure 3.1. The mathematics in this thesis requires a right handed frame, fixed to a point that is rigidly connected to the vehicle, but other choices that satisfy this requirement are possible. Sensors are located with respect to b, by the constant offset denoted $\mathbf{r}_s^b$, which is defined as:

$$\mathbf{r}_s^b \equiv \begin{bmatrix} r_{s,x}^b \\ r_{s,y}^b \\ r_{s,z}^b \end{bmatrix} \tag{3.2}$$

For clarity, this vector is denoted $\mathbf{r}$ to emphasise that it is a constant. It describes the translational position of the sensor frame origin, in the body frame. The rotation of the sensor is described by the three constant Euler angles (yaw about Z^b, pitch about Y^b and roll about X^b) needed to rotate frame b to align with frame s. For convenience, these parameters are grouped into a single vector, denoted $\mathbf{\Phi}_s^b$, given by:

$$\mathbf{\Phi}_s^b \equiv \begin{bmatrix} \phi_{s,z}^b \\ \phi_{s,y}^b \\ \phi_{s,x}^b \end{bmatrix} \tag{3.3}$$

It is important to note that this vector is defined for convenience only. It is used to group the Euler angles that describe a rotation and it has no direct application in linear algebra in this thesis. The ordering of the vector is used to represent the order in which the axial rotations are concatenated. This will be discussed in more detail below.

If multiple sensors are mounted on the one vehicle, then each has a unique frame *si* with a unique $\mathbf{r}^b_{si}$ and $\mathbf{\Phi}^b_{si}$ to describe the mounting location on the platform. In this way, the body frame provides a convenient point of reference with which to describe the *constant* position of each sensor. Together, the grouped term $\{\mathbf{r}^b_{si}, \mathbf{\Phi}^b_{si}\}$ describes the six degree of freedom (6DOF) constant sensor pose.

Navigation Frame

The navigation coordinate frame n is fixed to an arbitrary point on the Earth and is used to represent the stationary environment in which the vehicle operates. In this thesis, the frame is chosen such that the X^n axis is metres north of the origin, the Y^n axis is metres to the east and the Z^n axis is metres below the origin. The dynamic pose of the body frame within the navigation frame is defined in a similar way to the constant location of the sensors within b, defined above. Because the vehicle is mobile, the translation and rotation of the vehicle within n is non-constant. The translational position of the vehicle is denoted by $\mathbf{p}^n_b$, and is defined as:

$$\mathbf{p}^n_b \equiv \begin{bmatrix} p^n_{b,x} \\ p^n_{b,y} \\ p^n_{b,z} \end{bmatrix} \tag{3.4}$$

The rotation Euler angles of the vehicle are denoted by $\mathbf{\Psi}^n_b$ and are given by:

$$\mathbf{\Psi}^n_b \equiv \begin{bmatrix} \psi^n_{b,z} \\ \psi^n_{b,y} \\ \psi^n_{b,x} \end{bmatrix} \tag{3.5}$$

The terms $\{\mathbf{p}^n_b, \mathbf{\Psi}^n_b\}$ together describe the 6DOF location of the vehicle in the world, and together they are referred to as the navigation solution, or vehicle pose. These terms are estimated by the navigation (or localisation) system. The body frame provides a convenient reference to allow the specification of the sensor locations with respect to the platform only, which allows the navigation system to be completely independent of the sensor frame. Without the body frame, the navigation system would have to provide an estimate of the pose of *every* sensor in n.

Global positioning systems (GPS) commonly provide position estimates in an ellipsoidal coordinate system, as latitude, longitude and altitude above the mean sea level (MSL). In this thesis, the navigation frame is strictly Cartesian, so a transformation from ellipsoidal to Cartesian coordinates is required. Redfearn's formulae [25] are used to perform the Transverse Mercator projection using the Mapping Grid of Australia (MGA). This projection has better than millimetre accuracy everywhere in Australia, so uncertainty due to this transformation is disregarded [26].

Choice of Frame for Mapping

In Chapter 2, it was argued that a generic all-purpose perception system would be computationally intractable. To achieve computational tractability, assumptions about the structure of the environment must be leveraged, to allow for simplifications in the complexity of the perceptive algorithms and a reduction in the sensing requirements. The large variety of application domains has led to a large variety of algorithms. Similarly, the available assumptions in the application-specific domain, impact on the choice of frame in which to maintain mapped perceptive information, or indeed whether a map is even required. In Section 2.2.1, a cargo handling system was discussed in which a perceptive map was not actually required at all. This was because the environment was highly structured, static, and known in advance, reducing the need to perceive it. By contrast, in Section 2.2.2, the system was designed for operation in an unknown (or partially known) unstructured environment, requiring large amounts of perceptive reasoning and the mapping of this information. When a map is required, it can theoretically be maintained in s, b or n; the mathematics required to do so can be specified, but the implementation may be significantly more or less practical. The optimal choice of mapping frame depends heavily on the type of sensor, the type of platform, and the nature of the environment in which the vehicle operates.

This thesis is concerned with the sub-set of sensing and mapping systems where the sensor cannot perceive the entire environment at one time or from one perspective, requiring the mobile platform to be used as a scanner to successively build the map over time. In this scenario, the sensor information could theoretically be maintained in any of the three frames s, b or n, but if either s or b were used, the static contents of the map would have to be shifted as the frame moves. Mathematically this is possible, but any practical implementation would be inefficient [8, §4.4]. For this reason, all of the successive map building examples in Chapter 2 use n for the mapping frame and this is common in the literature generally.

The origin of n can be chosen to align to a *standard* global coordinate system (such as MGA) or it can be set to an *arbitrary* fixed point on the ground. Both are equivalent in terms of the mathematics that describes the frames. However, the choice between the two depends on many factors and has some significant impacts on the system. If there is no source of global information in the entire system, then it is not possible to estimate the vehicle position with a standard origin for n. In this case, the localisation must be done with respect to some convenient arbitrary origin. This thesis refers to arbitrary choices for n as *local* navigation frames (convenient choices for the origin are often found within the operational environment of the vehicle, although not necessarily), and standard choices for n are referred to as *global* navigation frames.

If there is a source of global information, then the choice to use a global frame will depend on the quality and availability of this information. In Section 3.2, the sensitivity of the transformation from s to n with respect to the navigation solution is calculated. This in turn enables the system designer to specify how accurate the navigation system must be, in order to keep the mapping accuracy above a chosen threshold. If the quality and availability of navigation information is high enough that the global uncertainty is always below the desired threshold (such as from a DGPS), then it may be possible to permanently use a standard global frame for n. If the quality *or* the availability of the global component of the navigation system is insufficient, then a locally fixed frame may be preferable. For example, it may be possible to calculate the vehicle trajectory from one point to another with much higher accuracy than the entire trajectory can be located in global coordinates. There are several key advantages to using a standard global frame. Maps made from different platforms can be easily shared without the need to correlate map features for alignment and they can also be easily stored in a database and re-used at a later time. Places of interest such as waypoints or targets can be specified more easily with global coordinates. On the other hand, sudden rifts due to fluctuations in the GPS signal, or due to loop closures in SLAM are common and often difficult to manage, particularly in terrain mapping contexts.

3.1.2 Coordinate Transformations

The coordinate transformation of a point $\mathbf{p}^i$ in frame i to a point $\mathbf{p}^j$ in frame j is parameterised by the offset and rotation between the two frames. Specifically, the offset $\mathbf{p}_i^j = [p_{i,x}^j, p_{i,y}^j, p_{i,z}^j]^T$ is the location of frame i in frame j, and $\mathbf{\Phi}_i^j = [\phi_{i,z}^j, \phi_{i,y}^j, \phi_{i,x}^j]$ denotes the ordered Euler rotation angles about the axes of j required to rotate frame j to align with frame i.

The notation used here is similar to [27] and [28]. Each Euler angle specifies a rotation of the frame about an orthogonal axis, while the point remains fixed in space. This is known as a *passive* rotation [29]. Other equivalent parameterisations are commonly used for rotations, such as quaternions [30, p373], Cayley-Klein parameters [31, p154], 4×4 rotation and translation matrices [29][23], and *active* Euler angle rotations (in which the point is considered to rotate instead of the frame) [29]. For Euler angle rotations there are twelve possible conventions for the ordering of the angles [31, p154]. The 'xyz convention' Euler angle parameterisation is used in this thesis to allow for 'human interpretable' parameters, although any of the alternatives could be used instead. This convention is commonly used in engineering applications, particularly when used to express vehicular motion [31, p154].

The three passive rotations are combined by first rotating the frame about its Z axis.

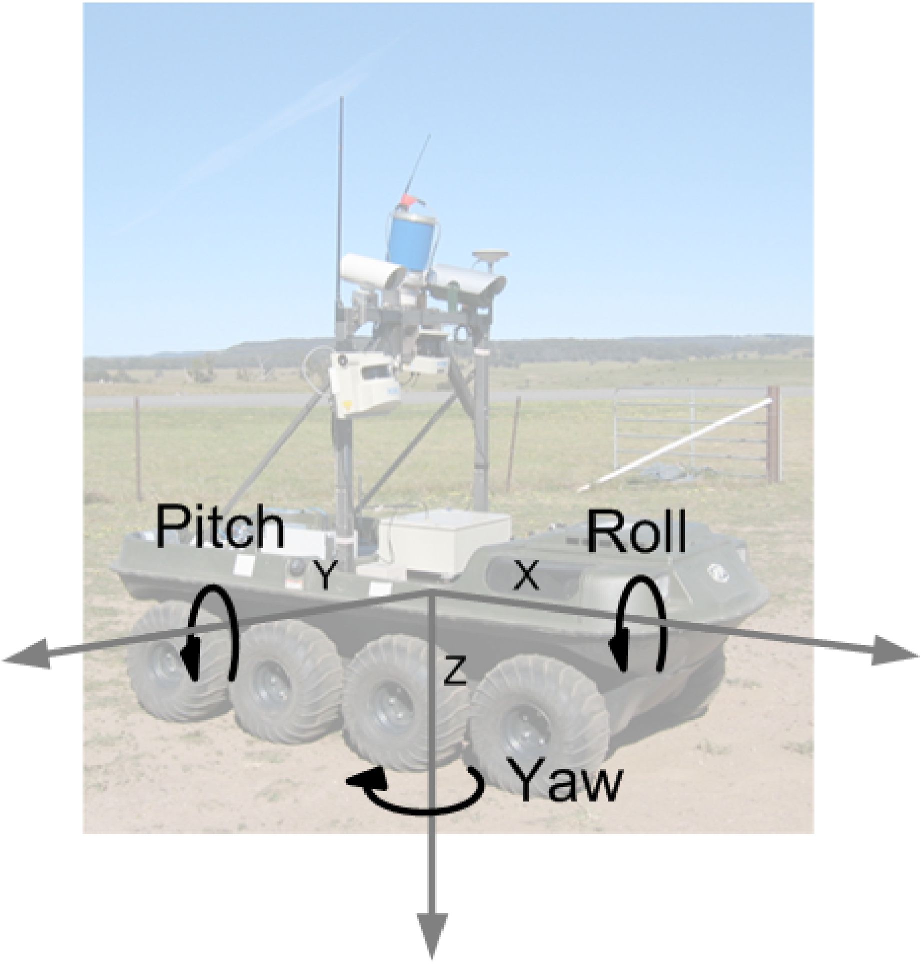


Figure 3.3: Positive Roll (ϕ_x), Pitch (ϕ_y) and Yaw (ϕ_z) angles shown about the positive axes of the vehicle body frame.

The frame is then rotated about its Y axis, which may have moved as a result of the Z rotation, then finally about its X axis, which may have moved due to either of the previous actions. This parameterisation and ordering is well suited to vehicle based coordinate frames, because it yields familiar 'human readable' descriptions of the Euler angles; ϕ_z corresponds to yaw, ϕ_y corresponds to pitch and ϕ_x corresponds to roll as shown in Figure 3.3. If another ordering were chosen, such as rotating about Z, X then Z again, an equivalent but numerically different set of three angles could be chosen to parameterise a rotation, but they would not have understandable descriptors such as yaw, pitch and roll.

Each orthogonal rotation is specified by a 3×3 rotation matrix $\mathbf{C}$, annotated with a single subscript (x,y or z) indicating the orthogonal axis of rotation:

$$\begin{aligned}
\mathbf{C}_z &= \begin{bmatrix} \cos(\phi_z^j) & \sin(\phi_z^j) & 0 \\ -\sin(\phi_z^j) & \cos(\phi_z^j) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
\mathbf{C}_y &= \begin{bmatrix} \cos(\phi_y^j) & 0 & -\sin(\phi_y^j) \\ 0 & 1 & 0 \\ \sin(\phi_y^j) & 0 & \cos(\phi_y^j) \end{bmatrix} \\
\mathbf{C}_x &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi_x^j) & \sin(\phi_x^j) \\ 0 & -\sin(\phi_x^j) & \cos(\phi_x^j) \end{bmatrix}
\end{aligned} \tag{3.6}$$

With sin() and cos() replaced with s() and c() respectively, the orthogonal Euler angle rotations combine to produce the 3×3 Direction Cosine Matrix (DCM), denoted $\mathbf{C}_i^j$, indicating a rotation from frame i to frame j [27, p22]:

$$\mathbf{C}_i^j = [\mathbf{C}_x \mathbf{C}_y \mathbf{C}_z]^T \tag{3.7}$$

$$\mathbf{C}_i^j = \begin{bmatrix} c(\phi_z)c(\phi_y) & c(\phi_z)s(\phi_y)s(\phi_x) - s(\phi_z)c(\phi_x) & c(\phi_z)s(\phi_y)c(\phi_x) + s(\phi_z)s(\phi_x) \\ s(\phi_z)c(\phi_y) & s(\phi_z)s(\phi_y)s(\phi_x) + c(\phi_z)c(\phi_x) & s(\phi_z)s(\phi_y)c(\phi_x) - c(\phi_z)s(\phi_x) \\ -s(\phi_y) & c(\phi_y)s(\phi_x) & c(\phi_y)c(\phi_x) \end{bmatrix} \tag{3.8}$$

The coordinate transformation from i to j is given by this rotation and the addition of the translational offset:

$$\mathbf{p}^j = \mathbf{C}_i^j \mathbf{p}^i + \mathbf{p}_i^j \tag{3.9}$$

Two coordinate transformations are combined to transfer the sensor information in s, via the body frame b to the navigation frame n. A spatially located sensor sample $\mathbf{p}^s$ in the sensor frame can be transformed into the vehicle body frame:

$$\mathbf{p}^b = \mathbf{C}_s^b \mathbf{p}^s + \mathbf{r}_s^b \tag{3.10}$$

where $\mathbf{r}_s^b$ and $\mathbf{C}_s^b$ are determined from the mounted sensor location (offset and rotation respectively) in the body frame. This transformation must be done whenever the sensor information is to be interpreted with respect to the vehicle.

Given the vehicle body pose $\mathbf{p}_b^n$ and $\mathbf{C}_b^n$ in n from the navigation system, a point $\mathbf{p}^b$ in the body frame can be transformed into the navigation frame:

$$\mathbf{p}^n = \mathbf{C}_b^n \mathbf{p}^b + \mathbf{p}_b^n \tag{3.11}$$

By combining Equations 3.10 and 3.11 the sensor information can be *mapped*; a point in s can be transformed to n:

$$\mathbf{p}^n = \mathbf{C}_b^n (\mathbf{C}_s^b \mathbf{p}^s + \mathbf{r}_s^b) + \mathbf{p}_b^n \tag{3.12}$$

This equation is required whenever sensory information from a mobile platform is placed in an earth fixed frame such as n, and will be used regardless of the choice of mapping algorithm or representation. Any errors in the input parameters of Equation 3.12 will result in corresponding errors in the output $\mathbf{p}^n$ and hence in the map. This equation depends on:

- The spatial sensor measurements $\mathbf{p}^s$, subject to sensor noise.
- The sensor position and alignment to the body frame $\mathbf{r}_s^b$ and $\mathbf{\Phi}_s^b$, subject to measurement or calibration error.
- The navigation system estimate of the vehicle pose $\mathbf{p}_b^n$ and $\mathbf{\Psi}_b^n$, subject to estimation uncertainty.

To minimise errors in the map, the errors in the quantities above must be minimised. The relative importance of the different sources of error and the level to which they must be reduced depends on the accuracy requirements of the map and on the sensitivity of Equation 3.12 to these parameters. The sensitivity of $\mathbf{p}^n$ to errors in each of the input parameters will be analysed in Section 3.2.

3.2 Sensitivity Analysis

Section 3.1 introduced the coordinate frames and transformations required for mobile exteroceptive sensor mapping applications. It was shown that Equation 3.12 is required in all cases where the sensor information is to be mapped in the navigation frame, regardless of the particular choice of mapping algorithm. In Section 3.2.1 the sensitivity of the output of Equation 3.12 with respect to errors in the input is quantified. This is extended in Section 3.2.2 to consider the *timing* implications that arise from any practical implementation of the system. This analysis enables the system designer to know how accurate the sensor, calibration, localisation and timing must be, in order to achieve a desired mapping accuracy.

3.2.1 Coordinate Transformation Sensitivity

In this Section, a sensitivity analysis is done with respect to the coordinate transformation from the sensor to navigation frame, Equation 3.12. The sensitivity of the spatial location of points in the navigation frame $\mathbf{p}^n$ with respect to errors in the input of this equation is determined. As shown in Section 3.1.2, errors in the input of this equation arise from:

- Sensor measurement noise or sensor measurement errors,
- Measurement or calibration errors of the mounted sensor position,
- Localisation estimation errors.

Consequently, the effects on the map due to the sensor, calibration and the navigation system can be determined from this analysis.

Calculating the sensitivity of the transformation equation yields two distinct and important benefits. Firstly, if the uncertainty of the input parameters is known, then the uncertainty of the output (in this case the map) can be estimated online (during real-time operation). This enables a basic form of fault detection by ensuring that the uncertainty in the map remains below a pre-defined threshold. Secondly, the sensitivity functions can be analysed off-line, in order to highlight likely sources of mapping error due to the geometry of the sensor and platform. This enables the designer to make reasonable decisions about where to focus effort or money to achieve a required accuracy in the navigation frame. For example, the designer can determine whether money should be spent on higher accuracy navigation or sensing equipment, or whether time and effort should be spent on accurately mounting and calibrating the sensing equipment.

Sensitivity

The sensitivity of an equation to variation in the input parameters is given by the partial derivatives of the outputs with respect to the individual input parameters. When calculated analytically, this is the 'direct differentiation method' described in [32]. For a given function $\mathbf{y} = f(\boldsymbol{\rho})$, the sensitivity is given by:

$$S_{\rho_j}^{y_i}(\boldsymbol{\rho}) = \frac{\partial y_i}{\partial \rho_j}, y_i \in \mathbf{y}, \rho_j \in \boldsymbol{\rho} \tag{3.13}$$

where $S_{\rho_j}^{y_i}(\boldsymbol{\rho})$ is the sensitivity of the output y_i with respect to the input parameter ρ_j, and is potentially a function of all of the input parameters. This represents the effect on the output due to an infinitesimal variation in the chosen input parameter.

The meaning of the partial derivatives of the single transformation in Equation 3.9 becomes clearer when the definition of the derivative is considered [33, p100]. With Equation 3.9 represented by $f()$, its derivatives $\frac{\partial \mathbf{p}^j}{\partial \rho}, \rho \in \{\mathbf{\Phi}_i^j, \mathbf{p}_i^j, \mathbf{p}^i\}$ are defined as:

$$\begin{aligned}\frac{\partial \mathbf{p}^j}{\partial \rho} &= \lim_{\rho_{error}\to 0} \frac{f(\rho+\rho_{error})-f(\rho)}{\rho_{error}} \\ \frac{\partial \mathbf{p}^j}{\partial \rho} &= \lim_{\rho_{error}\to 0} \frac{\hat{\mathbf{p}}^j-\mathbf{p}^j}{\rho_{error}}\end{aligned} \tag{3.14}$$

In this notation, the hat above a quantity indicates that it is an estimate of the true quantity, subject to error. The estimate $\hat{\mathbf{p}}^j$ differs from the true point $\mathbf{p}^j$ due to an error in the parameter ρ of size ρ_{error}. So the partial derivatives form an infinitesimal vector, as illustrated in Figure 3.4 for the 2D case. The partial derivatives are the gradients of the multidimensional transformation function. The transformation Equations 3.9, 3.10, 3.11 and 3.12 are non-linear, but for a small ρ_{error}, $\frac{\partial \mathbf{p}^j}{\partial \rho}$ can be thought of as the amplification of the output error due to the input error, or the *sensitivity.* If ρ_{error} is sufficiently small, linearisation is acceptable, and the error vector can be approximated by:

$$\hat{\mathbf{p}}^j - \mathbf{p}^j \approx \frac{\partial \mathbf{p}^j}{\partial \rho} \times \rho_{error} \tag{3.15}$$

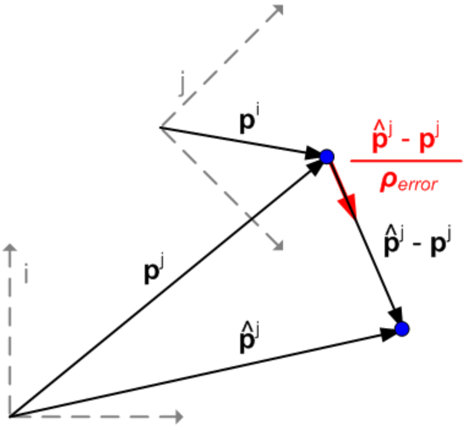


Figure 3.4: The error after transforming a point $\mathbf{p}^i$ from frame i to frame j, shown for the 2D case for simplicity. The vector $\hat{\mathbf{p}}^j$ is the result of an error in one of the parameters that describes the position and rotation of frame i in frame j. The vector interpretation of $\frac{\partial \mathbf{p}^j}{\partial \rho}$ is shown in red.

Online Estimation of Mapping Error

In this Section, the sensitivity of the spatial location of points in the navigation frame $\mathbf{p}^n$ with respect to all of the input parameters to Equation 3.12 is quantified. If the uncertainty of the input parameters is known while the system is running, then the uncertainty in the navigation frame at any instant can be calculated. The calculation of a mean estimate $\hat{\mathbf{p}}^n$ and covariance $\mathbf{\Sigma}_{\mathbf{p}^n}$ of the true location $\mathbf{p}^n$ is derived below. By maintaining the uncertainty of the estimated points in the navigation frame, the accuracy of the map is known at all times. In some applications the estimation of instantaneous uncertainty is sufficient for robustness and unrelated to safety. For example, if the task is to sample and map data from the environment at whatever accuracy is available, then recording the accuracy is *useful*, so that the quality of the information can be assessed, but not necessarily *critical* to mission success. In other applications where the mapped information is to be used online to determine appropriate control actions, such as the predictive control architecture seen in Chapter 2, then estimating the instantaneous uncertainty is critical for robustness and safety. Recall from Chapter 2 that predictive controllers predict the future state of the vehicle by simulating possible control actions over the perceived environment and this may not be capable of handling arbitrarily large uncertainty in the map. In this case, calculation of $\mathbf{\Sigma}_{\mathbf{p}^n}$ enables fault detection in a basic form, where a fault is defined as the instantaneous uncertainty of $\hat{\mathbf{p}}^n$ exceeding some predefined threshold. It also enables the use of policies for handling varying degrees of mapping accuracy, or for manipulating the vehicle position or sensor configuration to improve the accuracy.

The input parameters of Equation 3.12 consist of the spatial sensor data $\mathbf{p}^s$, the location of the sensor in the body frame $\{\mathbf{r}_s^b, \mathbf{\Phi}_s^b\}$, and the navigation solution $\{\mathbf{p}_b^n, \mathbf{\Psi}_b^n\}$. This is a total of 5 input vectors, each containing 3 components (either spatial distances $\{p_x, p_y, p_z\}$ or rotation angles $\{\phi_x, \phi_y, \phi_z\}$), yielding a total of 15 scalar inputs. The single output vector $\mathbf{p}^n$ has three spatial components $\{p_x^n, p_y^n, p_z^n\}$. Consequently there are a total of $3 \times 15 = 45$ sensitivity functions, one for every combination of input to output. These are given by each of the individual terms of the Jacobian matrix for Equation 3.12, shown with sub-matrices for visual separation of the 5 input vectors:

$$\mathbf{J} = \left[\begin{vmatrix} \frac{\partial p_x^n}{\partial p_x^s} & \frac{\partial p_x^n}{\partial p_y^s} & \frac{\partial p_x^n}{\partial p_z^s} \\ \frac{\partial p_y^n}{\partial p_x^s} & \frac{\partial p_y^n}{\partial p_y^s} & \frac{\partial p_y^n}{\partial p_z^s} \\ \frac{\partial p_z^n}{\partial p_x^s} & \frac{\partial p_z^n}{\partial p_y^s} & \frac{\partial p_z^n}{\partial p_z^s} \end{vmatrix} \begin{vmatrix} \frac{\partial p_x^n}{\partial r_{s,x}^b} & \frac{\partial p_x^n}{\partial r_{s,y}^b} & \frac{\partial p_x^n}{\partial r_{s,z}^b} \\ \frac{\partial p_y^n}{\partial r_{s,x}^b} & \frac{\partial p_y^n}{\partial r_{s,y}^b} & \frac{\partial p_y^n}{\partial r_{s,z}^b} \\ \frac{\partial p_z^n}{\partial r_{s,x}^b} & \frac{\partial p_z^n}{\partial r_{s,y}^b} & \frac{\partial p_z^n}{\partial r_{s,z}^b} \end{vmatrix} \begin{vmatrix} \frac{\partial p_x^n}{\partial \phi_{s,x}^b} & \frac{\partial p_x^n}{\partial \phi_{s,y}^b} & \frac{\partial p_x^n}{\partial \phi_{s,z}^b} \\ \frac{\partial p_y^n}{\partial \phi_{s,x}^b} & \frac{\partial p_y^n}{\partial \phi_{s,y}^b} & \frac{\partial p_y^n}{\partial \phi_{s,z}^b} \\ \frac{\partial p_z^n}{\partial \phi_{s,x}^b} & \frac{\partial p_z^n}{\partial \phi_{s,y}^b} & \frac{\partial p_z^n}{\partial \phi_{s,z}^b} \end{vmatrix} \begin{vmatrix} \frac{\partial p_x^n}{\partial p_{b,x}^n} & \frac{\partial p_x^n}{\partial p_{b,y}^n} & \frac{\partial p_x^n}{\partial p_{b,z}^n} \\ \frac{\partial p_y^n}{\partial p_{b,x}^n} & \frac{\partial p_y^n}{\partial p_{b,y}^n} & \frac{\partial p_y^n}{\partial p_{b,z}^n} \\ \frac{\partial p_z^n}{\partial p_{b,x}^n} & \frac{\partial p_z^n}{\partial p_{b,y}^n} & \frac{\partial p_z^n}{\partial p_{b,z}^n} \end{vmatrix} \begin{vmatrix} \frac{\partial p_x^n}{\partial \psi_{b,x}^n} & \frac{\partial p_x^n}{\partial \psi_{b,y}^n} & \frac{\partial p_x^n}{\partial \psi_{b,z}^n} \\ \frac{\partial p_y^n}{\partial \psi_{b,x}^n} & \frac{\partial p_y^n}{\partial \psi_{b,y}^n} & \frac{\partial p_y^n}{\partial \psi_{b,z}^n} \\ \frac{\partial p_z^n}{\partial \psi_{b,x}^n} & \frac{\partial p_z^n}{\partial \psi_{b,y}^n} & \frac{\partial p_z^n}{\partial \psi_{b,z}^n} \end{vmatrix} \right] \tag{3.16}$$

The Jacobian matrix can be expressed in vector form, yielding:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \mathbf{p}^n}{\partial \mathbf{p}^s} & \frac{\partial \mathbf{p}^n}{\partial \mathbf{r}^b} & \frac{\partial \mathbf{p}^n}{\partial \mathbf{\Phi}_s^b} & \frac{\partial \mathbf{p}^n}{\partial \mathbf{p}_b^n} & \frac{\partial \mathbf{p}^n}{\partial \mathbf{\Psi}_b^n} \end{bmatrix} \tag{3.17}$$

Each of the 45 individual functions in the Jacobian matrix can be evaluated by the numerical approximation of the derivative in Equation 3.14, with Equation 3.12 for $f()$:

$$\frac{\partial \mathbf{p}^n}{\partial \rho} \approx \frac{\hat{\mathbf{p}}^n - \mathbf{p}^n}{\rho_{error}}, \ \rho_{error} \approx 0 \tag{3.18}$$

If the sensitivity calculations are done with the numerical derivative of Equation 3.18, then this corresponds to the 'brute force method' in [32]. The problem with this approach is that it is difficult to know what value to choose for ρ_{error} to guarantee the accuracy of the calculation. The analytical derivation of all of the partial derivatives in the Jacobian matrix of Equation 3.17 is presented in Appendix A, and should be used in preference to Equation 3.18 for greater accuracy.

The uncertainty of all of the input parameters can be represented by a 15 × 15 covariance matrix $\mathbf{Q}$. An example of the structure of $\mathbf{Q}$ is given by:

$$\mathbf{Q} = \begin{bmatrix} \boldsymbol{\sigma}^2_{\mathbf{p}^s} & 0 & 0 & 0 & 0 \\ 0 & \boldsymbol{\sigma}^2_{\mathbf{r}^b_s\mathbf{r}^b_s} & \boldsymbol{\sigma}^2_{\mathbf{r}^b_s\boldsymbol{\Phi}^b_s} & 0 & 0 \\ 0 & \boldsymbol{\sigma}^2_{\boldsymbol{\Phi}^b_s\mathbf{r}^b_s} & \boldsymbol{\sigma}^2_{\boldsymbol{\Phi}^b_s\boldsymbol{\Phi}^b_s} & 0 & 0 \\ 0 & 0 & 0 & \boldsymbol{\sigma}^2_{\mathbf{p}^n_b\mathbf{p}^n_b} & \boldsymbol{\sigma}^2_{\mathbf{p}^n_b\boldsymbol{\Psi}^n_b} \\ 0 & 0 & 0 & \boldsymbol{\sigma}^2_{\boldsymbol{\Psi}^n_b\mathbf{p}^n_b} & \boldsymbol{\sigma}^2_{\boldsymbol{\Psi}^n_b\boldsymbol{\Psi}^n_b} \end{bmatrix} \tag{3.19}$$

where the off block-diagonal elements are all zero and each bold term represents a 3×3 sub-matrix. For example:

$$\boldsymbol{\sigma}^2_{\mathbf{r}^b_s\boldsymbol{\Phi}^b_s} = \begin{bmatrix} \sigma^2_{r^b_{s,x}\phi^b_{s,x}} & \sigma^2_{r^b_{s,x}\phi^b_{s,y}} & \sigma^2_{r^b_{s,x}\phi^b_{s,z}} \\ \sigma^2_{r^b_{s,y}\phi^b_{s,x}} & \sigma^2_{r^b_{s,y}\phi^b_{s,y}} & \sigma^2_{r^b_{s,y}\phi^b_{s,z}} \\ \sigma^2_{r^b_{s,z}\phi^b_{s,x}} & \sigma^2_{r^b_{s,z}\phi^b_{s,y}} & \sigma^2_{r^b_{s,z}\phi^b_{s,z}} \end{bmatrix} \tag{3.20}$$

The matrix $\mathbf{Q}$ describes the variance of all of the individual parameters on the diagonal, and the cross-correlation between all of the permutations of parameter pairs on the off-diagonals. In the example above, it is assumed that the sensor information is not affected by the location of the sensor on the platform, nor by the location of the platform in the environment. The mounted sensor location is also uncorrelated to the platform location. However, the translation and rotation errors of the sensor location may be correlated, depending on how they are calculated, and the translation and rotation errors of the vehicle position are usually correlated. The independence assumptions are expressed mathematically by the zero cross-correlation terms between the uncorrelated parameters in $\mathbf{Q}$. This version of $\mathbf{Q}$ is written as an example, and the independence assumptions may be relaxed or tightened further. For example, an auto-calibration method might be used to refine the estimate of the sensor location on the platform, by first allowing the location to be non-constant, then accepting a non-zero cross-correlation between it, the sensor data and the vehicle location. In other words, sensor observations about the world may lead to an improved estimate of the sensor location. If simultaneous localisation and mapping (SLAM) is used ([8, §37] for an overview), then the correlations between the sensor data and the platform pose will likely be non-zero. Both of these examples require some understanding of the structure in the operating environment of the vehicle. In the most general case, where no structural assumptions are made, these correlations are virtually impossible to model due to the complexity of the world. The mathematics in this thesis does not rely on any of the assumptions relating to the structure of $\mathbf{Q}$. It is merely assumed that appropriate values for $\mathbf{Q}$, for a given application, can be determined.

If the uncertainty of all of the inputs is available in the form of $\mathbf{Q}$, then the linearised uncertainty of $\hat{\mathbf{p}}^n$ can be calculated:

$$\mathbf{\Sigma}_{\mathbf{p}^n} = \mathbf{J}\mathbf{Q}\mathbf{J}^T \tag{3.21}$$

This yields a linearised three dimensional Gaussian uncertainty ellipse for the estimate of $\mathbf{p}^n$:

$$\hat{\mathbf{p}}^n \sim \mathcal{N}(\mathbf{p}^n, \mathbf{\Sigma}_{\mathbf{p}^n}) \tag{3.22}$$

Equation 3.22 can be used to quantify the mapping accuracy of a system, given all of the sources of uncertainty, which are normally available.

- The uncertainty due to the sensor $\boldsymbol{\sigma}^2_{\mathbf{p}^s}$ is typically supplied by the sensor model.
- The uncertainty due to sensor misalignment $\boldsymbol{\sigma}^2_{\mathbf{r}^b_s\mathbf{r}^b_s}$, $\boldsymbol{\sigma}^2_{\mathbf{r}^b_s\mathbf{\Phi}^b_s}$ and $\boldsymbol{\sigma}^2_{\mathbf{\Phi}^b_s\mathbf{\Phi}^b_s}$ are given by the results of calibration (as will be shown in Section 4.1.2), or manual estimates of measurement error.
- The navigation uncertainty $\boldsymbol{\sigma}^2_{\mathbf{p}^n_b\mathbf{p}^n_b}$, $\boldsymbol{\sigma}^2_{\mathbf{p}^n_b\mathbf{\Psi}^n_b}$ and $\boldsymbol{\sigma}^2_{\mathbf{\Psi}^n_b\mathbf{\Psi}^n_b}$ is provided by the navigation system.

This can be used to track the mapping accuracy at run-time, to dynamically detect when certain elements of the system are not accurate enough to meet the mapping requirements. For systems that use the map as an integral part of the control loop, a *minimum* requirement for a robust and safe system, is that this equation be used for comparison with some predefined threshold of mapping accuracy *and* a policy must be enacted whenever the accuracy drops below this threshold. In its simplest form the policy can call for a system halt and possibly user intervention. More complex, autonomous approaches may be used to alter the control methodology to cope with the increase in perception uncertainty. In some cases it may be possible to plan the vehicle trajectory or sensor configuration by explicitly considering uncertainty, in order to maintain the requisite level of accuracy. This is the approach taken by Partially Observable Markov Decision Processes (POMDPs) [34], or ad-hoc yet tractable equivalents such as [35] in which the expected future uncertainty is considered as part of the planning process. For perception systems in general, observing and managing uncertainty is a core requirement for robustness and safety. For perception systems that use a mobile platform and an exteroceptive sensor capable of providing complete spatial information, Equation 3.22 is a core component for robustness and safety when coupled with an appropriate policy.

Offline Estimation of Mapping Error

In this section, the coordinate transformation of Equation 3.12 is analysed in order to *generalise* about where the majority of error in $\hat{\mathbf{p}}^n$ is likely to come from. It will be shown that angular errors due to sensor misalignment and navigation uncertainty, are multiplied by a maximum of the distance to the sensor information in the sensor and body frames respectively. The translational errors due to misalignment and navigation uncertainty contribute directly to the final error magnitude with no amplification and are independent of the sensor range.

In the previous Section, the sensitivity of the sensor to navigation transformation of Equation 3.12 was calculated, and each specific sensitivity function can be seen in Appendix A. These functions are useful for computing the uncertainty of the estimate $\hat{\mathbf{p}}^n$ at runtime when all of the input parameters are numerically quantified. For *general* analytical purposes, however, where no particular vehicle and sensor geometry is assumed, they are too highly coupled. Most of the partial derivatives are a function of most of the input parameters:

$$\frac{\partial \mathbf{p}^n}{\partial \rho} = f(\mathbf{p}^s, \mathbf{r}_s^b, \mathbf{\Phi}_s^b, \mathbf{p}_b^n, \mathbf{\Psi}_b^n), \ \ \rho \in \{\mathbf{p}^s, \mathbf{r}_s^b, \mathbf{\Phi}_s^b, \mathbf{p}_b^n, \mathbf{\Psi}_b^n\} \tag{3.23}$$

so it is difficult to *generalise* about the effect of ρ. Appendix A provides the exact parameter lists for every partial derivative term.

Several things are done in this Section for simplification. Firstly, the focus is placed on the magnitude of error sensitivity rather than the direction, and the maximum sensitivity to error is calculated as a 'worst case scenario'. In other words, *how much* can the output be affected, regardless of the direction of the error. Secondly, the error sensitivity of the double transformation of Equation 3.12 is determined by expanding the equation, then considering the rotations by separation into the orthogonal components of Equation 3.6.

The partial derivative terms for the orthogonal rotations in Equation 3.6, when applied to a point $\mathbf{p}$ are given by:

$$\begin{aligned}
\frac{\partial \mathbf{C}_{\phi z}\mathbf{p}}{\partial \phi_z^j} &= \begin{bmatrix} \sin(\phi_z^j)p_x - \cos(\phi_z^j)p_y \\ \cos(\phi_z^j)p_x + \sin(\phi_z^j)p_y \\ 0 \end{bmatrix} \\
\frac{\partial \mathbf{C}_{\phi y}\mathbf{p}}{\partial \phi_y^j} &= \begin{bmatrix} \sin(\phi_y^j)p_x + \cos(\phi_y^j)p_z \\ 0 \\ -\cos(\phi_y^j)p_x + \sin(\phi_y^j)p_z \end{bmatrix} \\
\frac{\partial \mathbf{C}_{\phi x}\mathbf{p}}{\partial \phi_x^j} &= \begin{bmatrix} 0 \\ \sin(\phi_x^j)p_y - \cos(\phi_x^j)p_z \\ \cos(\phi_x^j)p_y + \sin(\phi_x^j)p_z \end{bmatrix}
\end{aligned} \tag{3.24}$$

These terms describe the infinitesimal error vector due to errors in the rotation angle when a point is rotated about a single orthogonal axis. The magnitude of the vectors is given by:

$$
\begin{aligned}
\left|\frac{\partial \mathbf{C}_{\phi z}\mathbf{p}}{\partial \phi_z^j}\right| &= \sqrt{(\sin(\phi_z^j)p_x - \cos(\phi_z^j)p_y)^2 + (\cos(\phi_z^j)p_x + \sin(\phi_z^j)p_y)^2} \\
&= \sqrt{p_x^2 + p_y^2} \\
\left|\frac{\partial \mathbf{C}_{\phi y}\mathbf{p}}{\partial \phi_y^j}\right| &= \sqrt{(\sin(\phi_y^j)p_x + \cos(\phi_y^j)p_z)^2 + (-\cos(\phi_y^j)p_x + \sin(\phi_y^j)p_z)^2} \\
&= \sqrt{p_x^2 + p_z^2} \\
\left|\frac{\partial \mathbf{C}_{\phi x}\mathbf{p}}{\partial \phi_x^j}\right| &= \sqrt{(\sin(\phi_x^j)p_y - \cos(\phi_x^j)p_z)^2 + (\cos(\phi_x^j)p_y + \sin(\phi_x^j)p_z)^2} \\
&= \sqrt{p_y^2 + p_z^2} \qquad (3.25)
\end{aligned}
$$

so a single orthogonal rotation has a sensitivity equal to the distance to the point in the plane of rotation. The magnitude of a vector remains unchanged after rotation [31, p156], so if two points in frame i are rotated to frame j, the magnitude of the vector between those points is also not altered:

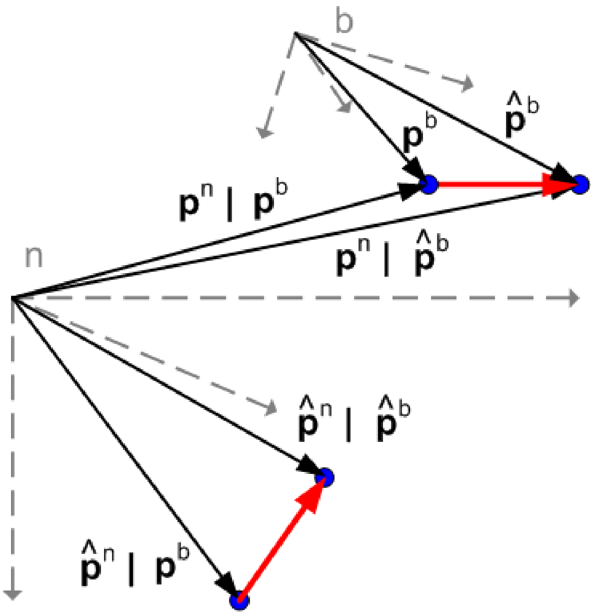

b
$\hat{\mathbf{p}}^b$
$\mathbf{p}^b$
$\mathbf{p}^n \mid \mathbf{p}^b$
n
$\mathbf{p}^n \mid \hat{\mathbf{p}}^b$
$\hat{\mathbf{p}}^n \mid \hat{\mathbf{p}}^b$
$\hat{\mathbf{p}}^n \mid \mathbf{p}^b$

Figure 3.5: This figure illustrates a transformation of a point from the body frame to the navigation frame. An error is already present in the body frame due to the transformation from the sensor frame. This is represented by $\hat{\mathbf{p}}^b$. The *magnitude* of this error is maintained through the next transform from b to n, whether this transform introduces an additional error or not. (The pipe symbol 'given' is borrowed from the probabilistic community)

$$
\begin{aligned}
\mathbf{p}_2^j - \mathbf{p}_1^j &= (\mathbf{C}_i^j \mathbf{p}_2^i + \mathbf{p}_i^j) - (\mathbf{C}_i^j \mathbf{p}_1^i + \mathbf{p}_i^j) \\
&= \mathbf{C}_i^j (\mathbf{p}_2^i - \mathbf{p}_1^i) \\
|\mathbf{p}_2^j - \mathbf{p}_1^j| &= |\mathbf{p}_2^i - \mathbf{p}_1^i|
\end{aligned}
\tag{3.26}
$$

Therefore, when multiple rotations are combined as in Equation 3.7, the *subsequent* rotations do not alter the magnitude of the partial derivative vectors. This is illustrated in Figure 3.5. Recall from Equation 3.7 that the direction cosine matrix $\mathbf{C}_i^j$ is formed by combining the three orthogonal rotations in turn. The input point for each orthogonal rotation is the output of the previous rotation. Therefore all that can be said for a *general* geometric configuration is that the sensitivity of the complete rotation $\mathbf{C}_i^j$ to errors in each of the Euler angles, varies from a maximum of the distance to the original point $\mathbf{p}^i$ to a minimum of complete insensitivity:

$$
\begin{aligned}
min(S_\phi^{\mathbf{p}^j}) &= 0 \\
max(S_\phi^{\mathbf{p}^j}) &= |\mathbf{p}^i|
\end{aligned}
\tag{3.27}
$$

This depends entirely on the specific geometry of the point, the coordinate frames and the set of Euler angles. For a particular orthogonal rotation, if the result of the previous rotation leaves the point directly on the current axis of rotation, then no amount of rotation will alter the coordinate of that point. In this case the sensitivity with respect to the current Euler angle is zero. The sensitivity reaches a maximum when the previous rotation leaves the point such that the vector from the origin to the point is perpendicular to the current axis of rotation.

Equation 3.12 can be expanded:

$$\mathbf{p}^n = \mathbf{C}_b^n \mathbf{C}_s^b \mathbf{p}^s + \mathbf{C}_b^n \mathbf{r}_s^b + \mathbf{p}_b^n \tag{3.28}$$

From this equation and the logic above, it can be seen that the Euler angles of rotation from the body to sensor frame $\mathbf{\Phi}_s^b$, have a maximum sensitivity equal to the sensor range $|\mathbf{p}^s|$. The navigation Euler angles $\mathbf{\Psi}_b^n$ have a maximum sensitivity of the addition of the sensor range and the sensor offset $|\mathbf{p}^s| + |\mathbf{r}_s^b|$.

$$\begin{aligned} min(S_{\mathbf{\Phi}}^{\mathbf{p}^n}) &= 0 \\ max(S_{\mathbf{\Phi}}^{\mathbf{p}^n}) &= |\mathbf{p}^s| \\ min(S_{\mathbf{\Psi}}^{\mathbf{p}^n}) &= 0 \\ max(S_{\mathbf{\Psi}}^{\mathbf{p}^n}) &= |\mathbf{p}^s| + |\mathbf{r}_s^b| \end{aligned} \tag{3.29}$$

If a translational error vector $\mathbf{e}$ is added to $\mathbf{p}^s$, $\mathbf{r}_s^b$ or $\mathbf{p}_b^n$, then by Euler's theorem it is clear that $|\mathbf{e}|$ will propagate without change through the transformation (as in Figure 3.5), resulting in an error vector of the same size in $\mathbf{p}^n$, so the sensitivity of these terms is 1. This result is also verified analytically by calculating the magnitude of the derivatives for Equation 3.12 directly:

$$\left|\frac{\partial \mathbf{p}^n}{\partial p_x^s}\right| = \left|\frac{\partial \mathbf{p}^n}{\partial p_y^s}\right| = \left|\frac{\partial \mathbf{p}^n}{\partial p_z^s}\right| = \left|\frac{\partial \mathbf{p}^n}{\partial r_{s,x}^b}\right| = \left|\frac{\partial \mathbf{p}^n}{\partial r_{s,y}^b}\right| = \left|\frac{\partial \mathbf{p}^n}{\partial r_{s,z}^b}\right| = \left|\frac{\partial \mathbf{p}^n}{\partial p_{b,x}^n}\right| = \left|\frac{\partial \mathbf{p}^n}{\partial p_{b,y}^n}\right| = \left|\frac{\partial \mathbf{p}^n}{\partial p_{b,z}^n}\right| = 1 \tag{3.30}$$

$$S_{\mathbf{p}^s}^{\mathbf{p}^n} = S_{\mathbf{r}_s^b}^{\mathbf{p}^n} = S_{\mathbf{p}_b^n}^{\mathbf{p}^n} = 1 \tag{3.31}$$

The derivation of this result is straightforward, but excessively large to print. It can be obtained by performing the matrix multiplication in Equation 3.28. The magnitude is determined as $\left|\frac{\partial \mathbf{p}^n}{\partial \rho}\right| = \sqrt{(\frac{\partial \mathbf{p}^n_x}{\partial \rho})^2 + (\frac{\partial \mathbf{p}^n_y}{\partial \rho})^2 + (\frac{\partial \mathbf{p}^n_z}{\partial \rho})^2}$. This can then be simplified with the identity: $\cos^2 + \sin^2 = 1$.

In most cases, the translational offset between the sensor and body frame is small when compared to the operational range of the sensor; the term multiplied by the sensor range will typically dominate. For these applications, the general sensitivity analysis can be summarised:

- The magnitude of angular errors is amplified by the sensor range.
- The magnitude of translational errors is propagated with no amplification.

For ground vehicle applications where the sensor range is significantly larger than the translational navigation uncertainty, the sensor offsets, and the distance between the body and sensor frames, the angular errors are dominant. As an example, for a sensor measurement at 60m range and an angular error (in either the navigation solution or the sensor alignment) of 1 degree, this will result in a 'worst case' error of 1.0m in the map, using Equations 3.29 and 3.15.

3.2.2 Sensitivity to Timing Errors

The coordinate transformation in Equation 3.12 has been analysed to determine the sensitivity to spatial measurement errors (errors in the measurement of the sensor mount location and vehicle pose estimation errors from the navigation system). This Equation combines $\{\mathbf{p}^s, \mathbf{r}^b_s, \mathbf{\Phi}^b_s\}$ from the sensor and $\{\mathbf{p}^n_b, \mathbf{\Psi}^n_b\}$ from the navigation system. In almost all *practical* implementations of mapping with a sensor mounted on a mobile platform, these two systems are physically separate. This usually means the systems have separate timing sources, which creates a possible error in *time* that does not feature in the basic transformation equations. This is not the case if the only source of information to the navigation system is the sensor that is also used for

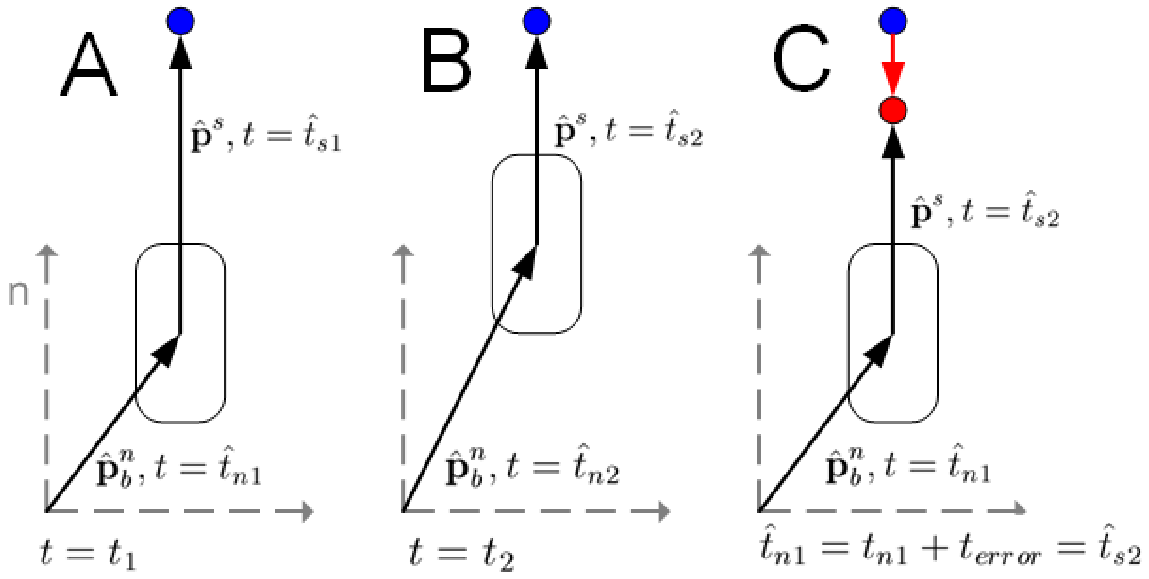

A
$\hat{\mathbf{p}}^s, t = \hat{t}_{s1}$
n
$\hat{\mathbf{p}}^n_b, t = \hat{t}_{n1}$
$t = t_1$
B
$\hat{\mathbf{p}}^s, t = \hat{t}_{s2}$
$\hat{\mathbf{p}}^n_b, t = \hat{t}_{n2}$
$t = t_2$
C
$\hat{\mathbf{p}}^s, t = \hat{t}_{s2}$
$\hat{\mathbf{p}}^n_b, t = \hat{t}_{n1}$
$\hat{t}_{n1} = t_{n1} + t_{error} = \hat{t}_{s2}$

Figure 3.6: The error in $\mathbf{p}^n$ due to a timing error between the sensor and navigation information sources. In this figure, the vehicle is represented by the rectangles, the blue circles represent the true location of a feature in the environment and the red circle represents the erroneous perceived location. Figure A shows the *true* geometry of the system at t_1. Figure B shows the *true* geometry at some later time t_2, after the vehicle has moved. Figure C shows the *perceived* geometry, when inaccurate estimates of time are used. In Figure C, the navigation solution from A is combined with the sensor observation from B, because the time-stamps on that pair of data are incorrectly measured as equal. $\hat{t}_{si}$ and $\hat{t}_{ni}$ denote the estimated sensor and navigation time-stamps for measurements that were actually taken at time t_i. Between $t1$ and $t2$ the vehicle moves forward, so the sensor range to the fixed target decreases. In this example, the mismatch in timing creates the false perception that the object in the map is nearer than it really is.

mapping. For example, a SLAM navigation application that uses 2D scan matching from the sensor and no other information will have completely simultaneous sensing, navigation and mapping. If any source of information with a separate timing system is used to aid in navigation, then errors in timing between the sensor used for mapping and the sensor used for navigation can exist. This section analyses the sensitivity of $\mathbf{p}^n$ to timing errors, by first extending the transformation equations to include the effect of the different time sources, then to analyse the partial derivatives with respect to timing error.

Because we are interested in the effect of timing on the *spatial* component of the data, a constant time offset of the whole system has no effect. For example, if all of the

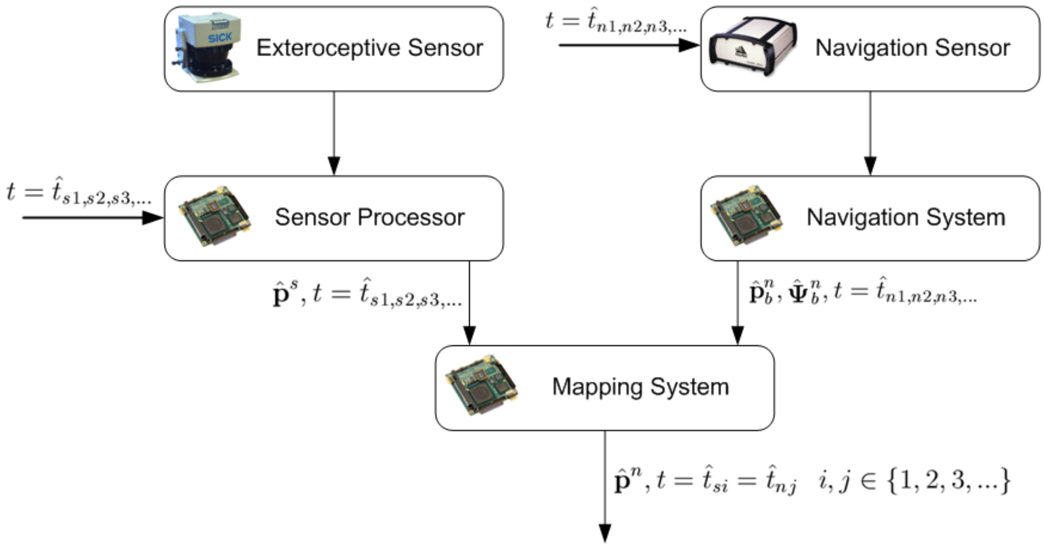

Exteroceptive Sensor
$t = \hat{t}_{n1,n2,n3,...}$
Navigation Sensor
$t = \hat{t}_{s1,s2,s3,...}$
Sensor Processor
Navigation System
$\hat{\mathbf{p}}^s, t = \hat{t}_{s1,s2,s3,...}$
$\hat{\mathbf{p}}^n_b, \hat{\mathbf{\Psi}}^n_b, t = \hat{t}_{n1,n2,n3,...}$
Mapping System
$\hat{\mathbf{p}}^n, t = \hat{t}_{si} = \hat{t}_{nj} \quad i, j \in \{1, 2, 3, ...\}$

Figure 3.7: Timing sources for a typical implementation of a mapping system, using a 2D range scanning laser and a GPS based navigation system. In this case, the laser information is time-stamped by the sensor processor, whereas the GPS provides its own highly accurate timing signal. The mapping system may be running with a third timing source, but it uses the time-stamps provided by the other systems. Other configurations are also possible. $\hat{t}_{si}$ and $\hat{t}_{ni}$ denote the estimated sensor and navigation time-stamps for measurements that were actually taken at time t_i.

system clocks are out by a week, then the whole map will have an error in time, but the spatial integrity will be preserved. System timing errors only cause a spatial error in the map when different sources of information required for the transformation in Equation 3.12 are time-stamped with different errors with respect to the true time. This is illustrated in Figure 3.6. Even if the whole system has a varying timing error, it will not cause spatial errors if the offset is the same for all timing sources. If the implementation separates the sensor and navigation sub-systems and each has a separate time source, as shown in Figure 3.7, then a spatial error will be caused whenever the sensor timing error differs from the navigation timing error. A sensor measurement that was made at time t_s has an estimated time-stamp denoted by $\hat{t}_s$. Similarly, $\hat{t}_n$ denotes the estimate of the true time t_n, corresponding to the output of the navigation system. If there is an error in the sensor timing system, then $\hat{t}_s$ will differ from t_s and if there is an error in the navigation timing system, then $\hat{t}_n$ will differ from t_n. This difference of timing errors is denoted t_e:

$$\begin{aligned} t_e &= (\hat{t}_s - t_s) - (\hat{t}_n - t_n) \\ t_e &= (\hat{t}_s - \hat{t}_n) - (t_s - t_n) \end{aligned} \tag{3.32}$$

When using Equation 3.12 to combine the sensor and navigation information, the matching synchronous data are paired together by selecting equal estimated time-stamps. With $\hat{t}_s = \hat{t}_n$ applied to Equation 3.32:

$$t_e = t_n - t_s \tag{3.33}$$

The timing difference error t_e is the difference between the true sample times of sensor and navigation information when the data are transformed to the map. This quantity is not measurable in typical real-world systems, however the sensitivity of the map with respect to t_e is analytically calculable. Calculation of this sensitivity allows the designer to specify bounds on the timing sub-systems, given a desired mapping accuracy.

For a particular timing error, the faster the vehicle is moving (translational and rotational), the worse the spatial error will be, as can be seen in Figure 3.6. The dependence of the mapping error on the vehicle velocity is therefore intuitively clear. In fact, the effect of timing errors can be thought of in terms of the sensitivity due to navigation errors. The amount that the vehicle moves during the time difference corresponds directly to a spatial error in the vehicle pose $\mathbf{p}_b^n$ and $\mathbf{\Psi}_b^n$ with respect to the sensor data. This in turn can be added to any errors already present in these navigation estimates $\hat{\mathbf{p}}_b^n$ and $\hat{\mathbf{\Psi}}_b^n$, and the conclusions about the sensitivity of $\mathbf{p}^n$ from Section 3.2.1 can be applied.

This is shown analytically by applying the chain rule [33, p794] to Equation 3.12, reproduced below and substituted by f():

$$\begin{aligned}\mathbf{p}^n &= \mathbf{C}_b^n(\mathbf{C}_s^b\mathbf{p}^s + \mathbf{r}_s^b) + \mathbf{p}_b^n \\ \mathbf{p}^n &= f(\mathbf{p}^s, \mathbf{r}_s^b, \mathbf{\Phi}_s^b, \mathbf{p}_b^n, \mathbf{\Psi}_b^n) \qquad (3.34)\end{aligned}$$

Considering the practical implementation of this equation to process sensor data for a single chosen time-stamp, $\mathbf{p}_b^n$ and $\mathbf{\Psi}_b^n$ become a function of the time synchronisation error t_e:

$$\mathbf{p}^n = f(\mathbf{p}^s, \mathbf{r}_s^b, \mathbf{\Phi}_s^b, \mathbf{p}_b^n(t_e), \mathbf{\Psi}_b^n(t_e)) \qquad (3.35)$$

and from the chain rule for partial differential equations, this yields:

$$\begin{aligned}\frac{\partial \mathbf{p}^n}{\partial t_e} &= \frac{\partial \mathbf{p}^n}{\partial \mathbf{p}_b^n}\frac{\partial \mathbf{p}_b^n}{\partial t_e} + \frac{\partial \mathbf{p}^n}{\partial \mathbf{\Psi}_b^n}\frac{\partial \mathbf{\Psi}_b^n}{\partial t_e} \\ \frac{\partial \mathbf{p}^n}{\partial t_e} &= \frac{\partial \mathbf{p}^n}{\partial \mathbf{p}_b^n}\dot{\mathbf{p}}_b^n + \frac{\partial \mathbf{p}^n}{\partial \mathbf{\Psi}_b^n}\dot{\mathbf{\Psi}}_b^n \qquad (3.36)\end{aligned}$$

The timing error causes an absolute rotation and translation error, due to the motion of the vehicle during the erroneous period of time. The sensitivity of the map with respect to these absolute errors was calculated in Section 3.2.1, so the conclusions about the worst case sensitivity from Equations 3.29 and 3.30 can be used to provide bounds for the timing error sensitivity. Combining these equations with Equation 3.36, the minimum and maximum sensitivities with respect to t_e are given by:

$$\begin{aligned}min(S_{t_e}^{\mathbf{p}^n}) &= |\dot{\mathbf{p}}_b^n| \\ max(S_{t_e}^{\mathbf{p}^n}) &= |\dot{\mathbf{p}}_b^n| + |\dot{\mathbf{\Psi}}_b^n| \times (|\mathbf{p}^s| + |\mathbf{r}_s^b|) \qquad (3.37)\end{aligned}$$

The timing difference error can be incorporated into the real-time calculation of mapping uncertainty by extending the Jacobian matrix of Equation 3.17 to include the partial derivatives with respect to t_e:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \mathbf{p}^n}{\partial \mathbf{p}^s} & \frac{\partial \mathbf{p}^n}{\partial \mathbf{r}^b} & \frac{\partial \mathbf{p}^n}{\partial \mathbf{\Phi}_s^b} & \frac{\partial \mathbf{p}^n}{\partial \mathbf{p}_b^n} & \frac{\partial \mathbf{p}^n}{\partial \mathbf{\Psi}_b^n} & \frac{\partial \mathbf{p}^n}{\partial t_e} \end{bmatrix} \tag{3.38}$$

This represents a 3 × 16 matrix. The uncertainty of t_e is then added to the system uncertainty matrix $\mathbf{Q}$ (now a 16 × 16 matrix):

$$\mathbf{Q} = \begin{bmatrix} \boldsymbol{\sigma}^2_{\mathbf{p}^s} & 0 & 0 & 0 & 0 & 0 \\ 0 & \boldsymbol{\sigma}^2_{\mathbf{r}_s^b\mathbf{r}_s^b} & \boldsymbol{\sigma}^2_{\mathbf{r}_s^b\mathbf{\Phi}_s^b} & 0 & 0 & 0 \\ 0 & \boldsymbol{\sigma}^2_{\mathbf{\Phi}_s^b\mathbf{r}_s^b} & \boldsymbol{\sigma}^2_{\mathbf{\Phi}_s^b\mathbf{\Phi}_s^b} & 0 & 0 & 0 \\ 0 & 0 & 0 & \boldsymbol{\sigma}^2_{\mathbf{p}_b^n\mathbf{p}_b^n} & \boldsymbol{\sigma}^2_{\mathbf{p}_b^n\mathbf{\Psi}_b^n} & 0 \\ 0 & 0 & 0 & \boldsymbol{\sigma}^2_{\mathbf{\Psi}_b^n\mathbf{p}_b^n} & \boldsymbol{\sigma}^2_{\mathbf{\Psi}_b^n\mathbf{\Psi}_b^n} & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma^2_{t_e} \end{bmatrix} \tag{3.39}$$

With $\mathbf{J}$ and $\mathbf{Q}$ taken from Equation 3.38 and 3.39 respectively, Equation 3.21 now provides the uncertainty of a point in n, including the effect of the timing difference error t_e. The individual partial derivative terms are shown in Appendix A.

When the translational offset between the sensor and body frame is small in comparison to the operational range of the sensor, the temporal sensitivity analysis can be summarised:

> The magnitude of timing error is amplified by the vehicle's translational velocity plus the rotational velocity multiplied by the sensor range.

As an example, for a sensor measurement at 60m range an angular velocity of 10 degrees per second and a 10ms timing error, this will result in a 'worst case' error of 0.10m in the map, using Equations 3.37 and 3.15.

3.3 Empirical Error Analysis

In Section 3.1 the requisite coordinate transformations for exteroceptive sensor mapping with a mobile platform were given and in Section 3.2 the sensitivity of these transformations with respect to the inputs was determined. This enabled off-line calculation of the expected mapping error for a given system configuration, and online estimation of mapping uncertainty by using real-time estimates of the uncertainty of the individual inputs. In this section, real data from a UGV system with an exteroceptive sensor are used, and experiments are done to emphasise the effect of individual sources of error. This serves to validate the analysis in previous sections, and to illustrate the nature of the errors for a typical sensor configuration.

Recall that the only assumption made in this chapter is that the exteroceptive sensors supply complete three dimensional spatial information. Provided this assumption is met, the equations and analysis given so far are application independent; they are not affected by the particular choice of sensor, mobile platform, environment or mapping algorithm. In this section, real exteroceptive sensor data from a mobile platform are used, so the figures and discussions here are specific to the configuration of this platform. This section shows that the chosen configuration conforms to the earlier mathematical predictions about error sensitivity, and additionally it provides a qualitative insight into the nature of the errors that arise in a real system. Three dimensional point cloud representations are used exclusively to visualise the mapped sensor data. As a result, individual sensor samples remain separate after the transformation to the navigation frame. Therefore, no assumptions are made about the mapping algorithm or filtration method. For example, filtration methods such as elevation maps or occupancy grids act to smooth the mapped data, potentially obscuring the interpretation.

3.3.1 Experimental Configuration

The CAS Outdoor Research Demonstrator platform is used as the mobile data gathering platform for the experiments in this section. As shown in Figure 3.8, data are

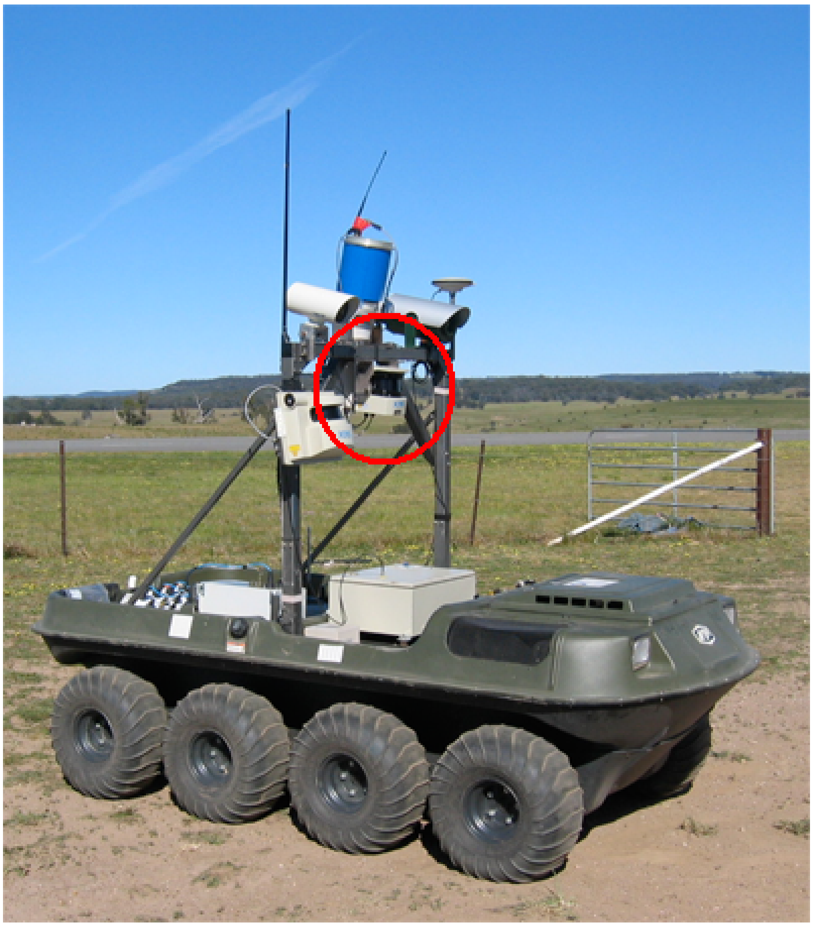


Figure 3.8: The CAS outdoor research demonstrator platform, with multiple sensors attached. The circled SICK LMS291 2D laser range scanner is used as the source of data for the this section. As shown, the laser is mounted at the top centre of the platform, looking forwards. The laser is pitched down 8 degrees from horizontal, and is mounted to align to the frame with zero roll and yaw.

selected from a single two dimensional laser range scanner (SICK-LMS291), mounted at the top centre position of the sensor frame. The laser is positioned with close to zero yaw and roll angle with respect to the frame, and has approximately 8 degrees of pitch down from horizontal. As the vehicle moves through the environment, it acts as a 'push broom' scanner, successively building up the profile of the terrain in front of the vehicle. This is illustrated in Figure 3.9. The sensor, body and navigation frames s, b and n are positioned as shown earlier in Figures 3.1 and 3.3.

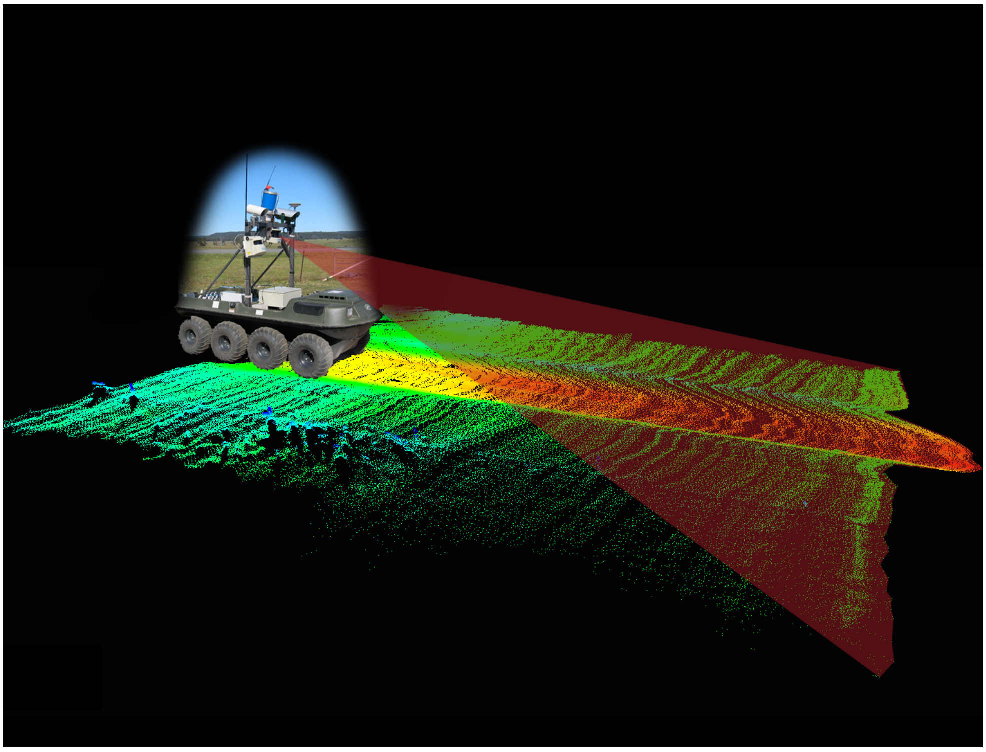


Figure 3.9: Building a terrain map with a single SICK LMS-291 laser. The laser is mounted at the top centre section of the sensor frame, and is angled down by 8 degrees from horizontal. In this 'push-broom' configuration, the vehicle scans the terrain as it moves, successively building the terrain map.

The sources of information are as follows:

- The SICK-LMS291 laser scanner provides range and bearing information, which is converted from polar coordinates to the cartesian values of $\mathbf{p}^s$ as per Equation 3.1 and Figure 3.2.
- The pose of the laser in the body frame $\{\mathbf{r}_s^b, \mathbf{\Phi}_s^b\}$ is calculated by the extrinsic calibration method, that is presented later in Chapter 4.
- The navigation solution $\{\hat{\mathbf{p}}_b^n, \hat{\mathbf{\Psi}}_b^n\}$ is provided by a Novatel DGPS/GPS/INS system, employing a Novatel ProPak-G2plus GPS receiver and a Honeywell HG1700 AG17 inertial measurement unit (IMU).
- Timing is provided by computers that are synchronised using the Network Time Protocol (NTP) [36, 37], which is described in more detail in Chapter 4.

To examine the effect of individual errors in isolation, it is necessary to minimise *all* errors, so that they can be re-introduced independently. In this section, the navigation system errors are minimised by adding DGPS infrastructure in the region of the experiment. With the differential corrections, the navigation solution error is significantly smaller than the individual errors being tested. The experiments that require DGPS are over a smaller time-scale and distance than typical outdoor autonomous operations. For large scale operations, this solution is not always applicable due to the infrastructure requirements. In Chapter 4 methods to manage the errors when DGPS is not permanently available are discussed and the techniques that are required here to minimise sensor misalignment and the timing difference t_e are presented. For this section, it will suffice to say that these errors have been reduced by enough, so that mapping errors seen in the visualisations are almost entirely due to the particular input error under examination. The purpose of this section is to illustrate the effect of the individual errors, and information about the error minimisation methods can be found in the following chapters.

3.3.2 Sensor Misalignment

To show the effect of sensor misalignment, an artificial environment is constructed by placing a single vertical pole on a relatively flat surface. The vehicle gathers sensor and navigation information from two straight passes through the environment, and the sensor information is transformed into the navigation frame using Equation 3.12. Only one set of sensor and navigation data is acquired, but the transformation is performed twice: once using the ‘correct’ sensor pose $\{\mathbf{r}_s^b, \mathbf{\Phi}_s^b\}$, and a second time using an incorrect sensor pose, obtained by adding a two degree error to the yaw angle $\phi_{s,z}^b$. The effect of the yaw angle error is illustrated in Figure 3.10 and Figure 3.11 shows this effect with respect to the two pass trajectory used in this experiment.

The point cloud produced by the ‘correct’ transformation is shown in Figure 3.12. The pole appears vertical as in the real environment. With two degrees of error added to $\phi_{s,z}^b$, the data are processed again. Figure 3.13 shows the point cloud after

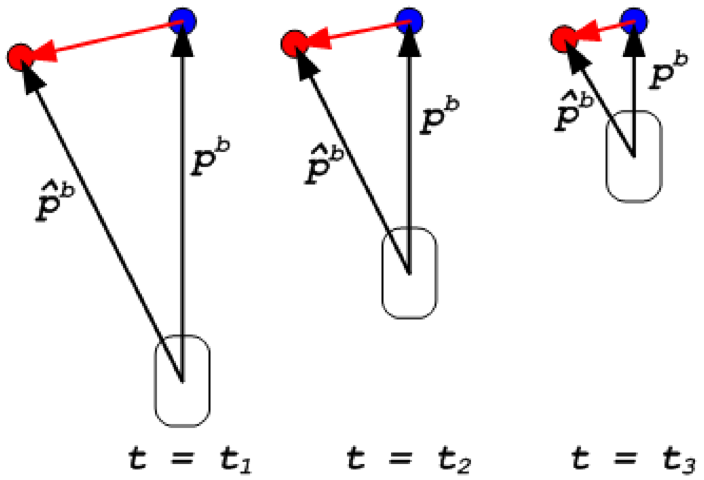


Figure 3.10: An illustration of the effect of an error in the yaw angle of the laser with respect to the platform $\phi^b_{s,z}$. The blue circle represents a vertical pole at a fixed location. The red circles represent the apparent position of the pole in the body frame due to the error. The vectors $\mathbf{p}^b$ and $\hat{\mathbf{p}}^b$ are separated by the yaw angle error, so the effect on the map is reduced as the vehicle approaches the pole.

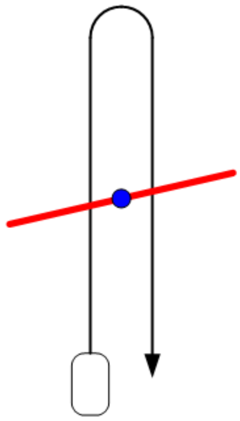


Figure 3.11: The effect of an error in the yaw angle of the laser with respect to the platform $\phi^b_{s,z}$. The trajectory of the vehicle around the pole is shown in black. The resulting set of errors in the navigation frame is shown (exaggerated) in red. Refer to Figure 3.10 for more detail.

the vehicle has passed the pole once (the first half of the data-set is processed only), and now the pole appears to be leaning. The complete data-set in which the vehicle passes the pole twice is shown in Figure 3.14. Two separate poles are visible and each one is angled to join at the top. This is precisely the phenomenon described by the two dimensional illustrations in Figures 3.10 and 3.11. Recall from Section 3.2 that in general, angular errors such as $\phi^b_{s,z}$ are magnified by a maximum of the

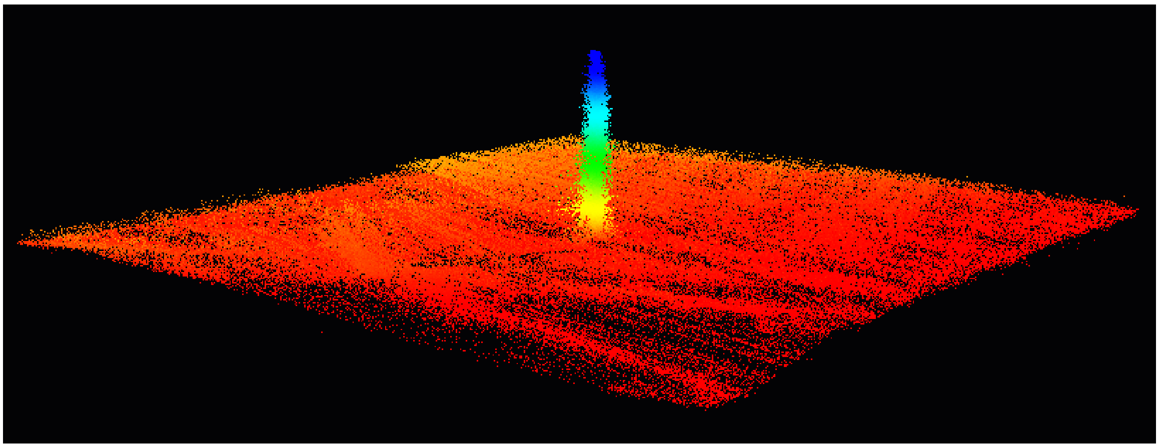


Figure 3.12: A three dimensional point cloud visualisation of laser data, coloured by elevation, taken from an environment with a vertical pole on flat ground. The laser data are transformed to the navigation frame (seen here) by using Equation 3.12, with optimised values for the sensor pose $\{\mathbf{r}_s^b, \mathbf{\Phi}_s^b\}$.

sensor range. Due to the 'push-broom' scanning configuration, the bottom of the pole is scanned from a longer range then the top of the pole, so the error decreases from a maximum at the base to a minimum at the top. As the sensor misalignment error is constant and the data were obtained with a relatively smooth trajectory, the mapping error also appears non-random in nature. Although the pole is leaning, it still contains significant structure. This structure will be used in Chapter 4 to reduce the systematic alignment error.

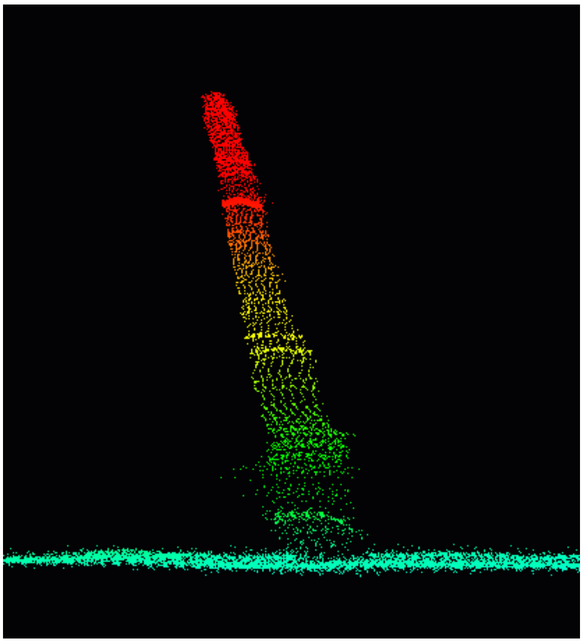


Figure 3.13: A three dimensional point cloud visualisation of laser data, coloured by elevation, taken from the same environment as Figure 3.12. An error of two degrees is added to the laser yaw $\phi^b_{s,z}$. The vehicle passes the pole *once*, causing the incorrect appearance of a single leaning pole.

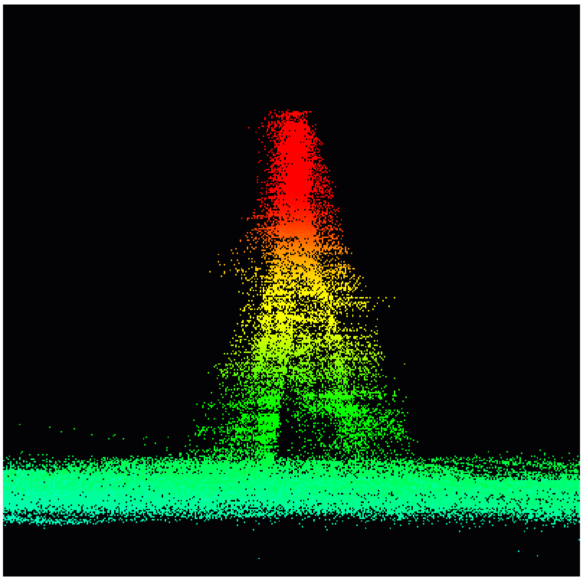


Figure 3.14: A three dimensional point cloud visualisation of laser data, coloured by elevation, taken from the same environment as Figure 3.12. An error of two degrees is added to the laser yaw $\phi^b_{s,z}$. The vehicle passes the pole *twice* from opposite directions, causing the incorrect appearance of two separate leaning poles.

3.3.3 Navigation Errors

This section shows the effect of typical navigation solution errors on the mapping of sensor data. The type of errors that can be expected depends on the type of navigation system that is employed. It is assumed that the navigation system employs some form of Kalman filter [7], [38] to fuse multiple sources of data, to provide the minimum mean squared error (MMSE) estimate of the vehicle pose. The MMSE estimate can then be incorporated directly into the linearised error model that was presented in Section 3.2. In the Kalman filtration framework, a process model is available to predict how the vehicle will move in the absence of any sensory information, and an observation model is used to apply relevant sources of sensor information to the filter. The uncertainty increases each time a prediction is made, and is reduced with each observation. The amount by which the uncertainty is reduced depends on the certainty of the observation with respect to the certainty of the current pose estimate.

In navigation systems, the observations are not necessarily made at regular intervals, and they often vary in certainty over time. In GPS based systems, environmental factors (such as occlusion, multipath and atmospheric interactions [9, §5]) affect both the availability and quality of information. In SLAM based systems (with or without GPS), the availability and quality of information depends on the location of the vehicle within the environment, because of the need to observe features. Navigation errors have an impact over a wide range of time scales from variations that occur in the order of seconds, to errors that are correlated in the short term and can only be detected over longer durations, in the order of tens of minutes to hours. For example, GPS multipath errors are caused when the environment adjacent to the GPS receiver reflects the signal, causing an ambiguity in the range estimate to the satellite [9]. This source of error can vary rapidly (in the order of seconds) as the vehicle moves past reflective structures or surfaces. Longer term errors (in the order of tens of minutes) are typified by GPS satellite observations, which have a source of error due to the interaction of the atmosphere with the signal propagation speed [9]. This error depends on the constellation of satellites in view of the vehicle and on the atmospheric conditions between the vehicle and these satellites.

Due to the fluctuating quality of the sources of navigation information, the uncertainty of the filtered navigation solution changes over time, resulting in a non-constant impact on the integrity of the map. This section is divided into short term effects due to rapid fluctuations of the navigation uncertainty with respect to the time taken to map a region of the environment, and long term effects that are only detectable over longer mapping durations.

Long Term Navigation Errors

Global positioning systems are often used as one of the sensors in outdoor ground vehicle navigation systems. [8, §20.5]. The GPS error due to environmental conditions varies over a time scale in the order of tens of minutes, so the effect on the map can only be seen over an equivalent amount of time. This error can be reduced significantly when differential corrections are applied (particularly when the vehicle is near the base station), because this enables the effects of atmospheric conditions and constellation geometry to be compensated for [9, §9.1]. However, this is not practical for arbitrarily large-scale outdoor autonomy, due to the infrastructure requirements. Differential corrections are not used for the navigation solutions for the data in this section.

To show the long term effects of navigation bias, a data-set is gathered by driving the platform one way along a road for approximately half an hour. It is then driven back along the same road to the starting position. Figures 3.15 and 3.16 show two clipped sections of the whole map. The lower half of the figure shows a section of road that was recorded on the outward journey, and the upper half shows the adjacent section of road recorded about one hour later on the return trip. If any section of the map from a brief temporal window is viewed (such as either the top or bottom half of Figure 3.15), the road appears to be mapped smoothly, due to the short term correlation of the error. However, the effect of the navigation bias is eventually revealed, because position estimates are uncorrelated when separated by enough time. Once the vehicle has driven back along the road, the two co-located sections of the map are separated by a large enough amount of time to reveal the navigation error.

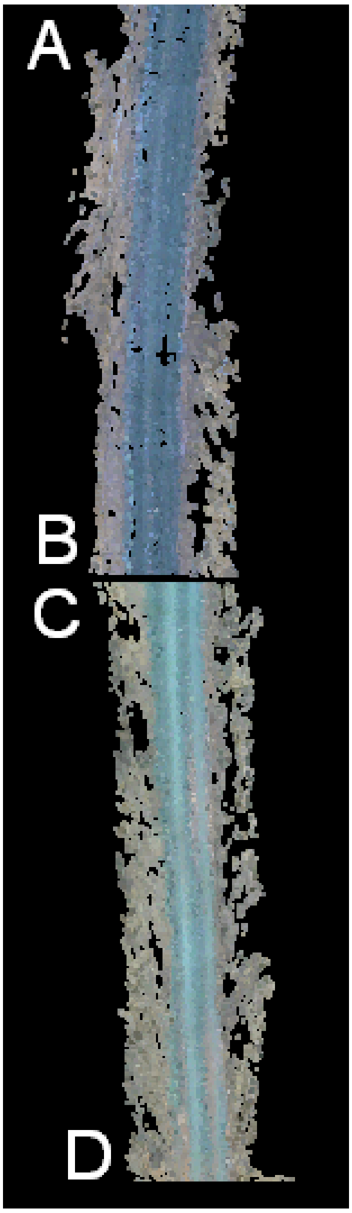


Figure 3.15: A three dimensional laser point cloud, viewed from above. (For human interpretation, colour is added by fusing local colour camera information at run-time using [39]). Data was collected as the robotic platform traveled approximately 2km along a road and then returned along the same route. The section ABCD marks one continuous section of road. Because the vehicle traveled in both directions, data pertaining to ABCD are gathered twice. In this figure, DC is produced from the data obtained on the way out. AB is produced from the data obtained on the return journey, one hour later. Although the map appears locally consistent in the short term, the long term error is seen clearly as the two maps do not meet precisely. This error is due to GPS bias in the navigation system. Figure 3.16 shows the same data from a low altitude perspective.

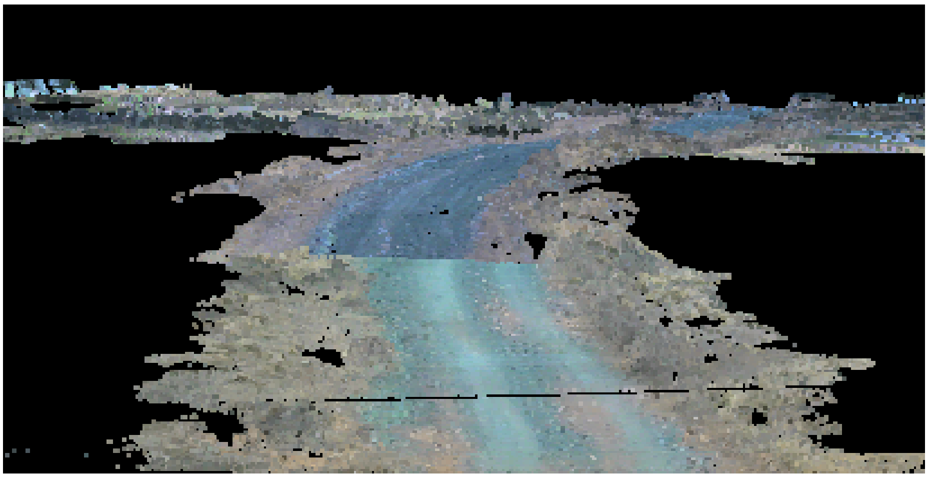


Figure 3.16: A three dimensional laser point cloud, viewed from a low altitude perspective. Figure 3.15 shows the same data, viewed from above.

Short Term Navigation Errors and Corrections

The fluctuation of navigation uncertainty can be rapid in comparison to the time taken for a mobile platform to pass through the environment. If highly certain sensor information suddenly becomes available to the navigation system (when a DGPS base station suddenly comes into transmission range, or a new SLAM feature is observed), then the optimal estimate of the vehicle pose may have a sudden correctional jump. This will cause a corresponding rift of a similar magnitude in the mapped data. Alternatively, if reliable navigation sensor information is unavailable for a period of time, the uncertainty can grow rapidly, as the system relies on the process model rather than the sensors. Both of these scenarios cause a rapid change in the navigation uncertainty, which may cause a reduction of integrity in the corresponding section of the map. As described in Section 3.2.1, the uncertainty of the map can be calculated at run-time, provided the navigation system calculates the uncertainty of the pose estimates.

To illustrate the effect of a fluctuating navigation uncertainty on mapped data, the vehicle is driven repeatedly through the artificial environment with a single vertical pole. This is repeated until the navigation system reports a significant fluctuation of uncertainty over a short time scale; the uncertainty must fluctuate during the time

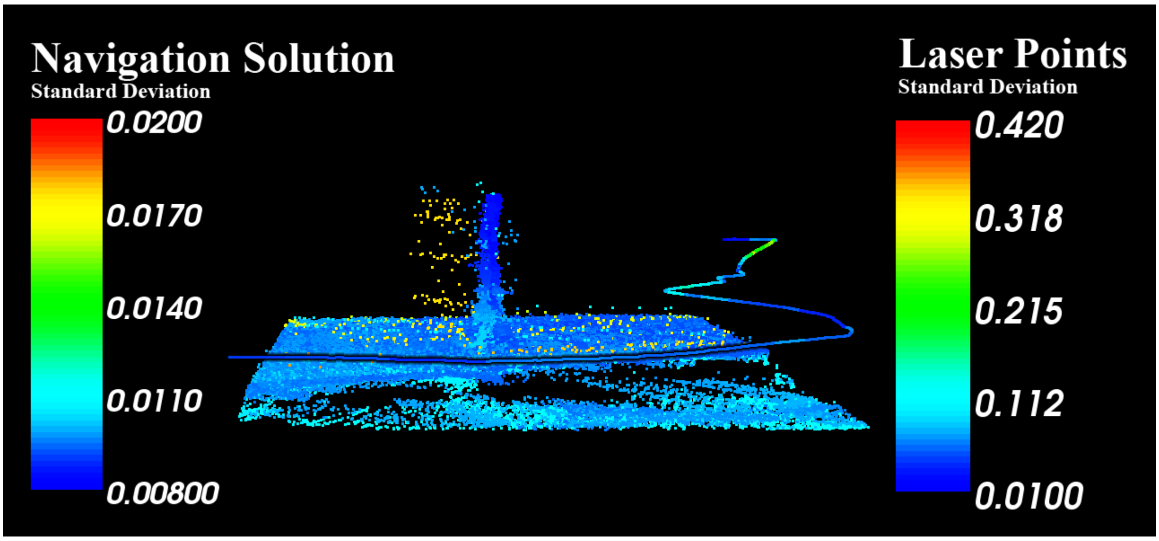


Figure 3.17: A map of the environment and the trajectory of the vehicle that produced it. The navigation system reports a variable uncertainty during the time taken to scan the environment. This causes a variation in the uncertainty of the mapped data. The figure illustrates that most of the erroneous points are known at run-time to have a higher uncertainty. The uncertainty in this figure does *not* take the timing error into consideration. This is done to prove that the uncertainty in this point cloud is almost entirely due to variations in the navigation uncertainty.

taken to pass through the environment once. The navigation data and the laser data are then extracted from this single pass and the laser data are transformed to the navigation frame. The resulting three dimensional point cloud and the estimated trajectory of the platform that produced it can be seen in Figure 3.17. The changes in navigation uncertainty have a direct impact on the uncertainty of points within the point cloud. The figure clearly shows that the erroneous points have a significantly higher uncertainty on average than the valid ones. The uncertainty of the points in this figure were produced explicitly *without* consideration of the effect of timing errors; Equations 3.17 and 3.19 were used for $\mathbf{J}$ and $\mathbf{Q}$ respectively. This is done to *prove* that for this point cloud, the navigation uncertainty alone is the cause of the highly uncertain erroneous points. If the timing uncertainty were to be included by using Equations 3.38 and 3.39 for $\mathbf{J}$ and $\mathbf{Q}$, it would not be possible to distinguish the source of the final point cloud uncertainty.

3.3.4 Timing Errors

To show the effect of the timing difference error t_e, the data-set containing the pole is used again. In this section, the data are processed twice: once with the complete transformation error model from Equations 3.38 and 3.39 for $\mathbf{J}$ and $\mathbf{Q}$ respectively, and a second time with the error model that omits the partial derivatives with respect to t_e. Because the same raw data are used in each case, the mean locations of the points in the point cloud is identical when processed with either of the error models. However, the *uncertainty* in the latter case is altered because it includes the additional uncertainty caused by t_e.

The entire data-set is viewed from above in Figure 3.18, without consideration of t_e, and with t_e incorporated in Figure 3.19. From this view-point, the most salient features that can be seen are the surrounding buildings and fence-lines. The entire scene occupies a square of approximately 100m sides. Figure 3.18 shows that the uncertainty increases with range, as expected from Section 3.2.1. The vehicle moved through the centre region of the data, so all of the outer points were observed at a longer sensor range. The solid walls and thin fence-lines appear over 2m thick in the data-set, despite an apparent uncertainty of only 19cm (largest orthogonal standard deviation). If there were no error, they should appear collinear, so this indicates that the sources of error are not modeled completely. With the additional effect of t_e included, Figure 3.19 shows a clear improvement in the calculation of uncertainty. Now that all of the sources of error have been included in the model, the thin walls and fence-lines can be seen amongst the noise. On average, the points that are further away from the line have a higher calculated uncertainty. Furthermore, the 1m standard deviation of the erroneous points is a reasonable reflection of the magnitude of the errors. The difference in these two figures clearly illustrates the effect that the timing error has on this data-set.

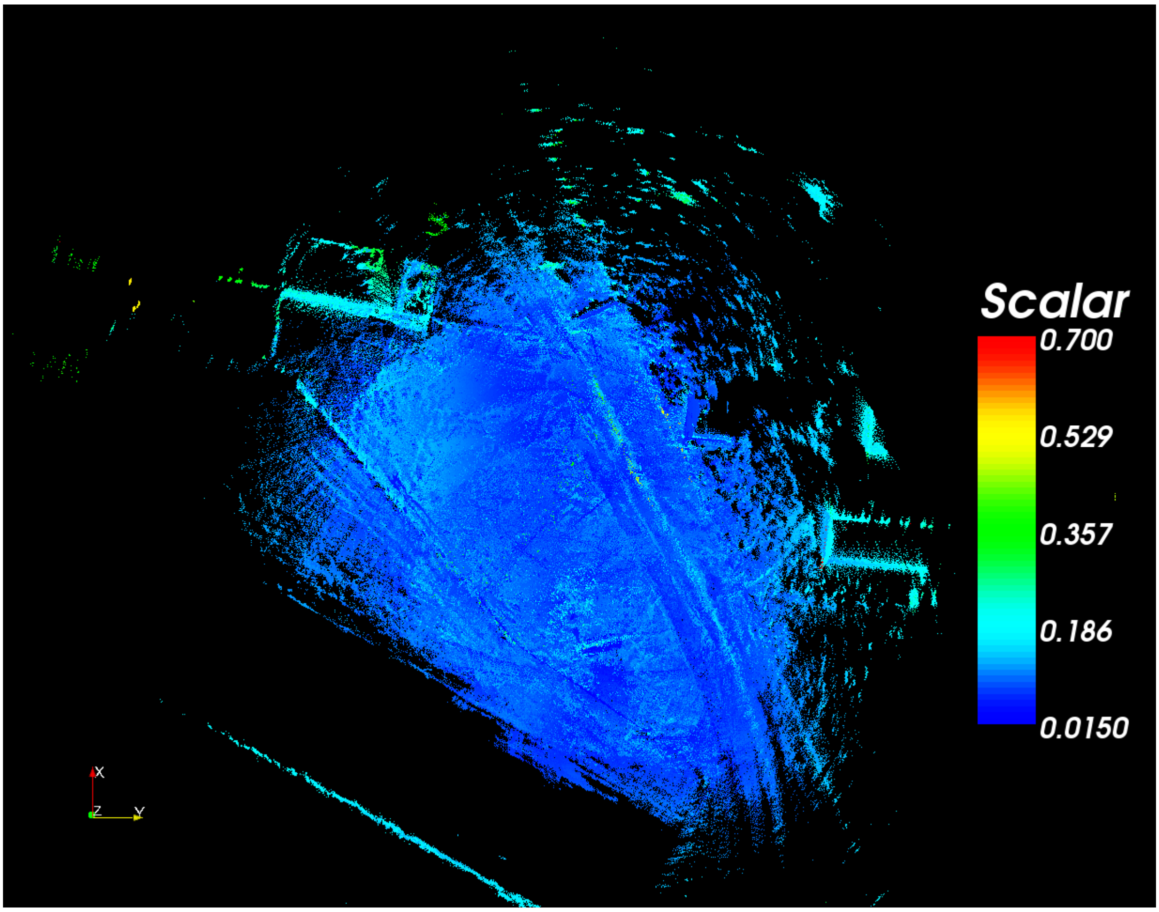


Figure 3.18: A three dimensional point cloud viewed from above. The points are coloured by their positional uncertainty (largest orthogonal standard deviation), but this is calculated without any consideration of the timing difference error t_e between the navigation and laser sensor information. The outer points have higher uncertainty due to the larger sensor range. The points that represent the walls form a thick cloud, which is not consistent with their relatively small uncertainty.

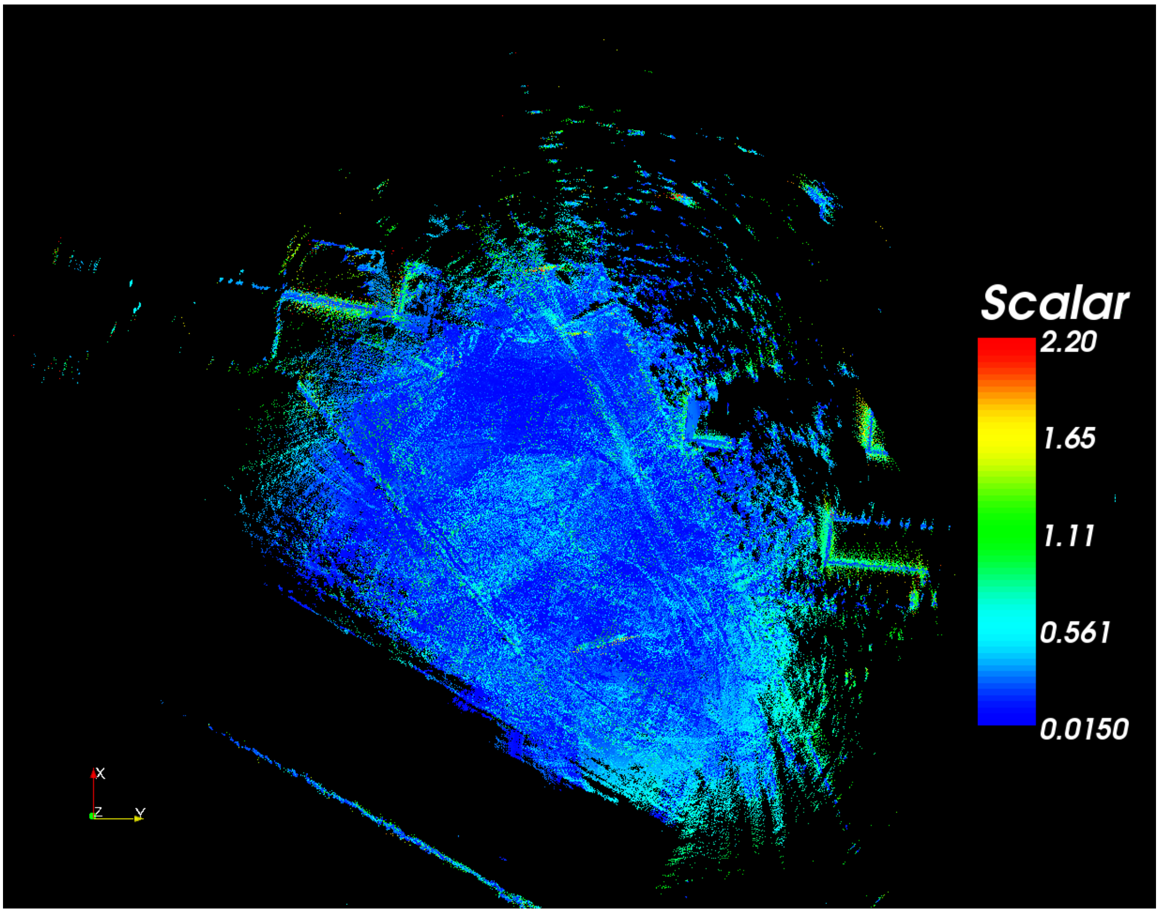


Figure 3.19: A three dimensional point cloud viewed from above. Points are coloured by the largest orthogonal standard deviation of uncertainty. The timing error t_e between the navigation and laser sensor information is included in the error model used to calculate the uncertainty of the points. The points that represent walls form a thick cloud, but the uncertainty of these points is consistent with their distance to the centre of the line.

Figures 3.20 and 3.21 show the same comparison between the error models, this time focussing on the vertical pole. In Figure 3.20, the effect of t_e has been ignored and there are many erroneous points that have an inconsistently small uncertainty. Figure 3.21 illustrates that the inclusion of t_e has corrected this. The uncertainty of the erroneous points now has a strong correlation with their distance to the pole, just as the points representing the walls and fence-lines above.

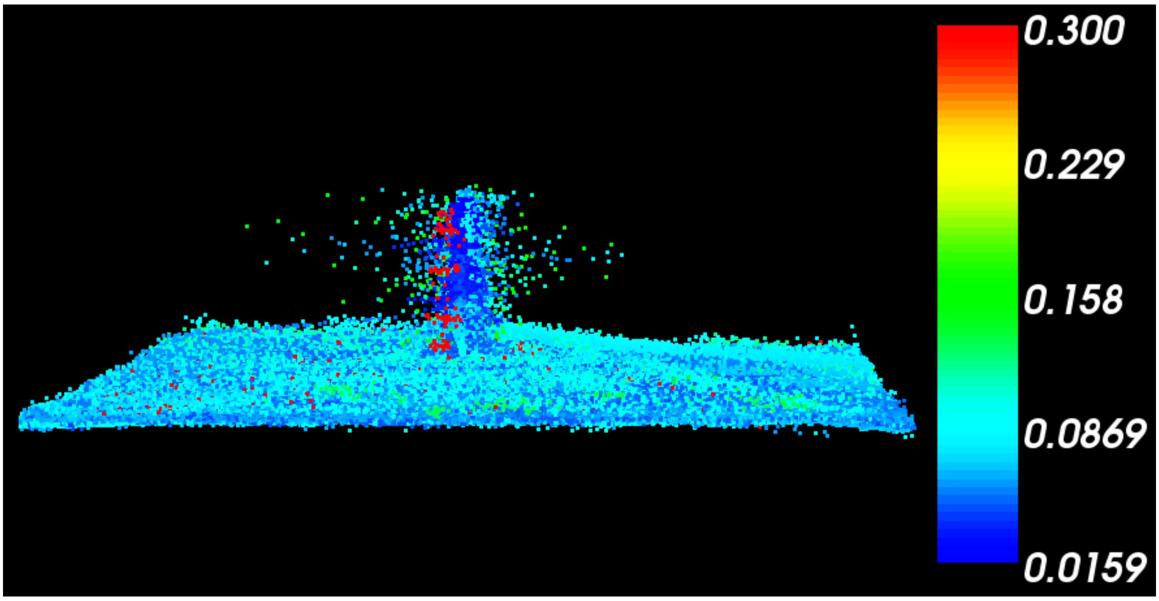


Figure 3.20: A 3D point cloud illustrating the uncertainty of points with respect to a vertical pole. The points are coloured by their uncertainty, but the error model ignores the effect of t_e. Some of the erroneous points are clearly inconsistent, because they are much further from the pole than is indicated by their calculated uncertainty.

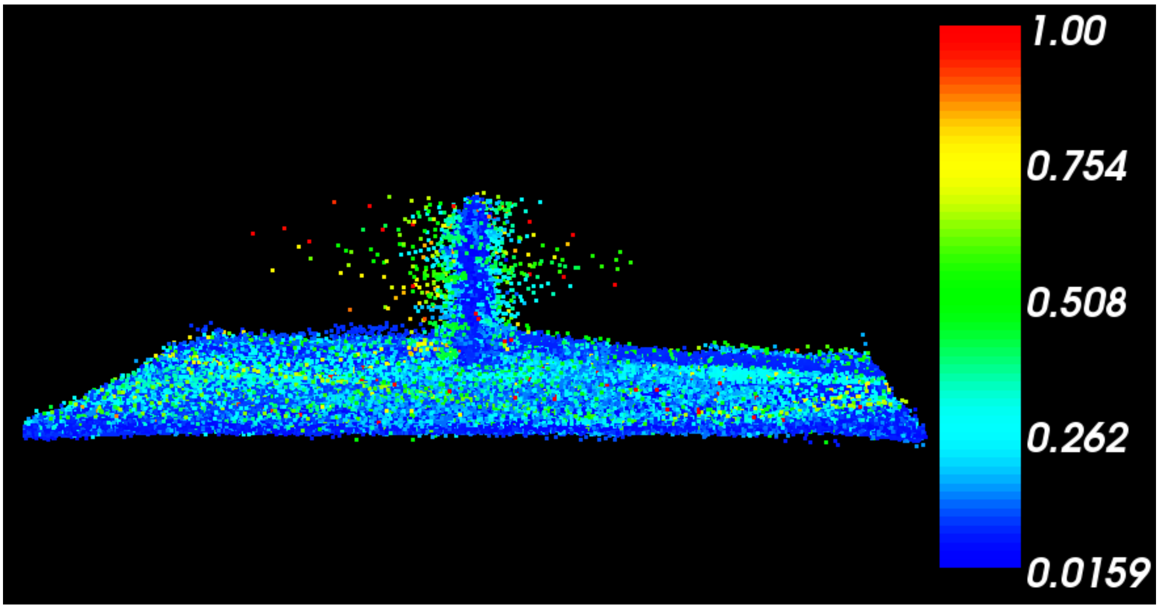


Figure 3.21: A 3D point cloud illustrating the uncertainty of points with respect to a vertical pole. The points are coloured by their uncertainty, including the effect of t_e. Due to the completeness of the error model, all of the points are consistent; their distance to the pole is within two standard deviations of their calculated uncertainty.

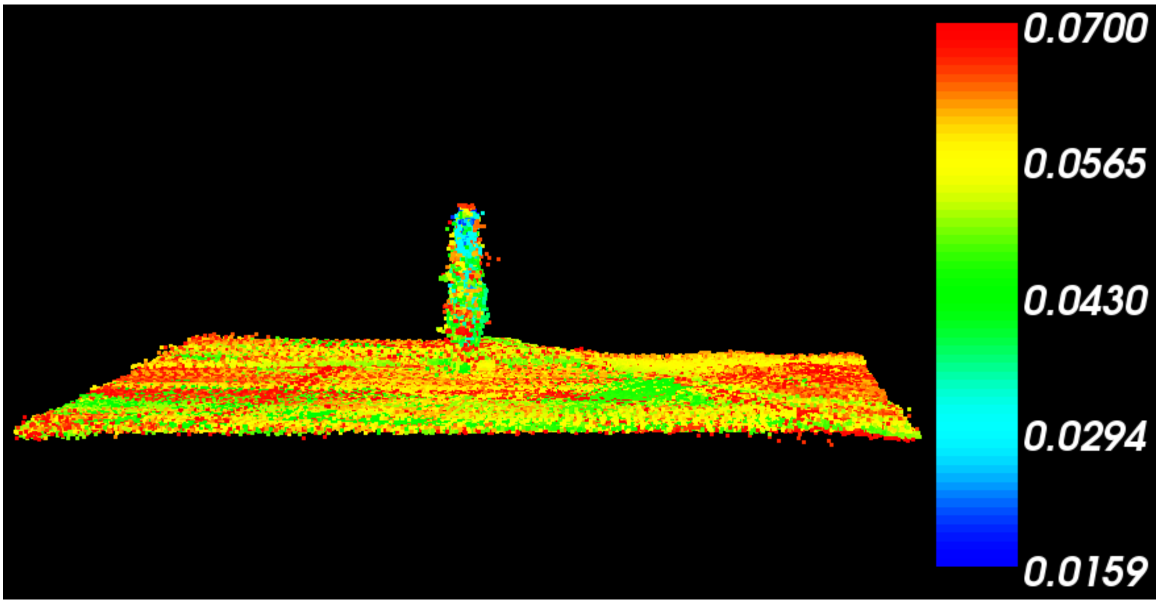


Figure 3.22: A threshold is applied to disregard points with a standard deviation of uncertainty larger than 7cm. The complete error model is used to calculate the uncertainty, so the remaining point cloud is much more accurate.

As the model now includes *all* of the sources of error, a simple threshold on uncertainty can be applied to extract a higher quality data-set. A threshold of 7cm for the largest orthogonal standard deviation is shown in Figure 3.22. Almost all of the erroneous points have now been removed. The accuracy of the error model is high enough to extract the accurate sections of data from the inaccurate samples, yielding a high quality map *despite* the errors. The minimisation of the sources of error that is presented in Chapter 4 will increase the percentage of high quality data points, allowing more of the data to be mapped for a given threshold on accuracy.

3.4 Model Failure

In this Chapter, the process of exteroceptive sensor mapping has been analysed from a geometric perspective. Section 3.1 presented the mathematics required to transform spatial information from the original sensor frame, via the vehicle body frame to which the sensor is attached, to the navigation frame in which the information is finally mapped. In Section 3.2, the sensitivity of these transformations was determined with respect to the geometric variables involved, including the sensor position and the vehicle position. This enabled off-line calculation of the expected mapping error for a given system configuration, and online estimation of mapping uncertainty by using real-time estimates of the uncertainty of the individual inputs. This means that regardless of the choice of environment, sensor and mapping representation, the uncertainty of that representation should in theory be calculable, because the geometric errors are observable. However, if an additional implicit or explicit interpretive layer is added to the representation, arbitrarily large unobservable errors may be added.

Recall the example from Chapter 2 of a laser sensor being used to build an occupancy grid representation of an indoor environment. Without any interpretation, the occupancy map stores a summary of the laser reflectivity in the environment. However, when the map is used as part of a model predictive controller to allow the vehicle to steer around obstacles or *occupied* regions, an implicit interpretive layer has been added to the map. In this case, no explicit processing is required to transform the reflectivity grid into an occupancy grid, but the interpretive layer is implied by the naming of the structure and its usage. The implicit interpretive layer in this case is the assumption that there is a one-to-one mapping between non-traversable areas of the environment and areas that reflect laser energy. This is violated by the presence of hard surfaces that are transparent at the particular laser wavelength such as glass. (Note that not all types of glass will be transparent to all types of laser, as it depends on the properties of both). Although the analysis in this chapter can be used to provide the spatial uncertainty of the map, theoretically allowing the vehicle to safely and robustly traverse the environment, the reality may not be so reliable, due to the potential inaccuracy of the interpretive assumptions. The implicit interpretation in this case can induce arbitrarily large unobservable errors in the map, despite the efforts of tracking the uncertainty through the transformations. Such an *interpretive violation* is considered to be a failure of the environment model.

For any perceptive system that couples some form of sensor with some form of interpretation, sensor failures (where there is a physical fault in the sensor hardware) and model failures (where the interpretation of the sensor information does not match the true environment due to a violated assumption) have very similar effects. Sensor failures can be caused by a physical failure of the hardware, such that the sensor reports spurious values or ceases to function altogether, or it can occur due to environmental effects such as a strong light source reducing the contrast of a camera, or dust affecting the performance of a laser range scanner. In this sense, the distinction between sensor and model failure is blurred. Both types of failure are considered by this thesis as a failure of the model to use sensor information to provide an internal representation that matches the external truth.

The nature and extent of model failures depends completely on the chosen model and how well it summarises the relationship between the sensor and the environment. Because of this, little can be identified in *general* about the effect of such errors on the mapping process, suffice to say that the possibility for such errors present an enormous caveat to the uncertainty calculations of this chapter. Unless careful consideration is given to potential failure modes of the chosen environment model, the reported uncertainty of a map cannot be trusted. In Chapter 5, some general design methods will be discussed in order to mitigate against model failure (including various types of sensor hardware failure), and some specific examples of failure mitigation will be given.

Chapter 4

Mitigation & Management of Mapping Errors

In Chapter 2 the three dominant sources of error for mapping were identified as:

- Systematic errors due to sensor misalignment and synchronisation,
- Stochastic uncertainty due to sensor measurement errors,
- Sensor interpretation errors due to unmodeled environmental conditions.

The effect and nature of these errors was analysed in detail in Chapter 3. In this chapter, solutions are presented to minimise and manage the first two primary sources of mapping error. The third error source (interpretation) will be discussed in Chapter 5. This chapter is divided into four sections. The systematic errors due to sensor misalignment are minimised in Section 4.1 by a new calibration methodology. This technique calculates the sensor alignment by comparing a known environment to the map. Section 4.2 discusses methods for reducing the timing synchronisation errors. In Section 4.3, an overview of current methods for managing terrain uncertainty is given and discussed. Finally, in Section 4.4, a local frame navigation filter is used to evaluate its potential for producing consistent, smooth and sharply focussed maps.

When used together, the solutions presented in this chapter provide increased accuracy and robustness to the most common forms of exteroceptive sensor and mapping algorithms, allowing them to be used safely for feedback control of AUGV systems.

4.1 Minimisation of Sensor Misalignment

The effect of the sensor alignment errors on mapping was analysed in detail in Chapter 3. It was shown that translational errors are propagated directly, and that rotational errors are maximally amplified by the sensor range. This enabled the mapping accuracy to be improved by discarding data with an unacceptably high uncertainty. In this section, the uncertainty of *all* of the mapped data is reduced, by refining the sensor alignment. The data that remains highly uncertain can still be removed, but the reduction in error allows a higher proportion of the data to be kept.

It can be difficult to precisely mount sensor equipment on a mobile platform. Once a sensor is mounted, accurately measuring its location in the vehicle body frame b in six degrees of freedom is also a challenge. In addition, when equipment must repeatedly be dismantled and reassembled, or when different mounting configurations are possible (such as a range scanner that can be mounted at any pitch angle) the sensor position must be accurately recalculated each time. It is assumed that the sensor is physically fixed at some imprecisely known location in b. Actuated sensors (ones that can be moved or directed in some way, with respect to b) are also considered to be mounted at a constant location, by treating the entire actuated assembly as the sensor.

There are various approaches to this problem in the literature. Some application-specific solutions use features of the particular robot to constrain the problem. The mounting position of a laser is determined for a legged robot in [40]. The robots own leg is measured by the laser and compared to the kinematic calculation of the leg position. This solution requires a part of the platform to be visible to the sensor, which is not always possible (or desirable). In [41] a laser is mounted on a movable arm. The position of the laser relative to the arm is found by observing a fixed plane in the environment, while undergoing precisely controlled motions through six degrees of freedom. This is not generally applicable to mobile platforms, because manoeuvrability in all degrees of freedom is not guaranteed. In particular, UGVs are typically constrained to the ground support surface. The approach taken here is similar to that method, but it allows for incomplete manoeuvrability and imperfect localisation. The technique of using a sensor to observe a geometric object such as a planar surface, in order to perform data-centric calibration, is also common for other types of sensor. For example, colour cameras can be calibrated with a fixed camera making observations of a plane that is free to move in six degrees of freedom [42], or the Kalman filtration method of [43] in which the plane is fixed and the robot provides the requisite motion.

In this section, an optimal sensor calibration method is described that measures the six degree of freedom alignment $\{\mathbf{r}_s^b, \mathbf{\Phi}_s^b\}$ of the sensor in the body frame, and provides a covariance matrix for the confidence of the estimate. An artificial environment is created, and mapped by the mobile platform, so that a comparison can be made between the map and the known ground truth. An optimisation is then performed to minimise the difference between the two, by refining the alignment parameters. The calibration procedure is described in Section 4.1.1 and then analysed for an example sensor application in Section 4.1.3. A complete worked example of calibrating a single sensor is given in Section 4.1.4 and multiple sensors in 4.1.5.

4.1.1 The Sensor Calibration Procedure

An artificial environment must be constructed so that a comparison can be made between this ground truth and the map produced by the mobile platform. This environment can have any form, subject to two requirements:

- A scalar cost function must be available to compare the mapped sensor data with the known ground truth. The cost function should return higher costs for maps that are not well matched to the environment, and the lowest cost for the nominal ‘best’ match.

- The cost function must be *sensitive* to changes in each of the parameters of $\{\mathbf{r}^b_s, \boldsymbol{\Phi}^b_s\}$ that are to be optimised.

Implicit in these two requirements is the need for appropriate physical structure, such that the perception of the environment will not be invariant to any of the parameters.

The mobile platform is driven through the known environment E, while recording the exteroceptive sensor data, the corresponding navigation solutions and the acquisition times. This forms a calibration data-set denoted D_E, defined by:

$$D_E \equiv \{\mathbf{p}^s_t, \hat{\mathbf{p}}^n_{b,t}, \hat{\boldsymbol{\Psi}}^n_{b,t}\}, \;\; t \in \{t_1, t_2, ..., t_N\} \tag{4.1}$$

for a set of N discrete sample times. If an estimate of the sensor location $\{\hat{\mathbf{r}}^b_s, \hat{\boldsymbol{\Phi}}^b_s\}$ can be provided (by manual measurement for example), then a map M_E can be produced from D_E by applying the transformation Equation 3.12. Additional application-specific processing can also be applied and there is no restriction imposed on the type of algorithm. For example, lines, planes or other primitive geometric structures may be fitted to the data as part of the mapping process. For generality and simplicity, the implementations in this chapter use the transformation from s to n to produce point clouds, and no further processing is done. The mapping function that combines Equation 3.12 and any additional application-specific processing is denoted $map()$, and the cost function that provides the comparison cost C_E is denoted $cost()$:

$$\begin{aligned} M_E &\equiv map(D_E, \{\hat{\mathbf{r}}^b_s, \hat{\boldsymbol{\Phi}}^b_s\}) \\ C_E &\equiv cost(E, M_E) \end{aligned} \tag{4.2}$$

The process of calibration is to calculate an optimal estimate $\{\overset{*}{\mathbf{r}}{}^b_s, \overset{*}{\boldsymbol{\Phi}}{}^b_s\}$ of the sensor location in b. This can be expressed mathematically by:

$$\begin{aligned} \{\overset{*}{\mathbf{r}}{}^b_s, \overset{*}{\mathbf{\Phi}}{}^b_s\} &\equiv \arg\min_{\{\hat{\mathbf{r}}^b_s, \hat{\mathbf{\Phi}}^b_s\}} \; cost(E, M_E) \\ &= \arg\min_{\{\hat{\mathbf{r}}^b_s, \hat{\mathbf{\Phi}}^b_s\}} \; cost(E, map(D_E, \{\hat{\mathbf{r}}^b_s, \hat{\mathbf{\Phi}}^b_s\}) \end{aligned} \tag{4.3}$$

where D_E and E are constant, and the sensor location estimate is altered to achieve a minimum value for C_E. The values $\{\overset{*}{\mathbf{r}}{}^b_s, \overset{*}{\mathbf{\Phi}}{}^b_s\}$ that yield the minimum cost are considered optimal with respect to the calibration data-set.

Extension to Multiple Sensors

The calibration procedure is described above for a single sensor, and the same procedure can be extended for use with multiple sensors. The simplest option is to repeat the entire process for each sensor, yielding the optimal pose for each ($\{\overset{*}{\mathbf{r}}{}^b_{si}, \overset{*}{\mathbf{\Phi}}{}^b_{si}\}$ for sensor i), with respect to the data-set obtained by that sensor D_{Esi}. However, when fusing information from multiple sensors, errors in the sensor alignment values cause not only a systematic error for each sensor, but a *systematic contradiction* between sensors. The errors for each sensor may differ such that the combined information is permanently contradictory. A better solution is to optimise all of the sensors together, which is possible if the sensors view the same environment (or at least overlapping sections). If optimised in unison, the final solution for each sensor is optimal with respect to *all* of the data from *all* of the sensors. This not only minimises the systematic errors for each sensor, but it also explicitly minimises the systematic contradiction. In addition, the individual sensor calibration should be improved because of the increased total amount of data available. In situations where the physics or geometry of a particular sensor makes it difficult to obtain enough data to perform the calibration accurately, a second sensor with a different modality may be used to calibrate both accurately. In Section 4.1.6, it is shown that a particular radar sensor is difficult to calibrate independently, but that the joint calibration with a laser makes precise calibration possible.

In this framework, the data from all of the sensors are mapped together:

$$\begin{aligned} M_E &= \sum_i map(D_{Esi}, \{\hat{\mathbf{r}}^b_{si}, \hat{\mathbf{\Phi}}^b_{si}\}) \\ C_E &= cost(E, M_E) \end{aligned} \tag{4.4}$$

where the sum operator refers to the *combination* of maps as appropriate.

The combined calibration is performed by minimising C_E by varying the offsets of all of the sensors:

$$\begin{aligned} \{\overset{*}{\mathbf{r}}{}^b_{si}, \overset{*}{\mathbf{\Phi}}{}^b_{si}\} &\equiv \underset{\{\hat{\mathbf{r}}^b_{si}, \hat{\mathbf{\Phi}}^b_{si}\}}{\arg\min} \; cost(E, M_E) \\ &= \underset{\{\hat{\mathbf{r}}^b_{si}, \hat{\mathbf{\Phi}}^b_{si}\}}{\arg\min} \; cost(E, \sum_i map(D_E, \{\hat{\mathbf{r}}^b_{si}, \hat{\mathbf{\Phi}}^b_{si}\}) \end{aligned} \tag{4.5}$$

For k sensors, the optimisation has a total of $6 \times k$ free parameters, so it is likely to take significantly longer to process this way. The multi-sensor variant of the procedure is validated experimentally, by comparing the calibration results of a single sensor in Section 4.1.4 and for four sensors in Section 4.1.5.

4.1.2 Calculation of Parameter Uncertainty

Once the optimal sensor location $\{\overset{*}{\mathbf{r}}{}^b_s, \overset{*}{\mathbf{\Phi}}{}^b_s\}$ has been determined from Equation 4.3, it is desirable to calculate the uncertainty of the six optimal parameters, in the form of a 6×6 sensor pose covariance matrix $\mathbf{Q}_{\{r^b_s, \Phi^b_s\}}$. This can then be used as part of the system covariance matrix $\mathbf{Q}$ in Equation 3.19 or 3.39. There are several techniques that can be used to approximate the sensor pose covariance matrix. The Laplace approximation [44, §27], [45, §4.4] has been proposed to calculate the covariance matrix for scan matching algorithms [46, 47, 48], which are similar to this calibration technique because they estimate a coordinate transformation given a data-set. In [48] it is argued that the Laplace method can be sensitive to the numerical calculation of the Hessian matrix, so an alternative Monte Carlo importance sampling technique is proposed instead. Furthermore, it states that the importance sampling technique can be arbitrarily accurate at the cost of computation time, which is not necessarily prohibitive for this application. Appendix C demonstrates the application of the Laplace approximation for the individual parameters in isolation (to estimate the diagonals of the covariance matrix), but this relies on large sample numbers for each parameter. It is computationally intractable to extend this to the full six dimensional joint probability due to the 'curse of dimensionality', and more efficient Hessian calculation methods are not used due to the potential sensitivity to the data.

Monte Carlo Importance Sampling

This section describes the application of importance sampling to calculate the covariance matrix for the six sensor location parameters. For notational simplicity and similarity to [48], a vector $\mathbf{x}$ is formed with the unification of the six parameters of $\{\mathbf{r}_s^b, \mathbf{\Phi}_s^b\}$:

$$\mathbf{x} = [r_{s,x}^b, r_{s,y}^b, r_{s,z}^b, \phi_{s,x}^b, \phi_{s,y}^b, \phi_{s,z}^b]^T \tag{4.6}$$

With the optimal sensor location calculated, N sample locations $\mathbf{x}_i$ are drawn at random from a Gaussian distribution, centred at $\mathbf{x}_{max} = \{\overset{*}{\mathbf{r}}{}_s^b, \overset{*}{\mathbf{\Phi}}{}_s^b\}$, with sufficient variance to cover the actual distribution to be determined. A likelihood function $\Lambda(\mathbf{x})$ is calculated by considering the error model for this problem $P(\mathbf{p}^n|\mathbf{x})$, which expresses the probability of a particular observation (one of the samples in the calibration dataset) given an assumed value for the sensor location $\mathbf{x}$. This function will depend on the choice of artificial environment, and should be related to the cost function from Equation 4.2. In some cases, the relation may be direct, where the cost function can be specified as $-\Lambda(\mathbf{x})$, but this is not strictly necessary. A specific example of a likelihood function $\Lambda(\mathbf{x})$ is given in Section 4.1.4.

The N random sample locations are drawn from a Gaussian distribution $\mathbf{x}_i \sim \mathcal{N}(\mathbf{x}_{max}, \mathbf{\Sigma})$, where $\mathbf{\Sigma}$ is chosen to span the likely distribution of the parameters. This distribution focuses the samples in the region of interest, allowing greater accuracy with fewer samples, as long as the sample distribution approximately covers the true distribution. Practically, this may have to be done more than once if the initial estimate of $\mathbf{\Sigma}$ is small with respect to the outcome of the importance sampling technique. Initial values for $\mathbf{\Sigma}$ can be produced from the uncertainty associated with the hand measurements of the sensor pose, as the calibration reduces the uncertainty from this starting point. Weights w_i are produced by the likelihood function divided by the likelihood of the sample location (which in this case is an unnormalised Gaussian probability density function (PDF)), and then normalised:

$$\begin{aligned}
w_i &= \frac{\Lambda(\mathbf{x}_i)}{\mathcal{N}(\mathbf{x}_i - \mathbf{x}_{max}, \mathbf{\Sigma})} \\
&= \frac{\Lambda(\mathbf{x}_i)}{\exp(-\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_{max})^T \mathbf{\Sigma}^{-1} (\mathbf{x}_i - \mathbf{x}_{max}))} \\
w_i^* &= w_i / \sum_{j=1}^{N} w_j \qquad (4.7)
\end{aligned}$$

The weighted mean $\hat{\mathbf{x}}$ and covariance $\mathbf{Q}_{\{r_s^b, \Phi_s^b\}}$ can then be calculated:

$$\begin{aligned}
\hat{\mathbf{x}} &= \sum_{i=1}^{N} w_i^* \mathbf{x}_i \\
\mathbf{Q}_{\{r_s^b, \Phi_s^b\}} &= \sum_{i=1}^{N} (w_i^* \mathbf{x}_i \mathbf{x}_i^T) - \hat{\mathbf{x}}\hat{\mathbf{x}}^T \qquad (4.8)
\end{aligned}$$

This provides a measure of the uncertainty of the sensor location parameters and can be incorporated directly into the system covariance matrix $\mathbf{Q}$, in Equation 3.19 or 3.39. $\mathbf{Q}_{\{r_s^b, \Phi_s^b\}}$ is an approximation that depends on the particular data-set D_E that is used for the calibration, and will only be a true reflection of the accuracy of the solution if enough data have been collected, as discussed further in Section 4.1.4.

4.1.3 Application to UGVs and 2D Scanning Range Sensors

In Section 4.1.1, the calibration method was described in general terms, but in order to implement the procedure, several specific choices must be made:

- An artificial environment E must be designed, to meet the functional requirements for calibration.
- An appropriate mapping algorithm $M_E = \mathit{map}(D_E, \{\hat{\mathbf{r}}_s^b, \hat{\mathbf{\Phi}}_s^b\})$ must be chosen.
- A cost function $C_E = \mathit{cost}(E, M_E)$ must be specified to compare the environment to the map.

The choice of these items depends largely on the sensor to be calibrated, so this section presents a specific example, together with the corresponding choice of the environment and the mapping and cost functions. The sensor to be calibrated is a SICK LMS-291 2D laser range scanner, mounted at some arbitrary location on an outdoor UGV. Appropriate choices for E, $\mathit{map}()$ and $\mathit{cost}()$ are now specified for this sensor, and an analysis is done to ensure that these choices meet the functional requirements specified in Section 4.1.1.

The Artificial Environment

The two functional requirements that the artificial environment must meet are that a cost function can be specified to compare E to M_E, and that this function is sensitive to each of the six parameters of $\{\mathbf{r}_s^b, \mathbf{\Phi}_s^b\}$. The calibration procedure must be performed each time the sensor configuration is altered due to remounting or repositioning, so an important non-functional requirement is that the environment be simple to prepare, and relatively quick to map.

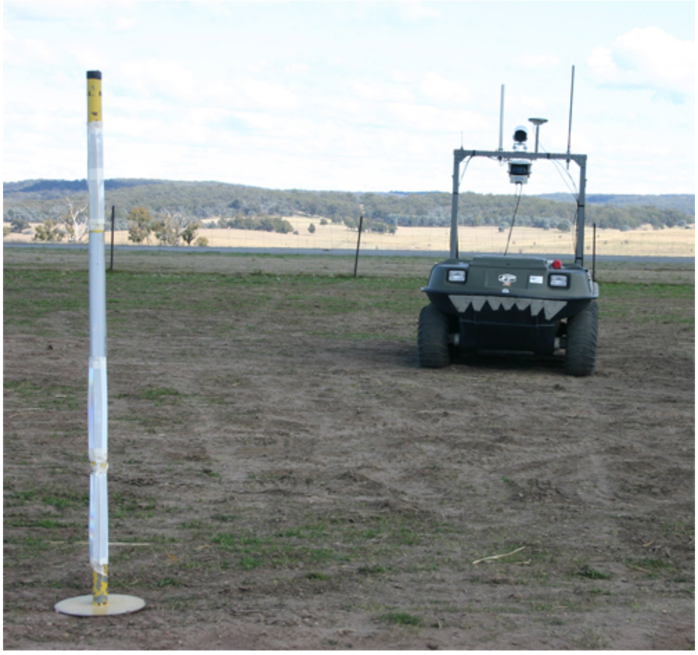


Figure 4.1: An artificial environment is constructed by placing a vertical pole on a section of relatively flat ground.

The environment used for calibration in this thesis consists of a single vertical pole, placed on a relatively flat section of ground. The base of the pole is perpendicular, and no special equipment is used for placement. In addition, an adjacent section of rough terrain is desirable, as it will be shown that some aspects of the calibration are not solvable unless the vehicle can have non-zero roll or pitch while acquiring the data. A photograph of the environment is given in Figure 4.1. The same environment was used in Chapter 3 as an illustration of the effect of misalignment on the map. From the earlier analysis, it should be intuitively clear that a function can be specified to quantify the extent of deviation between E and M_E. The former is known to be a single straight pole on a flat surface, whereas the map will have duplicated poles and a blurred support surface due to the alignment error. The process in Chapter 3 of inducing an alignment error and visualising the result is driven in reverse in this chapter to reduce the misalignment.

The pole and the ground are treated as two separate components. In fact, the ground and the pole can be scanned in completely different locations if necessary, provided that the platform and sensor geometry remains constant. The complete data-set D_E

is split into the subset of data corresponding to the pole D_{pole} and to the ground D_{ground}. One way to segment the data is to map the whole set (i.e. to produce M_E using $map(E, D_E)$), with an approximate measurement of the sensor location. If the errors are sufficiently small that the pole and the ground can be visualised, then the points corresponding to each can be manually segregated into M_{pole} and M_{ground}. The correspondence between D_E and M_E is kept, so that the segmentation of the map can be back-propagated to produce D_{pole} and D_{ground}.

Automatic segmentation is also possible in many circumstances. In this example, the pole is covered in a retro reflective material that the SICK LMS-291 laser scanner measures as highly reflective. Figure 4.2 illustrates the extraction of the reflective pole from the low reflectivity of the ground. Some of the points that belong to the pole have a low reflectivity and are not selected, but *all* of the highly reflective points belong to the pole, so there are no false positives. This technique depends on the specific physical properties and features of the sensor, so it may not always be applicable. If there is no clutter in the environment, then segmentation may be based on the geometric relationship of the sensor to the environment alone. Figure 4.3 shows a calibration map M_E from a ground based perspective. For this example environment, the pole is scanned at a shorter range than is possible for anything else. Therefore it can be easily segmented by proximity in the sensor frame, or even by uncertainty, given the correlation between scan range and uncertainty that was shown in Chapter 3.

The Corresponding Cost Function

With the calibration data segmented into two subsets D_{pole} and D_{ground}, it is possible to specify a separate cost function for each. The pole cost C_{pole} is defined to be the average squared perpendicular distance of each point in M_E to the pole. The pole is assumed to be vertical so the equation is the average squared 2D distance of each

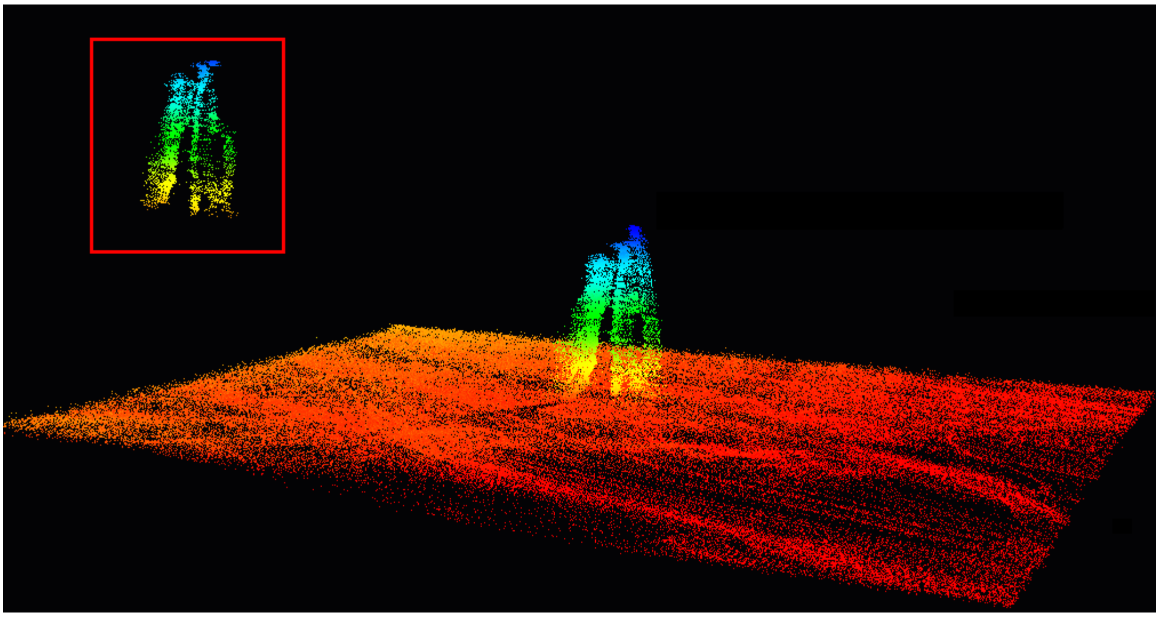


Figure 4.2: The automatic segmentation of the map, to separate the pole and the ground. Prior to calibration, a calibration data-set D_E is produced, and mapped as a point cloud M_E, shown here. The pole is covered with a retro-reflective material that is detectable by the SICK LMS-291 laser range scanner. The inset shows the points that exceed a reflectivity threshold. There are no false positives (ground points seen as part of the pole) because the reflectivity signal to noise ratio for this sensor is extremely high.

pole point to the mean of the pole data. Given a map in point cloud form:

$$M_{pole} = \{\mathbf{p}^n_{pole,i}, ..., \mathbf{p}^n_{pole,N_p}\} \tag{4.9}$$

the pole cost function is given by:

$$\begin{aligned} C_{pole} &= \frac{\sum_i (p^n_{pole,x,i} - \overline{p^n_{pole,x}})^2}{N_p} + \frac{\sum_i (p^n_{pole,y,i} - \overline{p^n_{pole,y}})^2}{N_p} \\ C_{pole} &= \sigma^2_{pole,x} + \sigma^2_{pole,y} \end{aligned} \tag{4.10}$$

where N_p is the sample size of M_{pole} and $\overline{p^n_{pole,x}}$ is the average X coordinate of these points. This is the sum of the variances of the two non vertical dimensions of the pole data M_{pole}.

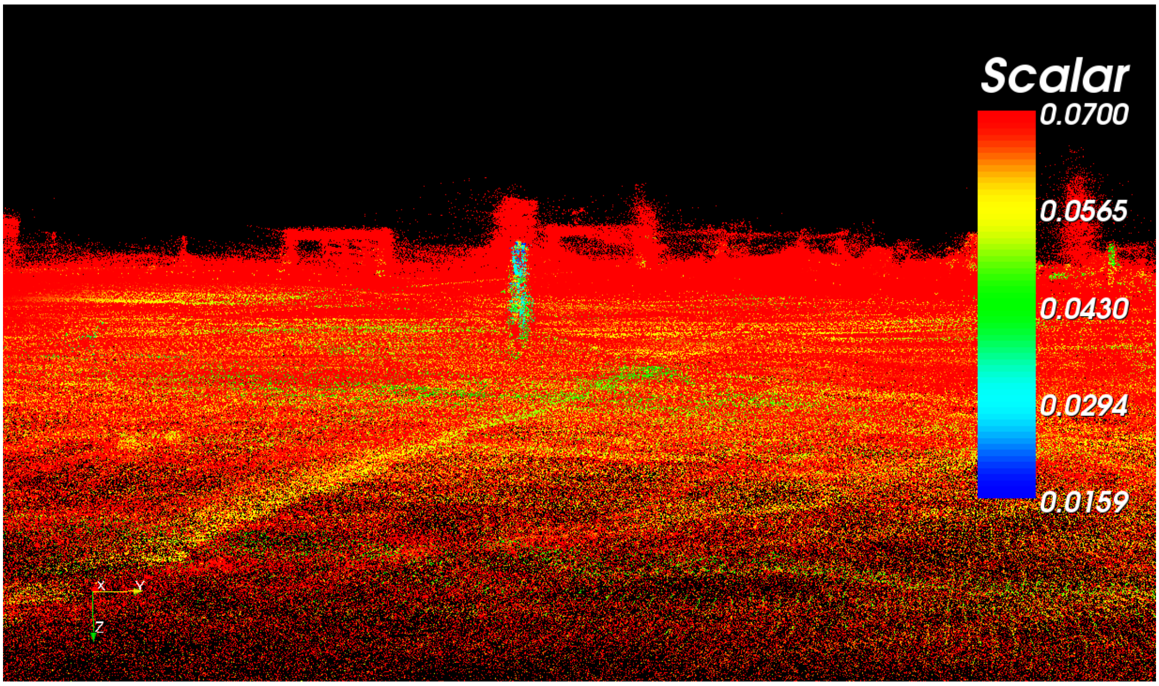


Figure 4.3: The automatic segmentation of the pole and ground can be done based on the geometry of the system and the environment. The map M_E is coloured by uncertainty, which is significantly lower for the pole due to the close proximity of the sensor during the scan. Alternatively, a threshold can be set on the range in the sensor frame to achieve the same result.

The function for the ground cost C_{ground} is similarly defined as the average squared distance of the points in M_{ground} to the ground plane. The plane is assumed to be horizontal, so the equation is the average squared 1D distance of each point to the mean of the ground data. Relying on a point cloud form:

$$M_{ground} = \{\mathbf{p}^n_{ground,i}, ..., \mathbf{p}^n_{ground,N_g}\} \tag{4.11}$$

the cost function for the ground is given by:

$$\begin{aligned} C_{ground} &= \frac{\sum\limits_i (p^n_{ground,z,i} - \overline{p^n_{ground,z}})^2}{N_g} \\ C_{ground} &= \sigma^2_{ground,z} \end{aligned} \tag{4.12}$$

where N_g is the sample size of M_{ground} and $\overline{p^n_{ground,z}}$ is the average Z coordinate of these points. This is the variance of the vertical dimension of the ground data.

The two cost functions are summed to give the overall cost function for this example environment.

$$\begin{aligned} C_E &= cost(E, M_E) \\ &= C_{pole} + C_{ground} \\ &= \sigma^2_{pole,x} + \sigma^2_{pole,y} + \sigma^2_{ground,z} \end{aligned} \tag{4.13}$$

Satisfaction of Functional Requirements

In this section, a sensitivity analysis is performed to determine how successful the optimisation is likely to be for the six parameters, if the example environment and cost function specified above is used. The specific case that the platform is constrained to the ground plane (as is the case for most ground vehicles) is considered. This restricts the roll and pitch of the vehicle to zero:

$$\psi^n_{b,x} = \psi^n_{b,y} = 0 \tag{4.14}$$

For this environment to yield a successful calibration of all of the parameters of $\{\mathbf{r}^b_s, \mathbf{\Phi}^b_s\}$, the cost function must be sensitive to each of them. This can be expressed mathematically as:

$$\frac{\partial C_E}{\partial \rho} \neq 0, \forall \rho \in \{\mathbf{r}^b_s, \mathbf{\Phi}^b_s\} \tag{4.15}$$

However, the cost function depends on the particular data in D_E, so it is difficult to calculate *a-priori*. When gathering the calibration data-set, the sensor location is fixed. The remaining free parameters are the spatial sensor data $\mathbf{p}^s$ (or for this example $\{r, \theta\}$), and the vehicle pose $\{\mathbf{p}^n_b, \mathbf{\Psi}^n_b\}$. This section will determine the necessary configurations of the free parameters for a data-set D_E to yield a successful calibration. For a single point in M_E, consider the sensitivity $\frac{\partial \mathbf{p}^n}{\partial \rho}$ with respect to the parameters of $\{\mathbf{r}_s^b, \mathbf{\Phi}_s^b\}$. If the sensitivity is zero for all variations in the free parameters, then changing ρ will not affect the point cloud for any D_E. Furthermore, if $\frac{\partial \mathbf{p}^n}{\partial \rho}$ is constant for all variations of the free parameters, then variations in ρ will cause the entire point cloud to move together with constant variance, which will not affect C_E in Equation 4.13. For the optimisation to perform as desired for parameter ρ, a vehicle pose and sensor return is required that yields a non zero and non-constant sensitivity:

$$\frac{\partial \mathbf{p}^n}{\partial \rho} \neq c, \;\; \exists \{\mathbf{p}_b^n, \mathbf{\Psi}_b^n, \mathbf{p}^s\}, \;\; \forall \rho \in \{\mathbf{r}_s^b, \mathbf{\Phi}_s^b\} \tag{4.16}$$

For this sensor configuration, the laser sensor geometry from Equation 3.1 is substituted into the partial derivatives of the s to n transformation (Equation 3.12), listed in Appendix A. The partial derivatives for $\mathbf{r}_s^b$ are then given by:

$$
\begin{aligned}
\frac{\partial p_x^n}{\partial r_{s,x}^b} &= \cos(\Psi_{b,z}^n)\cos(\Psi_{b,y}^n) \\
\frac{\partial p_y^n}{\partial r_{s,x}^b} &= \sin(\Psi_{b,z}^n)\cos(\Psi_{b,y}^n) \\
\frac{\partial p_z^n}{\partial r_{s,x}^b} &= -\sin(\Psi_{b,y}^n) \\
\frac{\partial p_x^n}{\partial r_{s,y}^b} &= -\sin(\Psi_{b,z}^n)\cos(\Psi_{b,x}^n) + \cos(\Psi_{b,z}^n)\sin(\Psi_{b,y}^n)\sin(\Psi_{b,x}^n) \\
\frac{\partial p_y^n}{\partial r_{s,y}^b} &= \cos(\Psi_{b,z}^n)\cos(\Psi_{b,x}^n) + \sin(\Psi_{b,z}^n)\sin(\Psi_{b,y}^n)\sin(\Psi_{b,x}^n) \\
\frac{\partial p_z^n}{\partial r_{s,y}^b} &= \cos(\Psi_{b,y}^n)\sin(\Psi_{b,x}^n) \\
\frac{\partial p_x^n}{\partial r_{s,z}^b} &= \sin(\Psi_{b,z}^n)\sin(\Psi_{b,x}^n) + \cos(\Psi_{b,z}^n)\sin(\Psi_{b,y}^n)\cos(\Psi_{b,x}^n) \\
\frac{\partial p_y^n}{\partial r_{s,z}^b} &= -\cos(\Psi_{b,z}^n)\sin(\Psi_{b,x}^n) + \sin(\Psi_{b,z}^n)\sin(\Psi_{b,y}^n)\cos(\Psi_{b,x}^n) \\
\frac{\partial p_z^n}{\partial r_{s,z}^b} &= \cos(\Psi_{b,y}^n)\cos(\Psi_{b,x}^n) \qquad (4.17)
\end{aligned}
$$

With the inclusion of the ground plane constraint in Equation 4.14, this reduces to:

$$
\begin{aligned}
\frac{\partial \mathbf{p}^n}{\partial r_{s,x}^b} &= [\cos(\Psi_{b,z}^n), \sin(\Psi_{b,z}^n), 0] \\
\frac{\partial \mathbf{p}^n}{\partial r_{s,y}^b} &= [-\sin(\Psi_{b,z}^n), \cos(\Psi_{b,z}^n), 0] \\
\frac{\partial \mathbf{p}^n}{\partial r_{s,z}^b} &= [0, 0, 1] \qquad (4.18)
\end{aligned}
$$

For all variations of the free parameters, $\frac{\partial \mathbf{p}^n}{\partial r_{s,z}^b} = [0, 0, 1]$ is constant, so $r_{s,z}^b$ fails the requirements in Equation 4.16. Therefore, $r_{s,z}^b$ can only be determined if the vehicle undergoes some roll or pitch. This makes intuitive sense; if the vehicle is limited to a planar surface, changing the Z offset of the sensor location will move the entire point cloud up or down, which will not affect $\sigma_{ground,z}^2$ in Equation 4.13. Also of interest is that the partial derivatives of p_z^n with respect to $r_{s,x}^b$ and $r_{s,y}^b$ are zero for this environment, so the calibration of these offsets is achieved using the pole data alone, and not the ground data. If the vehicle does undergo non-zero roll and pitch, then *all* of the terms of Equation 4.17 are dependent on Ψ. In this case, the pole and ground data-sets (D_{pole} and D_{ground}) will be useful for the calibration.

The same process is applied to the sensor rotation angles $\mathbf{\Phi}_s^b$. The laser sensor geometry of Equation 3.1 and the ground plane constraint of Equation 4.14 is substituted into the partial derivatives with respect to the sensor rotation angles $\frac{\partial \mathbf{p}^n}{\partial \mathbf{\Phi}_s^b}$. This is evaluated for *all* of the sensor rotation angles in Appendix B, but for brevity, only the analysis of sensor pitch is shown here. The sensitivity of the mapped points to the sensor pitch angle for this environment is given by:

$$\begin{aligned}\frac{\partial p_x^n}{\partial \phi_{s,y}^b} = & \\ & -\cos(\psi_{b,z}^n)\cos(\phi_{s,z}^b)\sin(\phi_{s,y}^b)r\cos(\theta) \\ & +\cos(\psi_{b,z}^n)\cos(\phi_{s,z}^b)\cos(\phi_{s,y}^b)\sin(\phi_{s,x}^b)r\sin(\theta) \\ & +\sin(\psi_{b,z}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,y}^b)r\cos(\theta) \\ & -\sin(\psi_{b,z}^n)\sin(\phi_{s,z}^b)\cos(\phi_{s,y}^b)\sin(\phi_{s,x}^b)r\sin(\theta)\end{aligned} \tag{4.19}$$

$$\begin{aligned}\frac{\partial p_y^n}{\partial \phi_{s,y}^b} = & \\ & -\sin(\psi_{b,z}^n)\cos(\phi_{s,z}^b)\sin(\phi_{s,y}^b)r\cos(\theta) \\ & +\sin(\psi_{b,z}^n)\cos(\phi_{s,z}^b)\cos(\phi_{s,y}^b)\sin(\phi_{s,x}^b)r\sin(\theta) \\ & -\cos(\psi_{b,z}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,y}^b)r\cos(\theta) \\ & +\cos(\psi_{b,z}^n)\sin(\phi_{s,z}^b)\cos(\phi_{s,y}^b)\sin(\phi_{s,x}^b)r\sin(\theta)\end{aligned} \tag{4.20}$$

$$\begin{aligned}\frac{\partial p_z^n}{\partial \phi_{s,y}^b} = \quad & -\cos(\phi_{s,y}^b) r \cos(\theta) \\ & -\sin(\phi_{s,y}^b)\sin(\phi_{s,x}^b) r \sin(\theta) \end{aligned} \tag{4.21}$$

The dependence of $\frac{\partial \mathbf{p}^n}{\partial \phi_{s,y}^b}$ on the vehicle heading $\psi_{b,z}^n$ and the scan angle θ means that viewing the environment from different headings and different scan angles will provide non-constant partial derivatives as required by Equation 4.16, as long as the derivatives are not equal to zero. Assuming that the data are collected from all permutations of vehicle heading and scan angle, then by analysing Equations 4.19, 4.20 and 4.21, the set of sensor rotation angles that result in non-zero partial derivative term can be determined. This will indicate the sensor configurations that permit calibration of the sensor pitch angle with this technique. A similar process is then done for the sensor roll and yaw, shown in Appendix B.

Equations 4.19 and 4.20 are analysed first, to discover how the pole data-set will perform, and the results are shown in Tables 4.1 and 4.2. Table 4.1 is constructed by considering each term in Equations 4.19 and 4.20. For each term, the conditions for $\mathbf{\Phi}_s^b$ that ensure a non-zero value are added as a row in the table. The rows are then logically compressed to form Table 4.2, where the first row is formed by combining rows 1 & 3 from Table 4.2 and the second row is formed from rows 2 & 4. If the sensor angles conform to the conditions in *any* row, then $\phi_{s,y}^b$ can be calibrated successfully with D_{pole}. Conversely, Table 4.3 shows the *only* configuration that cannot yield successful calibration of $\phi_{s,y}^b$ with D_{pole}. If the sensor has zero roll and pitch with respect to the body frame, then the pole will not be sufficient for calibration.

In Appendix B, the same process applied to $\frac{\partial \mathbf{p}^n}{\partial \phi_{s,x}^b}$ shows that roll calibration fails under the same unique condition as pitch and analysis of $\frac{\partial \mathbf{p}^n}{\partial \phi_{s,z}^b}$ shows that there is no failing configuration for yaw. With zero sensor roll and pitch, the partial derivatives of p_z^n in Equation 4.21 are non-zero, but it is impossible to obtain ground data with a sensor mounted parallel to the ground plane.

It has been shown that subject to the ground plane constraint, the pole section of

Table 4.1: Sensor configuration required for calibration of sensor pitch

$\phi^b_{s,x}$	$\phi^b_{s,y}$	$\phi^b_{s,z}$
any	$\neq 0$	$\neq \pm 90$
$\neq 0$	$\neq \pm 90$	$\neq \pm 90$
any	$\neq 0$	$\neq 0$
$\neq 0$	$\neq \pm 90$	$\neq 0$

Table 4.2: Condensed requirement for calibration of sensor pitch

$\phi^b_{s,x}$	$\phi^b_{s,y}$	$\phi^b_{s,z}$
any	$\neq 0$	any
$\neq 0$	$\neq \pm 90$	any

Table 4.3: Unique failing configuration for calibration of sensor pitch

$\phi^b_{s,x}$	$\phi^b_{s,y}$	$\phi^b_{s,z}$
$= 0$	$= 0$	any

this artificial environment will suffice for calibration of all parameters, other than $r^b_{s,z}$, provided that the sensor has a non-zero roll or pitch angle with respect to the vehicle body frame. Furthermore, D_{ground} cannot be used to fix the failing configurations. To allow for all parameters to be calculated, the platform must undergo non-zero roll or pitch manoeuvres while observing the environment. To achieve this, a section of non-flat terrain adjacent to the environment can be used, so that the vehicle roll or pitch can be varied. In this case, the ground plane constraint can be ignored, and the calibration becomes sensitive to both the pole and ground data-sets. Although D_{ground} is not strictly required, the larger surface area allows for more data to be obtained in practice, so it is recommended to incorporate both data-sets. Section 4.1.4 will experimentally confirm that the calibration provides reasonable values for the sensor location with just the pole data, but that the results are improved with the pole and ground used together.

This analysis relies on the constraints imposed by the geometry of the example environment, so the conclusions cannot be extrapolated to any arbitrary sensor or platform configuration. It is likely that different robotic applications will have different constraints and that in turn, they may require a different type of structure in the artificial calibration environment. However, the same form of analysis shown in this section can be used to calculate requirements for any such configurations and the overall calibration method can still be applied.

4.1.4 Calibration Results for a Single Sensor

The general calibration procedure described in Section 4.1.1 was then analysed further in Section 4.1.3 for the application to a UGV with a 2D laser range scanner. This section presents the results of calibration for a single sensor. The configuration that was used in Chapter 3, as described in Section 3.3.1, and illustrated in Figures 3.8 and 3.9, is used again here. The SICK LMS-291 laser scanner is configured with a range of 80m and ±90°, with a resolution of 1cm and 0.25°. A sequential quadratic programming (SQP) method implemented in [49] is used for the numerical optimisation of Equation 4.3.

In addition to the basic calibration, three other experiments are performed in this section:

- The performance of the pole and ground data is compared to the results obtained with the pole only. This confirms that the ground data are not strictly required, but that better calibration results are obtained when they are used, as predicted in Section 4.1.3.
- Data-sets of different sizes are used to show the *quantity* of data that is required for calibration.
- The effect of the uncertainty of the points in the calibration data-set is determined, by comparing the results of calibration with a highly certain sub-set of the data, to the results when all of the data are used.

The Calibration Data-Set

The success of the calibration procedure depends on the particular data-set D_E that is used. The analysis in Section 4.1.3 found that the calibration would be successful for

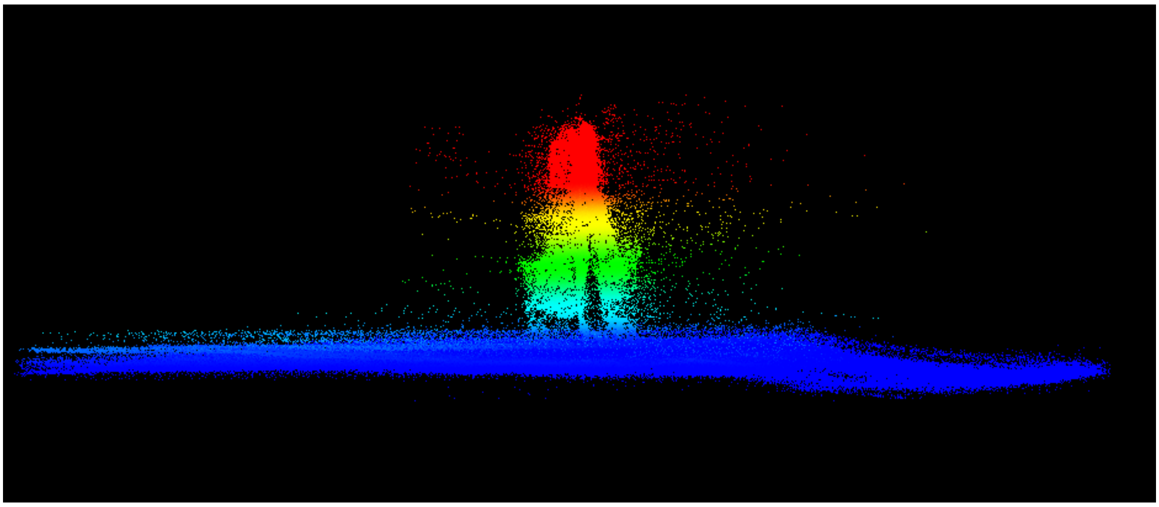


Figure 4.4: A three dimensional point cloud visualisation M_E of the calibration data-set D_E, coloured by elevation. The sensor location estimate $\{\hat{\mathbf{r}}^b_s, \hat{\mathbf{\Phi}}^b_s\}$ that was used to generate M_E has been measured as precisely as possible by hand, but there is clearly an error in the map. Despite this, it is still possible to manually segment the pole from the ground by elevation.

most of the parameters, as long as the environment was viewed with varying vehicle yaw angles. In addition, varying vehicle roll and pitch is required to determine the vertical sensor offset. This section checks that D_E conforms to these requirements.

The data-set must first be split into D_{pole} and D_{ground}. D_E is converted to a point cloud M_E with Equation 3.12, with initial hand-measured values for $\{\mathbf{r}^b_s, \mathbf{\Phi}^b_s\}$. Although the measurement is done as precisely as possible, a considerable amount of error can be seen in M_E, shown in Figure 4.4. Despite the error, the figure shows that it is still possible to manually segment the pole points from the ground points by using the elevation. Alternatively, the automatic segmentation techniques discussed in Section 4.1.3 can be used.

A histogram of the vehicle pose at the time of acquisition of each point is presented for D_{pole} in Figure 4.5, and for D_{ground} in Figure 4.6. These figures illustrate that a spread of vehicle angles $\mathbf{\Psi}^n_b$ has been achieved, including a wide distribution of vehicle heading $\psi^n_{b,z}$, and vehicle roll and pitch angles of up to eight degrees. This violates the ground plane constraint, although some parameter insensitivity may remain due to the small maximum roll and pitch angles. From the histograms, D_E is likely to

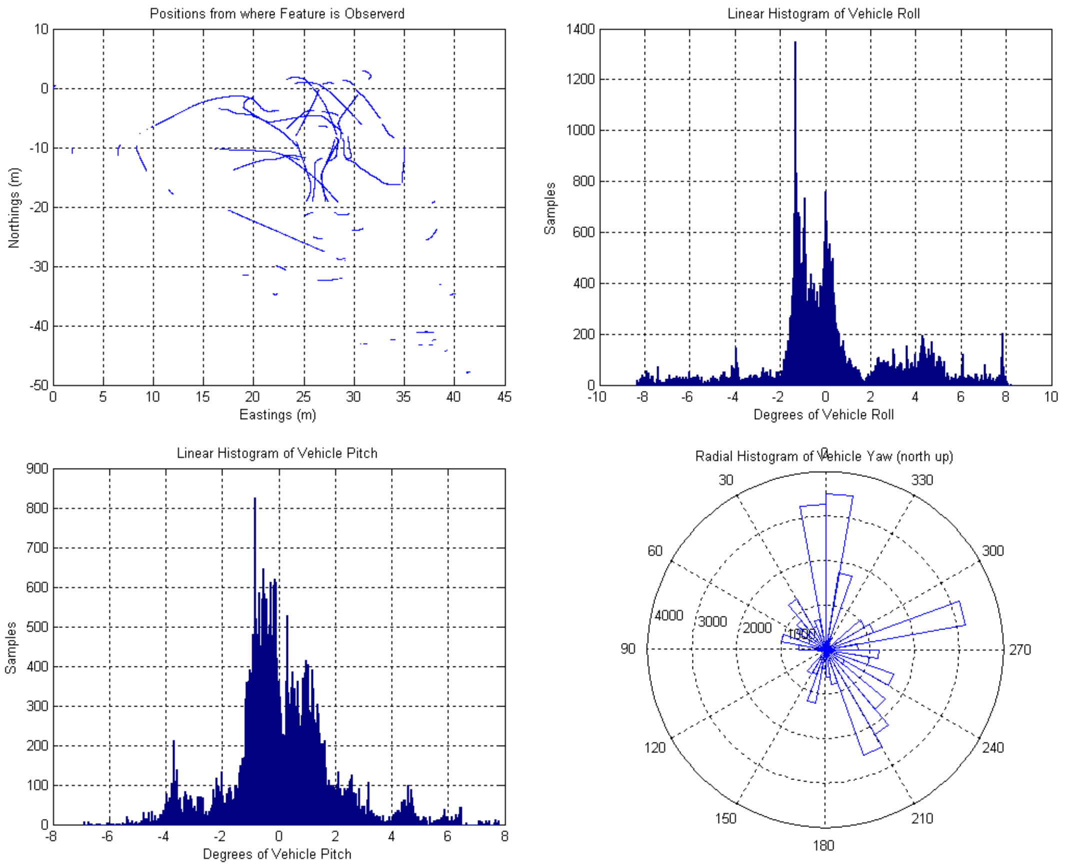


Figure 4.5: The vehicle poses at which the samples in the calibration data-set D_{pole} were obtained. A reasonable spread of vehicle positions, and angles has been achieved.

yield a successful calibration, because the constraints and assumptions Section 4.1.3 are satisfied.

The Calibration Results

The calibration procedure provides the optimal estimate of the sensor location $\{\overset{*}{\mathbf{r}}{}^b_s, \overset{*}{\mathbf{\Phi}}{}^b_s\}$ after approximately two hundred seconds. The first two rows of Table 4.4 show the hand-measured offset and the optimal estimate. The hand measured offsets are used as the initial values for the numerical optimisation and bounds of ±5° and ±0.1m are specified, for the elements of $\overset{*}{\mathbf{r}}{}^b_s$ and $\overset{*}{\mathbf{\Phi}}{}^b_s$ respectively. For comparison, the optimisation cost metric from Equation 4.13 is calculated for the hand-measured sensor pose as

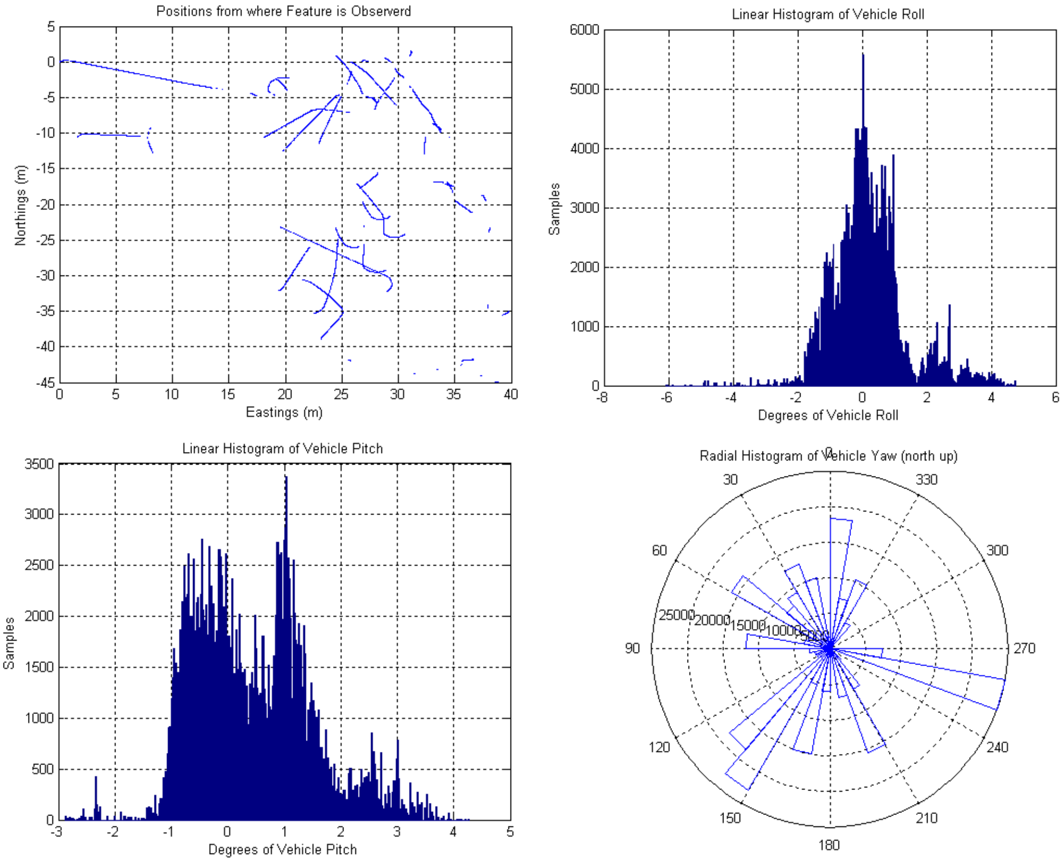


Figure 4.6: The vehicle poses at which the samples in the calibration data-set D_{ground} were obtained. A reasonable spread of vehicle positions, and angles has been achieved.

well as the optimal pose from calibration. The cost has been lowered by an order of magnitude. The point cloud map is reproduced from D_E with the optimal values, and is shown in Figure 4.7. It is clear from inspection that the optimal values are much more accurate than the measured ones, because the pole now appears to be singular and vertical, and the ground plane appears to be thin.

To test the contribution of the pole in isolation, the calibration is repeated, this time using D_{pole} on its own, and the results are listed on the third row of Table 4.4. Similar estimates for the sensor location are produced (in 14 seconds due to the decreased sample size) and the point cloud is shown in Figure 4.8. The pole appears very similar to the one that was produced using the pole and ground data for calibration, yet interestingly, the ground plane is visibly less accurate. This is because the individual

Table 4.4: Measured and Optimised Sensor Pose

	N	C_E	$r^b_{s,x}(m)$	$r^b_{s,y}(m)$	$r^b_{s,z}(m)$	$\phi^b_{s,z}(^\circ)$	$\phi^b_{s,y}(^\circ)$	$\phi^b_{s,x}(^\circ)$
Measured	296096	0.06094	0	0	-1	0	-8	0
$D_{pole} + D_{ground}$	296096	0.00982	0.053	-0.004	-0.998	-1.741	-7.066	-0.072
D_{pole}	37010	0.01086	0.061	-0.004	-0.997	-1.720	-7.822	-0.395
$D_{low\ uncertainty}$	136832	0.00739	0.057	-0.004	-0.999	-1.742	-7.054	-0.059

N is the total number of samples in the data-set. The cost metric C_E is calculated from Equation 4.13, using the sensor pose shown in the table.

features within D_E yield calibration results that are optimal with respect to *those* features. By removing the ground plane data from the calibration, the *mapped* ground plane becomes less accurate. Both calibration results yield a much more accurate sensor pose than can be measured by hand, but for the maximum accuracy, the pole and ground data-sets should both be used.

To test the effect of uncertainty in the calibration data-set, the calibration is performed with a sub-set of D_{pole} and D_{ground}. The data in D_E are ranked by the corresponding uncertainty in M_E, and the more highly uncertain half of the data-set is discarded. The results of calibration with this set are shown on the fourth row of Table 4.4. This method produces a reduction in the final cost metric, because the points with the largest error magnitude have been removed. However, the calibration result is almost identical to the result produced from all of the data. Because the data are gathered from a wide range of vehicle poses, the discarded data are approximately zero-mean about the true geometry, so it has little effect on the results. This technique can be used to reduce the quantity of data to speed up the calibration procedure (28 seconds, down from 200), but is unlikely to have a significant effect on the actual calibration result.

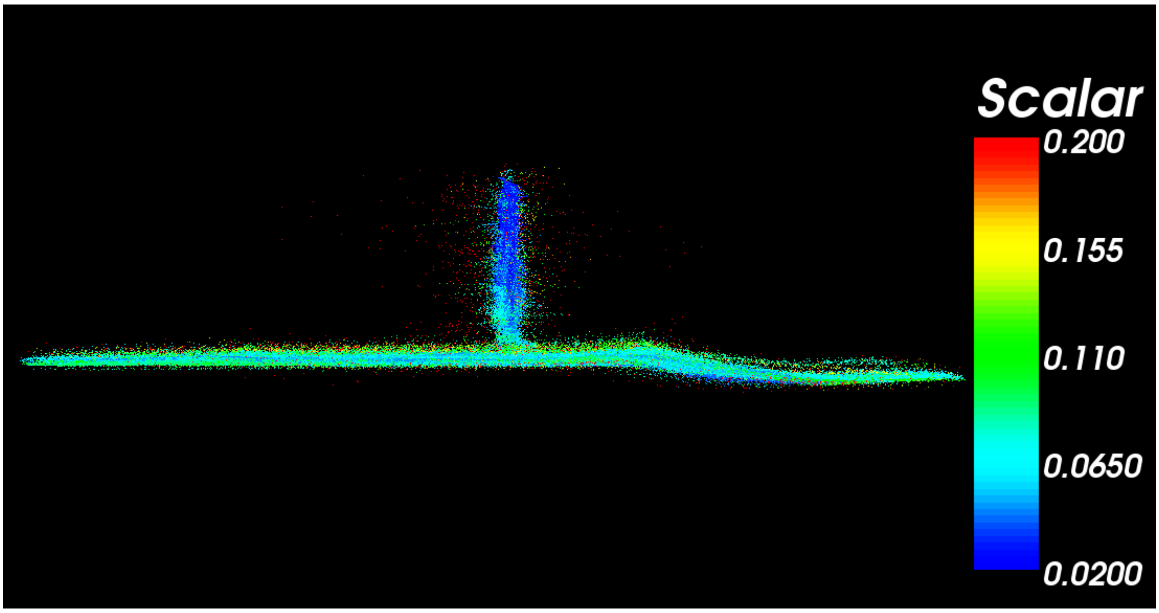


Figure 4.7: The three dimensional point cloud, produced using the optimal sensor pose provided by calibration with D_{pole} and D_{ground}. The pole appears singular and vertical and the ground surface appears to be thin and focussed. The points are coloured by the largest orthogonal standard deviation of uncertainty. The contoured road edge that was used to achieve non-zero vehicle roll and pitch can be seen to the right of the pole.

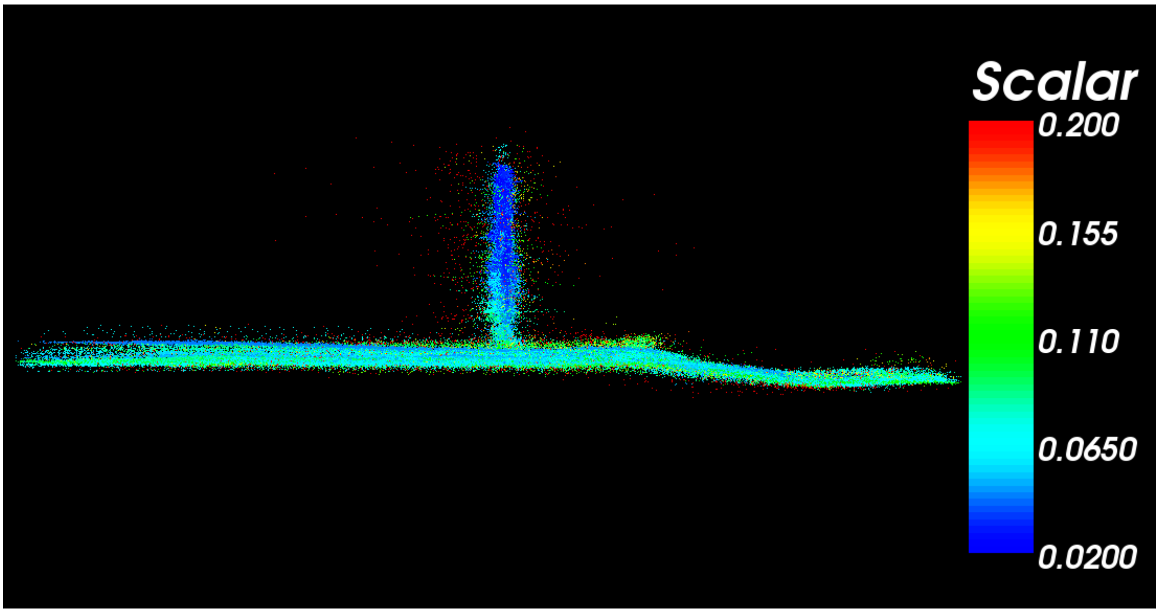


Figure 4.8: The three dimensional point cloud, produced using the optimal sensor pose provided by calibration with D_{pole} only. The pole appears singular and vertical, but the ground surface appears less accurate than in Figure 4.7. The contoured road edge that was used to achieve non-zero vehicle roll and pitch can be seen on the right.

Quantity of Data Required for Calibration

The calibration was done with all of the data in D_E, which contains a total of 296,096 samples, comprised of 37,010 samples in D_{pole} and 259,086 samples in D_{ground}. This section determines if this was enough data to produce globally optimal calibration results, or whether more data are needed. Global optimality is achieved when the values of $\{\overset{*}{\mathbf{r}}{}^b_s, \overset{*}{\mathbf{\Phi}}{}^b_s\}$ are found, such that C_E could not possibly be minimised further. It is important to note that this cost is still dependant on the calibration data-set D_E. Correct values for $\{\overset{*}{\mathbf{r}}{}^b_s, \overset{*}{\mathbf{\Phi}}{}^b_s\}$ can only be determined if D_E is an appropriate data-set (conforming to all constraints and assumptions in Section 4.1.3) *and* a globally optimal solution is found. It is shown here that better calibration results could not be expected with additional data and that only approximately 10,000 samples within D_E are absolutely required.

The calibration was performed repeatedly on progressively smaller random sub-sections of D_E. When sub-sampling, if the discarded data contain redundant information that

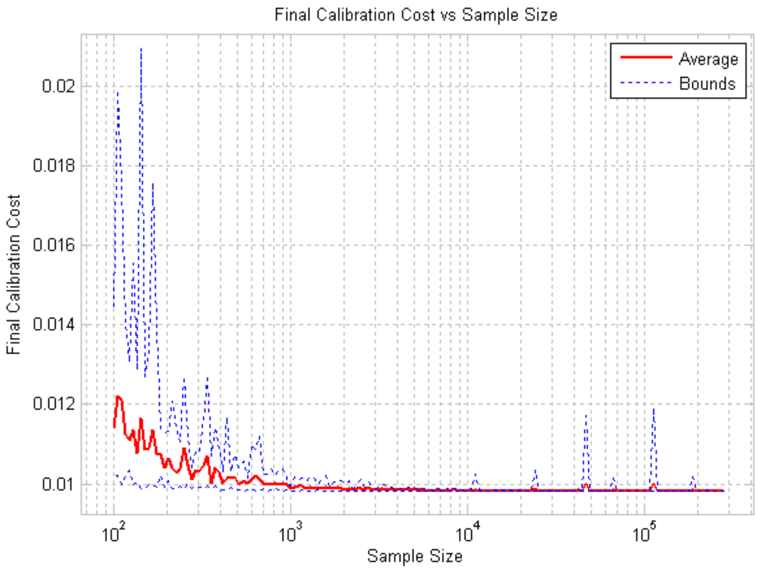


Figure 4.9: The calibration is performed using varying sized sub-sets of the data D_E. The calibration results are then evaluated by calculating the cost metric in Equation 4.13 with respect to the *complete* data-set. For each sample size, ten random sub-sets of that size are selected from D_E. The minimum, maximum and average cost are shown plotted against the sample size.

is replicated in the remaining sub-set, then the calibration will not be significantly affected. On the other hand, if the discarded samples contain unique information this will have a larger effect. Due to the random selection process, a smaller sub-set may perform better than a larger set. To avoid skewing the results, the calibration was run ten times for each sample size, with a different random set discarded each time. The resulting sensor pose from each of the ten calibrations was then used to map the *entire* set, and the cost function was calculated for that. The calibration was run on the sub-set, but evaluated on all of the available data, to determine whether the sub-set spanned the complete range of free parameters. Figure 4.9 shows the evaluated cost metric for each sample size. For each of the ten runs, the average, largest and smallest cost is plotted against the sample size. The cost asymptotes, indicating that after approximately ten thousand samples, little improvement in calibration can be obtained. This suggests that the calibrated solution has converged to the correct *global* minima. If the solution settled on a local minima with respect to one of the sub-sets, a higher cost would be expected when additional data (in this case the data from D_E that were discarded) were added. This effect can also be seen in Figure 4.10, which shows the optimal values for each of the six parameters of the sensor location $\{\mathbf{r}_s^b, \mathbf{\Phi}_s^b\}$, as provided by calibration with the varying sized sub-sets. After ten thousand samples, the fluctuation of the values has settled, indicating a steady state has been reached. The objective function of Equation 4.3 is certainly non-convex for small sample sizes, as seen by the convergence to local minima, yet as the sample size increases, the function is smoothed until convergence to a consistent global minima is possible.

The fact that this particular calibration data-set requires ten thousand samples cannot be generalised to all data-sets from all artificial environments. For example, a similar method was used in [50] to determine that a larger sample size of approximately sixty thousand was required for calibration. However, the sub-sampling methodology that is used here is generally applicable, and is a necessary step to ensure that the globally optimal sensor location has been determined.

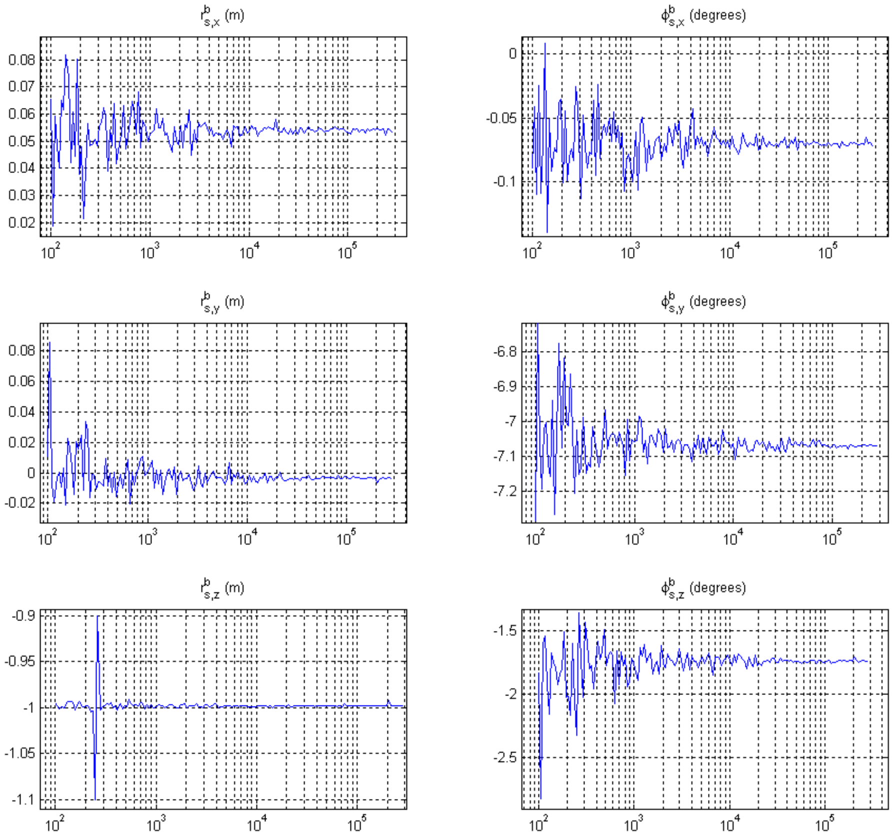


Figure 4.10: The calibration is performed using varying sized sub-sets of the data D_E. The six calibrated parameters of $\{\mathbf{r}^b_s, \mathbf{\Phi}^b_s\}$ are plotted against the sample size (the X axis for all figures), illustrating the convergence to the final solution that occurs once enough data are obtained.

Processing Time

The processing time required for the calibration as a function of sample size is shown in Figure 4.11. The time taken to process one iteration of the numerical minimisation is linear with respect to the sample size, however the number of iterations can vary significantly due to the random sampling process. To process the entire data-set takes fewer than four minutes on a 2GHz personal computer, so it is not recommended to use the sub-sampling technique to reduce the processing time. If sub-sampling is required to reduce processing time, enough data must remain to create the asymptote seen in Figure 4.9.

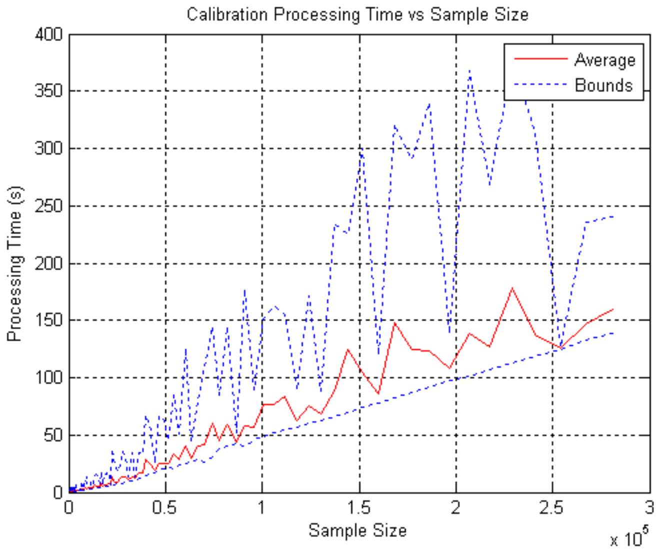


Figure 4.11: The calibration processing time for varying sample sizes. For each sample size, ten random sub-sets of that size are selected from D_E, and the calibration procedure is timed for all ten. The minimum, maximum and average times are shown. The processing time for one iteration of the numerical minimisation is linear with respect to sample size, but the number of iterations required can change as a result of the random selection process.

Calculation of Parameter Uncertainty

The optimal solution was given in Table 4.4, and it was established with reference to Figures 4.9 and 4.10 that enough data were available in the calibration data-set to consider the solution valid. It is therefore appropriate to use the same data-set to determine the uncertainty of the solution given the data, using the Monte Carlo importance sampling technique that was discussed in Section 4.1.2.

The importance sampling technique requires a likelihood function $\Lambda(\mathbf{x})$ to be specified. This must be performed for each application because of the dependance on the particular choice of the artificial environment and on the corresponding cost function. For this application, the likelihood of a single point in the calibration data-set D_E is given with respect to the Euclidean distance of the point to its corresponding feature, and the uncertainty of the point due to all sources *other than* the sensor pose location itself. To achieve this, the uncertainty due to the sensor location is temporarily ignored by setting the corresponding variance terms for $\mathbf{Q}$ in Equation 3.39 to zero sub-matrices:

$$\boldsymbol{\sigma}^2_{\mathbf{r}^b_s\mathbf{r}^b_s} = \boldsymbol{\sigma}^2_{\mathbf{r}^b_s\boldsymbol{\Phi}^b_s} = \boldsymbol{\sigma}^2_{\boldsymbol{\Phi}^b_s\mathbf{r}^b_s} = \boldsymbol{\sigma}^2_{\boldsymbol{\Phi}^b_s\boldsymbol{\Phi}^b_s} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.22}$$

Equation 3.21 provides the covariance matrix for each point. To simplify the implementation of the importance sampling technique, the uncertainty $\boldsymbol{\Sigma}_{\mathbf{p}^n}$ of all N points in D_E is calculated by Equation 3.21, without consideration of the sensor pose uncertainty, and then averaged according to:

$$\sigma_{D_E} = \frac{\sum_{i=1}^{N} \sqrt{max(eigenvalue(\boldsymbol{\Sigma}_{\mathbf{p}^n,i}))}}{N} \tag{4.23}$$

where for each point, the maximum eigenvalue is taken as a worst case variance. The root of these values are averaged over the entire calibration data-set to provide an approximate average standard deviation, which is conservative because the maximum variance for each point is considered.

The likelihood function is then specified by the combined probability of each point being located at the Euclidean distance d from its corresponding feature, given the average uncertainty due to all other sources of error:

$$\begin{aligned} \Lambda(\mathbf{x}) &= \prod_{i=1}^{N} \mathcal{N}(d_i, 2\sigma^2_{D_E}) \\ \log\Lambda(\mathbf{x}) &= -\sum_{i=1}^{N} d_i^2/\sigma^2_{D_E} \end{aligned} \tag{4.24}$$

The variance of the Gaussian in this equation is given as twice the average that was determined above, because the location of the environment is determined from the data, not from absolute coordinates. It is therefore plausible that the distance d is due to the sum of an error in the point location and the location of the feature itself. The variance is doubled to account for the convolution of these two errors, each with a variance of $\sigma^2_{D_E}$. Log probabilities are required for large sample sizes, because the product of N probabilities becomes numerically unstable for large N.

The complete covariance matrix is calculated using importance sampling with 100,000 samples yielding:

$$\mathbf{Q}_{\{r^b_s,\Phi^b_s\}} = 10^{-3} \times \begin{bmatrix} 0.00946 & -0.00137 & -0.04024 & 0.00017 & -0.00012 & 0.00004 \\ -0.00137 & 0.01464 & 0.02519 & -0.00134 & -0.00009 & 0.00001 \\ -0.04024 & 0.02519 & 4.53530 & -0.00188 & -0.00014 & -0.00011 \\ 0.00017 & -0.00134 & -0.00188 & 0.00038 & -0.00001 & 0.00001 \\ -0.00012 & -0.00009 & -0.00014 & -0.00001 & 0.00016 & -0.00003 \\ 0.00004 & 0.00001 & -0.00011 & 0.00001 & -0.00003 & 0.00002 \end{bmatrix} \tag{4.25}$$

This can be provided as part of the whole system uncertainty $\mathbf{Q}$ in Equation 3.19 or 3.39, with:

$$\mathbf{Q}_{\{r^b_s,\Phi^b_s\}} = \begin{bmatrix} \boldsymbol{\sigma}^2_{\mathbf{r}^b_s\mathbf{r}^b_s} & \boldsymbol{\sigma}^2_{\mathbf{r}^b_s\boldsymbol{\Phi}^b_s} \\ \boldsymbol{\sigma}^2_{\boldsymbol{\Phi}^b_s\mathbf{r}^b_s} & \boldsymbol{\sigma}^2_{\boldsymbol{\Phi}^b_s\boldsymbol{\Phi}^b_s} \end{bmatrix} \tag{4.26}$$

The orthogonal standard deviations are calculated as $\sqrt{diag(\mathbf{Q}_{\{r^b_s,\Phi^b_s\}})}$, and then converted from radians to degrees. Table 4.5 shows the weighted average pose and standard deviations, with the optimal pose repeated for comparison. The mean value is very close to optimal location, indicating that the distribution is close to zero mean about this region. The standard deviations are all in the order of millimetres and hundredths of a degree, other than the vertical offset which has a standard deviation of 6.7cm. This result confirms the lower sensitivity of the calibration to this parameter that was determined in Section 4.1.3. The Laplace method was also performed for each of the six parameters in isolation in Appendix C, using the same likelihood function. As expected, the resulting standard deviations seen in Table 4.5 are very similar to the diagonals of the MC covariance matrix.

Table 4.5: Sensor Pose: Optimised and Monte Carlo Mean and Standard Deviation

	$r^b_{s,x}(m)$	$r^b_{s,y}(m)$	$r^b_{s,z}(m)$	$\phi^b_{s,z}(^\circ)$	$\phi^b_{s,y}(^\circ)$	$\phi^b_{s,x}(^\circ)$
$D_{pole} + D_{ground}$	0.053	-0.004	-0.998	-1.741	-7.066	-0.072
$\hat{\mathbf{x}}$ (MC)	0.054	-0.003	-0.990	-1.742	-7.063	-0.071
$\boldsymbol{\sigma}$ (MC)	0.0031	0.0038	0.0673	0.0353	0.0232	0.0085
$\boldsymbol{\sigma}$ (Laplace)	0.0029	0.0030	0.0673	0.0258	0.0202	0.0062

The weighted mean sensor location $\hat{\mathbf{x}}$ and standard deviations $\boldsymbol{\sigma}$ from the Monte Carlo (MC) sampling process in Equation 4.8 and the standard deviations from the Laplace method in Appendix C. The optimal pose from Table 4.4 is repeated here for comparison.

Several assumptions are made in this section to simplify the calculation of the covariance matrix:

- The maximum eigenvalue is taken as an upper estimate for the global spatial uncertainty of each point.
- The uncertainty of the Euclidean distance of each point to the matching environment feature is averaged over the calibration data-set, rather than using the specific uncertainty of each point.
- The distance uncertainty is doubled because it is comprised of the unknown location of the feature to the unknown location of the point.

The importance sampling technique fits a multivariate Gaussian distribution to the data, which is only an approximation of the true distribution. Although each of the assumptions above further reduces the accuracy of the final covariance matrix, they do so in a way that *overestimates* the amount of error, thus providing a conservative estimate. The true variances are smaller than the ones provided by this method. The assumptions above can be reduced at the cost of implementation and processing time, but this is unlikely to provide a significantly more accurate covariance matrix, given that the Gaussian distribution only approximately represents the true distribution.

4.1.5 Calibration Results for Multiple Sensors

The single and multiple sensor calibration procedures were described in Section 4.1.1 and the single calibration was experimentally validated in Section 4.1.4. This section presents the calibration results for four sensors, confirming that the process improves the estimate of the sensor locations. It is shown that in addition to minimising the systematic errors for each sensor, the *systematic contradiction* between all of the sensors is also reduced, improving the results of any sensor fusion technique.

The configuration that was used for the single sensor calibration is extended, with the addition of three more SICK LMS laser range scanners, as shown in Figure 4.12. The multi-sensor calibration data-set $\{D_{Es1}, D_{Es2}, D_{Es3}, D_{Es4}\}$ includes the same data for the first sensor that was used in Section 4.1.4. The calibration is implemented by numerically optimising Equation 4.5 with the same SQP method [49] used previously.

The Calibration Results

The multi-sensor calibration procedure provides the optimal estimate of all four sensor locations $\{\overset{*}{\mathbf{r}}{}^b_{si}, \overset{*}{\mathbf{\Phi}}{}^b_{si}\}, i \in \{1, 2, 3, 4\}$ after 27 iterations and approximately 48 minutes. This is much longer than the 200 seconds taken to calibrate a single sensor, due to the increase in the dimensionality of the optimisation from 6 to 24 parameters. Table 4.6 shows the initial hand-measured sensor locations, and Table 4.7 presents the optimally calibrated results. Recall that the combined multi-sensor optimisation technique uses Equation 4.5, in which the data from *all* sensors are fused into a single map M_E. The cost to be minimised is calculated on this fused map, so there is one cost for the set of sensors. The cost metric in each row of the tables is calculated from Equation 4.13. This is the cost of the map produced from the individual sensor's data, not the complete cost metric for the whole calibration. This cost value can be

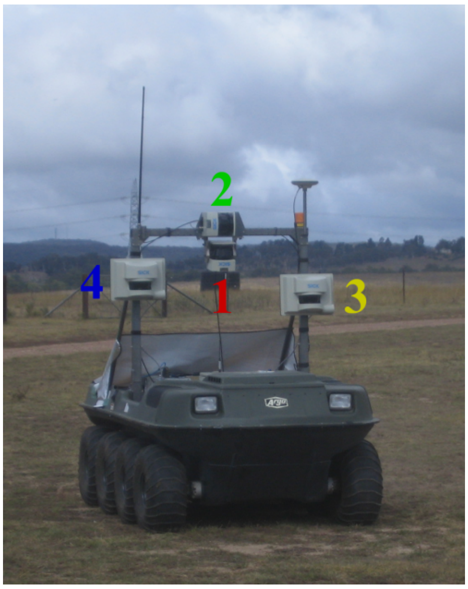


Figure 4.12: The CAS Outdoor Research Demonstrator platform with four enumerated SICK LMS 2D laser range scanners. The numbers are coloured to allow point clouds in subsequent figures to be coloured according to the sensor from which they originate. Sensor 1 was calibrated independently in Section 4.1.4. In this section, the four sensors are calibrated jointly and independently for comparison. Refer to Table 4.6 for the approximate positions of the sensors with respect to the vehicle body frame.

compared directly with the results from the single sensor calibration. The cost metric of all of the sensors has been reduced by an order of magnitude with respect to the initial hand measured values.

Figure 4.13 shows a comparison of the mapped data, before and after calibration, with the points coloured according to the sensor that produced them. These colours match the enumeration in Figure 4.12. It can clearly be seen that the calibration has improved the point cloud, and that after calibration, all four sensors produce a consistent picture of the pole. A similar comparison can be made in Figure 4.14, with the point cloud seen from the top. This figure shows that after calibration, the points from the sensors are mapped to the same consistent ground plane. Before calibration, this is not the case, as the occlusion of the blue points shows that they are *consistently* mapped beneath the data from the other sensors. Figures 4.13 and 4.14 illustrate a reduction in both systematic error (as the pole is singular and vertical and the ground plane is flat) and systematic contradiction between sensors, because they all map to

Table 4.6: Measured Sensor Poses

	N	C_E	$r^b_{s,x}(m)$	$r^b_{s,y}(m)$	$r^b_{s,z}(m)$	$\phi^b_{s,z}(^\circ)$	$\phi^b_{s,y}(^\circ)$	$\phi^b_{s,x}(^\circ)$
Sensor 1	296096	0.06094	0	0	-1	0	-8	0
Sensor 2	93684	0.02408	0	0	-1.15	0	0	90
Sensor 3	39598	0.03434	0.12	-0.55	-0.68	0	5	0
Sensor 4	87555	0.14058	0.12	0.55	-0.75	0	-5	0

Table 4.7: Calibrated Sensor Poses (Combined Calibration)

	N	C_E	$r^b_{s,x}(m)$	$r^b_{s,y}(m)$	$r^b_{s,z}(m)$	$\phi^b_{s,z}(^\circ)$	$\phi^b_{s,y}(^\circ)$	$\phi^b_{s,x}(^\circ)$
Sensor 1	296096	0.00987	0.054	-0.002	-0.900	-1.741	-7.175	-0.029
Sensor 2	93684	0.00823	0.092	0.003	-1.219	-1.174	-0.270	88.469
Sensor 3	39598	0.00612	0.148	-0.554	-0.78	-1.761	4.726	0.288
Sensor 4	87555	0.01058	0.170	0.546	-0.823	-2.259	-4.722	0.135

Table 4.8: Individually Calibrated Sensor Poses (Single Sensor Calibration)

	N	C_E	$r^b_{s,x}(m)$	$r^b_{s,y}(m)$	$r^b_{s,z}(m)$	$\phi^b_{s,z}(^\circ)$	$\phi^b_{s,y}(^\circ)$	$\phi^b_{s,x}(^\circ)$
Sensor 1	296096	0.00982	0.053	-0.004	-0.998	-1.741	-7.066	-0.072
Sensor 2	93684	0.00782	0.060	0.006	-1.148	-1.184	-0.284	88.999
Sensor 3	39598	0.00574	0.149	-0.551	-0.682	-1.776	4.423	0.344
Sensor 4	87555	0.01038	0.165	0.550	-0.749	-2.280	-4.784	0.211

N is the total number of samples in the data-set D_{Esi}. The cost metric C_E is calculated from Equation 4.13, using the sensor pose shown in the table.

the same geometric representation of the world after calibration.

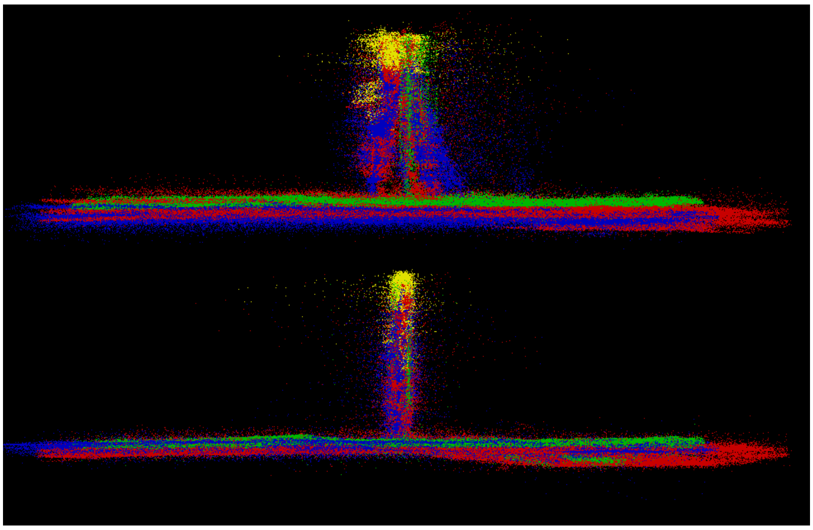


Figure 4.13: A side view of the point cloud produced by four laser range sensors before calibration (top) and after calibration (bottom). The points are coloured according to the sensor that produced them: Red 1, Green 2, Yellow 3, Blue 4. Refer to Table 4.6 for the approximate locations of the sensors on the mobile platform. After calibration, all sensors map data to a single, consistent, vertical pole.

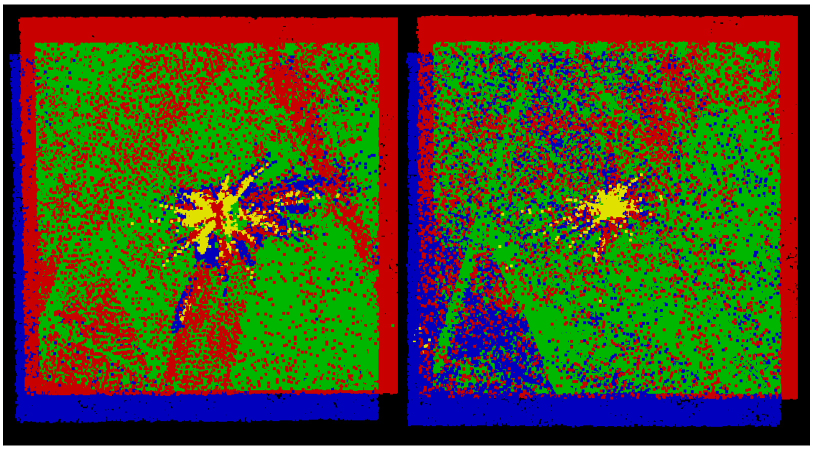


Figure 4.14: A top down view of the point cloud produced by four laser range sensors before calibration (left) and after calibration (right). The points are coloured according to the sensor that produced them: Red 1, Green 2, Yellow 3, Blue 4. Refer to Table 4.6 for the approximate locations of the sensors on the mobile platform. Before calibration, the 'blue' sensor data are mapped to a plane that is beneath the others, as the blue points are occluded. After calibration, all of the sensor data map to the same plane, as seen by the interwoven red, green and blue points. The 'yellow' sensor is angled up by five degrees, and does not produce data at the base of the pole.

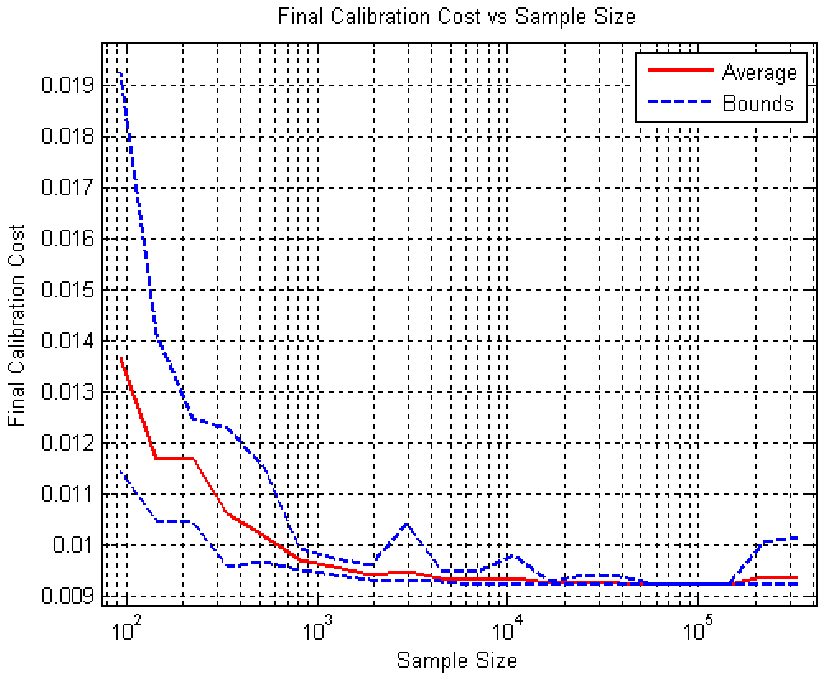


Figure 4.15: The combined multi-sensor cost metric from Equation 4.4, calculated for various sizes of subset of the calibration data.

To confirm that enough data are available for the calibration, the technique of subsampling from Section 4.1.4 is performed. Recall that this technique calibrates the sensor pose with varying sized subsets of the calibration data. These results are then used to produce the fused map and combined cost metric according to Equation 4.4. Although the calibration is performed with a subset of data, the cost metric is evaluated with the *whole* data-set to determine if the subset was able to span the necessary input space. Figure 4.15 shows the combined multi-sensor cost metric for different quantities of data. An asymptote has been reached at approximately ten thousand samples, which suggests that there are enough data for the multi-sensor calibration.

Benefit of Joint Sensor Calibration

In Section 4.1.1, the general calibration procedure for multiple sensors was described. The sensors can either be calibrated individually using the single sensor technique, or all of the sensors can be calibrated together using the multi-sensor calibration procedure. The joint calibration procedure of minimising the error in the *fused* map with Equation 4.5 is recommended, in order to reduce the *systematic contradiction* between sensors. This section compares the results of using the single sensor calibration multiple times, to using the combined multiple sensor calibration technique once. It will be seen that both techniques reduce the systematic error of each sensor, but that the combined technique is required to minimise the contradiction between the sensors, allowing for reliable multi-sensor fusion algorithms.

The single sensor calibration was performed for each of the four sensors and the results are shown in Table 4.8. Note that the results for the first sensor in this table are identical to the second row of Table 4.4, because the same data and process were used in both cases. The results of the individual calibrations are similar to the sensor locations provided by the multi-sensor calibration. The individual costs are lower for the single calibrations because the objective function is chosen to minimise the individual cost metric, whereas the multi-calibration is designed to minimise the error in the fused map. In other words, the multi-sensor calibration achieves a reduction in error in the fused map, at a slight cost to the individual optimality. This is best illustrated by visualising the fused point cloud map. Figure 4.16 shows the combined map from the four sensors, coloured according to the source of the points, according to the enumeration in Figure 4.12. The single sensor calibration results are used to produce the map seen on top, and the joint calibration produces the map on the bottom. In both cases, if the data from a single sensor (a single colour) are considered, then the pole appears singular and vertical, and the ground plane is thin and flat, indicating that the calibration has succeeded in reducing the systematic error of *each* sensor. However, when the data from *all* of the sensors are considered together, the joint calibration yields more consistent results. The data from all of the sensors produce a consistent and aligned map, which shows that

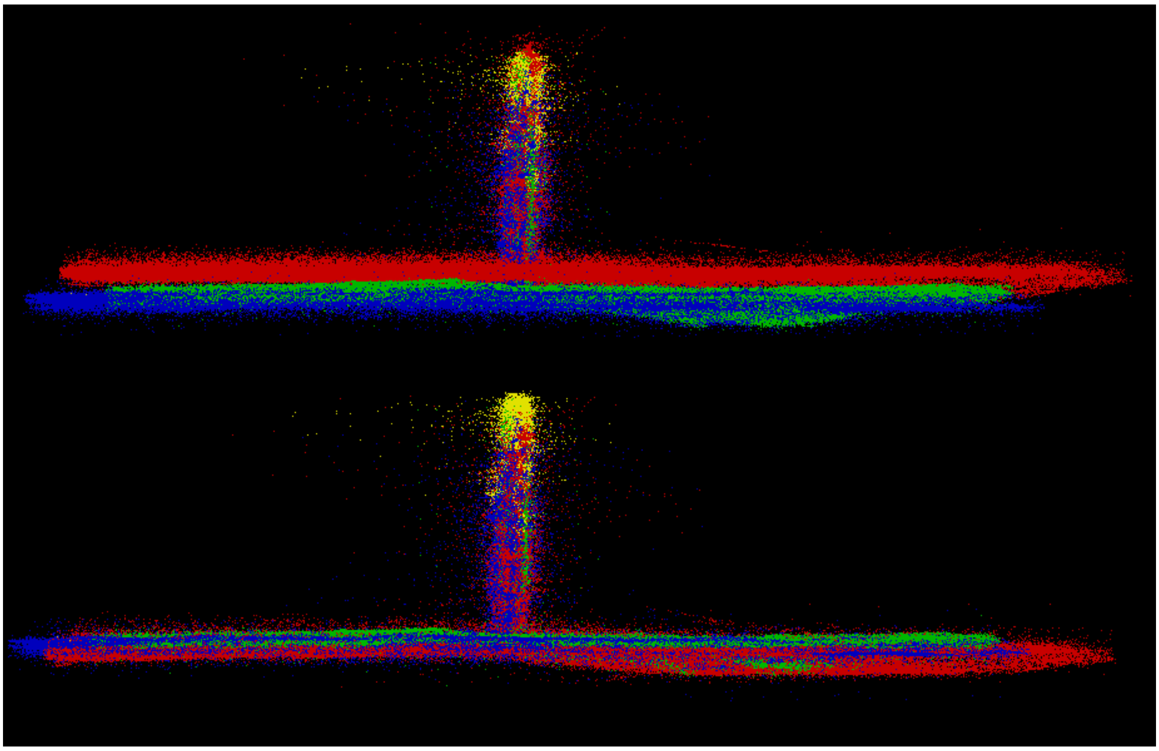


Figure 4.16: A side view of the point cloud produced by four laser range sensors, after calibrating each one individually (top) and after calibrating all of them together (bottom). The points are coloured according to the sensor that produced them, referring to the enumeration in Figure 4.12. Both figures illustrate that the *individual* sensor data are mapped well, because the pole is singular and vertical, and the ground plane is thin and flat. However, the combined calibration (bottom) provides a more consistent map when *all* of the data are fused together.

the systematic contradiction has been minimised. Whenever data fusion is to be performed, the results will be greatly improved by performing the joint calibration technique described in this chapter.

The calibration procedure in this section allows the data from multiple sensors to be reliably fused, despite the sensors having potentially very different geometric configurations on the mobile platform. The procedure minimises the systematic error by accurately calculating the sensor positions and allows for any remaining error to be tracked. These sensor pose parameters can now be used to map all of the data from all of the sensors in a single consistent fused map. This is illustrated in Figures 4.17, 4.18 and 4.19, in which the mapped point clouds from the four laser sensors show the

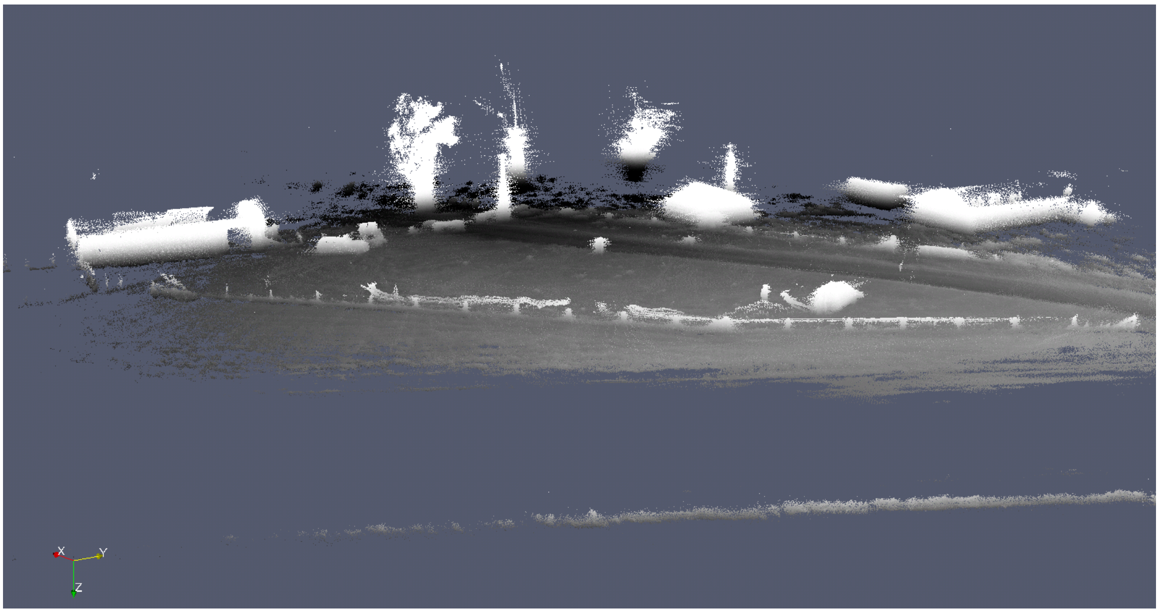


Figure 4.17: The combined mapped point clouds from the four laser range scanners, showing the wider scene around the calibration pole. All of the data are shown, with no threshold on uncertainty.

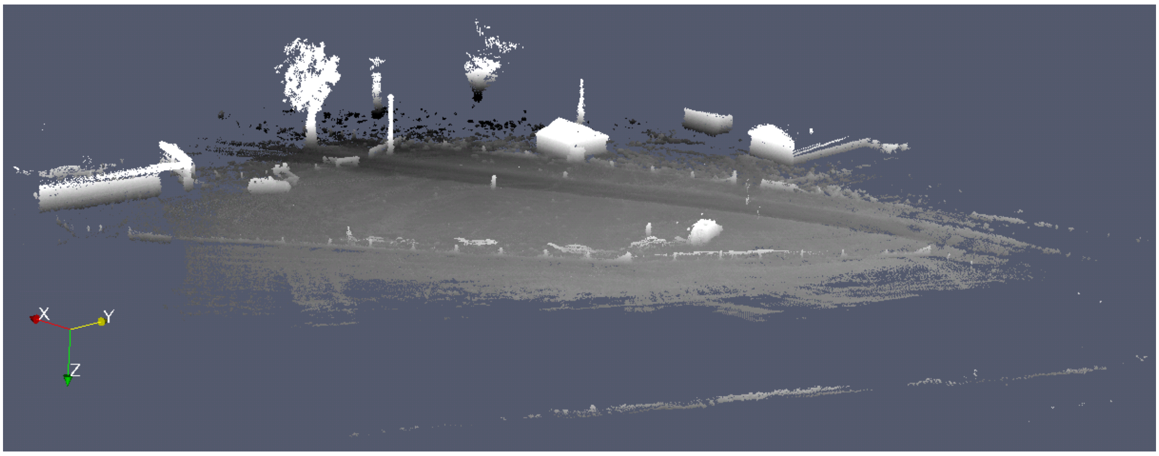


Figure 4.18: The combined mapped point clouds from the four laser range scanners, showing the wider scene around the calibration pole. A threshold of 20cm is applied to the largest orthogonal standard deviation of spatial uncertainty. Compared to Figure 4.17, this map appears to be more focussed.

wider calibration scene. In Figure 4.17, all of the data can be seen, whereas in Figures 4.18 and 4.19, a threshold of 20cm is applied to the standard spatial deviation of the points. With the threshold applied, the maps are visibly more focussed and accurate, illustrating the importance of rigorously calculating the mapped uncertainty.

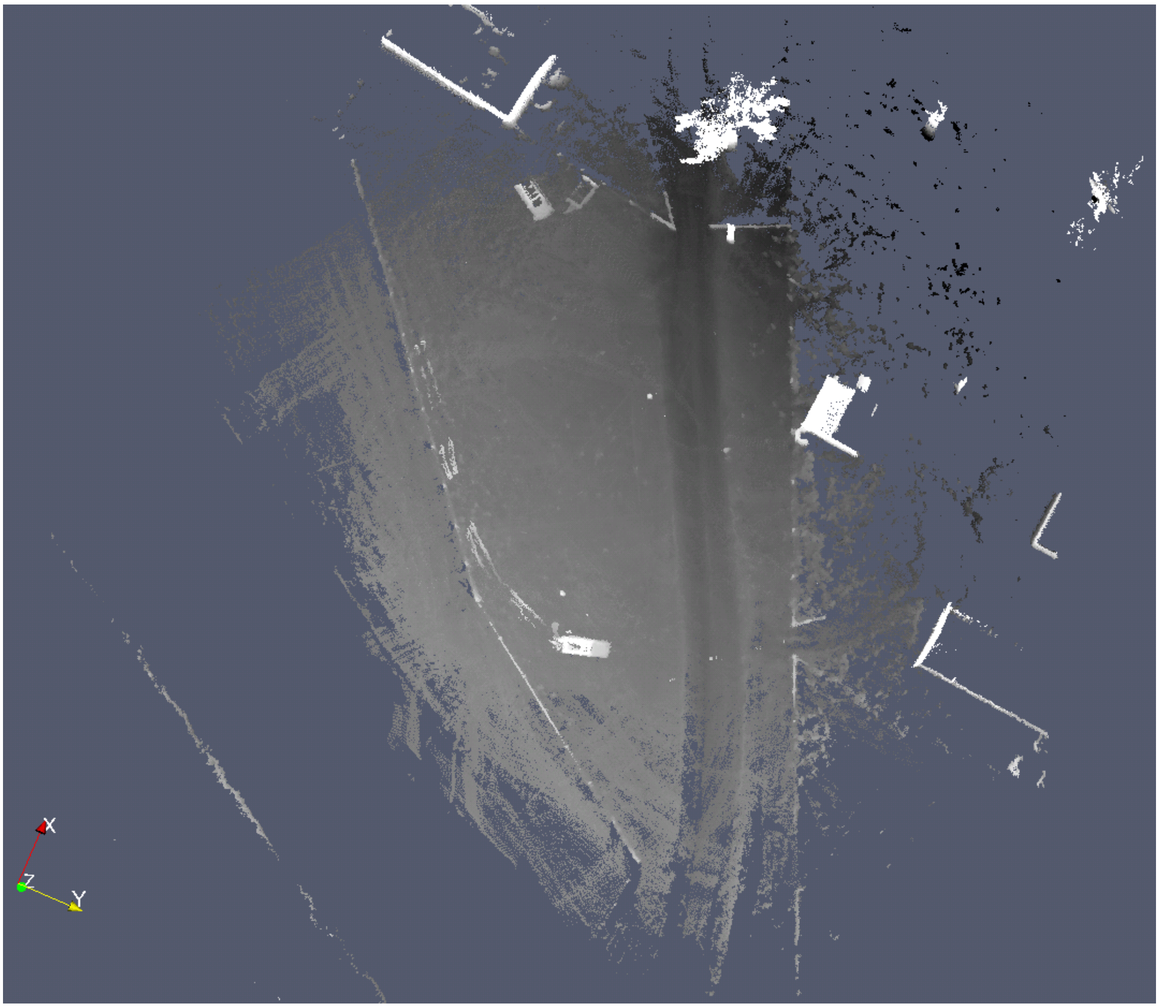


Figure 4.19: The combined mapped point clouds from the four laser range scanners, showing the wider scene around the calibration pole, seen from above. A threshold of 20cm is applied, as in Figure 4.18. The thin, focussed edges of walls and fence lines reveal the accuracy of the mapped data.

4.1.6 Calibration Results for Heterogenous Sensors

The single and multiple sensor calibration procedures were described in Section 4.1.1 and experimentally evaluated in Sections 4.1.4 and 4.1.5. In the latter, four homogenous sensors were calibrated independently and jointly and the calibration results were compared. It was shown that the independent calibrations are capable of minimising the systematic errors in each sensor, but that the joint calibration is required to minimise the systematic contradictions between sensors, to enable precise data fusion. In this section two heterogenous sensors (a radar and a laser scanner) are calibrated. The physical properties of the radar create difficulties in observing some of the features in the environment, resulting in a poor independent calibration result for that sensor. It is shown that by jointly calibrating the laser and the radar, the additional laser information allows *both* sensors to be calibrated more precisely. This is analogous to using the laser to survey the environment to construct the 'known' calibration features for the radar, but this is optimally achieved by the joint calibration procedure without any modification to the algorithm. Neither sensor is calibrated before the algorithm is run, both are calibrated together by the procedure.

Figure 4.20 shows the location of the scanning radar and scanning laser that are used in this section. The radar is a custom built 94GHz frequency modulated carrier wave (FMCW) 2D range scanner. It provides intensity of the reflected radar energy from the environment along the radar beam axis, in approximately 25cm bins. This can be used to generate a range estimate similar to the laser scanner, by extracting the range of the highest intensity peak. This range estimate is refined to a higher precision than the 25cm bin spacing by fitting a quadratic about the peak. In this configuration, the radar data are similar to the laser scanner data, although the range and angular precision is lower for the radar due to the wider beam width.

The laser is in approximately the same position as the one used in Section 4.1.4, however the equipment was remounted between the two experiments, causing a small change in the sensor pose. The radar is mounted in a similar way to the laser, looking forward and down from the horizon at approximately ten degrees. The radar scanner rotates in the opposite direction to the laser, so a right handed frame is constructed

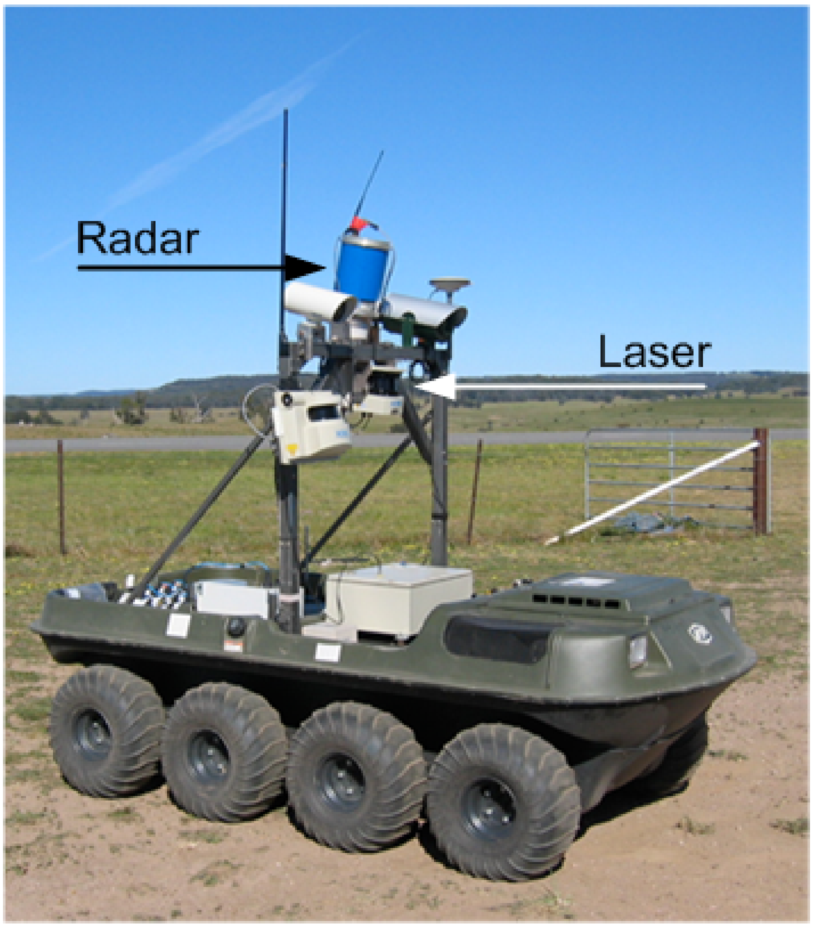


Figure 4.20: The CAS Outdoor Research Demonstrator platform. The location of the scanning radar and 2D laser range scanner are shown.

with the positive Z axis pointing upwards, the X axis pointing to the rear of the vehicle and the Y axis pointing to the right when facing forwards. The vertical pole features that were used to calibrate the laser are specular reflectors for the radar beam, so they return insufficient information for calibration. The vertical metal wall of a nearby shed was found to be a better feature, so the perpendicular wall and ground planes were used as calibration features. The location and orientation of the wall had been surveyed previously, so the average squared Euclidean distance of points to the wall was used for the cost function. The sensor calibration routine was applied to the radar independently as per Section 4.1.4 and the result is shown in Table 4.9. The measured pitch angle of 190 degrees corresponds to the 10 degrees of declination, but is added to 180 degrees due to the inverted scan axis.

Table 4.9: Measured and Independently Calibrated Radar Pose

	$r^b_{s,x}(m)$	$r^b_{s,y}(m)$	$r^b_{s,z}(m)$	$\phi^b_{s,z}(^\circ)$	$\phi^b_{s,y}(^\circ)$	$\phi^b_{s,x}(^\circ)$
Measured	0	0	-1.5	180	188	0
Optimised	0.1151	0.0438	-1.3002	177.2977	183.1803	-0.1550

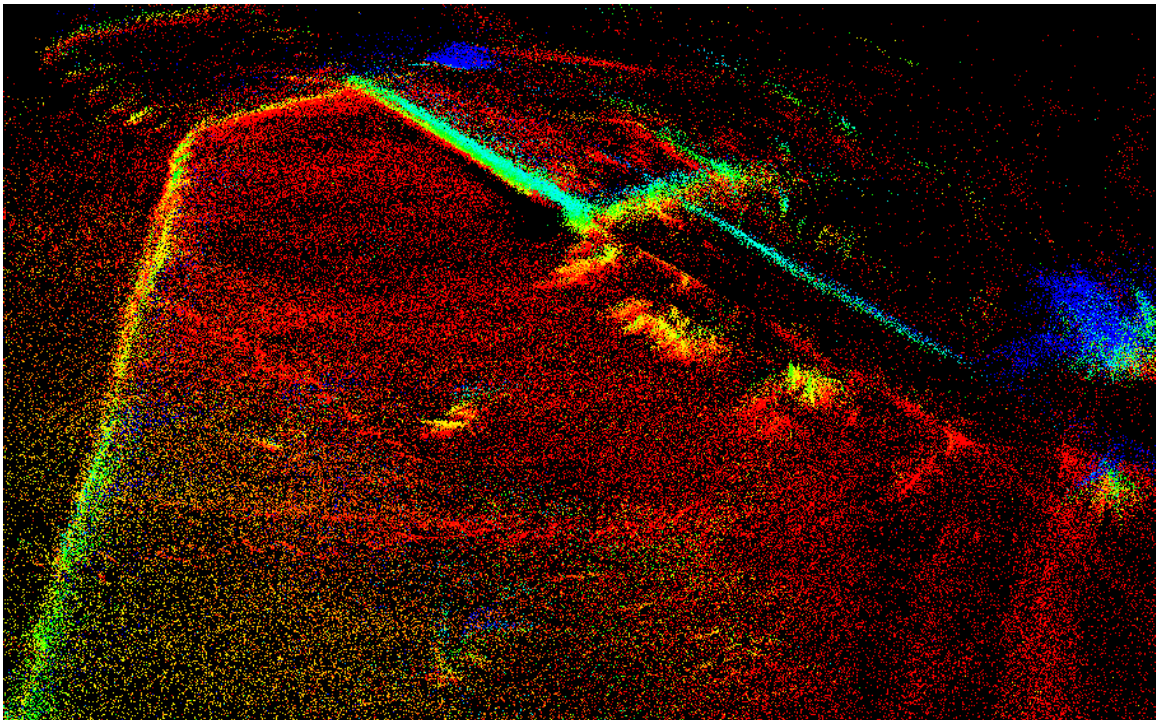


Figure 4.21: The mapped radar data coloured by elevation, using the results from the independent calibration procedure (see Table 4.9). The radar produces more sensor noise than the laser, but this figure indicates blurring to a greater extent than should be expected from the error analysis in Chapter 3.

The calibrated offsets are close to their measured value, but the pitch of 183 corresponds to a declination of only 3 degrees which is measurably incorrect. The radar data were mapped using the independent sensor calibration result from Table 4.9 and the point cloud map is displayed in Figure 4.21. The radar has a lower range and bearing accuracy than the laser, which can be modeled by larger values for $\boldsymbol{\sigma}^2_{\mathbf{p}^s}$ in Equations 3.19 and 3.39. However, the point cloud is blurred to a greater extent than should be expected from the error analysis in Chapter 3.

The calibration was redone jointly with the laser and the radar, using the multi-sensor calibration method from Section 4.1.5. The laser and radar are jointly optimised

Table 4.10: Jointly Calibrated Radar and Laser Pose

	$r^b_{s,x}(m)$	$r^b_{s,y}(m)$	$r^b_{s,z}(m)$	$\phi^b_{s,z}(^\circ)$	$\phi^b_{s,y}(^\circ)$	$\phi^b_{s,x}(^\circ)$
Measured Radar	0	0	-1.5	180	188	0
Optimised Radar	-0.0121	-0.0203	-1.4975	173.7784	189.8864	0.2369
Measured Laser	0	0	-0.9	-1.7	-8	0
Optimised Laser	-0.0077	-0.0334	-0.8996	-2.4567	-8.0178	-0.6420

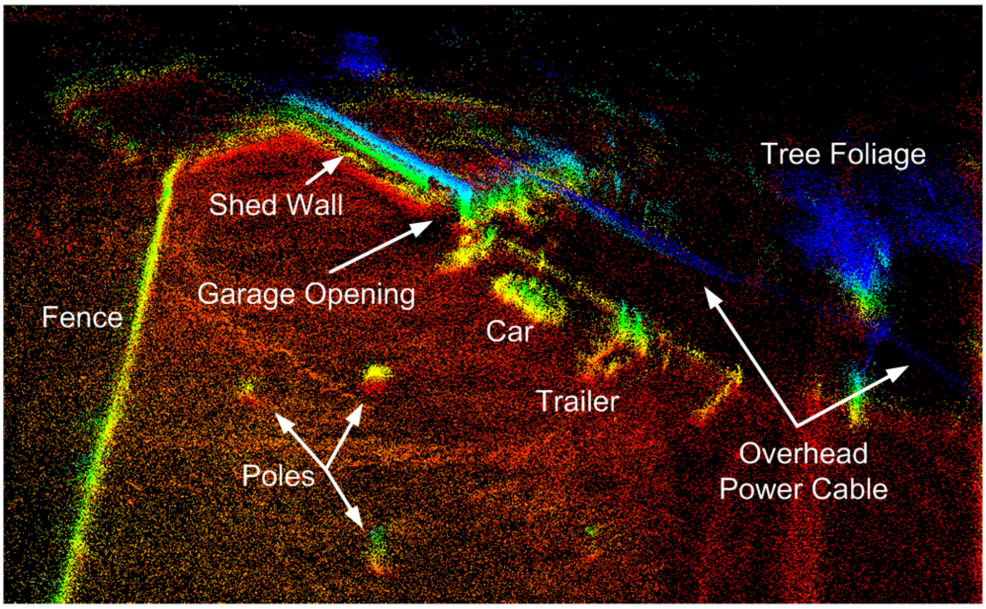


Figure 4.22: The mapped radar data coloured by elevation, using the results from the joint calibration procedure (see Table 4.10). This point cloud is more sharply focussed than the one in Figure 4.21, which was produced by calibrating the radar independently. Many features in the scene are now clearly identifiable.

against the metal shed wall and the ground plane. The resulting offsets for the laser and radar are shown in Table 4.10. Other than the yaw angle, the radar offsets and angles are closer to the measured values than the independent results in Table 4.9. The yaw angle differs from the measured value by more than was initially expected, but this is likely due to a known time synchronisation issue between the custom built radar scanner encoder and the radar range unit, that causes a constant angular offset about the rotation axis. The calibration routine has in effect compensated for this offset. The radar point cloud is mapped with these results and shown in Figure 4.22. The point cloud is now much more focussed, to the extent that many of the features in the scene are now clearly identifiable.

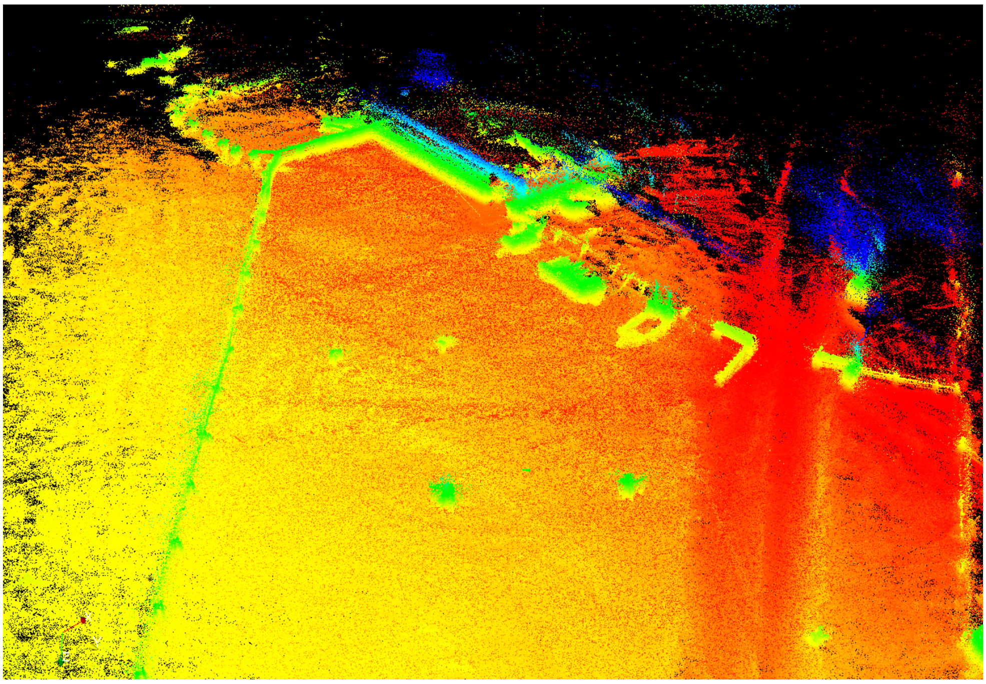


Figure 4.23: The mapped radar and laser data coloured by elevation, using the results from the joint calibration procedure (see Table 4.10). With the sensors calibrated jointly, the systematic errors and contradictions are minimised, so the data can be fused reliably.

As discussed in Section 4.1.1, because the sensors have been jointly calibrated, the systematic errors in each has been minimised and the radar sensor in particular has benefited from the quality of the laser data. The systematic contradiction has also been minimised so that all of the data from both sensors can be fused together reliably. The mapped laser and radar data are produced from the joint calibration results and displayed in Figure 4.23. All of the features can be seen without any significant blurring due to misalignment.

In situations where multiple sensors are to be used (whether different modalities or not) the joint calibration technique allows reliable sensor fusion. In addition, joint calibration provides a method of leveraging the precision of one sensor to assist the calibration of another. This allows the data from both to be fused, or the more precise sensor can be used temporarily for calibration and removed afterwards.

4.2 Minimisation of Timing Errors

It was shown in Chapter 3 that synchronisation is required whenever separate sensing and navigation systems are combined for mapping applications. The mapping error due to the timing synchronisation was modeled, allowing the quality of the map to be improved by discarding data with unacceptably high uncertainty. In this section methods of reducing the timing synchronisation error to acceptable levels are discussed, in order to lower the uncertainty for *all* of the data.

4.2.1 Calculation of Timing Accuracy Requirements

The sensitivity analysis in Section 3.2.2 determined the impact of the timing error on the map. Therefore, for a given mapping accuracy requirement, the analysis can be driven in reverse to determine the timing synchronisation accuracy that is required to meet the specification. The maximum sensitivity of a mapped point to the timing error t_e is given by Equation 3.37, so this is used as a 'worst case scenario'. For small values of t_e, this can be used in conjunction with Equation 3.15 to approximate the maximum magnitude of error in the map e for a given t_e:

$$e = max(S_{t_e}^{\mathbf{p}^n}) \times t_e \tag{4.27}$$

A threshold can be specified for the maximum allowable error in the map, denoted e_{max}, which yields a maximum allowable timing error:

$$\begin{aligned} e &\leq e_{max} \\ max(S_{t_e}^{\mathbf{p}^n}) \times t_e &\leq e_{max} \\ t_e &\leq \frac{e_{max}}{max(S_{t_e}^{\mathbf{p}^n})} \\ t_e &\leq \frac{e_{max}}{|\dot{\mathbf{p}}_b^n| + |\dot{\mathbf{\Psi}}_b^n| \times (|\mathbf{p}^s| + |\mathbf{r}_s^b|)} \end{aligned} \tag{4.28}$$

The sensitivity $S_{t_e}^{\mathbf{p}^n}$ depends on the sensor range, the location of the sensor on the mobile platform, and on the vehicle velocity at the time of acquiring the sensor data. The 'worst case' timing requirement can be calculated by specifying the bounds on all of these terms. It is important to note that Equation 4.28 provides a maximum t_e to constrain the mapping error *due to timing*. The total mapping error is the sum of all of the separate error sources, so t_e may need to be reduced further to compensate for these. For a particular vehicle, the maximum rotational and translational velocities can be determined experimentally, as can the operational range of the sensor. The location of the sensor in the vehicle body frame is constant and known *a-priori*. If the worst case timing requirement can be met, then data should never have to be discarded due to t_e. If it is not possible to meet the requirement, then it can be relaxed, causing some proportion of the mapped data to exceed the uncertainty threshold. This might still be acceptable, because the real-time calculation of uncertainty allows for a policy to manage the variable quality of the data. In the simplest case, data that exceed some uncertainty threshold can be discarded. The appropriateness of relaxing the timing requirement depends on the particular application. In many cases where accurately mapped data are needed during high speed manoeuvres, this requirement should be strictly met for safety and reliability.

The following worked example illustrates the calculation of the maximum allowable timing error t_e to for a particular application.

Example 4.1. *In this example, an appropriate specification for the maximum allowable timing error is chosen for the CAS Outdoor Research Demonstrator vehicle, given a design requirement for a maximum mapping error of ten centimetres:*

$$e_{max} = 0.1m \tag{4.29}$$

The upper limit for the timing error depends on the vehicle velocity and rotation rates. For a typical autonomous run of the AUGV under consideration, the navigation solutions are recorded. A histogram of the magnitude of velocity $|\dot{\mathbf{p}}_b^n|$ can be seen in Figure 4.24 and a histogram of the vehicle yaw rate $\dot{\psi}_{b,z}^n$ (which has a much wider

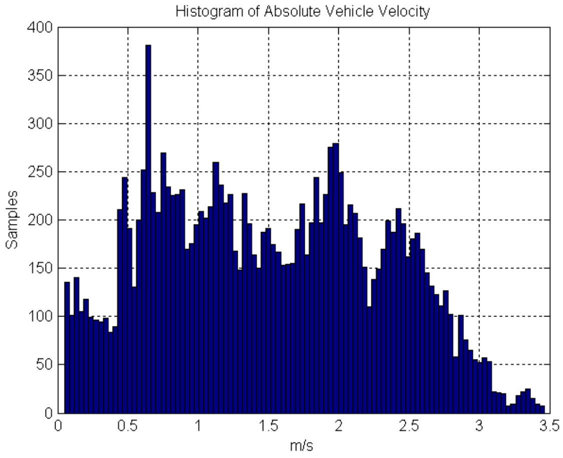


Figure 4.24: A histogram of the absolute translational vehicle velocity for a typical autonomous run of the CAS Outdoor Research Demonstrator platform.

range and large bounds than the roll and pitch rates) can be seen in Figure 4.25. The maximum velocity that this vehicle achieved is 3.5m/s (12.6km/h), which is relatively slow for ground vehicles in general, but the maximum yaw rate is 1.5rad/s ($86°/s$), which is relatively rapid. The AUGV is a skid steer vehicle, which frequently undergoes small rapid turns to steer. A histogram of the sensor range data $|\mathbf{p}^s|$ that was observed during the autonomous run is shown in Figure 4.26. The SICK LMS-291 sensor is physically capable of providing range returns of up to 80m according to its specification [51], but for this particular sensor configuration in a typical outdoor environment, 98% of the data have a range of less than 30m. The sensor is mounted approximately 1m from the origin of the vehicle body frame.

The recorded vehicle performance for a typical run provide bounds for the maximum velocities and turn rates, and the sensor range:

$$\begin{aligned} max(|\dot{\mathbf{p}}_b^n|) &= 3.5m/s \\ max(|\dot{\mathbf{\Psi}}_b^n|) &= 1.5rad/s \\ max(|\mathbf{p}^s|) &= 30m \\ |\mathbf{r}_s^b| &= 1m \end{aligned} \tag{4.30}$$

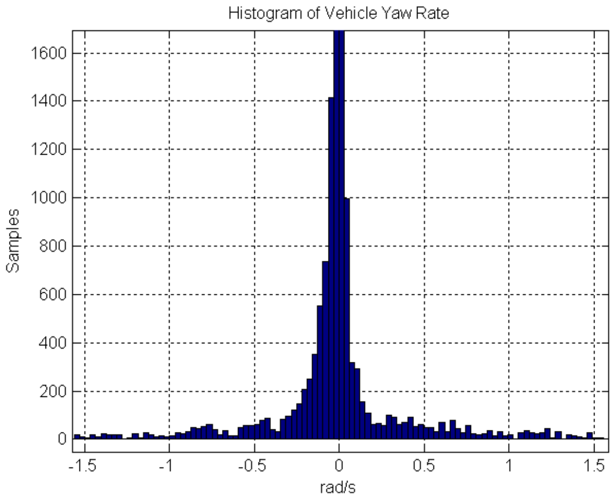


Figure 4.25: A histogram of the vehicle yaw rate for a typical autonomous run of the CAS Outdoor Research Demonstrator platform.

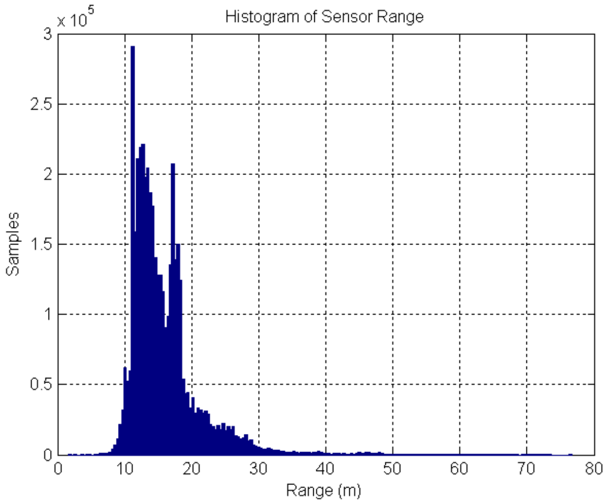


Figure 4.26: A histogram of the range of sensor data obtained during a typical autonomous run of the CAS Outdoor Research Demonstrator platform.

With the mapping error constraint from Equation 4.29 and the vehicle specific bounds in Equation 4.30 applied to the timing error inequality in Equation 4.28, the maximum acceptable timing error t_e can be calculated:

$$\begin{aligned}
t_e &\leq \frac{e_{max}}{|\dot{\mathbf{p}}_b^n| + |\dot{\mathbf{\Psi}}_b^n| \times (|\mathbf{p}^s| + |\mathbf{r}_s^b|)} \\
t_e &\leq \frac{0.1}{3.5 + 1.5 \times (30 + 1)} \\
t_e &\leq 0.002s \qquad (4.31)
\end{aligned}$$

If the timing error is kept at or below 2ms, then for a typical run, almost all of the data (98%) will satisfy the requirement of having less than 10cm error in the navigation frame. The remaining data may fail to meet this requirement, but this will be known at run-time and can be discarded if necessary.

4.2.2 Achieving Timing Accuracy Requirements

The previous section described how the required timing accuracy can be specified, given a desired mapping accuracy. This section discusses methods for reducing the timing error to meet this requirement. Recall from Section 3.2.2 that practical implementations of mobile sensing platforms frequently have separate sub-systems for navigation and exteroceptive sensing, each with their own time source. Mapping combines information from both sources, therefore synchronisation is required.

Because GPS receivers fuse time-of-flight measurements from multiple satellites, they require a very precise, synchronised timing system to maximise the accuracy of the range estimates. The satellites have highly stable atomic clocks, which are further corrected via communication with fixed ground stations. The corrections are then transmitted from the satellites to the ground based receivers. When a receiver observes the signals from four or more satellites, they are able to estimate local timing errors in addition to the global position. This enables receivers with relatively inexpensive timing equipment to achieve an accuracy in the order of a few nanoseconds, that would otherwise not be possible [9, §5.8,5.9]. GPS receivers typically provide a synchronised one pulse per second (PPS) hardware signal, that can be used as a base line for other equipment. This signal is accurate and globally referenced, meaning that even physically separated systems can be synchronised, although that is not an intrinsic requirement for a single mobile mapping platform. This section considers the synchronisation between *navigation* and other sensor systems, so GPS will often be available. Otherwise, inexpensive GPS units can be useful as a time base, even if the positioning information is not required. Other time sources are available, and an overview of synchronisation methods is given in [52], which points out that the use of GPS for timing has limitations because it can only be used outdoors, and can be problematic due to variations in signal quality due to environmental occlusions such as tree canopies. However, this only affects the synchronisation to the global time reference, and not the synchronisation between local sensors on a single mobile platform.

The optimal methodology for synchronising sensors to the navigation system uses a hardware timing reference signal to synchronise the data acquisition time of the mapping sensor with the navigation system. Although the navigation and sensor sub-systems remain separate modules, they are united by the hardware interface. This technology is available in 'high end' commercial laser systems such as the Riegl LMS-Q240 2D laser scanner [53], which has a PPS interface specifically to support mapping applications. This type of technology is required if the analysis in Section 4.2.1 concludes that sub-millisecond accuracy is needed, as is the case for systems with very fast dynamics, or long range sensing applications. If a particular mapping sensor does not have the requisite hardware timing interface, it may be possible to engineer an equivalent 'after-market' solution. For example, the most commonly used 'low end' laser scanning equipment is probably the SICK LMS-291 or similar variants, which do not include hardware for time synchronisation. However, it does include a feature to allow two adjacent sensors to be synchronised with each other, so that they sample out of phase to minimise interference [51]. It may be possible to use this hardware interface to synchronise a SICK laser to a PPS signal, but this is out of the scope of this thesis. In general, the best accuracy will be achieved if the navigation and sensor systems are designed with in-built synchronisation systems.

If direct hardware synchronisation is not possible, millisecond accuracy can still be achieved for the mapping sensor with standard computer hardware. Assuming that a PPS signal is available, then multiple computers can be synchronised to this reference using the Network Time Protocol (NTP) [36], to an accuracy in the order of microseconds [37]. The synchronised computer is then used to record the acquisition time of other sensors. The navigation information should have an accuracy in the order of nanoseconds as mentioned above, so the total timing error t_e is effectively determined by the ability of the computer software to correctly time-stamp the sensor data. This architecture was discussed in Chapter 3, and is illustrated in Figure 3.7. The software depends entirely on the specific features of the sensor. Example 4.1 showed how the timing requirement could be calculated for a mobile mapping system. The following example will show how this requirement can be achieved.

Example 4.2. *In Example 4.1, the timing requirement was calculated for the CAS Outdoor Research Demonstrator vehicle, resulting in a maximum allowable timing error of 2ms. In this example, data are acquired from a SICK LMS-291 2D laser scanner with a timing difference error t_e of less than 2ms, with respect to the PPS signal provided by the navigation system. This allows the laser and navigation information to be mapped, with an error due to timing that is always less than 10cm in the navigation frame, regardless of the motion of the vehicle.*

The SICK LMS-291 laser sensor is connected to an x86 machine via an RS-422 interface, at 500Kbaud. The system performs 181 scans over an angular range of 180 degrees at a scan rate of 75Hz. The unit can be configured to provide various angular ranges at different scan rates, but the sensor achieves these by performing multiple scans and combining the data in firmware. The data sheet for this sensor [54] provides timing information about the delays incurred for the different configurations, which indicate that the minimum delay of information occurs when using the ‘partial scan’ mode. In this mode, the sensor sends a maximum of one scan per transmission at 75Hz, regardless of the configuration. The scan data are then transmitted to the controlling computer via RS-422. The computer software then reacts to the reception of the data, and time-stamps it using its internal clock, that is synchronised to the PPS signal using NTP. The complete range scan is available to the processing

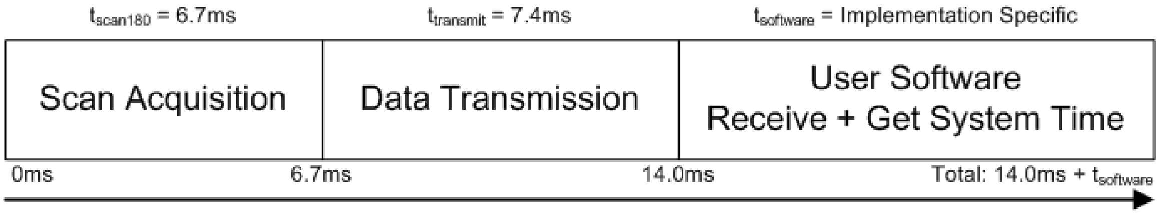


Figure 4.27: Time delays for the SICK LMS-291 laser scanner, between the start of a scan and the complete reception of the scan data.

software after a minimum scan and transmission delay of 14ms, assuming a constant baud rate and scan frequency. This is illustrated in Figure 4.27.

The delay caused by the processing is implementation specific. It depends on the operating system (OS) and its configuration, and the efficiency of the software. To minimise the processing lag, the acquisition software is written with a dedicated thread for time-stamping, that runs at a higher priority than other competing processes. The timing accuracy depends on the time-base of the OS kernel, and on its ability to pre-empt lower priority threads. If pre-emption is available, then the accuracy of the time-stamp is approximately equal to the time-base. It is difficult to measure the absolute error of the software time-stamp, because the ground truth is hard to obtain. However, the approximate error can be seen in terms of the observable timing jitter; the variation in the difference between successive time-stamps. The sensor transmits scans at a constant 75Hz, so it would be expected that each time-stamp is separated by 13.33ms (1/75s), subject to the resolution of the time-base. The Microsoft Windows OS has a time-base of 10ms, and the minimum standard Linux kernels have a minimum time-base of 1ms, restricting the maximum timing accuracy on these systems. By contrast, the software is implemented on the QNX Neutrino real-time operating system (RTOS) [55], which has a variable time-base that can be set as low as 10 microseconds. The jitter is recorded for a time-base of 0.2ms, and a histogram of the time difference error is shown in Figure 4.28.

QNX is a hard real-time operating system, so the timing shown in Figure 4.28 can be guaranteed, provided the balance of high priority threads on the system remains the same. The required maximum timing error of 2ms has been exceeded by a factor

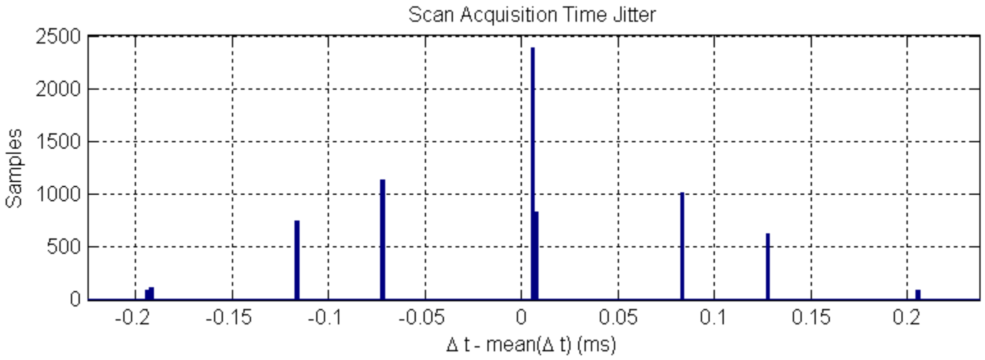


Figure 4.28: Jitter in sensor time-stamps, produced by software running on QNX RTOS, connected to a SICK LMS-291 laser scanner. On average, the scans are separated by a mean Δt of 13.333ms (75Hz). The OS time-base of 0.2ms results in jitter errors of a similar size.

of ten, with $max(t_e) = 0.0002s$, assuming the constancy of the transmission and scan rates.

4.3 Terrain Estimation with Uncertainty

Robust exteroceptive sensor mapping is critical for reliable and safe autonomy in AUGVs because it is a significant component in the model predictive control framework. In Chapter 3, a thorough analysis was performed to determine the effect of the primary sources of measurement error on typical mapping tasks. This established the sensitivity of the map to these sources of error, and provided the equations necessary to calculate the uncertainty in the map in real-time. Three dimensional point cloud representations were used as the most generic ‘map’ to avoid over-constraining the analysis. This form of representation is essentially a store of raw sensor information in a world fixed reference frame, and can be difficult to use in real-time for applications such as path planning and obstacle avoidance. To use the map in an MPC system, a more compressed representation of the environment is desirable, to allow for the vehicle model and forward predictions to be done in real-time. As discussed in Chapter 2, the chosen representation will depend on the specific application. In particular, structure in the environment may be leveraged to achieve the highest amount of compression and simplification. In all cases however, reliability and safety of the whole system can only be guaranteed if the measurement uncertainty given by Equation 3.22 is explicitly handled by the representation. It will not suffice to calculate the uncertainty of individual samples if it is ignored by the map. The uncertainty of the representation itself must explicitly reflect the uncertainty of the measurements that produced it. In this section, terrain representations that are useful for AUGV MPC are discussed, in particular with reference to their treatment of measurement uncertainty.

4.3.1 Grid Based Height Maps

Grid based height maps are commonly used in mobile robotic applications due to the simplicity of their implementation and their effectiveness. In this framework, the elevation of the terrain is assumed to be a function of the location, specified by two discretised spatial coordinates [8, §36.3]. An exteroceptive sensor mounted on the vehicle is used to gather data, which is then transformed into the reference frame of the grid map using the transformation of Equation 3.12. An example grid map showing a section of unsealed road is shown in Figure 4.29, with the average height at each cell drawn as a single point above the cell centre.

There are several significant problems with the typical grid based height map approach:

- The functional height map assumption means that complicated non-functional structures within the environment such as vertical surfaces and overhangs are not well represented.
- When updating the map with sensor data, the mapped spatial uncertainty (as described in Chapter 3) is often ignored, allowing data to be associated with incorrect cells. The mean location produced from Equation 3.12 is often used, without considering the associated uncertainty given by Equation 3.22.

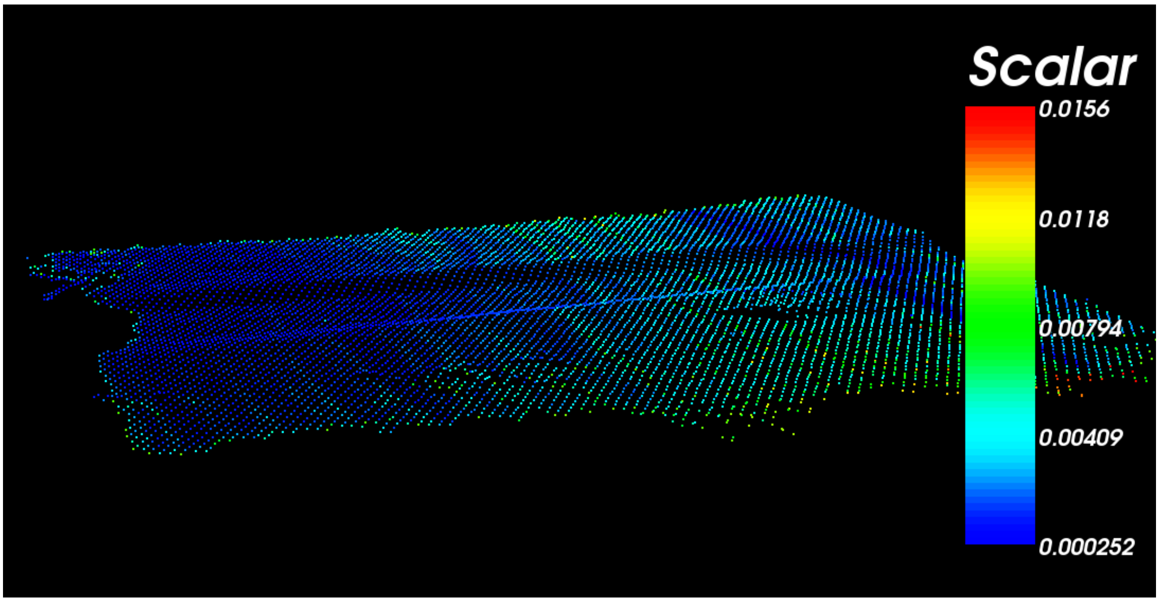


Figure 4.29: A grid based height map representing the surface of a section of unsealed road. The average height of each cell is drawn as a single point, located above the cell centre. The heights were produced by transforming the data from a SICK LMS-291 2D laser scanner into a global frame using Equation 3.12. The average height of all of the data in each cell is represented *without* consideration of the uncertainty from Equation 3.22. The colour of the dots represents the statistical standard deviation of the height of points in each cell, again without consideration of the measurement uncertainty of the individual points. As a result it is an over-confident measure of the true uncertainty. The resolution of the grid is $0.3m \times 0.3m$. This section of the environment is mapped well by this technique, because it conforms to the height function assumption.

- The inherent correlation between adjacent or nearby cells is typically not modeled, meaning that new information pertaining to one region has no effect on height estimates at any other location in the map, even if the environment is actually spatially correlated.

- The rigid structure of the cells is inflexible. Subsequent information may alter the validity of information that has already been included in the map, but the rigid structure does not easily allow large scale modifications.

Despite all of the problems with grid based height maps, they have been successfully employed in autonomous mobile robots in many cases, where the environment does not strongly violate the constraints and assumptions above [8, §36.3].

Extensions have been made to the basic representation in an attempt to address many of these issues. In [56], each grid cell maintains the probable height with a one dimensional Gaussian distribution, using one Kalman filter per cell. Although an ad-hoc method is used to calculate the uncertainty of the sensor measurements, rather than the complete error model presented in Chapter 3, the extension of the grid map to represent uncertainty is significant. However, when considering the measurement errors, a complete approach should not only explicitly deal with the *height* distribution, but also the uncertainty in the two dimensional spatial *domain*. Because each measurement has a potential error in three dimensions, choosing the correct cell to update is not trivial. This is the *data association* problem for cell based terrain estimation. Furthermore, navigation errors that are correlated in the short term (as discussed in Section 3.3.3) will lead to overly confident estimates of the terrain height. This is because successive correlated measurements will cause the Kalman filter to converge rapidly to a solution that is only falsifiable over larger time scales.

A more significant contribution of [56] is the extension of the basic grid height map representation to allow for some common non-functional features of the environment. By explicitly classifying regions of the environment as horizontal, vertical or overhanging structures, the technique greatly increases the scenarios in which the representation can be used reliably.

In [57], the world is discretised into ‘box shaped regions’, but unlike the standard height map approach in which each region is assumed independent of all others, the correlation between neighbouring regions of space is explicitly modeled. This allows the ground support surface to be estimated, despite occlusions due to vegetation above it. This leverages the structure in the particular operating environment of the vehicle to allow the uncertain sensor information to be filtered. The data association problem is avoided in this case by using highly accurate navigation equipment.

4.3.2 Gaussian Process Height Maps

To remove some of the constraints imposed by discretisation, and to incorporate explicit treatment of measurement uncertainty, Gaussian Process (GP) regression [44, §45] [45, §6.4] [58, 59] has been proposed for terrain representations [60]. GP models allow the terrain surface function to be regressed from exteroceptive sensor data (after being transformed to an Earth fixed frame), allowing interpolation and extrapolation of the surface. GPs explicitly handle measurement uncertainty and spatial correlations within the environment. Given a set of uncertain data points, the maximum likelihood terrain surface can be estimated, effectively learning the spatial correlations within the environment from the available data. An important characteristic of such models is that in addition to estimating the mean height of the surface with *explicit* consideration of measurement errors, they also provide the uncertainty of the height estimation. The uncertainty is higher in regions with less data, or where there is little correlation with well known areas. Sophisticated treatment of a non-stationary GP covariance function in [60] allows spatial correlation *trends* in the environment to be incorporated, enabling some areas to exhibit high spatial correlation in particular directions, and for other sections to remain uncorrelated, which greatly improves the extrapolation ability of the model.

The estimation of uncertainty in the maximum likelihood surface is arguably the most important feature of this technique. However, given the high computational cost for large data-sets ($\mathcal{O}(N^3)$ for the exact method, although less for approximate methods [58, 59]), the lack of a real-time incremental method to incorporate new data points to an existing model, and the high implementation complexity compared to height grids, it is worth considering what value is added over the grid based methods. Of particular interest is the comparison of GP surface models with grid maps that incorporate extensions to consider uncertainty such as [56], but that ignore the potential for spatial correlations under the assumption that they cannot *generally* be relied upon.

Figure 4.30 shows a section of terrain modeled with GP regression, using the *stationary* squared exponential covariance function. The model is learnt from the data, then evaluated over a fine regular grid. A Delaunay triangulation [61] is then per-

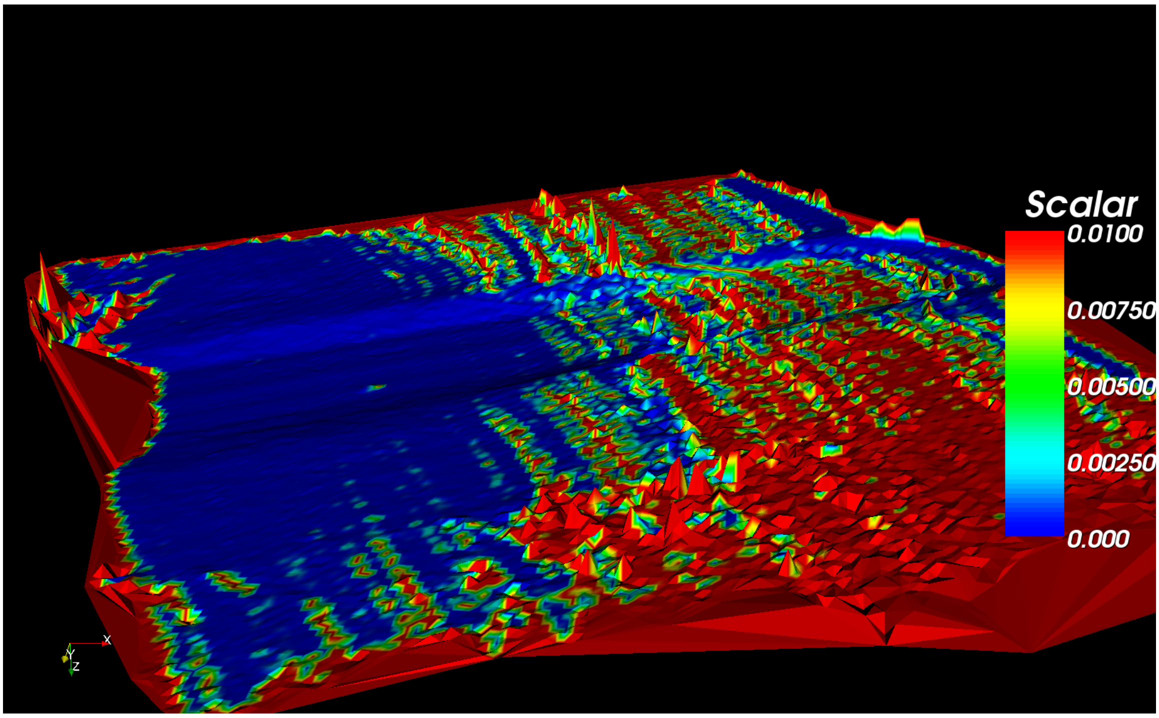


Figure 4.30: A Gaussian Process regression terrain model of a section of dirt road, learnt from laser data from a SICK LMS-291 2D laser scanner mounted on a UGV. A Delaunay triangulation is performed on the data to visualise the surface. At the left of the figure, the vehicle moves slowly (10km/h) achieving a high sample density, then speeds up (30km/h) on the right side of the figure, lowering the sample density. Due to the undulations of the terrain (in particular the ridges at the edges of the road) the model can only allow a small amount of spatial correlation. When the samples become less dense there is not enough predictive power in the model to interpolate between them and the terrain surface has a high amount of uncertainty.

formed over the evaluated points to form the surface visualisation seen in the figure. The surface is coloured by the uncertainty reported by the GP model. The data were obtained by a ground vehicle driving along a section of dirt road, first slowly (10km/h), then more rapidly (30km/h), in order to vary the density of the data. As expected, the uncertainty is lower where the data are more dense. In order to capture the undulations of the road (particularly at the edge where there is a large ridge), the GP model must allow for a relatively low spatial correlation. Because of this, the regions of lower data density do not have enough coverage to allow smooth interpolation between them, and the surface becomes very uncertain over a small spatial scale. In other words, although the model has allowed for spatial correlation, which can be

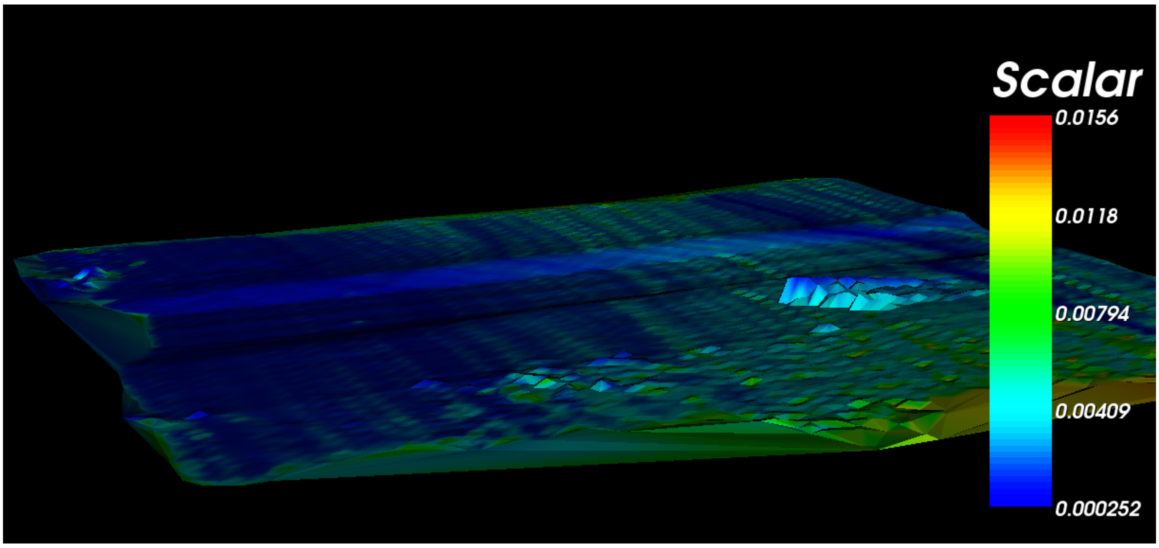


Figure 4.31: A height grid terrain model of a section of dirt road, produced from laser data from a SICK LMS-291 2D laser scanner mounted on a UGV. A Delaunay triangulation is performed on the data to visualise the surface. At the left of the figure, the vehicle moves slowly (10km/h) achieving a high sample density, then speeds up (30km/h) on the right side of the figure, lowering the sample density. Where there are fewer samples, the variance of the height estimates is less certain. Spatial correlation is not modeled by this representation, so adjacent cells could have vastly different heights. Each location of this surface exists as a result of direct observation by the sensor.

difficult to achieve in grid based structures, the allowable correlation is so small as to provide no real benefit in this case. The non-stationary covariance function of [60] would certainly help in this situation, but the fact remains that whenever the model of the environment incorporates the possibility of sudden local phenomena, there can be almost no predictive power. Just because a planar section of terrain extends for kilometers in every direction, there is no guarantee whatsoever that there will not be a tree, cliff or building in the next millimeter. The potential of such models is strictly limited due to the inherent complexity in natural environments, and the covariance estimates cannot be trusted for terrain applications in general.

For comparison, Figure 4.31 shows a Delaunay triangulation of the height grid data from Figure 4.29. The raw data for this figure are *identical* to the data that were used to produce the GP terrain surface in Figure 4.30, but this time a height grid is used instead. Although the grid completely ignores any potential spatial correlation in the environment, the resulting surface is very similar, both in terms of the surface heights *and the uncertainty measures*. In both cases, the uncertainty is known to be lower where there is less data, and although the GP model can account for spatial correlation, there is not enough correlation in the environment for this to be useful.

In certain situations where there is reliable structure in the environment and the complexity is reduced, models that allow a large amount of spatial correlation are more important. This is precisely the phenomenon used in [57] to predict the ground surface beneath occluding vegetation. The issue of uncertainty was avoided by that technique by using precise navigation equipment, so perhaps GP models would be a powerful alternative without the assumption of certain localisation. In the general case of a complex, unpredictable natural environment, it is likely that grid base representation will perform at least as well as Gaussian Process models of the terrain, but at a significantly lower computational and implementation cost.

4.3.3 Probabilistic Data Association Height Maps

One of the significant issues with using discrete grids to represent terrain given measurement uncertainty is the data association problem. When a sensor measurement is made by mobile platform and transformed to an Earth fixed frame with Equation 3.12, the spatial uncertainty given by Equation 3.22 is in three dimensions. This can be thought of as comprising a separable two dimensional *domain* uncertainty, reflecting the uncertainty about which region of the map the measurement belongs to, and a *value* uncertainty, reflecting the uncertainty in the measurement of height to attribute to that location. The height uncertainty can be filtered, such as in [56], but the choice of which cell to update given the domain uncertainty is not trivial.

The problem of data association is well known for applications such as simultaneous localisation and mapping (SLAM), where potentially ambiguous detections of discrete features must be clarified prior to applying the data in the update step of the filter [8, §37]. Typically a discrete set of hypotheses is formed as to the possible association of sensor information with features, including the possibility that the sensor information is associated with *none* of the features. The data association problem can then be viewed as the ranking of these hypotheses according their probability given the data [62] [44, §28].

In some applications this is done in a purely spatial context [63, 62, 8], whereby the association is accepted if it is statistically likely, given the current belief (posterior distribution) of the location of all features and the probability distribution of the location of observation. The ambiguity may also be reduced with a richer source of information, such as the unique visual appearance of specific features [64, 8]. In the context of terrain representations, these techniques are often not possible, due to the compact spatial arrangement of the features or cells. Spatial association via statistical significance tests requires very high certainty because the features are all in close proximity. If a terrain observation has a domain uncertainty that is larger than a single cell, it is not possible to determine precisely which cell it belongs to. Clarification with richer feature descriptors is difficult because proximal sections of terrain usually look very similar, regardless of the choice of sensor. This is due to the 'first law of geography', which states that "everything is related to everything else, but near things are more related than distant things" [65].

A similar problem exists in target tracking applications, where the target can be difficult to separate from 'clutter' of a similar appearance to the tracking sensor. The discrete hypotheses for whether observation data belong to the target or to clutter are analogous to the set of possible cell locations to which a spatially uncertain terrain observation belongs. Rather than accepting the most likely hypothesis, or rejecting the data altogether if they are unlikely, it is also possible to marginalise over the set of *possible* hypotheses. Such a technique is applied to the target tracking problem as a Probabilistic Data Association Filter (PDAF) in [66]. Each observation may be due to the true target, or to clutter and although the association cannot be clarified completely, probabilities are assigned to each hypothesis, and *all* hypotheses are used in the update step. For a system with a single Gaussian probability distribution for the state, and a Gaussian observation model, the marginalisation step generates a mixture of Gaussian posterior distribution, with a new mode created after each observation. This is intractable in most situations, so the PDAF method fits a single mode to represent the mixture after each update. This may be a poor approximation in applications which exhibit strong multi-modal behaviour, due to vastly different possible hypothesis.

In [23], PDAF is applied to a terrain representation that is similar to the Kalman filter (KF) per cell described in [56]. This technique calculates the measurement uncertainty according to Equation 3.22, then uses the PDAF mechanism from [66] to update the mean height and variance of each cell with non-zero association probability. This is one of the most complete treatments of uncertainty in an unstructured outdoor terrain representation to date. Figure 4.32 shows a comparison of the KF and PDAF representations for a simulated environment. In the 1D simulation, a UAV is flown over the environment, while making range measurements to the surface. Sensor measurements are generated by adding normally distributed random noise to the true position of the platform and to the range and bearing of the sensor. This information is then filtered with a KF and PDAF representation. The two plots are identical, but with emphasis given to the PDAF on top, and the KF terrain on the bottom, to declutter the figure. The solid lines represent the mean of the filter and the dotted lines show the two standard deviations (2σ) of uncertainty. Because the KF does not consider *domain* uncertainty (it assumes certain knowledge of the correspondence of measurements to cells) the 2σ bounds are over confident. By contrast, the uncertainty bounds in the PDAF filter are wider, to reflect the additional uncertainty due to the uncertain data association. The PDAF estimate is much smoother, because the model incorporates the possibility of any of the measurements actually belonging to the neighbouring cells due to the spatial domain uncertainty.

The PDAF map has a more consistent uncertainty than the KF map, because the model explicitly considers the location and height uncertainty resulting from the measurement process. The PDAF terrain estimate is effectively blurred to match the uncertainty in the measurements, to consistently ensure the estimate is not overconfident. However, the utility of the estimate is potentially reduced because of this. Also, the Gaussian approximation to the mixture of Gaussian distribution after each update step can induce a sampling order bias into the system. Unlike traditional

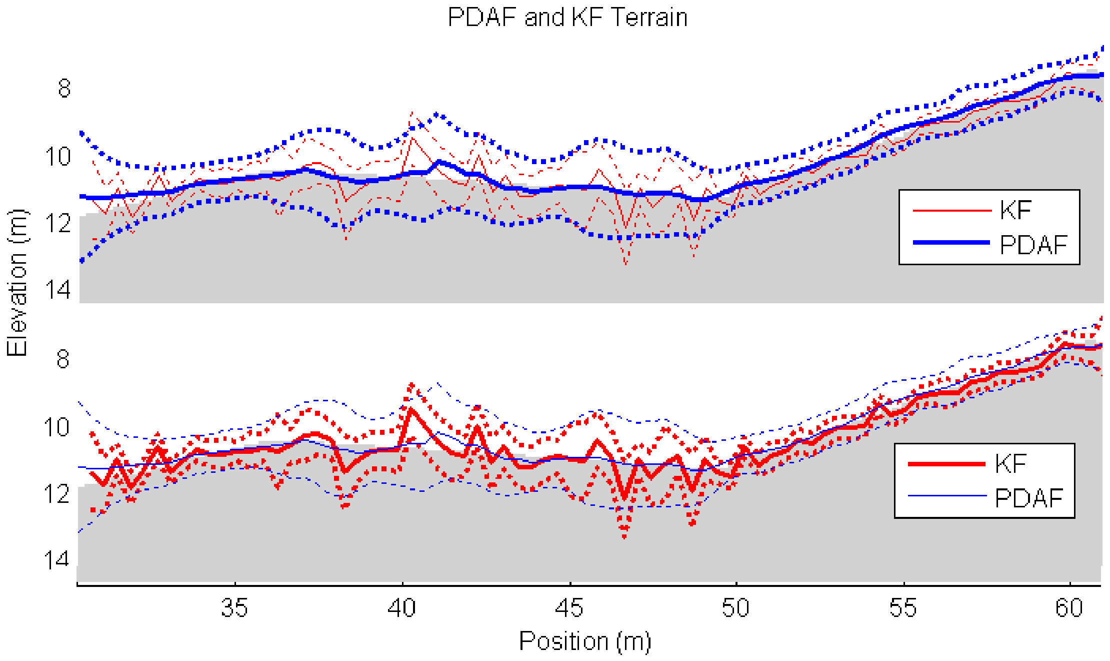
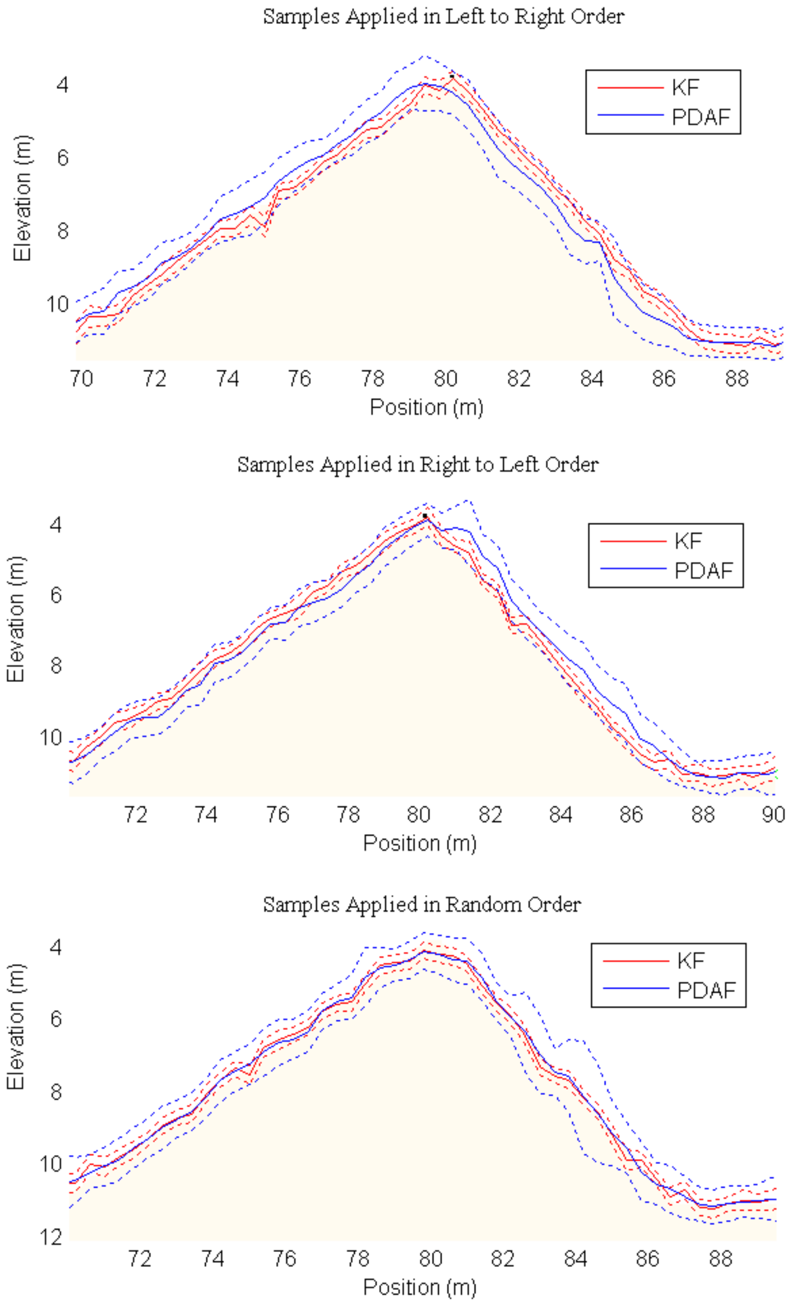


Figure 4.32: A 1D simulated terrain (in grey), modeled with a Probabilistic Data Association Filter (PDAF) method (emphasised on top) and a Kalman filter (KF) approach (emphasised on the bottom). A simulated UAV flies over the environment, taking range measurements of the terrain. Uncertainty in the UAV position and the range and bearing of the sensor measurements is modeled by normally distributed random noise. The two graphs are identical, but with different emphasis to declutter the figure. The solid lines show the mean terrain estimate, and the dotted lines illustrate two standard deviations (2σ) of uncertainty. Due to the measurement errors, the data association between observation and grid cell is not known with certainty. The KF assumes certain association, leading to overly confident 2σ bounds. The PDAF approach models the uncertain association, yielding larger 2σ bounds and a smoother, 'blurred' estimate of the ground.

Bayes updates, which are invariant to the order with which a set of information is applied (as shown in Appendix D), PDAFs with the single Gaussian approximation from [66] are not independent of the order of information. This effect can be seen in Figure 4.33. In the first figure, the terrain information is gathered from left to right, first ascending the hill, then descending. The resulting bias can clearly be seen. The opposite bias is seen when the samples are applied in reverse order (from right to left). Finally, the last graph shows that if a random order is used, the bias is reduced, but it is not practical to implement this in a real-time system, where measurements become available as the mobile platform progresses through the environment.

Although the consistency of the terrain uncertainty can be improved with techniques such as PDAF, the blurring of the information reduces its utility. An alternate approach to this problem is to use two different navigation frames, as was discussed in Chapter 2. A global frame provides global spatial consistency for high level concepts such as destinations or waypoints. A local frame uses only smooth sources of relative pose information such as accelerations from an IMU or velocities from TDCP, which can then be integrated to provide a smooth estimate of relative position. Although it drifts from the global truth over time, it does not incur sudden corrective jumps as global information sources such as GPS are incorporated. The local frame can then be used as the navigation frame for mapping to provide relatively smooth, consistent maps. In Section 4.4, the terrain modeling capabilities of a navigation system of this type is evaluated.



Samples Applied in Left to Right Order

Samples Applied in Right to Left Order

Samples Applied in Random Order

Figure 4.33: A 1D simulated terrain (shown in grey), modeled with a KF and PDAF method. The observations are made by a virtual UAV that flies over the environment, looking down. Random noise is added to the position estimates and to the range, bearing observations, to simulate the effect of an uncertain measurement process. The first graph shows the representation when the samples are applied from the left of the scene to the right, in the middle they are applied in reverse order, and a random order is used for the lower graph. The application of a single Gaussian to approximate the mixture after each update step has introduced a sampling order bias. Although the KF method has an overly confident estimation of uncertainty (the variance is smaller than it should be), it does not exhibit a sampling order bias.

4.4 Management of Navigation Uncertainty

In Chapter 3, the coordinate frames that are relevant to typical mapping applications were specified, and in Section 3.1.1, the navigation frame was chosen as the most appropriate and commonly used frame in which to store mapped data. The navigation frame could be aligned to a standard global reference (such as latitude and longitude) in which case it is termed a 'global frame', or it could have an arbitrary origin, termed a 'local frame' (as it is often aligned to a local feature such as a tree or building, or to the vehicle position at the time of the frame's creation). As discussed in Section 3.1.1, the decision to use a global or local frame depends on many factors. The benefits of using a global navigation frame are significant, yet the errors due to GPS based navigation systems were seen in Section 3.3.3 to have a significant detrimental impact on the map. Chapter 3 provides the capability to quantify these errors, so that although the map is adversely affected, the magnitude of the effect is known at run-time. However, this is often insufficient in practical applications due to the frequency with which navigation errors can occur. In [10] it is stated that the physics of GPS prevents it being reliable enough for a stand-alone navigation system, and many recent efforts to build fully autonomous AUGVs have cited difficulties with the use of GPS based navigation systems for mapping [4, 12, 13, 14, 2, 15].

Two main approaches to this problem have emerged in the literature. Firstly, the accumulation of sensor information from multiple measurements at different vehicle poses should be minimised, because of the detrimental blurring effect of imprecise navigation during the data gathering process [4]. Of course this is only possible when sensors have a sufficiently large field of view with respect to the perception requirements. This is often not the case, particularly when using 2D range scanners such as the popular SICK LMS laser sensor. Secondly, improvements can be made to the navigation system, to increase the duration over which data can be accumulated and mapped without significant error. Navigation systems are typically configured to provide the optimal pose estimate in the global frame, where the optimal pose is the most likely, given all available information. However, this is not necessarily an optimal estimate with respect to mapping, in which smooth pose estimates are often more valuable than absolute global accuracy. For this reason, many AUGV implementations use a local frame for mapping tasks, and a global frame to provide the absolute spatial context of landmarks or waypoints [4, 13, 14, 2, 15]. If an optimal global frame estimator is used for mapping sensor data, then the location of each individual mapped sensor datum can also be considered optimal, in the sense that the mean mapped position is the most likely in the global frame. However, if data are gathered over time (however brief) and fused together to achieve a bigger picture, a locally smooth navigation estimate may be optimal because it provides the most likely *relative* mapped positions. With a smooth navigation solution in a local frame, the overall shape and structure of clusters of data are preserved optimally, rather than the absolute position of each datum under the assumption that they are unrelated. It was concluded in Section 3.2, that in general, angular errors (including those in the navigation solution) are amplified by the sensor range when transformed to the map and that translational errors (including the navigation position error) are propagated directly. If the relative uncertainty of a local navigation solution and the absolute uncertainty of a global one can be estimated at design-time, then this general error sensitivity can be used as a 'rule-of-thumb' to determine the effect of the choice on the map.

In this section, four navigation systems are compared with respect to their appropriateness for perceptual mapping tasks, when sensor accumulation and mapping is required. A comparison is made between commercial off-the-shelf (COTS) DGPS/INS and GPS/INS 'global' navigation systems and the smoother output obtained from 'local' navigation systems using stand-alone INS and the ASNF [16, 17] configured for TDCP/INS. As discussed in Chapter 2, predicting the motion of a vehicle is particularly difficult in unstructured environments due to the complexity of the interactions between the vehicle and the ground. Therefore, the navigation systems considered in this section *do not use a vehicle model*; they are black box solutions that operate equally well regardless of the type of vehicle to which they are attached. In contrast, the AUGV systems cited above that use local and global frames all rely on vehicle models to improve the navigation filter output. This is typically achieved by assum-

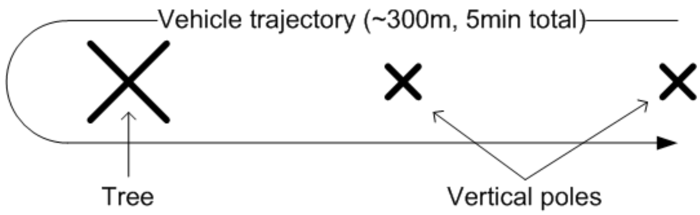


Figure 4.34: The vehicle follows a simple trajectory, passing two poles and a tree, then returning. If the navigation solution drifts in X or Y, the poles or tree will be mapped to a different place on the return journey. Similarly, if the Z position drifts, there will be an apparent separation of the ground surface.

ing there is no wheel slippage, which is an appropriate assumption when driving on tarmac roads at relatively low speeds, where the vehicle ground interaction is relatively simple. This is not generally extensible to unstructured environments where such assumptions are invalid. Wheel slippage is cited as a problem for navigation in [4], which is operating in an unstructured environment.

It is shown here that for mapping applications, although GPS position information cannot be reliably used in a global navigation frame due to the frequent jumps in position, GPS information in the form of TDCP velocity observations can be used in a local frame to provide a smooth relative pose estimate with a reduced drift rate when compared to stand-alone INS integration. This forms a smooth 'black box' local frame navigation solution that does not require a vehicle model and is appropriate for exteroceptive sensor mapping in unstructured environments.

The CORD platform is used again in this section, with a single downward looking SICK laser range scanner, as seen in Figure 3.8. The laser is in the same push-broom configuration that was illustrated in Figure 3.9. Data from this sensor is collected while the vehicle is driven autonomously along the trajectory shown in Figure 4.34. The 'blind' trajectory controller that was seen in Figure 2.3 is used so that the test can be repeated with different navigation systems under similar conditions.

The experiment is repeated four times, once for each configuration:

1. Novatel DGPS/INS (RTK corrections).
2. Novatel GPS/INS (no RTK corrections).

3. INS integration.

4. All Source Navigation Filter (ASNF) [16], configured for TDCP/INS.

All configurations use the same hardware, consisting of a Novatel ProPak-G2plus GPS receiver and a Honeywell HG1700 AG17 inertial measurement unit. The trajectory is approximately 300 metres in total length. The vehicle maintains a velocity of 1m/s and it completes each of the four runs in 305 seconds, ±5 seconds.

4.4.1 Commercial Off-the-Shelf DGPS/INS

For this experiment it would be useful to compare the output of the navigation systems to a ground truth that is known to be accurate, such as a geological survey of the site. In this case no such information was available, so the DGPS/INS solution was used as the reference against which the other systems were compared. At this test location, the DGPS/INS system reported an accuracy of greater than 5cm for the entire trajectory. The integrity of this result was empirically verified by mapping the data and observing that the features (the tree and the two poles) were aligned properly for all of the data from both the outward and returning parts of the path. The mapped data can be seen in Figure 4.35 and the alignment is visibly precise, with singular, vertical features.

The Z position (‘down’ or negative altitude) of the DGPS/INS navigation solution is shown in Figure 4.36. This single dimension is displayed as it is typically the least accurate in GPS based navigation systems and it is often considered the most important dimension when estimating the roughness of the terrain. The shape of this plot corresponds to the altitude of the ground surface beneath the vehicle trajectory. The graph is approximately symmetric, because the vehicle drives out and back along a similar path (a few metres adjacent, not identical). The largest jump or rift in altitude is approximately 1cm, and the final altitude is equal to the initial altitude, indicating the high accuracy and precision of the solution and its validity as a reference.

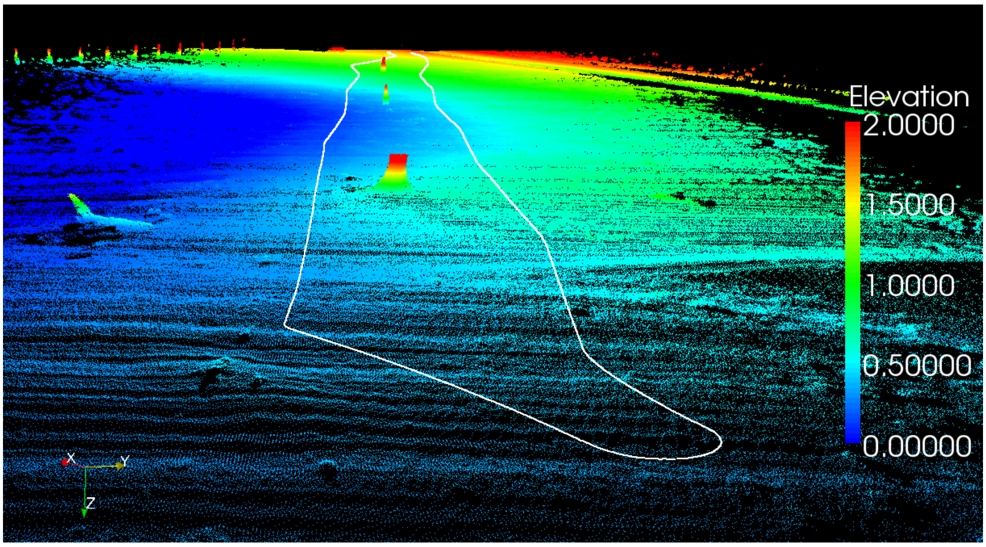


Figure 4.35: Mapped laser range scanner data using a commercial DGPS/INS navigation system. The CORD platform drove autonomously over the trajectory shown in Figure 4.34 and the exact trajectory that was driven is also seen in this image. The tree and the two poles have been mapped consistently during both the outward and return passes of the vehicle, indicating the high accuracy and precision of the DGPS based navigation solution.

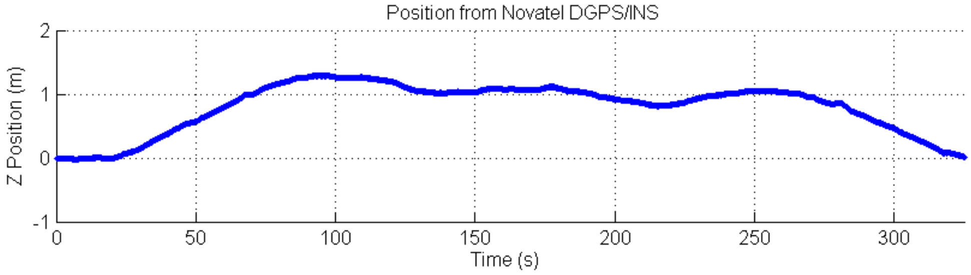


Figure 4.36: The Z position reported by the Novatel COTS DGPS/INS navigation solution, while autonomously driving the trajectory seen in Figure 4.34. The shape of this graph corresponds to the altitude of the ground beneath the vehicle trajectory. The graph is approximately symmetric, because the vehicle drives out and back along a similar path (although not identical).

4.4.2 Commercial Off-the-Shelf GPS/INS

The trajectory was repeated by the vehicle, without using the RTK differential corrections. The reported Z positions of the trajectory are shown in Figure 4.37. The altitude profile is substantially different from the profile that was produced by the DGPS/INS system, seen in Figure 4.36. The profile is asymmetric and the final altitude differs from the initial by approximately 50cm. Near the tree, between 140 and

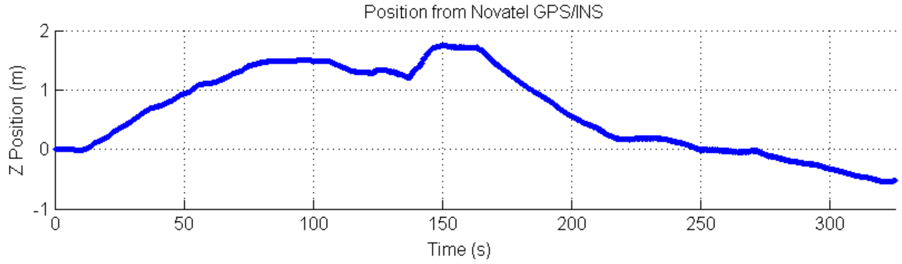


Figure 4.37: The Z position reported by the Novatel COTS GPS/INS navigation solution, while autonomously driving the trajectory seen in Figure 4.34. The shape of this graph differs from the DGPS estimates in Figure 4.36 and are clearly incorrect due to the asymmetry and the differing end points. It is likely that the large tree causes multipath effects on the GPS signal, as a sudden erroneous change in altitude is observed as the vehicle drives under the tree canopy at approximately 140 seconds.

150 seconds, an error of approximately 80cm is introduced, which causes an error of a similar order of magnitude in the map, by Equation 3.30. This sudden and unpredictable shift in the pose estimate is typical of navigation systems that operate in a global frame. Recall from Section 3.3.3 that global navigation systems report sudden jumps in position due to phenomena such as multipath with GPS or loop-closures in SLAM based systems. This results in substantial difficulties when attempting to accumulate and map data, even over relatively short time scales.

The mapped data can be seen in Figure 4.38. The consistency of the vertical features is poor (they appear to be leaning), when compared to the mapped data (DGPS/INS) in Figure 4.35. Furthermore, the estimated altitude has drifted by enough that the ground surface on the return journey is perceived as higher than the initial navigation solution located at the centre of the vehicle. This is clearly incorrect. The apparent difference in height of the ground surface can be measured at any location, by comparing the height from outgoing measurements to the height of measurements on the return journey. Figure 4.39 shows a top down view of the mapped data, located at the tree trunk. The sensor shadow from the tree visibly separates the two surfaces. In this case, the apparent ground height has deviated by approximately 50cm, in the 20 seconds taken for the vehicle to pass the tree and turn. It is clear that the GPS/INS solution is not appropriate for mapping due the occurrence of sudden and unpredictable shifts in the estimated pose of the vehicle.

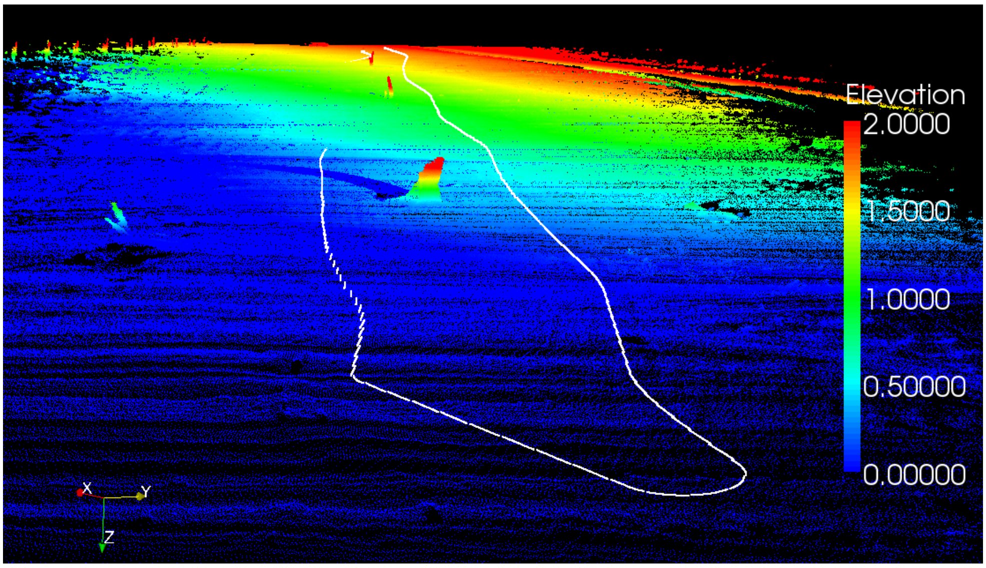


Figure 4.38: Mapped laser range scanner data using a commercial GPS/INS navigation system. The reported trajectory can also be seen in this image. Noticeable rifts are visible in the trajectory and the navigation solution deviates so far that the sensor data on the return journey is mapped above the initial vehicle trajectory. A shadow can be seen behind the tree trunk (the nearest feature), which is due to the drift in height of the mapped ground surface in the time taken to pass the tree and turn. This is shown in more detail in Figure 4.39.

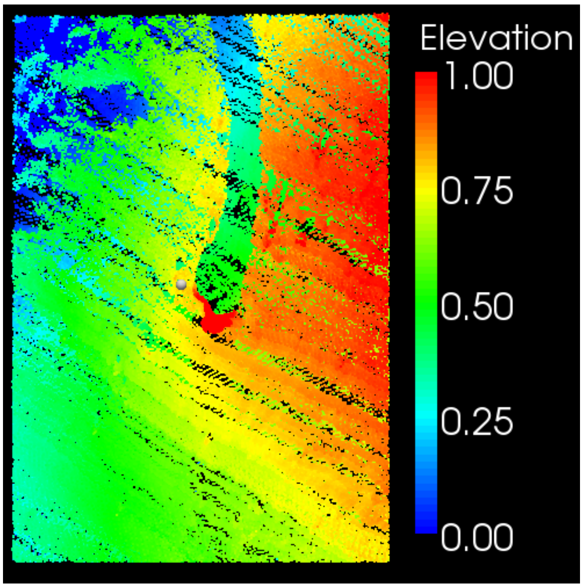


Figure 4.39: A top down view of the mapped data, using the commercial GPS/INS navigation system. A difference of 50cm in the mapped height of the ground surface appears in the 20 seconds taken for the vehicle to pass the tree and turn back.

4.4.3 All Source Navigation Filter TDCP/INS

The inclusion of TDCP measurements in the filter provides multiple benefits over stand-alone INS integration. The additional information reduces the rate at which the pose estimate drifts from the true location of the vehicle. Furthermore, the use of global information increases the observability of the attitude during vehicle motion (depending on the specific manoeuvres of the vehicle), such that the attitude estimate no longer drifts unbounded [67]. Unlike global position observations, the TDCP velocity observations must still be integrated once to obtain positions, so the pose estimates are less prone to sudden rifts. Like all GPS information, TDCP observations are affected by multipath, but the resulting errors are two orders of magnitude smaller than for global position observations [19]. In short, although the TDCP/INS pose estimate still drifts, the drift rate is reduced and the attitude is observable. The unpredictability due to characteristic GPS 'jumps' is also almost entirely avoided. The All Source Navigation Filter [16] can be used to fuse a variety of information sources with the INS, such as GPS, TDCP and feature tracking with SLAM according to the structure shown in Figure 4.40. In this section, only the TDCP component was used [19].

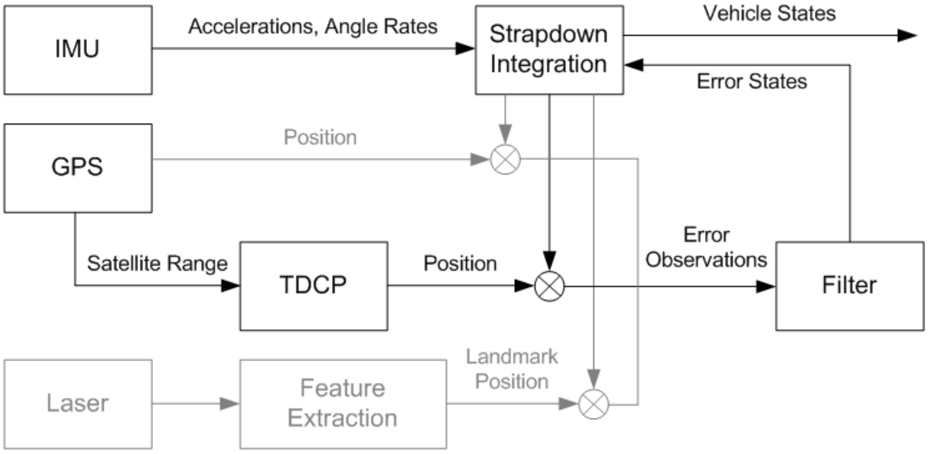


Figure 4.40: The All Source Navigation Filter [16] architecture that fuses INS, GPS position and TDCP information, and laser based feature detections, into a single tightly coupled structure. For the experiments in this section, the system is configured to provide smooth pose estimates in a local frame, using only the INS and TDCP.

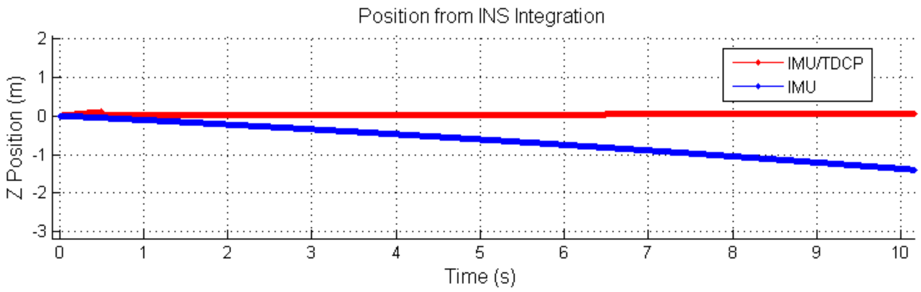


Figure 4.41: The position estimates produced by INS integration and by the ASNF TDCP/INS system. After only eight seconds and before the vehicle has moved, the stand-alone INS solution has drifted by a metre. After 5 minutes (not shown here) the INS Z position has drifted by more than 500 metres.

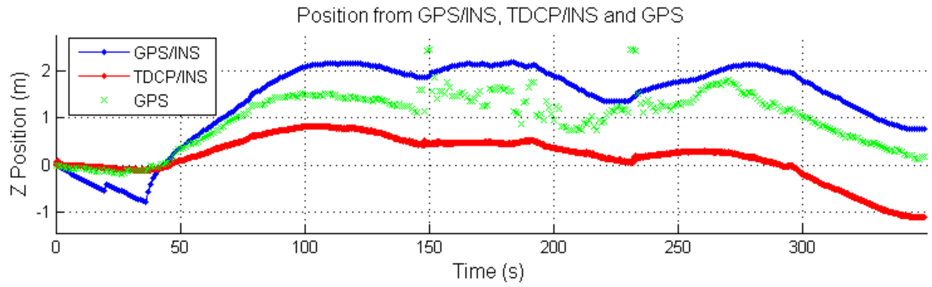


Figure 4.42: A comparison of the Z position estimates produced by the Novatel GPS/INS system, the ASNF TDCP/INS system and the raw GPS observations. The TDCP/INS solution has drifted by only 1 metre after more than 5 minutes.

In Figure 4.41, the Z position estimates from the stand-alone INS system are compared to the estimates produced by the ASNF TDCP/INS system. The data are shown for the initial ten second period during which the vehicle remained stationary. After only eight seconds, the INS solution has drifted by a metre and after the five minute trajectory is completed, the solution has drifted by more than 500 metres. The entire TDCP/INS Z position estimate of the trajectory is shown in Figure 4.42 and by contrast, the solution has drifted by only one metre after five minutes. Although the INS system provides smooth pose estimates, the drift occurs too rapidly for most mapping applications, allowing only one or two seconds of data accumulation before the error would corrupt the structure of the map. It is worth noting that higher grade IMUs are available at significant expense, which would buy more accumulation time, but TDCP provides a lower cost alternative.

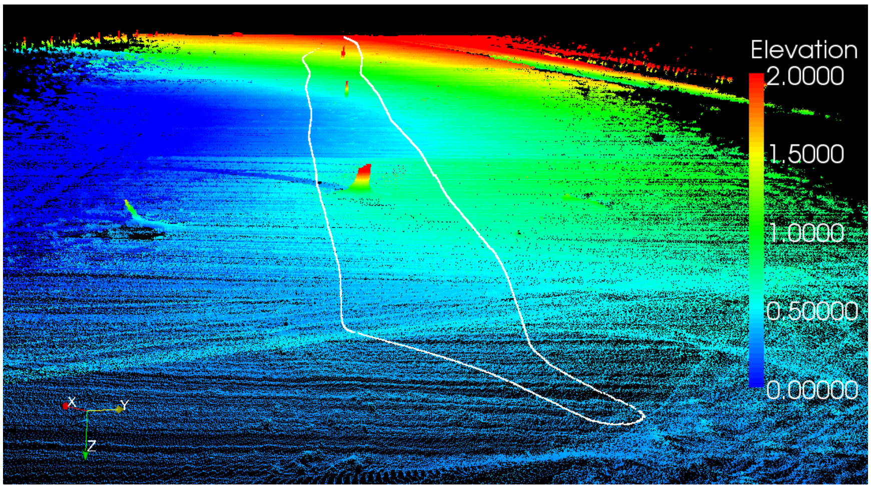


Figure 4.43: Mapped laser range scanner data using the ASNF. The reported trajectory can also be seen in this image. Compared to the commercial GPS/INS (see Figure 4.38), the ASNF has produced a much smoother estimate of the trajectory and the mapped data appear more consistent. The ground surface has separated less, although the drift can still be seen, particularly in the form of a sensor shadow at the tree.

For technical reasons it was not possible to run all of the navigation systems simultaneously, but it was possible to produce the COTS GPS/INS and the ASNF TDCP/INS solutions from *identical* data, during one run. A comparison between the two can be seen in Figure 4.42. This experiment was repeated several times and the results displayed are typical. In all cases, the GPS/INS estimates drifted suddenly when the vehicle passed near the tree, suggesting multipath errors. This is also suggested by the increased noise of the raw GPS observations in this region. The GPS/INS system produces a smoother result than the raw GPS observations due to the influence of the INS, but this appears to cause a drift from the ground truth. Both filtered solutions drift by approximately one metre over the entire trajectory, yet the TDCP/INS does not contain sudden rifts in the pose estimate that are a characteristic of GPS based systems. The TDCP/INS solution was used to map the laser data, and the resulting point cloud is shown in Figure 4.43. Unlike the mapped data from the GPS/INS solution in Figure 4.38, the vertical features appear to be vertical in the data and the ground planes are separated by less as the navigation solution is no longer occluded.

Figure 4.44 shows a top down view of the data surrounding the tree trunk. With the TDCP/INS navigation solution, the apparent ground height has deviated by

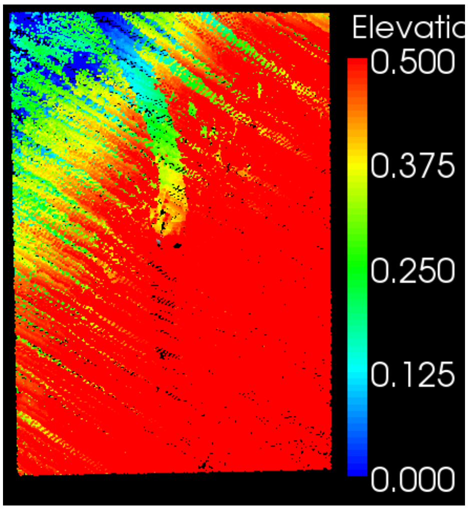


Figure 4.44: A top down view of the mapped data, using the ASNF TDCP/INS navigation system. A difference of 25cm in the mapped height of the ground surface appears in the 20 seconds taken for the vehicle to pass the tree and turn back, compared to the 50cm drift from the GPS/INS solution, seen in Figure 4.39.

approximately 25cm, compared with the 50cm drift using GPS/INS, seen in Figure 4.39. The TDCP/INS system uses no direct observations of the pose of the vehicle. The position estimate is produced by integration, after observing absolute velocities from TDCP and accelerations from the INS. Because the observations are imperfect, the position estimate will certainly drift. However, unlike the GPS/INS solution, the drift occurs in a predictable way and the solution is immune to the sudden rifts caused by global position updates.

Clearly the TDCP/INS local frame navigation solution cannot be used to accumulate and map data indefinitely, because of the drift in the position estimate. It does provide a significant decrease in drift rates compared with stand-alone INS integration, making it appropriate for the local frame solution in dual navigation frame systems, such as those described by [4, 13, 14, 2, 15]. The ASNF TDCP/INS system is capable of providing low drift rates without requiring a vehicle model, which makes it ideal for vehicles operating in unstructured environments, where the ground interaction is difficult to predict.

Chapter 5

Mitigation & Management of Interpretive Errors

Although Chapter 3 provided a geometry based sensitivity analysis, allowing mobile mapping systems to be designed to meet mapping accuracy requirements and track the uncertainty of the map in real-time, the significant issue of *model failure* was raised in Section 3.4, because such failures can cause arbitrarily large, unobservable errors to exists in the map.

The environment model is the interpretive layer added to the sensor information, that is used to *perceive* or infer some higher level information about the world from the sensor data, in order to facilitate some application-specific task. The words *perception* and *interpretation* are used to describe "task-oriented interpretation of sensor data" as in [8, §4]. This layer is almost always present (implicitly or explicitly) in AUGV systems (and autonomous systems in general), because it enables the relevant phenomena in the environment to be inferred from the sensor information, allowing the vehicle to achieve some form of control loop around its task. In the case of AUGVs, the loop can be considered as a model predictive controller, where the perception of the environment is used to predict the interactions of the vehicle and the terrain ahead of time, allowing appropriate control actions to be issued by the vehicle to maximise safety and reliability.

The mitigation of sensor and model failure cannot easily be solved for the general case, because the failure modes are likely to be dependent on the type of environment and sensor, and the specification of the model. In this section, some general design principles that are likely to be useful for a variety of applications are discussed. Although Chapters 3 and 4 have largely focused on spatially aware exteroceptive sensors such as laser range scanners and radars, this chapter pertains to exteroceptive sensors in general.

Some specific examples are given in this chapter:

- Improving a camera perception model to detect potential failure due to poor contrast in the image, caused by environmental effects including lighting conditions, occlusions, or atmospheric conditions such as fog or particulate dust.
- Developing and applying a generic form of fault detection for an thermal infrared camera and a colour camera, to detect unspecified faults.
- Combining laser and radar information to increase the robustness of the laser in poor atmospheric conditions including fog, rain and particulate dust.

Empirically, wind swept dust in the air has been one of the principle causes of failure for the CAS Outdoor Research Demonstrator platforms (seen in Figure 3.8) when relying on sensors in the visible and near infra-red spectrum (cameras and lasers). This has also been noted as a concern in other systems [4, 20, 21]. For this reason, the mitigation of errors in the interpretation of sensor data due to particulate dust in the atmosphere is used as a recurring example in this chapter.

5.1 Improvements to the Perception Model

Failure due to errors in the perception model can be avoided or reduced in severity by increasing the fidelity of the model. Improvements can be made by accounting for nuances of the environment and by including specific failure modes of the sensor. By conceiving of possible model deficiencies prior to deployment, or by rigorous testing to determine the causes of failure after they occur, additional complexity can potentially be incorporated into the model so that it better represents the true environment. With the inclusion of the fault models, the perception model failures are avoided and the sensor failures are detected (whether it is a hardware failure or a failure to produce useful information in the environmental context). This is similar to the 'Multiple Hypothesis Filters' approach to Fault Detection and Identification (FDI), where one filter is used for each pre-conceived fault, under the hypothesis that the particular fault has occurred. If the information matches the filter, the fault is detected and identified [11, §3.3].

As an example, consider a video camera operating in the visible spectrum. There are many phenomena in natural environments that cause a reduction in the utility of the sensor. These include low (or no) light conditions, near-field occlusions, or atmospheric effects such as dust or fog, all of which reduce the contrast in the images. If the perception model assumes that the images have usable information in each frame, relating to the surroundings, then the presence of these environmental effects may reduce the robustness of the interpretation of the sensor data. The fidelity of the model can be increased by explicitly incorporating the possibility of such conditions.

Information theoretic measures such as Shannon information or entropy (average Shannon information) [44, §2.4] of the raw sensor data can be used to determine when there is not enough information in the sensor stream to be useful for processing, with some independence from the algorithm that uses the data. Sensor performance metrics can be difficult to specify for complicated algorithms, whereas the information theoretic approach provides a general purpose measure of the utility of raw sensor information. Although this is generally true, the appropriateness of such metrics depends on the type of sensor and processing method, and it is certainly not guaranteed. For example, if an image processing algorithm is specifically designed to determine whether the images were taken at night or day, then an image with almost no Shannon information (all black) actually contains a large amount of useful information to that algorithm. In general, however, streams of raw sensor data that contain little

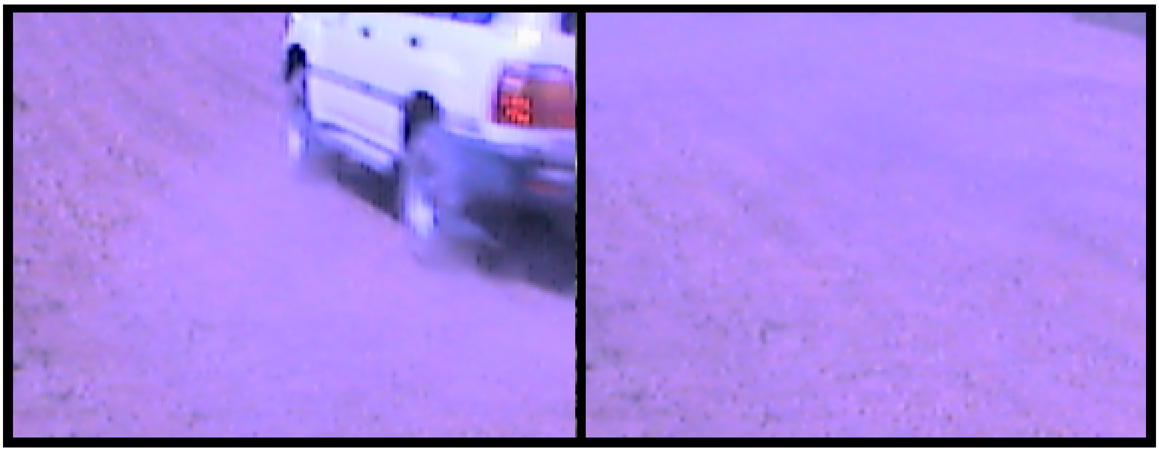


Figure 5.1: A vehicle is driven on a dirt road past the colour camera. The dust stirred behind the vehicle lingers in the atmosphere for several seconds, during which time the colour camera images appear washed out.

or no Shannon information are likely to be of little use to processing algorithms, and may even indicate a sensor fault.

5.1.1 Detecting Poor Contrast in Colour Video

In this section, Shannon information is used as a general performance metric for a colour video information stream. This will be used to detect the presence of environmental effects such as particulate airborne dust, that reduce the contrast of the images.

Colour video was recorded while a UGV was manually driven in a well lit area, free of dust and fog. A second video was recorded of a car driving along a dirt road, trailing a cloud of dust behind. Figure 5.1 shows two frames from the camera, with the vehicle and the resulting washed out image due to the airborne dust.

Given a discrete set of possible pixel values A_X, the average Shannon information $H(X)$ is given by:

$$H(X) = \sum_{x \in A_X} P(x) \log_2 \frac{1}{P(x)} \tag{5.1}$$

where $\log_2$ is used to obtain information in units of bits [44, §4, p73]. If the natural log is used, the units are termed 'nats'. When using bits as the unit of information, the quantities can be interpreted in natural terms as "the size of a file that encodes the outcomes of a random experiment" [44, §4].

For this example, the colour image was converted to an 8 bit integer grey scale (using the Matlab rgb2gray function), yielding $A_X \equiv [0, 255]$ and a maximum possible information content of 8 bits per pixel. The probabilities $P(x)$ were calculated by using a histogram. Given a single image, for each possible integer pixel value $x \in A_X$, $P(x)$ was calculated by counting the number of pixels in that image with the value x, and dividing by the total number of pixels. Equation 5.1 can then be used to calculate the average Shannon information in the image.

The entropy calculations are shown in Figure 5.2, for the data-set in which a car passed the sensor viewpoint at a progressively faster velocity on a dirt road. This resulted in an increasing amount of airborne dust trailing behind the vehicle. The figure shows that as the vehicle enters the scene, the amount of information increases. Once the vehicle leaves the frame, the information content is significantly lowered to less than 4 bits per pixel, due to the cloud of dust causing a drop in contrast in the image. Interestingly, the drop in information content appears to be correlated to the speed of the passing vehicle, due to the correlation of the velocity and the volume of dust that is stirred into the atmosphere. In Figure 5.3, the information content is shown for the manual run, with no airborne dust. At no point does the information content drop below 4 bits per pixel, so this value can be used as a minimum nominal information content. The contrast is even more significant for comparing day and night driving, for detecting when an object is placed directly over the camera lens, or when the camera is pointing at the sun, because there is an even lower amount of Shannon information content than for the dusty images. By incorporating this information theoretic measure to model several common causes of sensor failure in natural environments, the robustness of the model is increased. This is potentially a useful addition to any perceptive process involving colour cameras in natural environments.

Achieving reliability, safety and robustness by improving the perception model relies

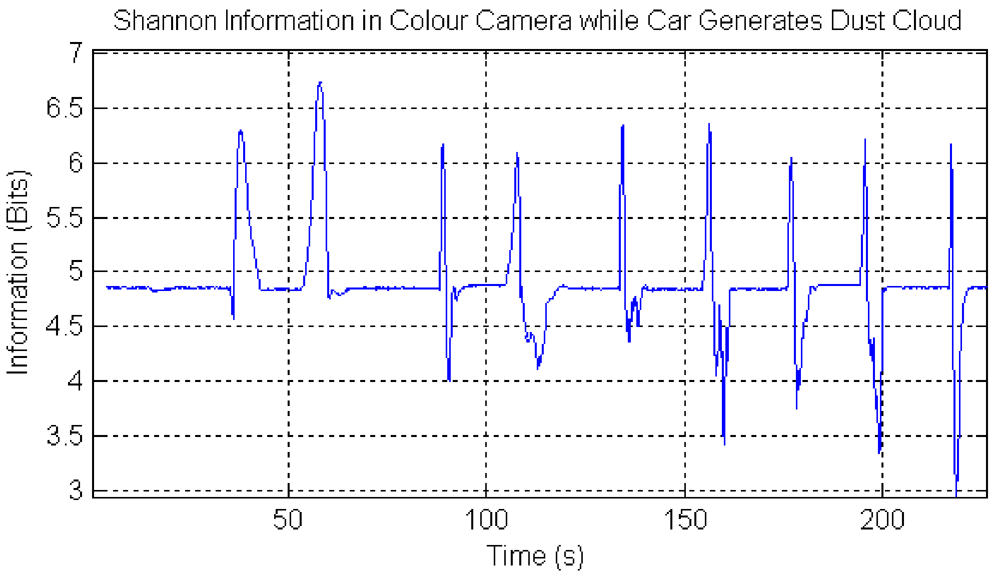


Figure 5.2: The entropy or average Shannon information of images in a colour video sequence, in which a vehicle repeatedly drives across the camera viewpoint along a dirt road. The appearance of the vehicle is relatively informative, whereas the dust cloud that follows is very uninformative. The vehicle drives faster by approximately 10km/h each time, creating a larger dust cloud and a corresponding decrease in information content. The first two passes at 40 and 59 seconds generate so little dust that the camera entropy is unaffected.

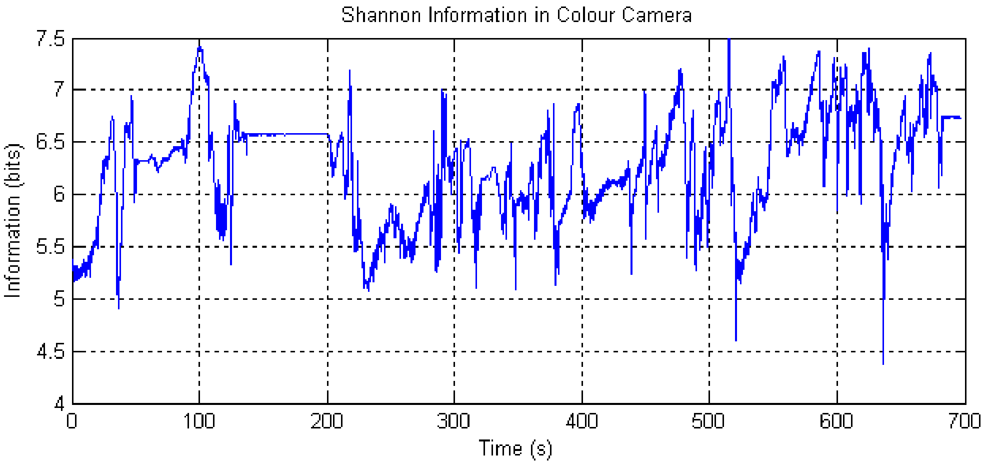


Figure 5.3: The entropy or average Shannon information of images in a colour video sequence, taken while driving manually through a natural environment. Care is taken to avoid any dusty areas. The images always have more than 4 bits per pixel of information. This is significantly higher than the information content of images with airborne dust, seen in Figure 5.2 to be below 4 bits per pixel.

on the ability to pre-conceive all possible faults, or to be willing to accept that faults *must occur* to be identified. Furthermore, this method requires that identified failure modes can indeed be modeled, which is not guaranteed. If the available sensor information does not span the decision boundary [68] for a particular fault, it cannot be modeled robustly. In other words, there may be no way to use information from a failing sensor to detect that this sensor has failed. If other sensors are available to the system, they may not provide relevant information either. In this case, some form of redundancy is required to address the failure mode.

5.2 Multi-Modal Hardware Redundancy

In many cases, improvements to the perception model will allow potential causes of sensor failure to be detected, thereby reducing the likelihood of model failure. However, in the common case that the system designer is not aware of all possible modes of failure in the system, or if known failure modes are unobservable, then additional hardware redundancy must be added.

Physical hardware redundancy can theoretically be created for any sensor based system by using duplicate sensors to measure the same quantity. If two *identical* sensors are used, then faults are detected when the measurements differ by more than an expected quantity of sensor noise. Two sensors yield robustness in the form of fault detection, but it is not necessarily possible to tell which sensor is at fault. Therefore continued operation after a fault may not be feasible. With three or more sensors, 'voting' can be used to additionally determine *which* sensor has failed [69], unless there are multiple simultaneous failures [11, §3.3]. Unlike the model improvement approach, where specific faults have to be modeled, or where faults are considered to be deviations from a highly accurate process model, hardware redundancy allows for a potentially simpler algorithm to be used, at the expense of the additional hardware. Because the direct comparison of identical sensors does not require explicit knowledge of all the ways in which the sensor can fail, the risk of failure in complicated systems due to unforeseen events is substantially reduced. However, any environmental phenomenon that covers the physical separation of all of the redundant sensors will have the same potentially hazardous impact, completely circumventing the redundancy. The benefit of using identical sensors to allow for simple comparison algorithms is likely to be outweighed by the prevalence of simultaneous model failure for *all* of the supposedly redundant sensors. Some local phenomena may still be detectable, such as an occlusion that covers one sensor only, however many environmental effects such as rain or fog will impact all sensors alike.

Rather than using identical sensors, a more robust approach to hardware redundancy can be achieved by using *different* sensors. In particular, sensors that use a completely different underlying physical process. This can be seen as an 'analytical redundancy' approach to FDI [70, 69]. In this framework, a mathematical model exists to relate the different sensors to some common underlying property. If inconsistencies arise in the estimation of that property, then a sensor may be at fault. For example, the surface temperature of an object could be sensed by a contact thermometer that converts temperature to voltage, and by a thermal infra-red camera, that measures the intensity of infra-red emissions from the object. A model exists to map the different types of sensor data (voltage and infra-red intensity) to the underlying estimate of surface temperature, which can then be compared directly. If there is a discrepancy, this indicates a fault in either one of the sensors, or the models used to translate the sensor data to the common property. This form of *multi-modal redundancy* is likely to be more robust than simple hardware redundancy, because the different physical processes used by the sensor are less likely to fail simultaneously due to unforseen environmental phenomena. Such phenomena may influence all of the sensors simultaneously, but the impact on the sensors is likely to differ as a result of their different physical properties. The practical implementation of analytical redundancy is usually more difficult than the simple comparison method of hardware redundancy, due to the need for accurate sensor models. This also creates additional scope for errors, because modeling inaccuracies may lead to incorrect conclusions about sensor faults.

A hybrid approach between hardware redundancy and analytical redundancy is possible. The different forms of sensor data can be directly compared, without the need for the underlying state to be modeled or estimated at all. Provided that there exists an underlying phenomenon in the environment that gives rise to the various forms of sensor information, this *causal* relationship should correlate the streams of data. This is represented by the simple graphical model in Figure 5.4.

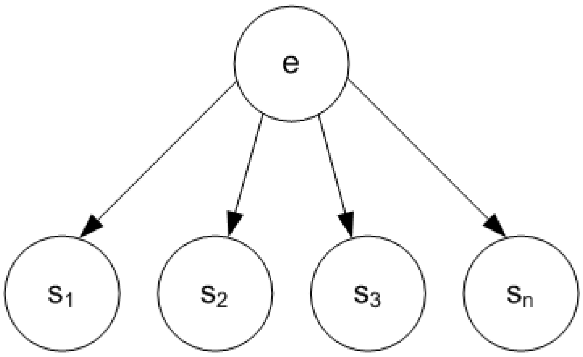


Figure 5.4: If there is an underlying phenomenon in the environment e that gives rise to all of the sensor measurements $\{s_1, s_2, ..., s_n\}$, then the sensor data are correlated by the environment.

This correlation can be measured by the amount of mutual information (MI) [44, §8] in any two sensor streams. The MI between two pieces of information X and Y is denoted $I(X;Y)$ and is given by:

$$\begin{aligned} I(X;Y) &= H(X) + H(Y) - H(X,Y) \\ I(X;Y) &= I(Y;X) \geq 0 \end{aligned} \tag{5.2}$$

where $H(X)$ and $H(Y)$ are the individual entropies of X and Y given by Equation 5.1 (in units of bits), and $H(X,Y)$ is the joint entropy.

MI is suggested in preference to the direct calculation of correlation, which assumes that the two sources of information are jointly Gaussian, because this is not necessarily a good approximation for all multi-modal sensor combinations. For example, consider the comparison of a colour camera and a thermal infra-red camera in an outdoor environment on a sunny day. Objects that *appear* dark to the colour camera are either hotter than the surroundings due to their absorbtion of solar energy (E.g.

tarmac roads in direct sunlight), or are significantly colder than the surroundings if the dark appearance is due to a lack of direct sunlight (E.g. sun shadows). There is still a large amount of mutual information in this relationship, but the joint Gaussian assumption is strongly violated.

The fact that some underlying property of the environment gives rise to the different sensor information streams can be used without ever having a model between the sensor data and the property. Anomalies in the mutual information between sensors can be used to indicate a fault in the same way that comparisons of identical sensors provide hardware redundancy. This form of multi-modal redundancy is better than the duplicate sensor approach, because the sensors are likely to have different failure modes. Furthermore, the extra modalities of sensor information are useful for discrimination or classification of the environment even *before* a sensor fails. In contrast, identical sensors provide no additional useful information *until* one of them fails. The reduction in common failure modes and the discrimination power are both due to exactly the same thing: the different physical properties of the sensors.

The benefit of multi-modal sensors can be seen in an information theoretic context. Two identical sensors with the same viewpoint will provide almost identical data, subject to the sensor noise. Although the sum of Shannon information from both sensors $H(X) + H(Y)$ will be twice the size of the Shannon information from either one on its own, the joint information $H(X, Y)$ will reveal that there is not actually any additional information from the second sensor due to the *redundancy.* The Equation for the joint entropy can be expressed as:

$$\begin{aligned} H(X,Y) &= H(X) + H(Y|X) \\ &= H(Y) + H(X|Y) \end{aligned} \tag{5.3}$$

This shows that when the sensors are not assumed to be independent, the total amount of Shannon information $H(X, Y)$ is equal to the amount provided by the first sensor, plus the additional amount provided by the second sensor, given that we already know something from the first. If the two sensors are identical, nothing is learnt from the second sensor: $H(Y|X) = H(X|Y) = 0$. Additional information is only given *after* a failure creates a difference between the sensors. Multi-modal sensor combinations are correlated by the environment, but they typically measure different properties, such that $H(X|Y) \neq 0$ and $H(Y|X) \neq 0$. If the sensors measure *unrelated* properties, then the two sources of sensor information are independent, such that:

$$\begin{aligned} H(X,Y) &= H(X) + H(Y) \\ H(X|Y) &= H(X) \\ H(Y|X) &= H(Y) \end{aligned} \tag{5.4}$$

so the total amount of information $H(X,Y)$ is indeed the sum of contributions from each sensor. This is the best possible case for classification purposes, because we have the largest amount of information available for discrimination. It is also likely to be the most robust scenario to prevent simultaneous sensor failure due to environmental effects. However, this is useless for redundancy based fault detection because there is no redundancy. The optimal configuration lies in between the two extremes. There should be enough correlation between the sensors to allow redundancy to be useful for fault detection, but there should be enough independence to prevent shared failure modes and to provide discriminative capabilities.

As an example of the discriminative power that two different sensor modes provide, consider a laser range scanner that is used to estimate the nearby terrain. The laser returns are typically interpreted as hard surfaces (ground or trees), and regions that the laser penetrates are considered free space. Figure 5.5 shows that certain phenomena in the environment (such as fog or glass) violate this assumption. There are in fact occupied ($O+$) and unoccupied ($O-$) regions that provide laser returns, and $O\pm$ regions that allow laser penetration. This degrades the ability of the sensor to distinguish between $O+$ and $O-$, which is a core reliability and safety requirement if this information is used for MPC.

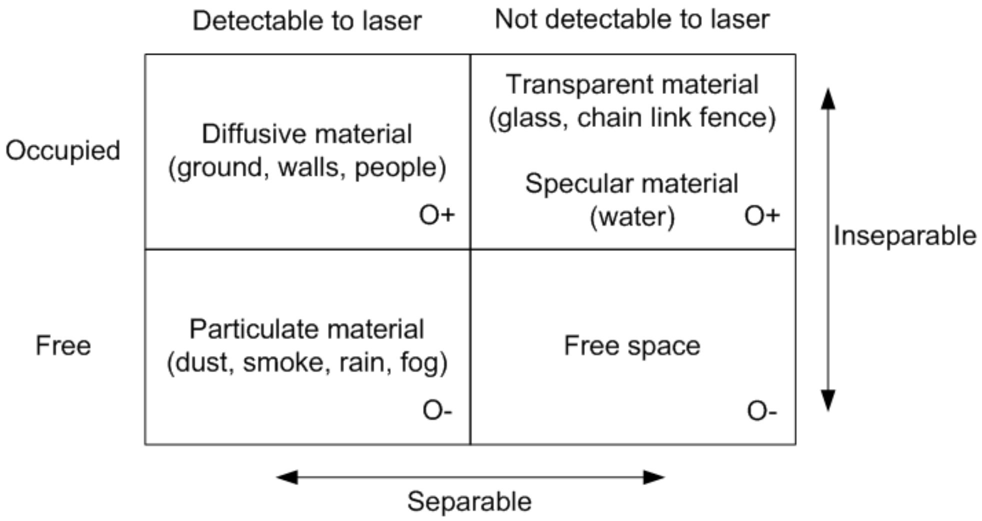
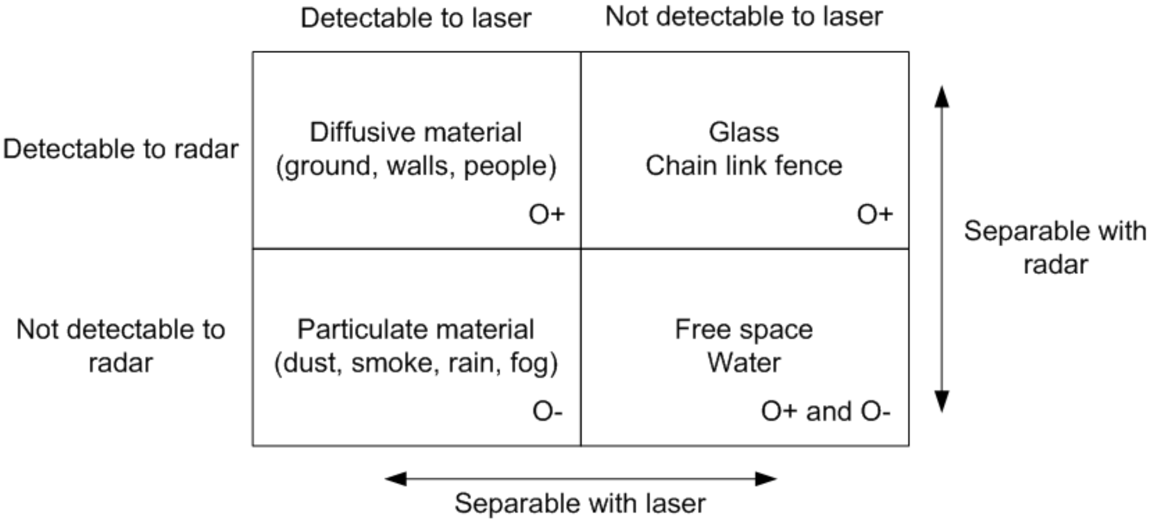


Figure 5.5: When using a laser range sensor, discrimination comes from the different response of the laser to the environment. The environment causes laser returns or allows laser penetration, and this difference forms the axis of discrimination or separability. The figure illustrates that certain phenomena in the environment degrade the ability to classify occupied ($O+$) regions and unoccupied ($O-$) regions.

By augmenting the system with an additional sensor mode in the form of a radar, the discrimination ability can be greatly improved. Figure 5.6 shows the separability matrix for the multi-modal sensor system. Particulate matter such as dust and fog, and items that are transparent to the laser and reflective to the radar such as glass and chain fences, can now be explicitly detected. The system is *almost* capable of detecting the difference between occupied and free space, but water still poses a problem. This could be addressed by the incorporation of a third sensor with a modality that is useful for discriminating the particular properties of water, such as the polarisation of light [71]. Note that the detectability of features such as glass and fence wire depends completely on the specific physical properties and geometry of the feature, and the specification of the sensor. Not all radars will detect reflected energy from all types of glass. Similarly not all lasers will penetrate all types of glass.

In addition to providing a much more accurate internal representation of the world, mutual information between the laser and radar systems can be used as a form of redundancy to detect some types of sensor and model failure. Multi-modal sensor combinations are a very powerful tool for perception, because they can provide a



	Detectable to laser	Not detectable to laser
Detectable to radar	Diffusive material (ground, walls, people) O+	Glass Chain link fence O+
Not detectable to radar	Particulate material (dust, smoke, rain, fog) O-	Free space Water O+ and O-

Figure 5.6: When using a laser *and* a radar, the discrimination ability is much improved over the use of either in isolation. It is possible to explicitly classify the labeled phenomena in separate boxes (although it is not possible to discriminate between items within a box). It is now only water that prevents discrimination between O+ and O- space. To detect water, a different combination of sensors is required, or perhaps the inclusion of a third. In addition to the gain in classification ability, the correlation between the sensors can be used as a form of redundancy to detect sensor and model failures.

richer understanding of the world due to the *differences* between the sensors, and they provide increased robustness and reliability due their *similarities*.

5.2.1 Mutual Information Redundancy for an IR and Colour Camera

In this section an example of mutual information redundancy is given for a thermal infrared camera and a colour video camera. Firstly, the MI between the two sensors is directly calculated to see whether this could provide enough redundancy to detect when one sensor ceases to function correctly. Secondly, a more robust approach is taken by calculating the MI as a function of the time difference between the sensors and comparing the results with the known time offset. This is a useful method to calculate the time synchronisation if it is not otherwise known, but more importantly, if the time synchronisation is known via methods discussed in Section 4.2, then the two estimates of the same quantity (time) are analytically redundant, and can be used for fault detection. This method assumes only that the time difference is known to the system, and that the redundant sensors are correlated by their view of the environment. This model can then be applied to any pair of sensors that satisfies these assumptions.

The sensors were mounted on the CAS Outdoor Research Demonstrator platform as shown in Figure 5.7. The cameras observed an overlapping region of the environment, so the information they provide is correlated, due to the common viewpoint, as illustrated by the graphical model in Figure 5.4. The magnitude of the correlation depends on the extent of the actual correlation of the underlying properties being measured by the sensor, which in this case are the colour and temperature of the terrain. Note that no explicit model of these relationships will be developed.

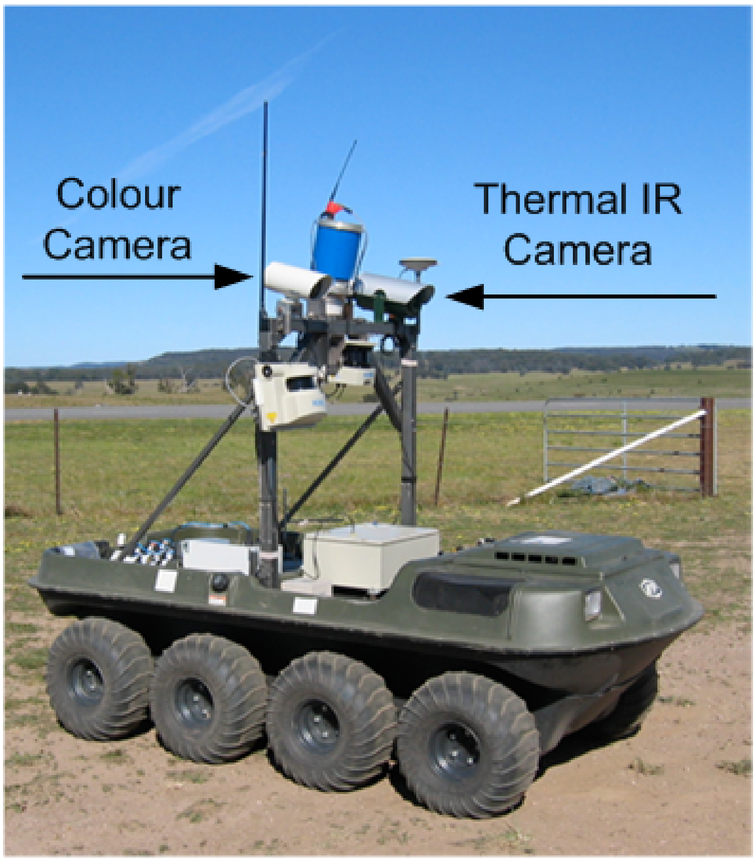


Figure 5.7: The CAS Outdoor Research Demonstrator platform. The location of the colour camera and thermal infrared camera are shown.

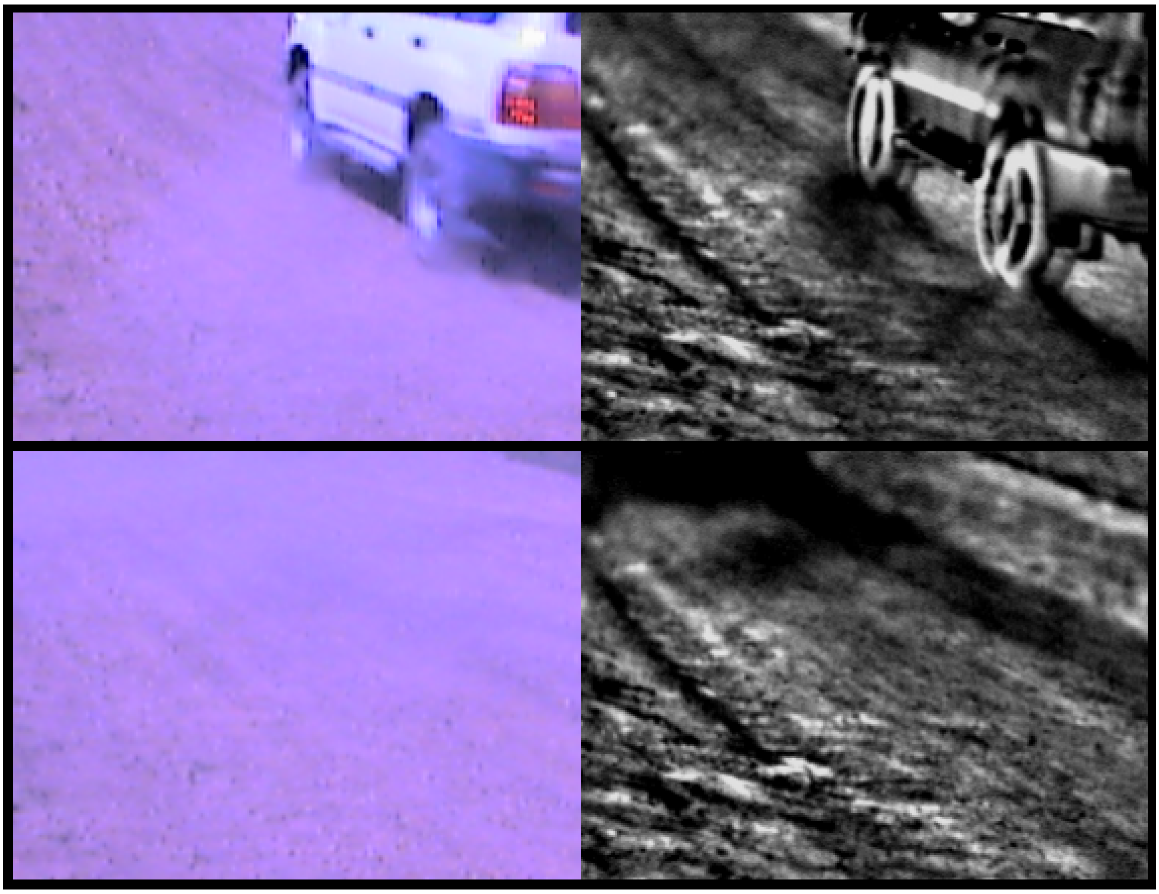


Figure 5.8: Two synchronised pairs of colour and thermal infrared camera images, shown after coarse alignment has been performed. Visibly the car is in the same place in both images, but because no intrinsic or extrinsic calibration has been done, the alignment quality may vary with the geometry of the environment. In the top image pair, a car can be seen driving across the scene. Moments later, the bottom pair shows the trail of dust behind the vehicle. The colour camera image appears to be more affected than the thermal image.

Mutual information is sensitive to the spatial alignment of the sensors. If the two sources of information are improperly aligned, then the correspondences between the data are blurred, causing a reduction in the correlation. However, in this section, a minimal effort was made to align the two cameras, to show the ease of implementation of this technique. Neither camera was calibrated for its intrinsic or extrinsic properties, the cameras were only approximately aligned to have the same field of view, and they had different lens and zoom settings. To align the information, accurate time synchronisation was provided as discussed in Section 4.2. Two synchronised pairs of IR and colour images were manually chosen from the data-sets and corresponding pixels within each were manually labeled. The IR camera was more highly zoomed than the colour camera, so the IR images correspond to a sub region of the colour images. The manually selected corresponding region was cropped from the colour images and the resolution was resampled using linear interpolation so that both sets of images are the same size. Because the pixel correspondence is done in only two images and no intrinsic or extrinsic camera parameters are used, this is likely to be a relatively poor alignment. As the geometry of the environment changes, so will the matching region. The alignment can be seen in Figure 5.8, in which two matching pairs of colour and infrared images are shown. Although a better alignment could be performed by calibrating the two cameras, the coarse alignment is sufficient for this application.

The colour images were transformed to an 8 bit grey scale as in Section 5.1.1. This forms a library of possible colour camera pixel values $A_C \equiv [0, 255]$. The thermal infrared camera provides 8 bit thermal readings, represented by $A_I \equiv [0, 255]$. The maximum possible individual entropies for either sensor is 8 bits per pixel. Jointly, the sensors provide grey and infrared pixel pairs, in the two dimensional space $A_{I,C} \equiv [(0, 0), (255, 255)]$, with a maximum entropy of 16 bits per pixel. The entropy of the individual sensors and the joint entropy are all calculated by forming histograms of the data, to provide frequentist probabilities.

The same data-sets that were used to show the effect of particulate dust in the atmosphere in Section 5.1.1 are used again here. The individual entropies $H(X)$ and $H(Y)$ are calculated, as is the joint information $H(X, Y)$ and the mutual information $I(X; Y)$ from Equation 5.2. This is plotted in Figure 5.9 for a typical run along a dirt road, and in Figure 5.10 for the view of the passing car and trailing dust.

On average, there are 0.5 bits per pixel of MI between these two sensors in a typical outdoor environment, although prominent features within the environment act to correlate the sensors, increasing the MI to over 1 bit per pixel. Large faults where one of the sensors is no longer measuring a valid property of the environment could be detected in this way, because the MI would be reduced. More subtle faults where the sensor performance is reduced but still reactive to the environment in some way

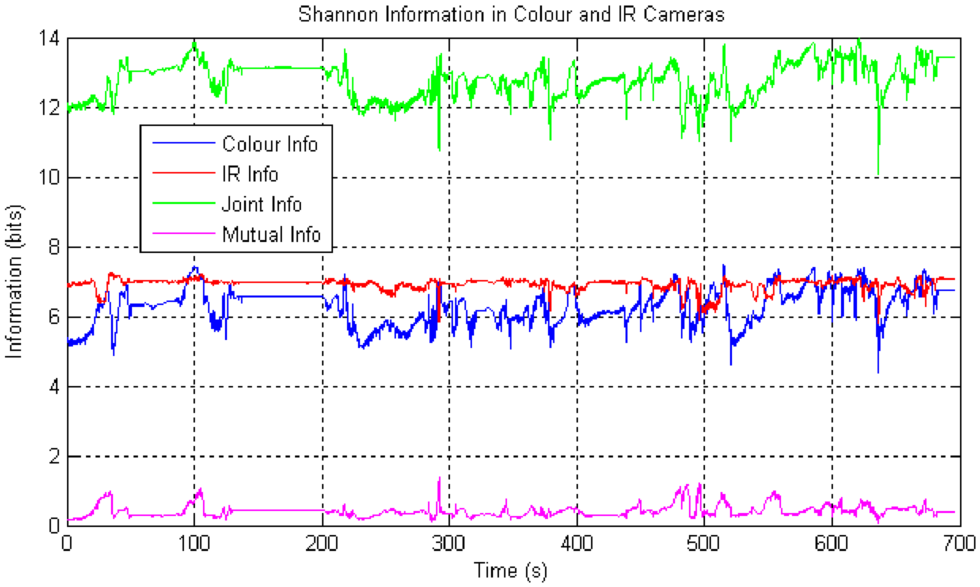


Figure 5.9: The average Shannon information (entropy) of colour and thermal IR images in a synchronised video sequence, taken while driving manually through a natural environment. Also shown are the joint entropy and MI of the two sensors. The correlation of the sensors depends on the variable nature of the environment, so the MI fluctuates throughout the run. The average MI over the run is quite low, at 0.42 bits per pixel.

may not be detectable, because the nominal amount of MI is quite low. This could be improved to some extent by carefully aligning the cameras, and by performing the intrinsic and extrinsic calibration, so if sensor degradation is possible, this would probably be necessary with the direct MI approach.

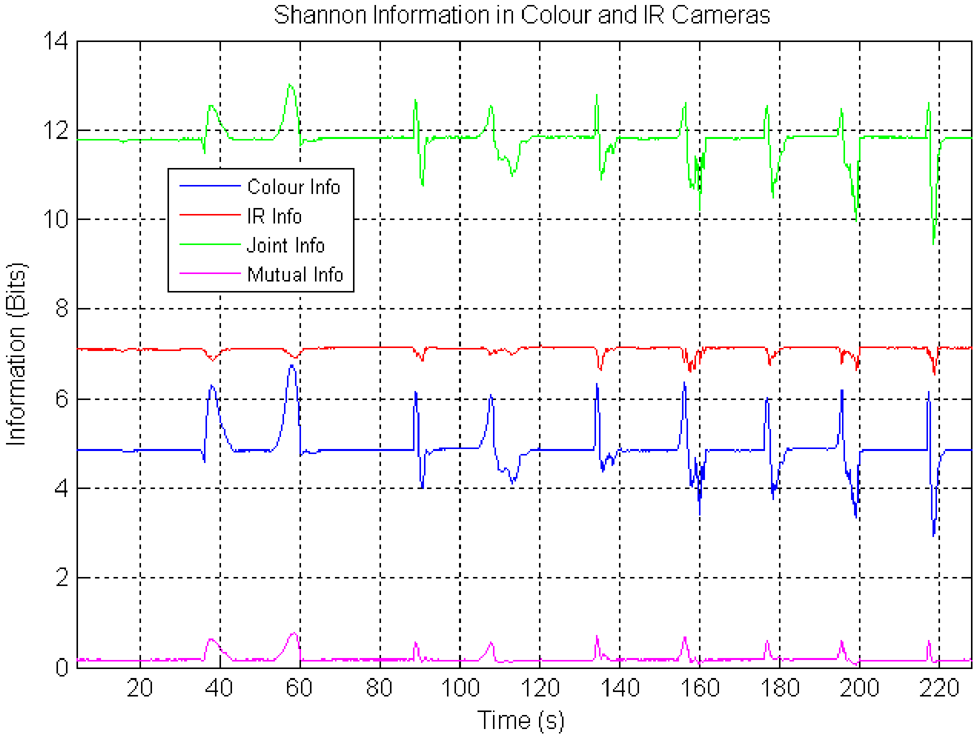


Figure 5.10: The average Shannon information of colour and thermal IR images in a synchronised video sequence, in which a vehicle repeatedly drives across the camera viewpoint along a dirt road. Also shown are the joint entropy and MI of the two sensors. The appearance of the vehicle is informative to the colour camera, but not to the IR camera. The MI is increased when the vehicle is in view, because the images of the passing car correlate the two sensors more effectively than the images of the road. The dust cloud that follows the vehicle is very uninformative to the colour camera, and only slightly uninformative to the IR camera. This may be due to the difference in the wavelength of visible and infrared light, with respect to the particle size of the dust. In Figure 5.8, the dust cloud appears to obscure the IR camera less than the colour camera.

5.2.2 Analytical MI Redundancy with Time Synchronisation

Rather than performing the time-consuming sensor alignment process, a form of analytical redundancy can be introduced to effectively raise the noise floor. In this section, the mutual information between the two sensors is calculated as a function of the time difference between the sensors. In theory, the MI should be highest when the two sensors are synchronised, because the source of correlation is the environment. If the sensor data acquisition time of the two sensors is offset, then as the vehicle moves through the environment (or the environment changes dynamically in view of the vehicle) the sensors are comparing different elements of the environment. In general, nearer things tend to be more highly correlated [65], so the MI is expected to reduce as the time error grows.

This technique can be used to calculate the time offset if it is not known. For the application to analytical redundancy based fault detection, this method relies on the assumption that the time offset *is known independently* of the MI calculations, and that the environment correlates the sensor readings due to a common viewpoint of some underlying property, as shown in Figure 5.4.

The MI between the colour and infrared camera is calculated for a sequence of 100 images for each sensor. At fifteen frames per second, the sequence covers about seven seconds. The MI is then calculated on this entire pixel pair set, by Equation 5.2. The sensors are synchronised to a temporal offset of approximately 0 seconds, ±10 milliseconds, using NTP without a hardware pulse [36]. The MI calculation is then repeated, with a varying temporal offset of ±5 seconds between the two sensors. Figures 5.11 and 5.12 (zoomed) show the resulting MI in bits as a function of the time offset. There is a significant peak near the nominal offset of zero seconds, with an error of only 13 milliseconds.

This result would be highly unlikely if the sensors were not correlated by the environment over the previous seven seconds. The calculation of MI has provided a highly accurate estimate of the temporal offset between the two sensors, which can be directly compared with the independently measured time estimate. Sensors by

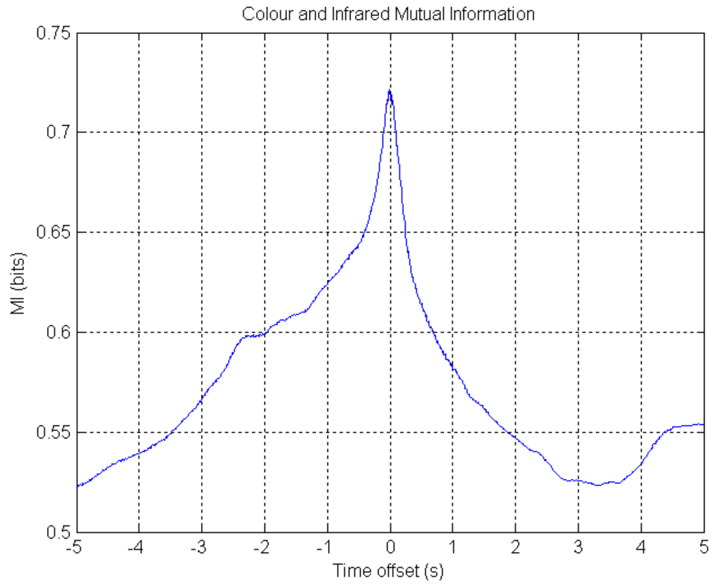


Figure 5.11: The mutual information between a colour camera and a thermal infrared camera, calculated as a function of the time offset between the two sources of information. The maximum correlation between the sensors occurs when the time stamps are correctly matched. If the time offset is known independently, then this can be used to check that the sensor is still correctly perceiving *some* aspect of the environment.

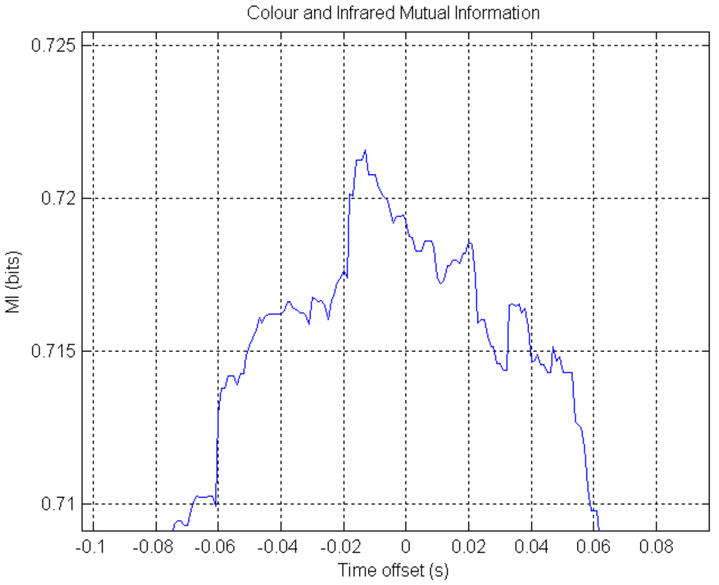


Figure 5.12: The mutual information between a colour camera and a thermal infrared camera, calculated as a function of the time offset between the two sources of information. This shows a magnified version of Figure 5.11. The MI peak occurs 13 milliseconds off the nominal value of 0 ms.

their very definition are used to sense some aspect of the environment. If a fault is defined to be any scenario in which a sensor ceases to be affected by its surroundings, then this technique can be used in real-time to detect sensor faults.

Furthermore, only three assumptions were made:

- The data from each sensor are related to some underlying property in the environment when the sensor is functioning properly.
- The environmental properties that are measured by each sensor are correlated (even if the relationship is not fully understood, modeled, or even able to be explicitly modeled), and this causes the data to be correlated.
- When the sensor is faulty, it no longer senses the relevant underlying property, thus removing the correlation between data from the two sensors.

This technique applies to any combination of sensors that meet these requirements. It is likely that the correlation in some environments will depend on the particular scene, and there may be situations where only a small sub-set of the sensor information is related. Appendix E shows the potential for rapidly calculating maximally informative joint sub-regions of the two data streams so that the technique shown here can be applied to a targeted sub-section of the data.

5.2.3 Discrimination of Dust using Laser and Radar

This section demonstrates the improvement of a perception model by fusing information from two sensors with different modalities. In section 5.1.1, the benefit of increasing the model fidelity by including specific sensor failure modes and nuances of the environment was discussed. Then in section 5.2, it was argued that incorporating different sensor modalities into a perception model provides two main benefits: redundancy is achieved from the similarities between the sensors, and discrimination or classification power is achieved from the differences. This section uses the discriminative power obtained from the fusion of a laser range scanner and a scanning radar to model the effect of airborne dust, which is a nuance in the environment that can cause significant misinterpretation of laser sensor data, when used in isolation.

The CAS Outdoor Research Demonstrator was used again for this experiment, with a 2D range scanning laser (SICK LMS-291) and a custom built 94GHz frequency modulated carrier wave (FMCW) 2D range scanning radar, as was shown in Figure 4.20. The radar provides intensity of the reflected radar energy from the environment along the radar beam axis, in approximately 25cm bins. This can be used to extract a range estimate similar to the laser scanner, by extracting the range of the highest intensity peak. This range estimate is refined to a higher accuracy than the 25cm bin spacing by fitting a quadratic about the peak. In this configuration, the radar data are similar to the laser scanner, but the transmitted radar energy is able to penetrate particulate matter in the atmosphere much better than the laser due to the different transmission wavelength. The attenuation of millimetre wave radar through air-borne dust is several orders of magnitude lower than that of 1 to 10μm wavelength infra-red laser, so "in contrast to the poor transmission through dust at the shorter wavelength, the attenuation in the millimetre wave band is fairly insignificant" [72]. The sensors are mounted near each other, and both point to the front of the mobile platform at a similar angle of declination from the horizon (8°).

To observe the effect of dust on the sensors in a controlled environment, data were gathered from the radar and laser during the same experiment that was described in Section 5.1.1. A car was driven at varying speeds along a dirt road in front of the sensors, while the sensor data were recorded. Although not used in this section, colour and infrared photos from the same experiment can be seen in Figure 5.8 for reference. To illustrate the effect of airborne dust on the laser scanner, the data are graphed as a time sequence of scans. Figure 5.13 shows the scans during a slow pass of the car, with no trailing dust, and Figure 5.14 shows the sequence for a pass at approximately 70km/h with a large trailing dust cloud. The graphs show individual 2D laser scans along the horizontal axis. The first scan is at the bottom of the figures, with increasing time up the vertical axis. The range is represented by the colour, and undetected sensor returns are represented coloured by the maximum

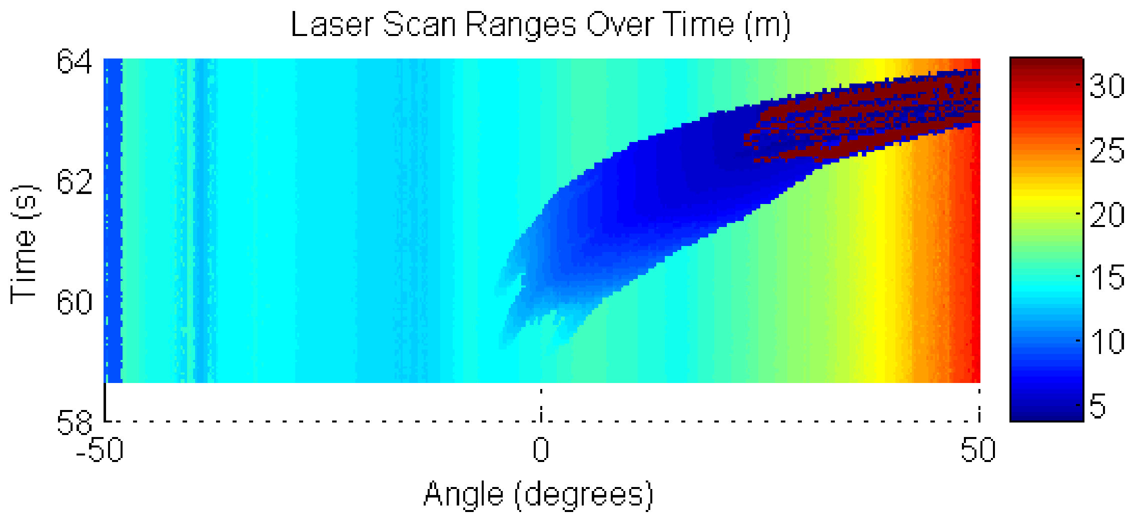


Figure 5.13: A vehicle drives slowly along a dirt road, passing in front of the laser scanner. The time sequence of range scans is shown. The range scans of ±50° are shown along the horizontal axis. The first scan is at the bottom of the figure, with time increasing in the vertical axis. Initially the wheels of the car can be seen, then the body. As the specular car body reaches a shallow angle with respect to the laser beams, some returns are lost, shown as the maximum range in red. The vehicle does not stir any dust on the road at this speed. For comparison, the effect of this pass on information content of the colour camera can be seen at 59 seconds in Figure 5.2.

range in red. Figure 5.13 shows the appearance to the laser of the slowly passing car, with no trailing dust. This can be used to identify the portion of Figure 5.14 that corresponds to the car, and it is labeled by hand accordingly. The difference between the two figures is due to the airborne dust that is stirred up when the car travels more quickly. As seen in Figure 5.14, the initial range measurements immediately behind the car are almost constant for about two seconds, indicating a strong signal return from the front of the dust cloud during this time. As the wind disperses the dust, the cloud becomes thinner and the laser is able to penetrate further. After approximately four seconds the laser is able to correctly perceive the ground behind the cloud. Because the dust is capable of returning such consistent range values, it would be difficult to discriminate between this and a fixed hard surface such as a wall. This is an example of the ambiguity represented in Figure 5.5, that prevents the laser from being a reliable discriminator between occupied and free space, as required for safe ground vehicle autonomy.

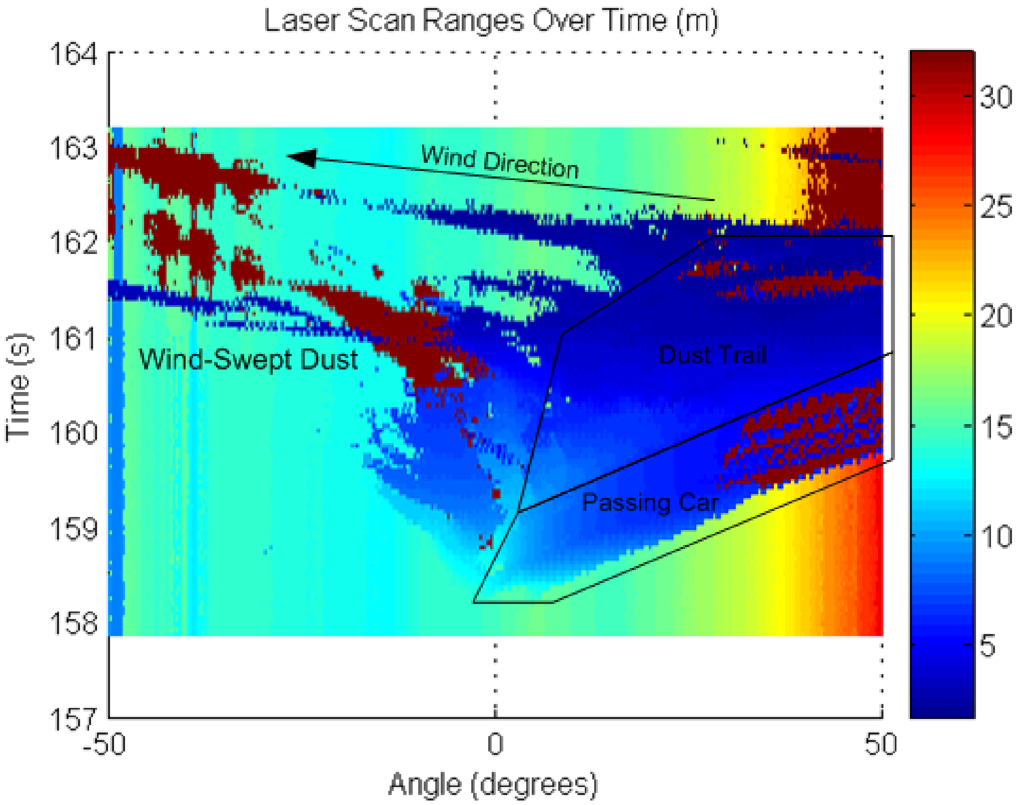


Figure 5.14: A vehicle drives at approximately $70km/h$ along a dirt road, passing in front of the laser scanner. The time sequence of range scans is shown as in Figure 5.13. The car appears identical in the range sequence as in Figure 5.13 (although compressed in the time axis due to the increased speed). A large cloud of dust follows immediately behind the car, and the 'front' of the cloud is sensed at an almost constant range. The wind disperses the dust, and as it thins, the laser penetrates past the front, through the body of the cloud, and eventually back to the ground behind. For comparison, Figure 5.2 shows the effect of the airborne dust on the information content of the colour camera. A reduction in entropy can be seen at 160 seconds, which is restored at 163 seconds once the cloud dissipates.

As shown in Figure 5.6, the discrimination can be improved by using both a laser and a radar sensor. Figure 5.15 shows a synchronised pair of 2D scans from both of these sensors, when the car is in view during the slow pass. The car and the ground reflect the transmitted radar and laser energy similarly, so the sensor profiles are well matched. This corresponds to the scenario in the top left quadrant of Figure 5.6. In Figures 5.16 and 5.17, the passing car traveled faster, leaving a cloud of dust above the dirt road, and this is sensed differently by the laser and radar. The front of the dust cloud returns laser energy, hence it is visible in the laser scan profile. The radar

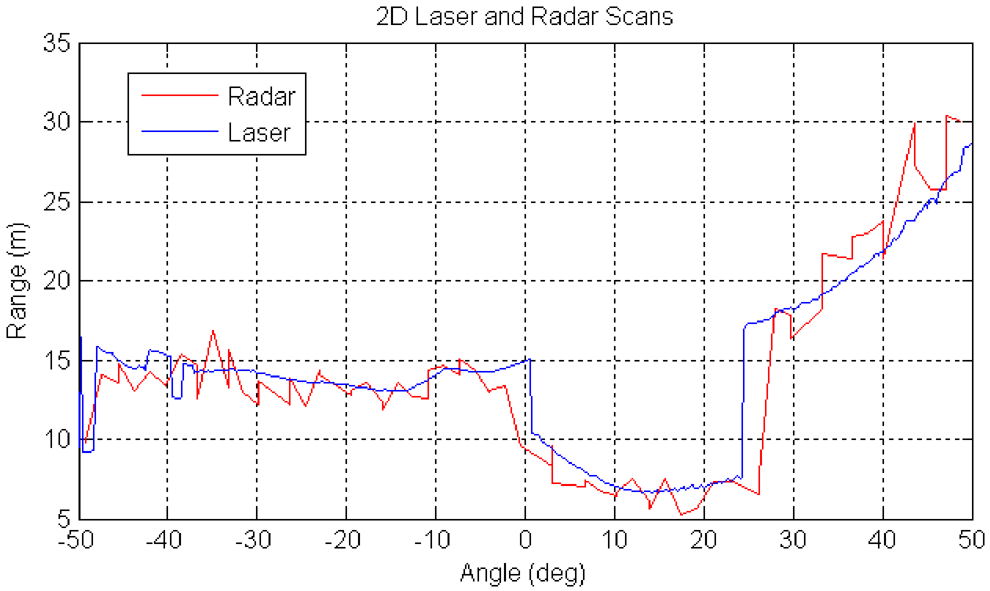


Figure 5.15: A synchronised 2D laser and radar scan, showing the ground intersection profile and the profile of a slowly passing car. Both scans appear similar, although not identical as the sensors are not perfectly co-located; they have a small vertical offset between them.

energy penetrates the dust, so the ground profile behind the cloud is visible in the radar scan. Figure 5.16 shows that the dust can appear similar to a consistent hard object, and Figure 5.17 shows the magnitude by which the laser scans can be affected. In either case, the radar scans are unaffected. The scans differ due to the effect of the environment on the different sensing modalities, corresponding to the bottom left quadrant of Figure 5.6. These two quadrants are separable, so it should be possible to classify scans that are affected by airborne dust, and scans that are due to surfaces that are diffusely reflective to the laser and radar.

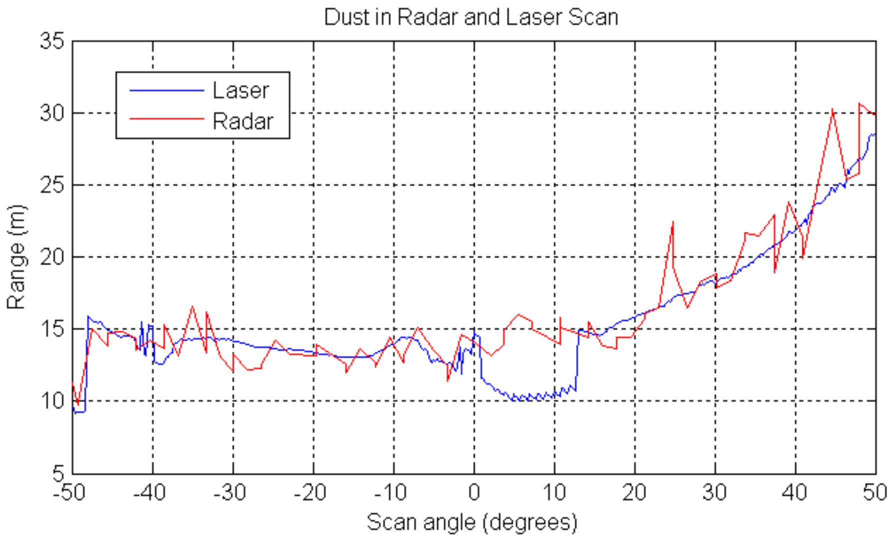


Figure 5.16: A synchronised 2D laser and radar scan, showing the ground intersection profile. The front of a cloud of airborne dust is profiled by the laser scanner, but the radar energy penetrates the cloud and reflects from the ground behind. The sensor data differ significantly due to the effect of the environment on the different sensing modalities.

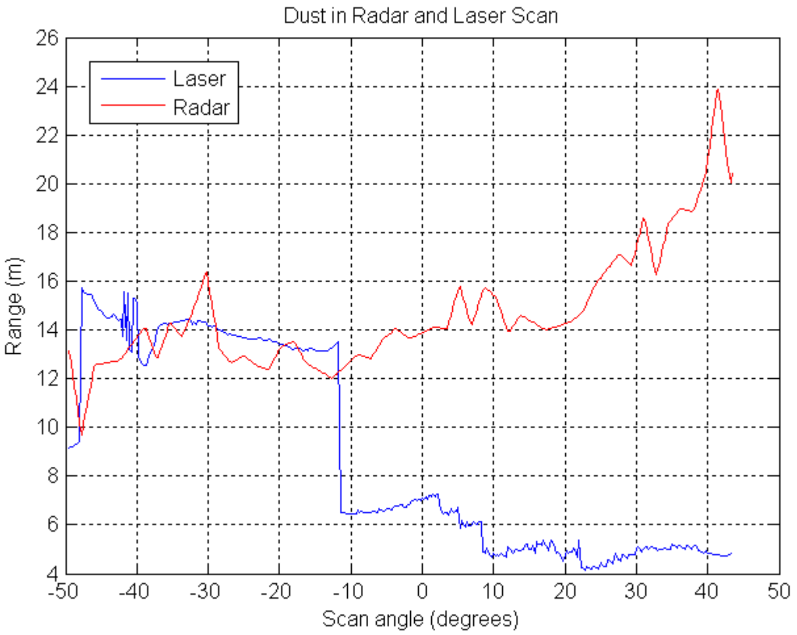


Figure 5.17: Similar to Figure 5.16, this figure shows a large amount of airborne dust. The laser profiles the front of the dust cloud, but the radar still has a clear view of the ground profile behind.

To classify laser returns that are affected by dust, a metric e_{rl} is formed for each matching laser and radar point within a scan pair. This metric is defined to be the squared error between the laser and radar range. When the environment affects the sensors differently, e_{rl} is large, and when the sensors agree, e_{rl} is small. Due to the difference in the angular sensor resolution, the radar scans are resampled using linear interpolation to match the laser bearings, resulting in both sensors having the same number of points per scan, at matching scan angles. The error introduced by this is small because the angular sampling resolution of both sensor is similar. For a laser range r_l, a radar range r_r, both at bearing θ, the error value is given by:

$$e_{rl} = (r_l - r_r)^2 \tag{5.5}$$

The radar laser metric is shown in Figure 5.18, corresponding to the data-set with repeated passes of a car at increasing velocity. For the first two passes of the car, little dust was stirred into the air. As a result, the laser and radar range profiles are similar and the error metric e_{rl} remains below the noise floor. Importantly, this shows that the movement of the car through the laser and radar viewpoint does not induce a significant error between the two sensors. This is because they are almost co-located and they are accurately time synchronised. Figure 5.15 also shows that the car can be seen by both sensors, and that the information from both is aligned. When the car passes more rapidly, stirring more dust into the air behind it, the presence of airborne dust in the viewpoint of the sensors causes a discrepancy, and a corresponding increase in e_{rl}. For all instances where there exists airborne dust, e_{rl} is above the noise floor, clearly demonstrating the discrimination power of this combination of sensors for this phenomenon.

A classification data-set of approximately $55,000$ laser and radar range pairs was gathered from a more typical operating environment of the vehicle. The vehicle was driven aggressively (with sharp turns), causing dust to be stirred into the atmosphere. This reduced the possibility of learning specific features of the controlled experiment above. Figure 4.20 illustrates that the sensors are not perfectly co-located, rather

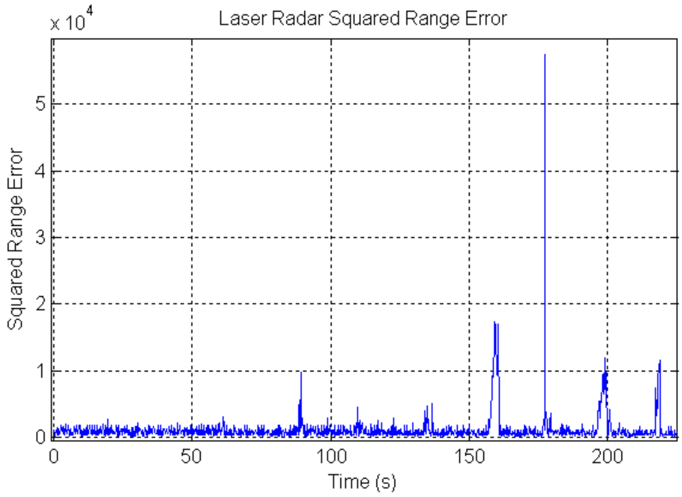


Figure 5.18: The squared range error between a 2D laser range scanner and a 2D radar scanner. A car is driven on a dirt road through the viewpoint of the sensors, at increasing velocities. The first two passes at 40 and 59 seconds do not create a large volume of airborne dust, and there is no corresponding increase in the error metric. This means that the presence of the car in the laser and radar data does not cause a discrepancy, because the sensors are synchronised and almost co-located. This was expected, as Figure 5.15 illustrated the similarity of the car in the data. As the car passes more rapidly, dust affects the laser range profile, but not the radar range, thus causing an error between the sensors. This diagram can be compared to Figure 5.2, which shows the effect of the dust on the colour camera. The first two passes do not stir enough dust to affect the camera either. The largest peak in this error plot corresponds to the scan pair shown in Figure 5.17

they are positioned to yield similar range values to the ground plane directly in front of the vehicle, when the vehicle is positioned on flat ground. Therefore, as the sensor bearing deviates from 0° (pointing directly in front), the error between the sensors increases. The error metric from Equation 5.5 was calculated for the entire classification data-set, and plotted against the bearing at which the measurement was made in Figure 5.19. The error due to the physical misalignment of the sensors can be seen as a curved base line in the figure. In general, the errors are much larger when the bearing deviates from zero degrees. However, the dust causes a significant additional error, and can be seen as a separable region with bearings of ±50°. For this reason, the classification will be restricted to this bearing region. Figure 5.20 shows the error values e_{rl} within this bearing range, plotted against time for the same data-set. When the laser is affected by airborne dust, the errors are significantly higher than the noise floor.

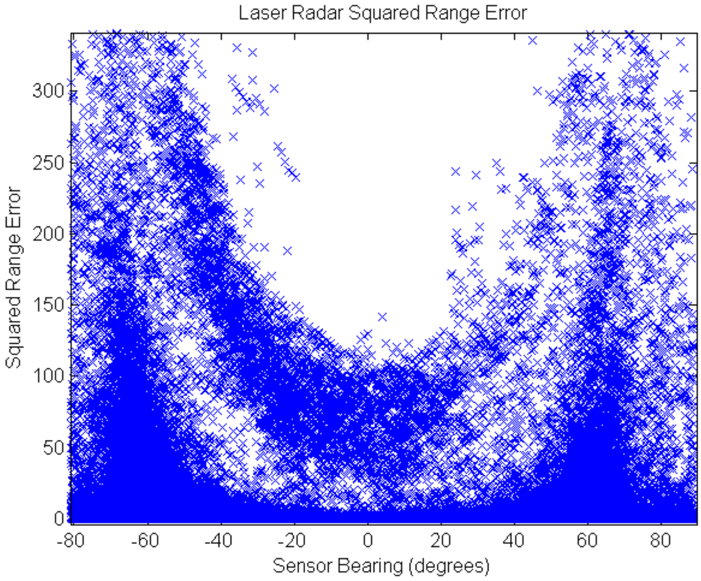


Figure 5.19: The squared range error e_{rl} between a 2D laser range scanner and a 2D radar scanner, from Equation 5.5. The errors are plotted against the sensor bearing. As the bearing deviates from 0°, the error between the two sensors increases because the sensors are not perfectly co-located. The region of dust affected scans can clearly be seen within the ±50° range.

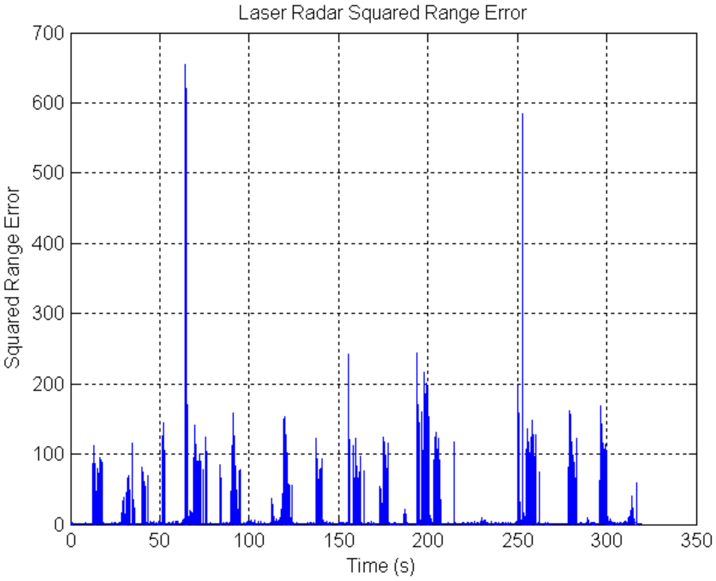


Figure 5.20: The squared range error e_{rl} between a 2D laser range scanner and a 2D radar scanner, from Equation 5.5. The errors are calculated between sensor bearings of ±50°. The vehicle was driven manually with fast turns, causing dust to be stirred, which in turn creates an error much higher than the noise floor.

Expectation maximisation (EM) [73, §20.3], [45, §9], [74] is used to learn a naïve Bayes binary classification for when the sensors agree or disagree, in the latter case due to airborne dust. From the classification data-set of 55,000 samples, a training subset of 2000 samples is used. The resulting Gaussian likelihood functions are difficult to visualise on the same graph because they vary significantly in scale, instead they are conveyed numerically:

$$\begin{aligned} Class &= [clear, dust] \\ P(e_{rl}|clear) &= \mathcal{N}(e_{rl} - 0.2043, 0.0816) \\ P(e_{rl}|dust) &= \mathcal{N}(e_{rl} - 39.6013, 10970) \end{aligned} \tag{5.6}$$

The difference in scale is due to the fact that e_{rl} is close to zero when the environment is clear, yet the dust induces a very large range of error values (with a standard deviation of $100m^2$) depending on factors such as the distance between the dust cloud and the ground behind.

Assuming a uniform prior, the probability of a sensor range being affected by dust is given by Bayes rule:

$$\begin{aligned} P(Class|e_{rl}) &= \frac{P(e_{rl}|class)P(class)}{P(e_{rl})} \\ P(dust|e_{rl}) &= \frac{P(e_{rl}|dust)}{P(e_{rl}|dust) + P(e_{rl}|clear)} \\ P(dust|e_{rl}) &= \frac{\mathcal{N}(e_{rl} - 39.6013, 10970)}{\mathcal{N}(e_{rl} - 39.6013, 10970) + \mathcal{N}(e_{rl} - 0.2043, 0.0816)} \\ P(clear|e_{rl}) &= 1 - P(dust|e_{rl}) \end{aligned} \tag{5.7}$$

The probability of the sensors being affected by dust is shown for different values of

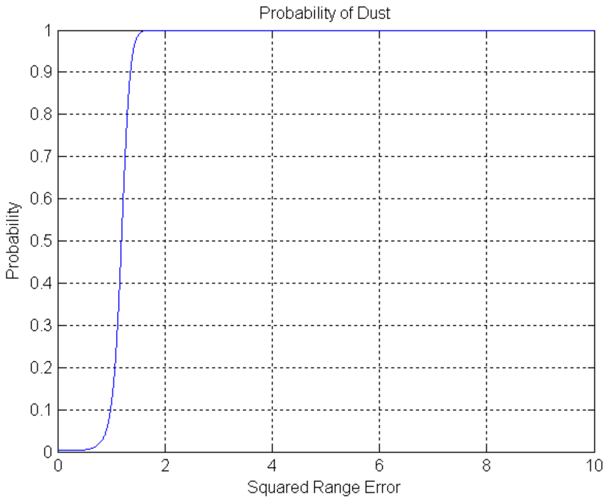


Figure 5.21: The learnt probabilistic model, describing the probability that an error in range between the laser and radar sensors is due to airborne dust, as opposed to noise.

e_{rl} in Figure 5.21. This figure illustrates the learnt laser radar dust model. The class boundary point occurs at $e_{rl} = 1.19m^2$, meaning that range errors smaller than this are more likely to be due to noise than due to dust.

The model is then applied to the entire classification data-set to filter laser points that have been affected by airborne dust. In Figure 5.22, the unfiltered laser data from the entire classification set are mapped to a three dimensional point cloud. In Figure 5.23, the laser points have been filtered using the classification model in Equation 5.7. The figures are coloured by elevation, so that the area affected by dust can easily be seen. The filter has succeeded in removing *every* sample that was affected by airborne dust, while leaving the ground, and other objects in the scene. The object to the centre left of both figures is a stationary car, and the linear objects are fence lines and building walls. Some points corresponding to the wall have been classified as dust due to the geometry of the sensors. When a vertical wall is viewed in close proximity by the laser and radar, the imperfect collocation causes an increased range discrepancy. These figures show that the model that was learnt from the training subset has successfully captured the classification boundary, and is generally applicable to the wider data-set.

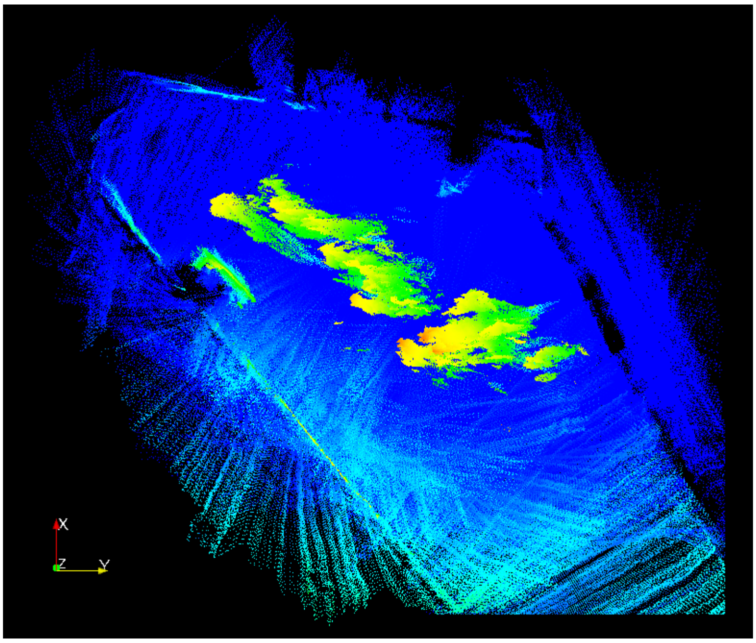


Figure 5.22: A 3D point cloud of all of the laser points from the dust classification data-set, within a bearing of ±50°, coloured by elevation. The area that is affected by airborne dust can be seen in the centre of the scene.

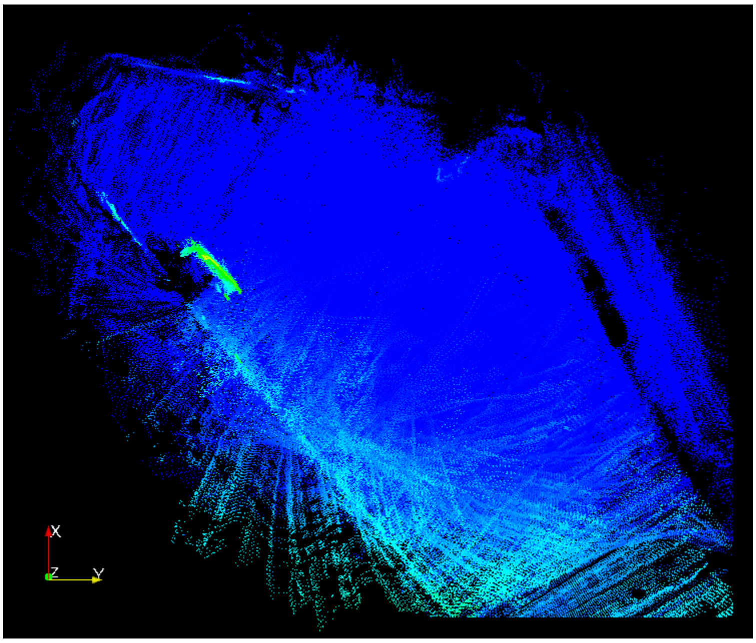


Figure 5.23: A 3D point cloud of all the laser points from the dust classification data-set that are classified as *clear*, using the classification model of Equation 5.7. The dust is completely removed, yet the vehicle to the left of the scene and most of the wall and fence line remain.

5.2.4 Real-time Implementation of Dust Filtration for an AUGV

In Section 5.2.3, it was shown that airborne dust particles that are commonly stirred from the wheels of an AUGV can have a detrimental impact on the ability to perceive the surrounding terrain, when using a scanning laser sensor. To detect the presence of airborne dust, a multi-modal sensor fusion algorithm was described, that combines information from a scanning radar and laser, to detect when the laser is compromised. In this section, a real-time implementation of the algorithm is tested on the CAS Outdoor Research Demonstrator (CORD) platform, to show that it is successful at allowing continued autonomous operation despite the presence of significant quantities of airborne particulate dust. It is likely that the technique will work equally well for other common similar atmospheric conditions such as rain and fog, but the method is only experimentally evaluated on dust because unlike rain and fog, dusty conditions are usually available.

In Chapter 2, general control strategies for AUGVs were discussed, and the specific control system architecture for the CORD platform was shown in Figures 2.3 and 2.6. In this section the control system is modified to incorporate the binary Bayes classification algorithm of Section 5.2.3 as a modular sensor filtration block, shown in Figure 5.24. The classifier could have been implemented in many different ways, but the filtration method was chosen for modularity, as it can be inserted into the control structure with no further modifications to the architecture or the algorithms within the blocks. As a result, the filter is independent of the specific algorithms inside the predictive block, meaning that it should be applicable to *any* perception system that relies on laser range scanner information. Although not strictly required, a brief description of the contents of *this* predictive controller is now given.

The range scanner information is combined with the vehicle pose estimate according to the transformation in Equation 3.12. This mapped information is then stored in a height map, where the mean heights above discrete grid cells are stored. This information is converted to a cost map (which maintains the predicted cost of traversing

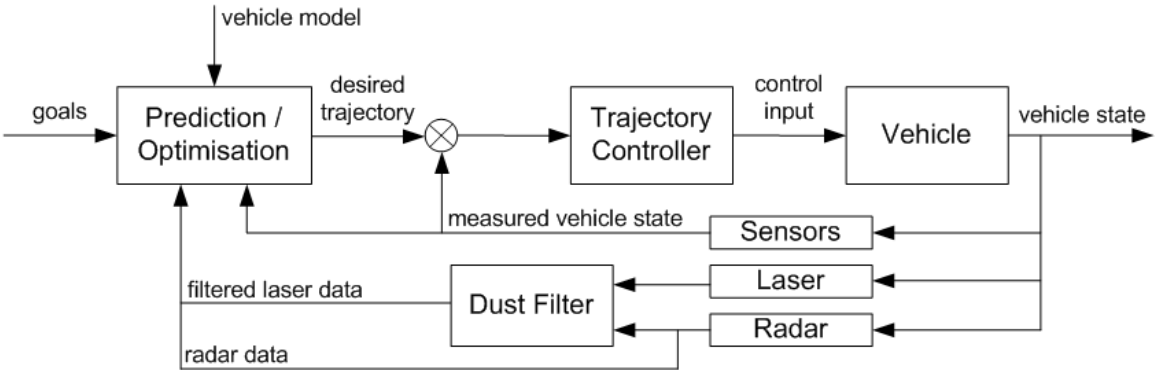


Figure 5.24: The dust model in Equation 5.7, that was learnt from the classification data is used in real-time to filter the laser sensor measurements. Measurements that are more likely to be affected by dust than not, are discarded to prevent them from being *perceived* as an obstacle to the vehicle. This is implemented as an in-line filter, allowing the rest of the control system to remain unmodified. This modularity means that the filter should work with any perception, mapping and planning algorithms contained within the optimisation block, that rely on laser information.

each discrete cell) by using a simple vehicle model, that considers the gradients of the height map. There is a linear cost of traversal ranging from zero to one, where zero is the cost of traversing completely flat terrain, and one is the cost of traversing terrain that is 'rougher' than a vehicle specific threshold. This cost information is dilated by the size of the vehicle to transform it to the configuration space of the vehicle [73, §25.4]. Finally, a combination of the A* [73, §4] and D* [75] heuristic search algorithms are used to plan trajectories over this space. A more detailed description of the control strategy can be found in [76, 77].

To test the ability of the filter to allow autonomous operation in dusty environments while using laser sensors, the vehicle is operated autonomously with and without the filter in the system. For testing purposes, the radar data are only used in the filter, and are not directly fed into the perception system. This is the only point of difference from the control system architecture, shown in Figure 5.24.

A very eroded section of ground was chosen for the test site as a 'worst case' scenario. Figure 5.25 shows the complete erosion of the ground at the site. This photo was opportunistically taken during an unrelated experiment in the same region, to show the ability of the wind to stir significant volumes of dust into the air, independent of

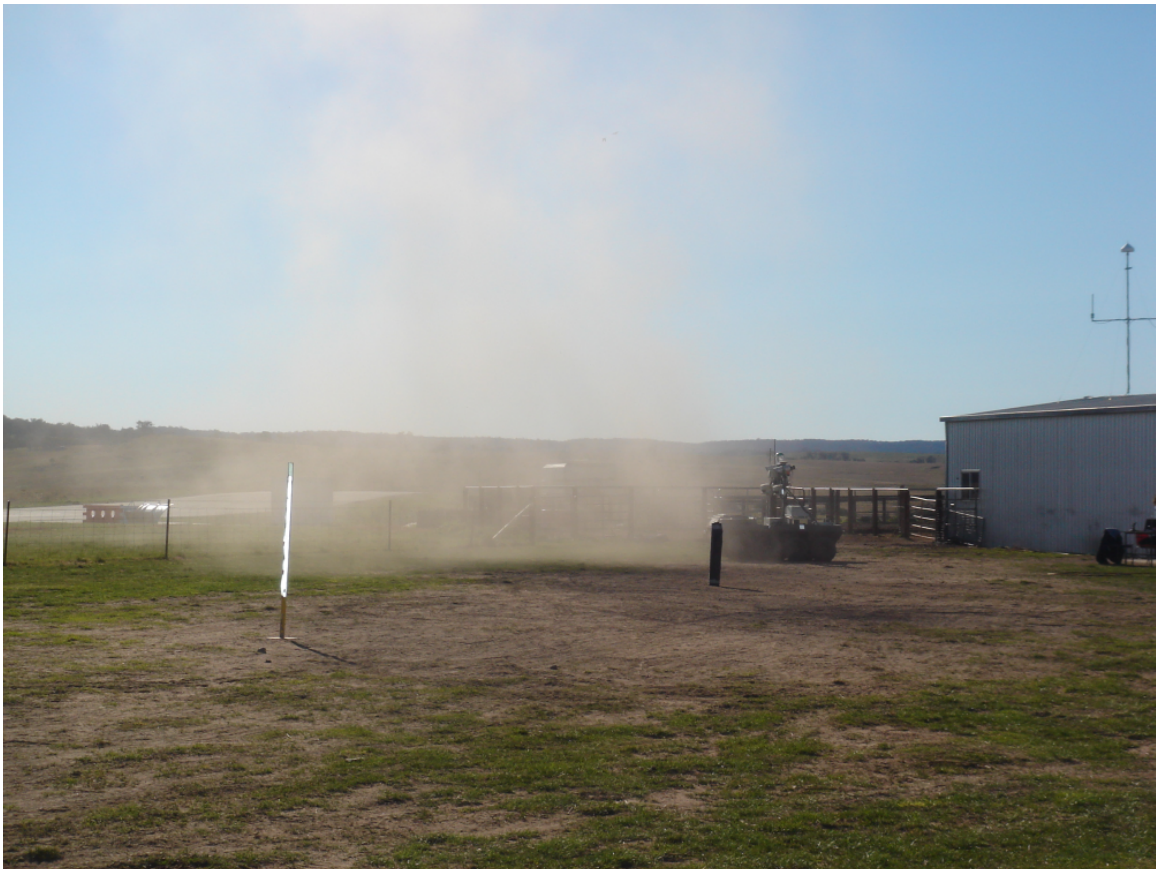


Figure 5.25: This eroded section of ground is used as a 'worst case' test site for the real-time performance of the dust filter. The figure shows the ability of the wind to raise significant volumes of dust into the air, independently of the vehicle. The interaction between the vehicle's tyres and the ground also causes dust to be stirred in the vicinity of the vehicle as can be seen in Figure 5.26. Airborne dust is the primary cause of sensor failure for the laser range scanners in this type of outdoor environment.

the vehicle. Ten waypoints were placed across the test site, to form a single line of approximately thirty metres length, requiring the vehicle to drive out and back for five 'round trips'. Every time the vehicle turned, and sometimes as it drives in a straight line, significant volumes of dust were stirred into the air due to the interaction of the tyres and the ground. In Figure 5.26, two still frames are selected from video footage of the experiment, to show the dust being stirred as the vehicle turns.

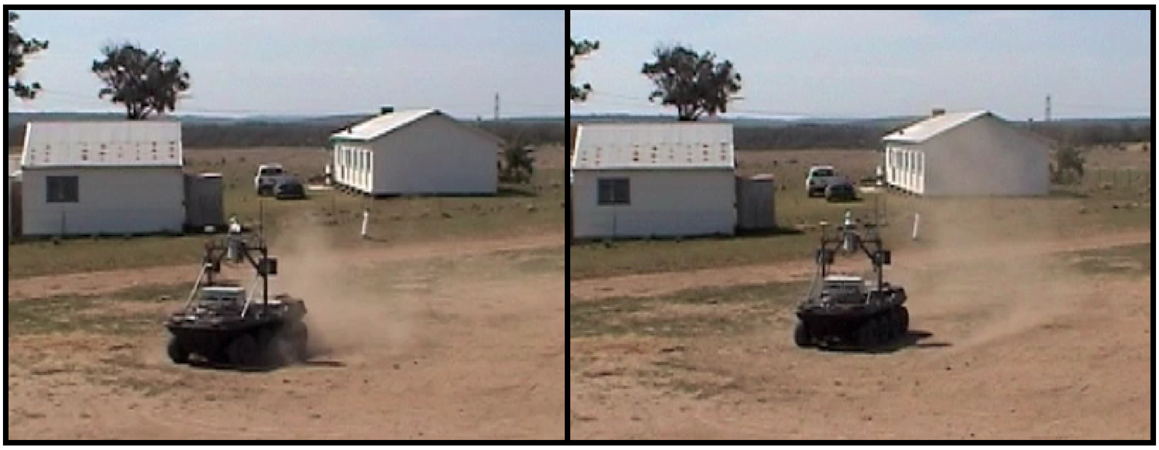


Figure 5.26: The two images have been selected from video footage of the real-time dust experiment, to illustrate the dust that is stirred into the air as the vehicle turns. This volume of dust is enough to reduce the ranges that are reported by the laser range scanner, causing the perception system to consider the region to be non-traversable. The dust model in Equation 5.7 that was learnt from the classification data is used in real-time to filter the laser sensor measurements. Measurements that are more likely to be affected by dust than not, are discarded to prevent them from being *perceived* as an obstacle to the vehicle.

The experiment was repeated three times with the filter and three times without. In all cases where the filter was used, the vehicle successfully reached the ten waypoints without requiring operator intervention. In all cases where the filter was not used, the vehicle failed to reach even the first waypoint. In these cases, as soon as the vehicle turned, enough dust was stirred that the predictive control system considered the immediate region around the vehicle to be non-traversable, and canceled the autonomous controller, requiring the human operator to intervene. Figure 5.27 illustrates representative cost maps from this region, superimposed with the waypoint locations that were used for the experiment. This figure shows the effect of the dust on the cost maps, but was produced by *manually* driving the vehicle over the same region. This was required as autonomous operation was not possible without the filter in place. The figure shows that the dust has been removed, although the car and walls remain.

The experiments unambiguously show the success of the filtration method at allowing continued autonomous operation in sporadically dusty environments using laser range scanners. Although this is a highly successful first pass implementation, further work

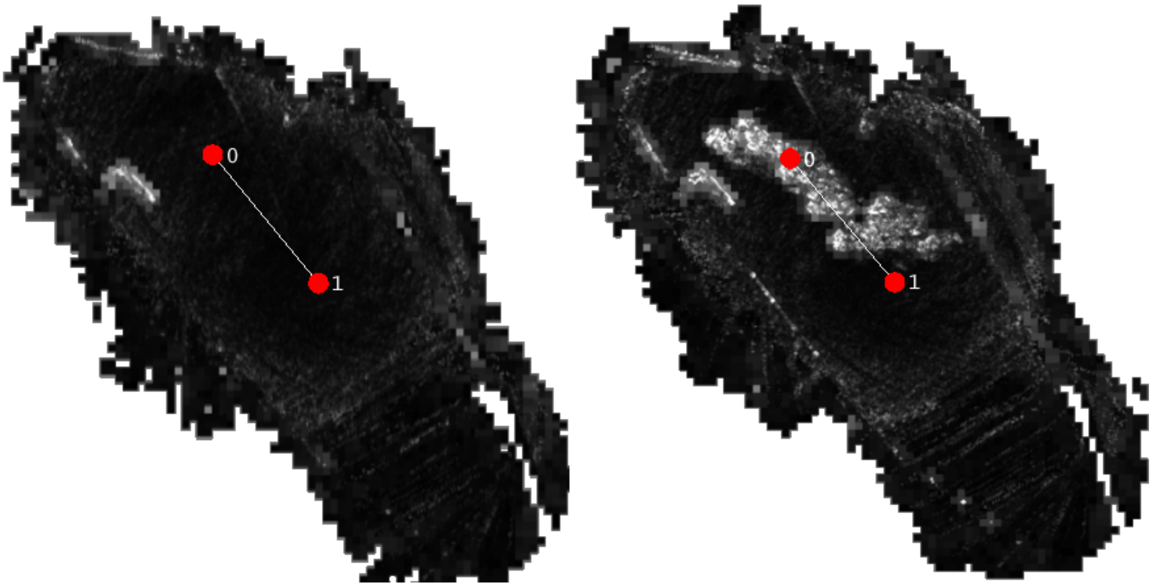


Figure 5.27: The laser and navigation data are transformed using Equation 3.12 to produce an estimate of the terrain surrounding the vehicle. This estimate is convolved with a simple vehicle model to produce a cost map. The two images illustrate the predicted cost of traversal, with completely flat terrain represented as black, and non-traversable terrain shown as white. The image on the left shows the predicted costs when the filter is included in the control loop (as shown in Figure 5.24). The figure on the right shows the prediction when the filter is not used. As expected, the dust creates the perception of non-traversable regions that prevent the waypoints from being reached, and the filter successfully prevents this from happening.

is still required to determine what environmental conditions could cause the filter to fail. The examples that were illustrated by the discrimination matrices in Figures 5.5 and 5.6 show that particular nuances of the environment may be responsible for systematic failures. Testing such a classification based system can therefore only be complete if it spans all such possible nuances. Rather than collecting further statistics on the success and failure of this system to achieve waypoints, a more thorough examination is required to find the types of environment that might cause the system to fail.

Chapter 6

Conclusion

The purpose of this thesis is to promote the reliability and safety of autonomous unmanned ground vehicles operating in unstructured environments. In this chapter, a summary of the thesis is given, followed in Section 6.2 by a list of the specific contributions that have been made. The thesis is concluded in Section 6.3, with a discussion of future work in the field.

6.1 Summary

This thesis analysed the algorithms and systems that are required to enable autonomous operation of an unmanned ground vehicle (UGV) in an unstructured and unknown environment. The primary sources of error in these systems were identified and new solutions to mitigate against their effect were developed. The scope of the thesis was limited to experiments on an AUGV platform, although much of the analysis and the development of algorithms and techniques are extensible to other forms of autonomous system (such as non ground vehicles or stationary mapping platforms).

In **Chapter 2**, the general control system architecture that is used by almost all AUGV systems was presented, with the argument that the robustness of the perception and mapping components is critical to achieving reliability and safety in the whole autonomous system. Two specific systems from the literature were discussed, with emphasis on how and to what extent they achieved reliability and safety. It was argued that any structure within the environment provides significant leverage for achieving robust algorithms, due to the assumptive power that the structure provides. As this thesis is focussed on reliability for systems that operate in unstructured environments, this assumptive power is not generally available. This discussion underlines the need for general purpose algorithms or methods that can increase the system reliability without making any assumptions about the nature of the environment.

In **Chapter 3**, an error analysis was performed to determine the primary sources of error in typical perception and mapping systems. A detailed model of the geometric and temporal sources of error was developed. This enables design-time decisions to be made to maximise achievable map quality, and it provides the ability to estimate the map quality at run-time. By providing a high fidelity error model, the quality of mapped data can be improved and the integrity can be increased because the quality is always known. Reliability is achieved not by requiring systems to have perfect output always, but by *knowing* when the quality of the output is not high enough. This enables some application-specific policy to be enacted (perhaps to halt, or ask an operator for assistance), to preserve the overall reliability and safety of the autonomous system. As a caveat to the fidelity of the geometric and temporal error model, the *interpretation* of sensor information that is performed by perceptive processes was identified in **Chapter 3** as a potential source of unmodeled error, that could decrease the overall system reliability.

With the high fidelity geometric and temporal model specified in **Chapter 3**, **Chapter 4** utilised this information by developing methods to minimise these sources of error, in order to increase the quality of the map, thus increasing the quality of the entire control system. The caveat imposed by perception was then explored in **Chapter 5**. Although a general error model cannot be given due to the application-specific nature of perception algorithms, some general principles were discussed. These were then applied to specific sensors. Multi-modal sensor fusion and information theoretic measures were used to combine thermal infra-red and colour cameras for fault detection, while laser was fused with radar to classify perceptive nuances in the environment. These algorithms were based on a general design principle and methodology that is extensible to other application domains.

The integrity of the perception and mapping components in control system architectures is critical to achieving reliable and safe autonomous systems. By identifying the primary sources of error in typical perception and mapping systems, and addressing these with new methods to calculate and to minimise them, the methods presented in this thesis work together to increase the reliability and safety of AUGVs that operate in unstructured environments.

6.2 Contributions

The major contribution of this thesis is the holistic approach to identify and mitigate the primary sources of error in typical AUGV sensor feedback systems, in order to promote reliability and safety. More specifically, this thesis contributes:

- An analysis of the sensitivity of mapped data to the primary sources of measurable error, both *stochastic* in the form of localisation errors and sensor noise and *systematic* in the form of spatial sensor misalignment and sensor timing synchronisation errors.
- An analysis and development of an error model to calculate the combined geometric map uncertainty, given the joint effect of localisation, noise, misalignment, and time synchronisation.
- A calibration method to learn the optimal sensor alignment parameters with a full six degrees of freedom and a method to calculate the precision of the learnt parameters.
- An extension of this method to learn the joint alignment parameters of multiple sensors, so that their information can reliably be fused into one consistent map.

- The rigorous testing of the calibration procedure for a single sensor, multiple homogenous sensors and multiple heterogenous sensors. This demonstrates the minimisation of systematic errors, the minimisation of systematic contradictions between sensors to allow data fusion and the leveraging of a high precision sensor to assist the calibration of a lower precision sensor.

- Examples of calculating and meeting system design requirements that are driven by a desired mapping accuracy.

- An analysis and discussion of multi-modal sensor fusion to reduce interpretive errors in perception algorithms and an analysis of multi-modal hardware redundancy as a model-free method of fault detection for arbitrary exteroceptive sensors.

- A specific multi-modal sensor fusion algorithm that uses a laser and a radar to detect when airborne dust particles are impeding the use of the laser.

- An evaluation of a navigation system architecture for mapping, that is designed to reduce the effect of stochastic global positioning errors.

6.3 Future Work

Achieving autonomous reliability in unstructured environments is a challenging problem that remains largely unsolved. Although the task becomes more tractable each year as computational power increases and researchers provide new algorithms, commercial scale reliability still seems to be a long way off for systems that perform high level perceptive tasks. Systems such as [4] that achieve long term mission success, do so by incorporating a tightly integrated human supervisor. The fact that the whole system, including the human, can be deemed reliable is an indication that the sensing and actuation of such state-of-the-art systems are sufficient. The bottleneck is the autonomous interpretation of sensor information, or *perception*. In the short to medium term it seems likely that a tightly integrated human supervisor will be needed whenever the environment of an autonomous vehicle lacks sufficient structure. The work in this thesis has been focussed at the lowest levels of sensor processing (low level perception and mapping), to promote reliability from the ground up. The algorithms and methods that have been presented, achieve an increased level of system reliability and safety and are widely applicable. These form a strong basis for further research into system reliability at ever higher levels, which is required for the commercial viability of unmanned systems in unstructured environments.

The particular methods that were developed in this thesis also lead to some immediate areas of future research and development:

- The extrinsic sensor calibration method from Chapter 4 can be extended to additionally provide intrinsic sensor parameters for particular sensors, such as scaling factors and offsets that are required for some radar systems, or possibly intrinsic camera parameters for stereo camera pairs.

- The calibration method is extensible to multiple sensors on *multiple platforms*. The use of a high precision laser to assist the calibration of a lower precision radar was demonstrated in Chapter 4. This type of heterogenous calibration could be spread across multiple platforms. For example, a laser on one vehicle will assist with the calibration of a radar on a separate vehicle. This is analogous to surveying features with one sensor platform, to provide the 'known' features for the calibration for other sensor platforms. The entire joint system of M vehicles with N sensors can be optimally jointly calibrated, with a minimal extension to the calibration method of Chapter 4.

- The point above should be explored for multi-sensor/multi-vehicle systems where the physics or geometry of a platform preclude it from seeing certain features well. For example, an aerial vehicle may not be able to see the entire profile of a vertical feature, yet calibration could be assisted by a ground vehicle that can jointly observe the feature from the ground.

- The mutual information based analytical redundancy technique that was developed in Section 5.2.2 provides a general framework for analytical redundancy, without requiring a model of the interaction between the sensor and the environment. Further experiments are required to validate this method for a wider variety of sensors in different types of environment.

- In Chapter 3, the effect of navigation errors on mapping was shown, and in Chapter 4, a reduction in this effect was shown to be possible using appropriate 'black box' navigation solutions. Investigation of *non* black box solutions, should be done, whereby a tight coupling between terrain perception and navigation may lead to a navigation solution that is optimal with respect to the perception task specifically. Navigation errors are apparent when building terrain maps because they invoke unlikely rifts in the surface. These rifts contain information that could potentially be fed back to the navigation system to refine both the pose estimate and the integrity of the map.

Appendix A

Transformation Sensitivities

The transformation equation to transform a point in the sensor frame s via the vehicle body frame b into the navigation frame n is given by Equation 3.12. The Jacobian matrix for this equation, is given by Equation 3.16, and with the inclusion of practical timing considerations in Equation 3.38. Each of the individual terms of this Jacobian is listed here. Although Equation 3.18 allows for the 'brute force' calculation of these terms, the 'direct differentiation method' provides better accuracy [32]. The partial derivative functions in this appendix should be used in preference to numerical approximation of the Jacobian.

$$\begin{aligned}
\frac{\partial p_x^n}{\partial r_{s,x}^b} &= cos(\psi_{b,z}^n)cos(\psi_{b,y}^n) \\
\frac{\partial p_y^n}{\partial r_{s,x}^b} &= sin(\psi_{b,z}^n)cos(\psi_{b,y}^n) \\
\frac{\partial p_z^n}{\partial r_{s,x}^b} &= -sin(\psi_{b,y}^n) \qquad \text{(A.1)}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial p_x^n}{\partial r_{s,y}^b} &= -sin(\psi_{b,z}^n)cos(\psi_{b,x}^n) + cos(\psi_{b,z}^n)sin(\psi_{b,y}^n)sin(\psi_{b,x}^n) \\
\frac{\partial p_y^n}{\partial r_{s,y}^b} &= cos(\psi_{b,z}^n)cos(\psi_{b,x}^n) + sin(\psi_{b,z}^n)sin(\psi_{b,y}^n)sin(\psi_{b,x}^n) \\
\frac{\partial p_z^n}{\partial r_{s,y}^b} &= cos(\psi_{b,y}^n)sin(\psi_{b,x}^n) \qquad \text{(A.2)}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial p_x^n}{\partial r_{s,z}^b} &= sin(\psi_{b,z}^n)sin(\psi_{b,x}^n) + cos(\psi_{b,z}^n)sin(\psi_{b,y}^n)cos(\psi_{b,x}^n) \\
\frac{\partial p_y^n}{\partial r_{s,z}^b} &= -cos(\psi_{b,z}^n)sin(\psi_{b,x}^n) + sin(\psi_{b,z}^n)sin(\psi_{b,y}^n)cos(\psi_{b,x}^n) \\
\frac{\partial p_z^n}{\partial r_{s,z}^b} &= cos(\psi_{b,y}^n)cos(\psi_{b,x}^n) \qquad \text{(A.3)}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial p_x^n}{\partial \phi_{s,x}^b} &= f(\mathbf{p}^s, \mathbf{\Phi}_s^b, \mathbf{\Psi}_b^n) \\
&= +p_y^s \cos(\psi_{b,z}^n) \cos(\psi_{b,y}^n) \sin(\phi_{s,z}^b) \sin(\phi_{s,x}^b) \\
&\quad +p_y^s \cos(\psi_{b,z}^n) \cos(\psi_{b,y}^n) \cos(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \cos(\phi_{s,x}^b) \\
&\quad +p_y^s \sin(\psi_{b,z}^n) \cos(\psi_{b,x}^n) \cos(\phi_{s,z}^b) \sin(\phi_{s,x}^b) \\
&\quad -p_y^s \sin(\psi_{b,z}^n) \cos(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \cos(\phi_{s,x}^b) \\
&\quad -p_y^s \cos(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \cos(\phi_{s,z}^b) \sin(\phi_{s,x}^b) \\
&\quad +p_y^s \cos(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \cos(\phi_{s,x}^b) \\
&\quad +p_y^s \sin(\psi_{b,z}^n) \sin(\psi_{b,x}^n) \cos(\phi_{s,y}^b) \cos(\phi_{s,x}^b) \\
&\quad +p_y^s \cos(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \cos(\psi_{b,x}^n) \cos(\phi_{s,y}^b) \cos(\phi_{s,x}^b) \\
&\quad +p_z^s \cos(\psi_{b,z}^n) \cos(\psi_{b,y}^n) \sin(\phi_{s,z}^b) \cos(\phi_{s,x}^b) \\
&\quad -p_z^s \cos(\psi_{b,z}^n) \cos(\psi_{b,y}^n) \cos(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \sin(\phi_{s,x}^b) \\
&\quad +p_z^s \sin(\psi_{b,z}^n) \cos(\psi_{b,x}^n) \cos(\phi_{s,z}^b) \cos(\phi_{s,x}^b) \\
&\quad +p_z^s \sin(\psi_{b,z}^n) \cos(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \sin(\phi_{s,x}^b) \\
&\quad -p_z^s \cos(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \cos(\phi_{s,z}^b) \cos(\phi_{s,x}^b) \\
&\quad -p_z^s \cos(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \sin(\phi_{s,x}^b) \\
&\quad -p_z^s \sin(\psi_{b,z}^n) \sin(\psi_{b,x}^n) \cos(\phi_{s,y}^b) \sin(\phi_{s,x}^b) \\
&\quad -p_z^s \cos(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \cos(\psi_{b,x}^n) \cos(\phi_{s,y}^b) \sin(\phi_{s,x}^b) \qquad \text{(A.4)}
\end{aligned}$$

$$
\begin{aligned}
\frac{\partial p_y^n}{\partial \phi_{s,x}^b} &= f(\mathbf{p}^s, \mathbf{\Phi}_s^b, \mathbf{\Psi}_b^n) \\
&= +p_y^s \sin(\psi_{b,z}^n) \cos(\psi_{b,y}^n) \sin(\phi_{s,z}^b) \sin(\phi_{s,x}^b) \\
&\quad +p_y^s \sin(\psi_{b,z}^n) \cos(\psi_{b,y}^n) \cos(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \cos(\phi_{s,x}^b) \\
&\quad -p_y^s \cos(\psi_{b,z}^n) \cos(\psi_{b,x}^n) \cos(\phi_{s,z}^b) \sin(\phi_{s,x}^b) \\
&\quad +p_y^s \cos(\psi_{b,z}^n) \cos(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \cos(\phi_{s,x}^b) \\
&\quad -p_y^s \sin(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \cos(\phi_{s,z}^b) \sin(\phi_{s,x}^b) \\
&\quad +p_y^s \sin(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \cos(\phi_{s,x}^b) \\
&\quad -p_y^s \cos(\psi_{b,z}^n) \sin(\psi_{b,x}^n) \cos(\phi_{s,y}^b) \cos(\phi_{s,x}^b) \\
&\quad +p_y^s \sin(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \cos(\psi_{b,x}^n) \cos(\phi_{s,y}^b) \cos(\phi_{s,x}^b) \\
&\quad +p_z^s \sin(\psi_{b,z}^n) \cos(\psi_{b,y}^n) \sin(\phi_{s,z}^b) \cos(\phi_{s,x}^b) \\
&\quad -p_z^s \sin(\psi_{b,z}^n) \cos(\psi_{b,y}^n) \cos(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \sin(\phi_{s,x}^b) \\
&\quad -p_z^s \cos(\psi_{b,z}^n) \cos(\psi_{b,x}^n) \cos(\phi_{s,z}^b) \cos(\phi_{s,x}^b) \\
&\quad -p_z^s \cos(\psi_{b,z}^n) \cos(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \sin(\phi_{s,x}^b) \\
&\quad -p_z^s \sin(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \cos(\phi_{s,z}^b) \cos(\phi_{s,x}^b) \\
&\quad -p_z^s \sin(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \sin(\phi_{s,x}^b) \\
&\quad +p_z^s \cos(\psi_{b,z}^n) \sin(\psi_{b,x}^n) \cos(\phi_{s,y}^b) \sin(\phi_{s,x}^b) \\
&\quad -p_z^s \sin(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \cos(\psi_{b,x}^n) \cos(\phi_{s,y}^b) \sin(\phi_{s,x}^b)
\end{aligned}
\tag{A.5}
$$

$$
\begin{aligned}
\frac{\partial p_z^n}{\partial \phi_{s,x}^b} &= f(\mathbf{p}^s, \mathbf{\Phi}_s^b, \mathbf{\Psi}_b^n) \\
&= -p_y^s \sin(\psi_{b,y}^n) \sin(\phi_{s,z}^b) \sin(\phi_{s,x}^b) \\
&\quad -p_y^s \sin(\psi_{b,y}^n) \cos(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \cos(\phi_{s,x}^b) \\
&\quad -p_y^s \cos(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \cos(\phi_{s,z}^b) \sin(\phi_{s,x}^b) \\
&\quad +p_y^s \cos(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \cos(\phi_{s,x}^b) \\
&\quad +p_y^s \cos(\psi_{b,y}^n) \cos(\psi_{b,x}^n) \cos(\phi_{s,y}^b) \cos(\phi_{s,x}^b) \\
&\quad -p_z^s \sin(\psi_{b,y}^n) \sin(\phi_{s,z}^b) \cos(\phi_{s,x}^b) \\
&\quad +p_z^s \sin(\psi_{b,y}^n) \cos(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \sin(\phi_{s,x}^b) \\
&\quad -p_z^s \cos(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \cos(\phi_{s,z}^b) \cos(\phi_{s,x}^b) \\
&\quad -p_z^s \cos(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \sin(\phi_{s,x}^b) \\
&\quad -p_z^s \cos(\psi_{b,y}^n) \cos(\psi_{b,x}^n) \cos(\phi_{s,y}^b) \sin(\phi_{s,x}^b) \qquad \text{(A.6)}
\end{aligned}
$$

$$\begin{aligned}
\frac{\partial p_x^n}{\partial \phi_{s,y}^b} &= f(\mathbf{p}^s, \mathbf{\Phi}_s^b, \mathbf{\Psi}_b^n) \\
&= -p_x^s \cos(\psi_{b,z}^n)\cos(\psi_{b,y}^n)\cos(\phi_{s,z}^b)\sin(\phi_{s,y}^b) \\
&\quad +p_x^s \sin(\psi_{b,z}^n)\cos(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,y}^b) \\
&\quad -p_x^s \cos(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\sin(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,y}^b) \\
&\quad -p_x^s \sin(\psi_{b,z}^n)\sin(\psi_{b,x}^n)\cos(\phi_{s,y}^b) \\
&\quad -p_x^s \cos(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\cos(\psi_{b,x}^n)\cos(\phi_{s,y}^b) \\
&\quad +p_y^s \cos(\psi_{b,z}^n)\cos(\psi_{b,y}^n)\cos(\phi_{s,z}^b)\cos(\phi_{s,y}^b)\sin(\phi_{s,x}^b) \\
&\quad -p_y^s \sin(\psi_{b,z}^n)\cos(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\cos(\phi_{s,y}^b)\sin(\phi_{s,x}^b) \\
&\quad +p_y^s \cos(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\sin(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\cos(\phi_{s,y}^b)\sin(\phi_{s,x}^b) \\
&\quad -p_y^s \sin(\psi_{b,z}^n)\sin(\psi_{b,x}^n)\sin(\phi_{s,y}^b)\sin(\phi_{s,x}^b) \\
&\quad -p_y^s \cos(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\cos(\psi_{b,x}^n)\sin(\phi_{s,y}^b)\sin(\phi_{s,x}^b) \\
&\quad +p_z^s \cos(\psi_{b,z}^n)\cos(\psi_{b,y}^n)\cos(\phi_{s,z}^b)\cos(\phi_{s,y}^b)\cos(\phi_{s,x}^b) \\
&\quad -p_z^s \sin(\psi_{b,z}^n)\cos(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\cos(\phi_{s,y}^b)\cos(\phi_{s,x}^b) \\
&\quad +p_z^s \cos(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\sin(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\cos(\phi_{s,y}^b)\cos(\phi_{s,x}^b) \\
&\quad -p_z^s \sin(\psi_{b,z}^n)\sin(\psi_{b,x}^n)\sin(\phi_{s,y}^b)\cos(\phi_{s,x}^b) \\
&\quad -p_z^s \cos(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\cos(\psi_{b,x}^n)\sin(\phi_{s,y}^b)\cos(\phi_{s,x}^b) \qquad \text{(A.7)}
\end{aligned}$$

$$
\begin{aligned}
\frac{\partial p_y^n}{\partial \phi_{s,y}^b} &= f(\mathbf{p}^s, \mathbf{\Phi}_s^b, \mathbf{\Psi}_b^n) \\
&= -p_x^s \sin(\psi_{b,z}^n)\cos(\psi_{b,y}^n)\cos(\phi_{s,z}^b)\sin(\phi_{s,y}^b) \\
&\quad -p_x^s \cos(\psi_{b,z}^n)\cos(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,y}^b) \\
&\quad -p_x^s \sin(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\sin(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,y}^b) \\
&\quad +p_x^s \cos(\psi_{b,z}^n)\sin(\psi_{b,x}^n)\cos(\phi_{s,y}^b) \\
&\quad -p_x^s \sin(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\cos(\psi_{b,x}^n)\cos(\phi_{s,y}^b) \\
&\quad +p_y^s \sin(\psi_{b,z}^n)\cos(\psi_{b,y}^n)\cos(\phi_{s,z}^b)\cos(\phi_{s,y}^b)\sin(\phi_{s,x}^b) \\
&\quad +p_y^s \cos(\psi_{b,z}^n)\cos(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\cos(\phi_{s,y}^b)\sin(\phi_{s,x}^b) \\
&\quad +p_y^s \sin(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\sin(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\cos(\phi_{s,y}^b)\sin(\phi_{s,x}^b) \\
&\quad +p_y^s \cos(\psi_{b,z}^n)\sin(\psi_{b,x}^n)\sin(\phi_{s,y}^b)\sin(\phi_{s,x}^b) \\
&\quad -p_y^s \sin(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\cos(\psi_{b,x}^n)\sin(\phi_{s,y}^b)\sin(\phi_{s,x}^b) \\
&\quad +p_z^s \sin(\psi_{b,z}^n)\cos(\psi_{b,y}^n)\cos(\phi_{s,z}^b)\cos(\phi_{s,y}^b)\cos(\phi_{s,x}^b) \\
&\quad +p_z^s \cos(\psi_{b,z}^n)\cos(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\cos(\phi_{s,y}^b)\cos(\phi_{s,x}^b) \\
&\quad +p_z^s \sin(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\sin(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\cos(\phi_{s,y}^b)\cos(\phi_{s,x}^b) \\
&\quad +p_z^s \cos(\psi_{b,z}^n)\sin(\psi_{b,x}^n)\sin(\phi_{s,y}^b)\cos(\phi_{s,x}^b) \\
&\quad -p_z^s \sin(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\cos(\psi_{b,x}^n)\sin(\phi_{s,y}^b)\cos(\phi_{s,x}^b) \qquad \text{(A.8)}
\end{aligned}
$$

$$\begin{aligned}
\frac{\partial p_z^n}{\partial \phi_{s,y}^b} &= f(\mathbf{p}^s, \mathbf{\Phi}_s^b, \mathbf{\Psi}_b^n) \\
&= +p_x^s \sin(\psi_{b,y}^n) \cos(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \\
&\quad -p_x^s \cos(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \\
&\quad -p_x^s \cos(\psi_{b,y}^n) \cos(\psi_{b,x}^n) \cos(\phi_{s,y}^b) \\
&\quad -p_y^s \sin(\psi_{b,y}^n) \cos(\phi_{s,z}^b) \cos(\phi_{s,y}^b) \sin(\phi_{s,x}^b) \\
&\quad +p_y^s \cos(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \cos(\phi_{s,y}^b) \sin(\phi_{s,x}^b) \\
&\quad -p_y^s \cos(\psi_{b,y}^n) \cos(\psi_{b,x}^n) \sin(\phi_{s,y}^b) \sin(\phi_{s,x}^b) \\
&\quad -p_z^s \sin(\psi_{b,y}^n) \cos(\phi_{s,z}^b) \cos(\phi_{s,y}^b) \cos(\phi_{s,x}^b) \\
&\quad +p_z^s \cos(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \cos(\phi_{s,y}^b) \cos(\phi_{s,x}^b) \\
&\quad -p_z^s \cos(\psi_{b,y}^n) \cos(\psi_{b,x}^n) \sin(\phi_{s,y}^b) \cos(\phi_{s,x}^b) \qquad \text{(A.9)}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial p^n_x}{\partial \phi^b_{s,z}} = &= f(\mathbf{p}^s, \boldsymbol{\Phi}^b_s, \boldsymbol{\Psi}^n_b) \\
&= -p^s_x \cos(\psi^n_{b,z}) \cos(\psi^n_{b,y}) \sin(\phi^b_{s,z}) \cos(\phi^b_{s,y}) \\
&\quad -p^s_x \sin(\psi^n_{b,z}) \cos(\psi^n_{b,x}) \cos(\phi^b_{s,z}) \cos(\phi^b_{s,y}) \\
&\quad +p^s_x \cos(\psi^n_{b,z}) \sin(\psi^n_{b,y}) \sin(\psi^n_{b,x}) \cos(\phi^b_{s,z}) \cos(\phi^b_{s,y}) \\
&\quad -p^s_y \cos(\psi^n_{b,z}) \cos(\psi^n_{b,y}) \cos(\phi^b_{s,z}) \cos(\phi^b_{s,x}) \\
&\quad -p^s_y \cos(\psi^n_{b,z}) \cos(\psi^n_{b,y}) \sin(\phi^b_{s,z}) \sin(\phi^b_{s,y}) \sin(\phi^b_{s,x}) \\
&\quad +p^s_y \sin(\psi^n_{b,z}) \cos(\psi^n_{b,x}) \sin(\phi^b_{s,z}) \cos(\phi^b_{s,x}) \\
&\quad -p^s_y \sin(\psi^n_{b,z}) \cos(\psi^n_{b,x}) \cos(\phi^b_{s,z}) \sin(\phi^b_{s,y}) \sin(\phi^b_{s,x}) \\
&\quad -p^s_y \cos(\psi^n_{b,z}) \sin(\psi^n_{b,y}) \sin(\psi^n_{b,x}) \sin(\phi^b_{s,z}) \cos(\phi^b_{s,x}) \\
&\quad +p^s_y \cos(\psi^n_{b,z}) \sin(\psi^n_{b,y}) \sin(\psi^n_{b,x}) \cos(\phi^b_{s,z}) \sin(\phi^b_{s,y}) \sin(\phi^b_{s,x}) \\
&\quad +p^s_z \cos(\psi^n_{b,z}) \cos(\psi^n_{b,y}) \cos(\phi^b_{s,z}) \sin(\phi^b_{s,x}) \\
&\quad -p^s_z \cos(\psi^n_{b,z}) \cos(\psi^n_{b,y}) \sin(\phi^b_{s,z}) \sin(\phi^b_{s,y}) \cos(\phi^b_{s,x}) \\
&\quad -p^s_z \sin(\psi^n_{b,z}) \cos(\psi^n_{b,x}) \sin(\phi^b_{s,z}) \sin(\phi^b_{s,x}) \\
&\quad -p^s_z \sin(\psi^n_{b,z}) \cos(\psi^n_{b,x}) \cos(\phi^b_{s,z}) \sin(\phi^b_{s,y}) \cos(\phi^b_{s,x}) \\
&\quad +p^s_z \cos(\psi^n_{b,z}) \sin(\psi^n_{b,y}) \sin(\psi^n_{b,x}) \sin(\phi^b_{s,z}) \sin(\phi^b_{s,x}) \\
&\quad +p^s_z \cos(\psi^n_{b,z}) \sin(\psi^n_{b,y}) \sin(\psi^n_{b,x}) \cos(\phi^b_{s,z}) \sin(\phi^b_{s,y}) \cos(\phi^b_{s,x}) \quad \text{(A.10)}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial p_y^n}{\partial \phi_{s,z}^b} &= f(\mathbf{p}^s, \mathbf{\Phi}_s^b, \mathbf{\Psi}_b^n) \\
&= -p_x^s \sin(\psi_{b,z}^n) \cos(\psi_{b,y}^n) \sin(\phi_{s,z}^b) \cos(\phi_{s,y}^b) \\
&\quad +p_x^s \cos(\psi_{b,z}^n) \cos(\psi_{b,x}^n) \cos(\phi_{s,z}^b) \cos(\phi_{s,y}^b) \\
&\quad +p_x^s \sin(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \cos(\phi_{s,z}^b) \cos(\phi_{s,y}^b) \\
&\quad -p_y^s \sin(\psi_{b,z}^n) \cos(\psi_{b,y}^n) \cos(\phi_{s,z}^b) \cos(\phi_{s,x}^b) \\
&\quad -p_y^s \sin(\psi_{b,z}^n) \cos(\psi_{b,y}^n) \sin(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \sin(\phi_{s,x}^b) \\
&\quad -p_y^s \cos(\psi_{b,z}^n) \cos(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \cos(\phi_{s,x}^b) \\
&\quad +p_y^s \cos(\psi_{b,z}^n) \cos(\psi_{b,x}^n) \cos(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \sin(\phi_{s,x}^b) \\
&\quad -p_y^s \sin(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \cos(\phi_{s,x}^b) \\
&\quad +p_y^s \sin(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \cos(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \sin(\phi_{s,x}^b) \\
&\quad +p_z^s \sin(\psi_{b,z}^n) \cos(\psi_{b,y}^n) \cos(\phi_{s,z}^b) \sin(\phi_{s,x}^b) \\
&\quad -p_z^s \sin(\psi_{b,z}^n) \cos(\psi_{b,y}^n) \sin(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \cos(\phi_{s,x}^b) \\
&\quad +p_z^s \cos(\psi_{b,z}^n) \cos(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \sin(\phi_{s,x}^b) \\
&\quad +p_z^s \cos(\psi_{b,z}^n) \cos(\psi_{b,x}^n) \cos(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \cos(\phi_{s,x}^b) \\
&\quad +p_z^s \sin(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \sin(\phi_{s,x}^b) \\
&\quad +p_z^s \sin(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \cos(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \cos(\phi_{s,x}^b) \qquad \text{(A.11)}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial p_z^n}{\partial \phi_{s,z}^b} &= f(\mathbf{p}^s, \mathbf{\Phi}_s^b, \mathbf{\Psi}_b^n) \\
&= +p_x^s \sin(\psi_{b,y}^n) \sin(\phi_{s,z}^b) \cos(\phi_{s,y}^b) \\
&\quad +p_x^s \cos(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \cos(\phi_{s,z}^b) \cos(\phi_{s,y}^b) \\
&\quad +p_y^s \sin(\psi_{b,y}^n) \cos(\phi_{s,z}^b) \cos(\phi_{s,x}^b) \\
&\quad +p_y^s \sin(\psi_{b,y}^n) \sin(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \sin(\phi_{s,x}^b) \\
&\quad -p_y^s \cos(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \cos(\phi_{s,x}^b) \\
&\quad +p_y^s \cos(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \cos(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \sin(\phi_{s,x}^b) \\
&\quad -p_z^s \sin(\psi_{b,y}^n) \cos(\phi_{s,z}^b) \sin(\phi_{s,x}^b) \\
&\quad +p_z^s \sin(\psi_{b,y}^n) \sin(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \cos(\phi_{s,x}^b) \\
&\quad +p_z^s \cos(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \sin(\phi_{s,x}^b) \\
&\quad +p_z^s \cos(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \cos(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \cos(\phi_{s,x}^b) \qquad \text{(A.12)}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial p_x^n}{\partial p_x^s} &= f(\mathbf{\Phi}_s^b, \mathbf{\Psi}_b^n) \\
&= +\cos(\psi_{b,z}^n) \cos(\psi_{b,y}^n) \cos(\phi_{s,z}^b) \cos(\phi_{s,y}^b) \\
&\quad -\sin(\psi_{b,z}^n) \cos(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \cos(\phi_{s,y}^b) \\
&\quad +\cos(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \cos(\phi_{s,y}^b) \\
&\quad -\sin(\psi_{b,x}^n) \sin(\phi_{s,y}^b) \sin(\psi_{b,z}^n) \\
&\quad -\cos(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \cos(\psi_{b,x}^n) \sin(\phi_{s,y}^b) \qquad \text{(A.13)}
\end{aligned}$$

$$
\begin{aligned}
\frac{\partial p_y^n}{\partial p_x^s} &= f(\mathbf{\Phi}_s^b, \mathbf{\Psi}_b^n) \\
&= +\sin(\psi_{b,z}^n)\cos(\psi_{b,y}^n)\cos(\phi_{s,z}^b)\cos(\phi_{s,y}^b) \\
&\quad +\cos(\psi_{b,z}^n)\cos(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\cos(\phi_{s,y}^b) \\
&\quad +\sin(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\sin(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\cos(\phi_{s,y}^b) \\
&\quad +\sin(\psi_{b,x}^n)\sin(\phi_{s,y}^b)\cos(\psi_{b,z}^n) \\
&\quad -\sin(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\cos(\psi_{b,x}^n)\sin(\phi_{s,y}^b) \qquad \text{(A.14)}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial p_z^n}{\partial p_x^s} &= f(\mathbf{\Phi}_s^b, \mathbf{\Psi}_b^n) \\
&= -\sin(\psi_{b,y}^n)\cos(\phi_{s,z}^b)\cos(\phi_{s,y}^b) \\
&\quad +\cos(\psi_{b,y}^n)\sin(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\cos(\phi_{s,y}^b) \\
&\quad -\cos(\psi_{b,y}^n)\cos(\psi_{b,x}^n)\sin(\phi_{s,y}^b) \qquad \text{(A.15)}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial p_x^n}{\partial p_y^s} &= f(\mathbf{\Phi}_s^b, \mathbf{\Psi}_b^n) \\
&= -\cos(\psi_{b,z}^n)\cos(\psi_{b,y}^n)\sin(\phi_{s,z}^b)\cos(\phi_{s,x}^b) \\
&\quad +\cos(\psi_{b,z}^n)\cos(\psi_{b,y}^n)\cos(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\sin(\phi_{s,x}^b) \\
&\quad -\sin(\psi_{b,z}^n)\cos(\psi_{b,x}^n)\cos(\phi_{s,z}^b)\cos(\phi_{s,x}^b) \\
&\quad -\sin(\psi_{b,z}^n)\cos(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\sin(\phi_{s,x}^b) \\
&\quad +\cos(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\sin(\psi_{b,x}^n)\cos(\phi_{s,z}^b)\cos(\phi_{s,x}^b) \\
&\quad +\cos(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\sin(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\sin(\phi_{s,x}^b) \\
&\quad +\sin(\psi_{b,z}^n)\sin(\psi_{b,x}^n)\cos(\phi_{s,y}^b)\sin(\phi_{s,x}^b) \\
&\quad +\cos(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\cos(\psi_{b,x}^n)\cos(\phi_{s,y}^b)\sin(\phi_{s,x}^b) \\
&\qquad \text{(A.16)}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial p_y^n}{\partial p_y^s} &= f(\boldsymbol{\Phi}_s^b, \boldsymbol{\Psi}_b^n) \\
&= -\sin(\psi_{b,z}^n)\cos(\psi_{b,y}^n)\sin(\phi_{s,z}^b)\cos(\phi_{s,x}^b) \\
&\quad +\sin(\psi_{b,z}^n)\cos(\psi_{b,y}^n)\cos(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\sin(\phi_{s,x}^b) \\
&\quad +\cos(\psi_{b,z}^n)\cos(\psi_{b,x}^n)\cos(\phi_{s,z}^b)\cos(\phi_{s,x}^b) \\
&\quad +\cos(\psi_{b,z}^n)\cos(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\sin(\phi_{s,x}^b) \\
&\quad +\sin(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\sin(\psi_{b,x}^n)\cos(\phi_{s,z}^b)\cos(\phi_{s,x}^b) \\
&\quad +\sin(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\sin(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\sin(\phi_{s,x}^b) \\
&\quad -\cos(\psi_{b,z}^n)\sin(\psi_{b,x}^n)\cos(\phi_{s,y}^b)\sin(\phi_{s,x}^b) \\
&\quad +\sin(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\cos(\psi_{b,x}^n)\cos(\phi_{s,y}^b)\sin(\phi_{s,x}^b) \qquad \text{(A.17)}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial p_z^n}{\partial p_y^s} &= f(\boldsymbol{\Phi}_s^b, \boldsymbol{\Psi}_b^n) \\
&= +\sin(\psi_{b,y}^n)\sin(\phi_{s,z}^b)\cos(\phi_{s,x}^b) \\
&\quad -\sin(\psi_{b,y}^n)\cos(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\sin(\phi_{s,x}^b) \\
&\quad +\cos(\psi_{b,y}^n)\sin(\psi_{b,x}^n)\cos(\phi_{s,z}^b)\cos(\phi_{s,x}^b) \\
&\quad +\cos(\psi_{b,y}^n)\sin(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\sin(\phi_{s,x}^b) \\
&\quad +\cos(\psi_{b,y}^n)\cos(\psi_{b,x}^n)\cos(\phi_{s,y}^b)\sin(\phi_{s,x}^b) \qquad \text{(A.18)}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial p_x^n}{\partial p_z^s} &= f(\mathbf{\Phi}_s^b, \mathbf{\Psi}_b^n) \\
&= +\cos(\psi_{b,z}^n)\cos(\psi_{b,y}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,x}^b) \\
&\quad +\cos(\psi_{b,z}^n)\cos(\psi_{b,y}^n)\cos(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\cos(\phi_{s,x}^b) \\
&\quad +\sin(\psi_{b,z}^n)\cos(\psi_{b,x}^n)\cos(\phi_{s,z}^b)\sin(\phi_{s,x}^b) \\
&\quad -\sin(\psi_{b,z}^n)\cos(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\cos(\phi_{s,x}^b) \\
&\quad -\cos(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\sin(\psi_{b,x}^n)\cos(\phi_{s,z}^b)\sin(\phi_{s,x}^b) \\
&\quad +\cos(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\sin(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\cos(\phi_{s,x}^b) \\
&\quad +\sin(\psi_{b,z}^n)\sin(\psi_{b,x}^n)\cos(\phi_{s,y}^b)\cos(\phi_{s,x}^b) \\
&\quad +\cos(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\cos(\psi_{b,x}^n)\cos(\phi_{s,y}^b)\cos(\phi_{s,x}^b) \qquad \text{(A.19)}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial p_y^n}{\partial p_z^s} &= f(\mathbf{\Phi}_s^b, \mathbf{\Psi}_b^n) \\
&= +\sin(\psi_{b,z}^n)\cos(\psi_{b,y}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,x}^b) \\
&\quad +\sin(\psi_{b,z}^n)\cos(\psi_{b,y}^n)\cos(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\cos(\phi_{s,x}^b) \\
&\quad -\cos(\psi_{b,z}^n)\cos(\psi_{b,x}^n)\cos(\phi_{s,z}^b)\sin(\phi_{s,x}^b) \\
&\quad +\cos(\psi_{b,z}^n)\cos(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\cos(\phi_{s,x}^b) \\
&\quad -\sin(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\sin(\psi_{b,x}^n)\cos(\phi_{s,z}^b)\sin(\phi_{s,x}^b) \\
&\quad +\sin(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\sin(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\cos(\phi_{s,x}^b) \\
&\quad -\cos(\psi_{b,z}^n)\sin(\psi_{b,x}^n)\cos(\phi_{s,y}^b)\cos(\phi_{s,x}^b) \\
&\quad +\sin(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\cos(\psi_{b,x}^n)\cos(\phi_{s,y}^b)\cos(\phi_{s,x}^b) \qquad \text{(A.20)}
\end{aligned}
$$

$$\begin{aligned}
\frac{\partial p_z^n}{\partial p_z^s} &= f(\boldsymbol{\Phi}_s^b, \boldsymbol{\Psi}_b^n) \\
&= -\sin(\psi_{b,y}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,x}^b) \\
&\quad -\sin(\psi_{b,y}^n)\cos(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\cos(\phi_{s,x}^b) \\
&\quad -\cos(\psi_{b,y}^n)\sin(\psi_{b,x}^n)\cos(\phi_{s,z}^b)\sin(\phi_{s,x}^b) \\
&\quad +\cos(\psi_{b,y}^n)\sin(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\cos(\phi_{s,x}^b) \\
&\quad +\cos(\psi_{b,y}^n)\cos(\psi_{b,x}^n)\cos(\phi_{s,y}^b)\cos(\phi_{s,x}^b) \qquad (A.21)
\end{aligned}$$

$$\frac{\partial \mathbf{p}^n}{\partial \mathbf{p}_b^n} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (A.22)$$

$$
\begin{aligned}
\frac{\partial p_x^n}{\partial \psi_{b,x}^n} &= f(\mathbf{p}^s, \mathbf{r}_s^b, \mathbf{\Phi}_s^b, \mathbf{\Psi}_b^n) \\
&= +p_x^s \sin(\psi_{b,z}^n) \sin(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \cos(\phi_{s,y}^b) \\
&\quad +p_y^s \sin(\psi_{b,z}^n) \sin(\psi_{b,x}^n) \cos(\phi_{s,z}^b) \cos(\phi_{s,x}^b) \\
&\quad +p_y^s \sin(\psi_{b,z}^n) \sin(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \sin(\phi_{s,x}^b) \\
&\quad -p_z^s \sin(\psi_{b,z}^n) \sin(\psi_{b,x}^n) \cos(\phi_{s,z}^b) \sin(\phi_{s,x}^b) \\
&\quad +p_z^s \sin(\psi_{b,z}^n) \sin(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \cos(\phi_{s,x}^b) \\
&\quad +r_{s,y}^b \sin(\psi_{b,z}^n) \sin(\psi_{b,x}^n) \\
&\quad +p_x^s \cos(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \cos(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \cos(\phi_{s,y}^b) \\
&\quad +p_y^s \cos(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \cos(\psi_{b,x}^n) \cos(\phi_{s,z}^b) \cos(\phi_{s,x}^b) \\
&\quad +p_y^s \cos(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \cos(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \sin(\phi_{s,x}^b) \\
&\quad -p_z^s \cos(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \cos(\psi_{b,x}^n) \cos(\phi_{s,z}^b) \sin(\phi_{s,x}^b) \\
&\quad +p_z^s \cos(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \cos(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \cos(\phi_{s,x}^b) \\
&\quad +r_{s,y}^b \cos(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \cos(\psi_{b,x}^n) \\
&\quad -p_x^s \sin(\psi_{b,z}^n) \cos(\psi_{b,x}^n) \sin(\phi_{s,y}^b) \\
&\quad +p_y^s \sin(\psi_{b,z}^n) \cos(\psi_{b,x}^n) \cos(\phi_{s,y}^b) \sin(\phi_{s,x}^b) \\
&\quad +p_z^s \sin(\psi_{b,z}^n) \cos(\psi_{b,x}^n) \cos(\phi_{s,y}^b) \cos(\phi_{s,x}^b) \\
&\quad +r_{s,z}^b \sin(\psi_{b,z}^n) \cos(\psi_{b,x}^n) \\
&\quad +p_x^s \cos(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \sin(\phi_{s,y}^b) \\
&\quad -p_y^s \cos(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \cos(\phi_{s,y}^b) \sin(\phi_{s,x}^b) \\
&\quad -p_z^s \cos(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \cos(\phi_{s,y}^b) \cos(\phi_{s,x}^b) \\
&\quad -r_{s,z}^b \cos(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \qquad \text{(A.23)}
\end{aligned}
$$

$$\begin{aligned}
\frac{\partial p_y^n}{\partial \psi_{b,x}^n} &= f(\mathbf{p}^s, \mathbf{r}_s^b, \mathbf{\Phi}_s^b, \mathbf{\Psi}_b^n) \\
&= -p_x^s \cos(\psi_{b,z}^n) \sin(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \cos(\phi_{s,y}^b) \\
&\quad -p_y^s \cos(\psi_{b,z}^n) \sin(\psi_{b,x}^n) \cos(\phi_{s,z}^b) \cos(\phi_{s,x}^b) \\
&\quad -p_y^s \cos(\psi_{b,z}^n) \sin(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \sin(\phi_{s,x}^b) \\
&\quad +p_z^s \cos(\psi_{b,z}^n) \sin(\psi_{b,x}^n) \cos(\phi_{s,z}^b) \sin(\phi_{s,x}^b) \\
&\quad -p_z^s \cos(\psi_{b,z}^n) \sin(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \cos(\phi_{s,x}^b) \\
&\quad -r_{s,y}^b \cos(\psi_{b,z}^n) \sin(\psi_{b,x}^n) \\
&\quad +p_x^s \sin(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \cos(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \cos(\phi_{s,y}^b) \\
&\quad +p_y^s \sin(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \cos(\psi_{b,x}^n) \cos(\phi_{s,z}^b) \cos(\phi_{s,x}^b) \\
&\quad +p_y^s \sin(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \cos(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \sin(\phi_{s,x}^b) \\
&\quad -p_z^s \sin(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \cos(\psi_{b,x}^n) \cos(\phi_{s,z}^b) \sin(\phi_{s,x}^b) \\
&\quad +p_z^s \sin(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \cos(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \cos(\phi_{s,x}^b) \\
&\quad +r_{s,y}^b \sin(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \cos(\psi_{b,x}^n) \\
&\quad +p_x^s \cos(\psi_{b,z}^n) \cos(\psi_{b,x}^n) \sin(\phi_{s,y}^b) \\
&\quad -p_y^s \cos(\psi_{b,z}^n) \cos(\psi_{b,x}^n) \cos(\phi_{s,y}^b) \sin(\phi_{s,x}^b) \\
&\quad -p_z^s \cos(\psi_{b,z}^n) \cos(\psi_{b,x}^n) \cos(\phi_{s,y}^b) \cos(\phi_{s,x}^b) \\
&\quad -r_{s,z}^b \cos(\psi_{b,z}^n) \cos(\psi_{b,x}^n) \\
&\quad +p_x^s \sin(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \sin(\phi_{s,y}^b) \\
&\quad -p_y^s \sin(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \cos(\phi_{s,y}^b) \sin(\phi_{s,x}^b) \\
&\quad -p_z^s \sin(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \cos(\phi_{s,y}^b) \cos(\phi_{s,x}^b) \\
&\quad -r_{s,z}^b \sin(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \qquad \text{(A.24)}
\end{aligned}$$

$$
\begin{aligned}
\frac{\partial p_z^n}{\partial \psi_{b,x}^n} &= f(\mathbf{p}^s, \mathbf{r}_s^b, \mathbf{\Phi}_s^b, \mathbf{\Psi}_b^n) \\
&= +p_x^s \cos(\psi_{b,y}^n)\cos(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\cos(\phi_{s,y}^b) \\
&\quad +p_y^s \cos(\psi_{b,y}^n)\cos(\psi_{b,x}^n)\cos(\phi_{s,z}^b)\cos(\phi_{s,x}^b) \\
&\quad +p_y^s \cos(\psi_{b,y}^n)\cos(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\sin(\phi_{s,x}^b) \\
&\quad -p_z^s \cos(\psi_{b,y}^n)\cos(\psi_{b,x}^n)\cos(\phi_{s,z}^b)\sin(\phi_{s,x}^b) \\
&\quad +p_z^s \cos(\psi_{b,y}^n)\cos(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\cos(\phi_{s,x}^b) \\
&\quad +r_{s,y}^b \cos(\psi_{b,y}^n)\cos(\psi_{b,x}^n) \\
&\quad +p_x^s \cos(\psi_{b,y}^n)\sin(\psi_{b,x}^n)\sin(\phi_{s,y}^b) \\
&\quad -p_y^s \cos(\psi_{b,y}^n)\sin(\psi_{b,x}^n)\cos(\phi_{s,y}^b)\sin(\phi_{s,x}^b) \\
&\quad -p_z^s \cos(\psi_{b,y}^n)\sin(\psi_{b,x}^n)\cos(\phi_{s,y}^b)\cos(\phi_{s,x}^b) \\
&\quad -r_{s,z}^b \cos(\psi_{b,y}^n)\sin(\psi_{b,x}^n) \qquad \text{(A.25)}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial p_x^n}{\partial \psi_{b,y}^n} &= f(\mathbf{p}^s, \mathbf{r}_s^b, \mathbf{\Phi}_s^b, \mathbf{\Psi}_b^n) \\
&= -p_x^s \cos(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \cos(\phi_{s,z}^b) \cos(\phi_{s,y}^b) \\
&\quad +p_y^s \cos(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \sin(\phi_{s,z}^b) \cos(\phi_{s,x}^b) \\
&\quad -p_y^s \cos(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \cos(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \sin(\phi_{s,x}^b) \\
&\quad -p_z^s \cos(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \sin(\phi_{s,z}^b) \sin(\phi_{s,x}^b) \\
&\quad -p_z^s \cos(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \cos(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \cos(\phi_{s,x}^b) \\
&\quad -r_{s,x}^b \cos(\psi_{b,z}^n) \sin(\psi_{b,y}^n) \\
&\quad +p_x^s \cos(\psi_{b,z}^n) \cos(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \cos(\phi_{s,y}^b) \\
&\quad +p_y^s \cos(\psi_{b,z}^n) \cos(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \cos(\phi_{s,z}^b) \cos(\phi_{s,x}^b) \\
&\quad +p_y^s \cos(\psi_{b,z}^n) \cos(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \sin(\phi_{s,x}^b) \\
&\quad -p_z^s \cos(\psi_{b,z}^n) \cos(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \cos(\phi_{s,z}^b) \sin(\phi_{s,x}^b) \\
&\quad +p_z^s \cos(\psi_{b,z}^n) \cos(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \sin(\phi_{s,z}^b) \sin(\phi_{s,y}^b) \cos(\phi_{s,x}^b) \\
&\quad +r_{s,y}^b \cos(\psi_{b,z}^n) \cos(\psi_{b,y}^n) \sin(\psi_{b,x}^n) \\
&\quad -p_x^s \cos(\psi_{b,z}^n) \cos(\psi_{b,y}^n) \cos(\psi_{b,x}^n) \sin(\phi_{s,y}^b) \\
&\quad +p_y^s \cos(\psi_{b,z}^n) \cos(\psi_{b,y}^n) \cos(\psi_{b,x}^n) \cos(\phi_{s,y}^b) \sin(\phi_{s,x}^b) \\
&\quad +p_z^s \cos(\psi_{b,z}^n) \cos(\psi_{b,y}^n) \cos(\psi_{b,x}^n) \cos(\phi_{s,y}^b) \cos(\phi_{s,x}^b) \\
&\quad +r_{s,z}^b \cos(\psi_{b,z}^n) \cos(\psi_{b,y}^n) \cos(\psi_{b,x}^n) \qquad \text{(A.26)}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial p_y^n}{\partial \psi_{b,y}^n} &= f(\mathbf{p}^s, \mathbf{r}_s^b, \mathbf{\Phi}_s^b, \mathbf{\Psi}_b^n) \\
&= -p_x^s \sin(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\cos(\phi_{s,z}^b)\cos(\phi_{s,y}^b) \\
&\quad +p_y^s \sin(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\sin(\phi_{s,z}^b)\cos(\phi_{s,x}^b) \\
&\quad -p_y^s \sin(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\cos(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\sin(\phi_{s,x}^b) \\
&\quad -p_z^s \sin(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,x}^b) \\
&\quad -p_z^s \sin(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\cos(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\cos(\phi_{s,x}^b) \\
&\quad -r_{s,x}^b \sin(\psi_{b,z}^n)\sin(\psi_{b,y}^n) \\
&\quad +p_x^s \sin(\psi_{b,z}^n)\cos(\psi_{b,y}^n)\sin(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\cos(\phi_{s,y}^b) \\
&\quad +p_y^s \sin(\psi_{b,z}^n)\cos(\psi_{b,y}^n)\sin(\psi_{b,x}^n)\cos(\phi_{s,z}^b)\cos(\phi_{s,x}^b) \\
&\quad +p_y^s \sin(\psi_{b,z}^n)\cos(\psi_{b,y}^n)\sin(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\sin(\phi_{s,x}^b) \\
&\quad -p_z^s \sin(\psi_{b,z}^n)\cos(\psi_{b,y}^n)\sin(\psi_{b,x}^n)\cos(\phi_{s,z}^b)\sin(\phi_{s,x}^b) \\
&\quad +p_z^s \sin(\psi_{b,z}^n)\cos(\psi_{b,y}^n)\sin(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\cos(\phi_{s,x}^b) \\
&\quad +r_{s,y}^b \sin(\psi_{b,z}^n)\cos(\psi_{b,y}^n)\sin(\psi_{b,x}^n) \\
&\quad -p_x^s \sin(\psi_{b,z}^n)\cos(\psi_{b,y}^n)\cos(\psi_{b,x}^n)\sin(\phi_{s,y}^b) \\
&\quad +p_y^s \sin(\psi_{b,z}^n)\cos(\psi_{b,y}^n)\cos(\psi_{b,x}^n)\cos(\phi_{s,y}^b)\sin(\phi_{s,x}^b) \\
&\quad +p_z^s \sin(\psi_{b,z}^n)\cos(\psi_{b,y}^n)\cos(\psi_{b,x}^n)\cos(\phi_{s,y}^b)\cos(\phi_{s,x}^b) \\
&\quad +r_{s,z}^b \sin(\psi_{b,z}^n)\cos(\psi_{b,y}^n)\cos(\psi_{b,x}^n) \qquad \text{(A.27)}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial p^n_z}{\partial \psi^n_{b,y}} &= f(\mathbf{p}^s, \mathbf{r}^b_s, \mathbf{\Phi}^b_s, \mathbf{\Psi}^n_b) \\
&= -p^s_x \cos(\psi^n_{b,y}) \cos(\phi^b_{s,z}) \cos(\phi^b_{s,y}) \\
&\quad +p^s_y \cos(\psi^n_{b,y}) \sin(\phi^b_{s,z}) \cos(\phi^b_{s,x}) \\
&\quad -p^s_y \cos(\psi^n_{b,y}) \cos(\phi^b_{s,z}) \sin(\phi^b_{s,y}) \sin(\phi^b_{s,x}) \\
&\quad -p^s_z \cos(\psi^n_{b,y}) \sin(\phi^b_{s,z}) \sin(\phi^b_{s,x}) \\
&\quad -p^s_z \cos(\psi^n_{b,y}) \cos(\phi^b_{s,z}) \sin(\phi^b_{s,y}) \cos(\phi^b_{s,x}) \\
&\quad -r^b_{s,x} \cos(\psi^n_{b,y}) \\
&\quad -p^s_x \sin(\psi^n_{b,y}) \sin(\psi^n_{b,x}) \sin(\phi^b_{s,z}) \cos(\phi^b_{s,y}) \\
&\quad -p^s_y \sin(\psi^n_{b,y}) \sin(\psi^n_{b,x}) \cos(\phi^b_{s,z}) \cos(\phi^b_{s,x}) \\
&\quad -p^s_y \sin(\psi^n_{b,y}) \sin(\psi^n_{b,x}) \sin(\phi^b_{s,z}) \sin(\phi^b_{s,y}) \sin(\phi^b_{s,x}) \\
&\quad +p^s_z \sin(\psi^n_{b,y}) \sin(\psi^n_{b,x}) \cos(\phi^b_{s,z}) \sin(\phi^b_{s,x}) \\
&\quad -p^s_z \sin(\psi^n_{b,y}) \sin(\psi^n_{b,x}) \sin(\phi^b_{s,z}) \sin(\phi^b_{s,y}) \cos(\phi^b_{s,x}) \\
&\quad -r^b_{s,y} \sin(\psi^n_{b,y}) \sin(\psi^n_{b,x}) \\
&\quad +p^s_x \sin(\psi^n_{b,y}) \cos(\psi^n_{b,x}) \sin(\phi^b_{s,y}) \\
&\quad -p^s_y \sin(\psi^n_{b,y}) \cos(\psi^n_{b,x}) \cos(\phi^b_{s,y}) \sin(\phi^b_{s,x}) \\
&\quad -p^s_z \sin(\psi^n_{b,y}) \cos(\psi^n_{b,x}) \cos(\phi^b_{s,y}) \cos(\phi^b_{s,x}) \\
&\quad -r^b_{s,z} \sin(\psi^n_{b,y}) \cos(\psi^n_{b,x})
\end{aligned} \tag{A.28}
$$

$$
\begin{aligned}
\frac{\partial p_x^n}{\partial \psi_{b,z}^n} &= f(\mathbf{p}^s, \mathbf{r}_s^b, \boldsymbol{\Phi}_s^b, \boldsymbol{\Psi}_b^n) \\
&= -r_{s,x}^b \sin(\psi_{b,z}^n)\cos(\psi_{b,y}^n) \\
&\quad -r_{s,y}^b \sin(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\sin(\psi_{b,x}^n) \\
&\quad -p_x^s \cos(\psi_{b,z}^n)\sin(\psi_{b,x}^n)\sin(\phi_{s,y}^b) \\
&\quad -r_{s,z}^b \sin(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\cos(\psi_{b,x}^n) \\
&\quad -p_x^s \sin(\psi_{b,z}^n)\cos(\psi_{b,y}^n)\cos(\phi_{s,z}^b)\cos(\phi_{s,y}^b) \\
&\quad +p_y^s \sin(\psi_{b,z}^n)\cos(\psi_{b,y}^n)\sin(\phi_{s,z}^b)\cos(\phi_{s,x}^b) \\
&\quad -p_y^s \sin(\psi_{b,z}^n)\cos(\psi_{b,y}^n)\cos(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\sin(\phi_{s,x}^b) \\
&\quad -p_z^s \sin(\psi_{b,z}^n)\cos(\psi_{b,y}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,x}^b) \\
&\quad -p_z^s \sin(\psi_{b,z}^n)\cos(\psi_{b,y}^n)\cos(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\cos(\phi_{s,x}^b) \\
&\quad -p_x^s \cos(\psi_{b,z}^n)\cos(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\cos(\phi_{s,y}^b) \\
&\quad -p_y^s \cos(\psi_{b,z}^n)\cos(\psi_{b,x}^n)\cos(\phi_{s,z}^b)\cos(\phi_{s,x}^b) \\
&\quad -p_y^s \cos(\psi_{b,z}^n)\cos(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\sin(\phi_{s,x}^b) \\
&\quad +p_z^s \cos(\psi_{b,z}^n)\cos(\psi_{b,x}^n)\cos(\phi_{s,z}^b)\sin(\phi_{s,x}^b) \\
&\quad -p_z^s \cos(\psi_{b,z}^n)\cos(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\cos(\phi_{s,x}^b) \\
&\quad -r_{s,y}^b \cos(\psi_{b,z}^n)\cos(\psi_{b,x}^n) \\
&\quad +r_{s,z}^b \cos(\psi_{b,z}^n)\sin(\psi_{b,x}^n) \\
&\quad -p_x^s \sin(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\sin(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\cos(\phi_{s,y}^b) \\
&\quad -p_y^s \sin(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\sin(\psi_{b,x}^n)\cos(\phi_{s,z}^b)\cos(\phi_{s,x}^b) \\
&\quad -p_y^s \sin(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\sin(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\sin(\phi_{s,x}^b) \\
&\quad +p_z^s \sin(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\sin(\psi_{b,x}^n)\cos(\phi_{s,z}^b)\sin(\phi_{s,x}^b) \\
&\quad -p_z^s \sin(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\sin(\psi_{b,x}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\cos(\phi_{s,x}^b) \\
&\quad +p_y^s \cos(\psi_{b,z}^n)\sin(\psi_{b,x}^n)\cos(\phi_{s,y}^b)\sin(\phi_{s,x}^b) \\
&\quad +p_z^s \cos(\psi_{b,z}^n)\sin(\psi_{b,x}^n)\cos(\phi_{s,y}^b)\cos(\phi_{s,x}^b) \\
&\quad +p_x^s \sin(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\cos(\psi_{b,x}^n)\sin(\phi_{s,y}^b) \\
&\quad -p_y^s \sin(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\cos(\psi_{b,x}^n)\cos(\phi_{s,y}^b)\sin(\phi_{s,x}^b) \\
&\quad -p_z^s \sin(\psi_{b,z}^n)\sin(\psi_{b,y}^n)\cos(\psi_{b,x}^n)\cos(\phi_{s,y}^b)\cos(\phi_{s,x}^b)
\end{aligned} \tag{A.29}
$$

$$
\begin{aligned}
\frac{\partial p^n_y}{\partial \psi^n_{b,z}} &= f(\mathbf{p}^s, \mathbf{r}^b_s, \mathbf{\Phi}^b_s, \mathbf{\Psi}^n_b) \\
&= -r^b_{s,y}\sin(\psi^n_{b,z})\cos(\psi^n_{b,x}) \\
&\quad +p^s_x\cos(\psi^n_{b,z})\cos(\psi^n_{b,y})\cos(\phi^b_{s,z})\cos(\phi^b_{s,y}) \\
&\quad +r^b_{s,x}\cos(\psi^n_{b,z})\cos(\psi^n_{b,y}) \\
&\quad -p^s_y\cos(\psi^n_{b,z})\cos(\psi^n_{b,y})\sin(\phi^b_{s,z})\cos(\phi^b_{s,x}) \\
&\quad +p^s_y\cos(\psi^n_{b,z})\cos(\psi^n_{b,y})\cos(\phi^b_{s,z})\sin(\phi^b_{s,y})\sin(\phi^b_{s,x}) \\
&\quad +p^s_z\cos(\psi^n_{b,z})\cos(\psi^n_{b,y})\sin(\phi^b_{s,z})\sin(\phi^b_{s,x}) \\
&\quad +p^s_z\cos(\psi^n_{b,z})\cos(\psi^n_{b,y})\cos(\phi^b_{s,z})\sin(\phi^b_{s,y})\cos(\phi^b_{s,x}) \\
&\quad +r^b_{s,y}\cos(\psi^n_{b,z})\sin(\psi^n_{b,y})\sin(\psi^n_{b,x}) \\
&\quad -p^s_x\sin(\psi^n_{b,z})\cos(\psi^n_{b,x})\sin(\phi^b_{s,z})\cos(\phi^b_{s,y}) \\
&\quad -p^s_y\sin(\psi^n_{b,z})\cos(\psi^n_{b,x})\cos(\phi^b_{s,z})\cos(\phi^b_{s,x}) \\
&\quad -p^s_y\sin(\psi^n_{b,z})\cos(\psi^n_{b,x})\sin(\phi^b_{s,z})\sin(\phi^b_{s,y})\sin(\phi^b_{s,x}) \\
&\quad +p^s_z\sin(\psi^n_{b,z})\cos(\psi^n_{b,x})\cos(\phi^b_{s,z})\sin(\phi^b_{s,x}) \\
&\quad -p^s_z\sin(\psi^n_{b,z})\cos(\psi^n_{b,x})\sin(\phi^b_{s,z})\sin(\phi^b_{s,y})\cos(\phi^b_{s,x}) \\
&\quad +p^s_x\cos(\psi^n_{b,z})\sin(\psi^n_{b,y})\sin(\psi^n_{b,x})\sin(\phi^b_{s,z})\cos(\phi^b_{s,y}) \\
&\quad +p^s_y\cos(\psi^n_{b,z})\sin(\psi^n_{b,y})\sin(\psi^n_{b,x})\cos(\phi^b_{s,z})\cos(\phi^b_{s,x}) \\
&\quad +p^s_y\cos(\psi^n_{b,z})\sin(\psi^n_{b,y})\sin(\psi^n_{b,x})\sin(\phi^b_{s,z})\sin(\phi^b_{s,y})\sin(\phi^b_{s,x}) \\
&\quad -p^s_z\cos(\psi^n_{b,z})\sin(\psi^n_{b,y})\sin(\psi^n_{b,x})\cos(\phi^b_{s,z})\sin(\phi^b_{s,x}) \\
&\quad +p^s_z\cos(\psi^n_{b,z})\sin(\psi^n_{b,y})\sin(\psi^n_{b,x})\sin(\phi^b_{s,z})\sin(\phi^b_{s,y})\cos(\phi^b_{s,x}) \\
&\quad -p^s_x\sin(\psi^n_{b,z})\sin(\psi^n_{b,x})\sin(\phi^b_{s,y}) \\
&\quad +r^b_{s,z}\cos(\psi^n_{b,z})\sin(\psi^n_{b,y})\cos(\psi^n_{b,x}) \\
&\quad +r^b_{s,z}\sin(\psi^n_{b,z})\sin(\psi^n_{b,x}) \\
&\quad +p^s_y\sin(\psi^n_{b,z})\sin(\psi^n_{b,x})\cos(\phi^b_{s,y})\sin(\phi^b_{s,x}) \\
&\quad +p^s_z\sin(\psi^n_{b,z})\sin(\psi^n_{b,x})\cos(\phi^b_{s,y})\cos(\phi^b_{s,x}) \\
&\quad -p^s_x\cos(\psi^n_{b,z})\sin(\psi^n_{b,y})\cos(\psi^n_{b,x})\sin(\phi^b_{s,y}) \\
&\quad +p^s_y\cos(\psi^n_{b,z})\sin(\psi^n_{b,y})\cos(\psi^n_{b,x})\cos(\phi^b_{s,y})\sin(\phi^b_{s,x}) \\
&\quad +p^s_z\cos(\psi^n_{b,z})\sin(\psi^n_{b,y})\cos(\psi^n_{b,x})\cos(\phi^b_{s,y})\cos(\phi^b_{s,x})
\end{aligned} \tag{A.30}
$$

$$\frac{\partial p_z^n}{\partial \psi_{b,z}^n} = 0 \tag{A.31}$$

$$\begin{aligned}
\frac{\partial p_x^n}{\partial t_e} &= f(\mathbf{p}^s, \mathbf{r}_s^b, \mathbf{\Phi}_s^b, \mathbf{\Psi}_b^n, \dot{\mathbf{\Psi}}_b^n, \dot{\mathbf{p}}_b^n) \\
&= \frac{\partial p_x^n}{\partial p_{b,x}^n}\frac{\partial p_{b,x}^n}{\partial t_e} + \frac{\partial p_x^n}{\partial p_{b,y}^n}\frac{\partial p_{b,y}^n}{\partial t_e} + \frac{\partial p_x^n}{\partial p_{b,z}^n}\frac{\partial p_{b,z}^n}{\partial t_e} \\
&\quad + \frac{\partial p_x^n}{\partial \psi_{b,x}^n}\frac{\partial \psi_{b,x}^n}{\partial t_e} + \frac{\partial p_x^n}{\partial \psi_{b,y}^n}\frac{\partial \psi_{b,y}^n}{\partial t_e} + \frac{\partial p_x^n}{\partial \psi_{b,z}^n}\frac{\partial \psi_{b,z}^n}{\partial t_e} \\
&= \dot{p}_{b,x}^n + \frac{\partial p_x^n}{\partial \psi_{b,x}^n}\dot{\psi}_{b,x}^n + \frac{\partial p_x^n}{\partial \psi_{b,y}^n}\dot{\psi}_{b,y}^n + \frac{\partial p_x^n}{\partial \psi_{b,z}^n}\dot{\psi}_{b,z}^n
\end{aligned} \tag{A.32}$$

$$\begin{aligned}
\frac{\partial p_y^n}{\partial t_e} &= f(\mathbf{p}^s, \mathbf{r}_s^b, \mathbf{\Phi}_s^b, \mathbf{\Psi}_b^n, \dot{\mathbf{\Psi}}_b^n, \dot{\mathbf{p}}_b^n) \\
&= \frac{\partial p_y^n}{\partial p_{b,x}^n}\frac{\partial p_{b,x}^n}{\partial t_e} + \frac{\partial p_y^n}{\partial p_{b,y}^n}\frac{\partial p_{b,y}^n}{\partial t_e} + \frac{\partial p_y^n}{\partial p_{b,z}^n}\frac{\partial p_{b,z}^n}{\partial t_e} \\
&\quad + \frac{\partial p_y^n}{\partial \psi_{b,x}^n}\frac{\partial \psi_{b,x}^n}{\partial t_e} + \frac{\partial p_y^n}{\partial \psi_{b,y}^n}\frac{\partial \psi_{b,y}^n}{\partial t_e} + \frac{\partial p_y^n}{\partial \psi_{b,z}^n}\frac{\partial \psi_{b,z}^n}{\partial t_e} \\
&= \dot{p}_{b,y}^n + \frac{\partial p_y^n}{\partial \psi_{b,x}^n}\dot{\psi}_{b,x}^n + \frac{\partial p_y^n}{\partial \psi_{b,y}^n}\dot{\psi}_{b,y}^n + \frac{\partial p_y^n}{\partial \psi_{b,z}^n}\dot{\psi}_{b,z}^n
\end{aligned} \tag{A.33}$$

$$\begin{aligned}
\frac{\partial p_z^n}{\partial t_e} &= f(\mathbf{p}^s, \mathbf{r}_s^b, \mathbf{\Phi}_s^b, \mathbf{\Psi}_b^n, \dot{\mathbf{\Psi}}_b^n, \dot{\mathbf{p}}_b^n) \\
&= \frac{\partial p_z^n}{\partial p_{b,x}^n}\frac{\partial p_{b,x}^n}{\partial t_e} + \frac{\partial p_z^n}{\partial p_{b,y}^n}\frac{\partial p_{b,y}^n}{\partial t_e} + \frac{\partial p_z^n}{\partial p_{b,z}^n}\frac{\partial p_{b,z}^n}{\partial t_e} \\
&\quad + \frac{\partial p_z^n}{\partial \psi_{b,x}^n}\frac{\partial \psi_{b,x}^n}{\partial t_e} + \frac{\partial p_z^n}{\partial \psi_{b,y}^n}\frac{\partial \psi_{b,y}^n}{\partial t_e} + \frac{\partial p_z^n}{\partial \psi_{b,z}^n}\frac{\partial \psi_{b,z}^n}{\partial t_e} \\
&= \dot{p}_{b,z}^n + \frac{\partial p_z^n}{\partial \psi_{b,x}^n}\dot{\psi}_{b,x}^n + \frac{\partial p_z^n}{\partial \psi_{b,y}^n}\dot{\psi}_{b,y}^n
\end{aligned} \tag{A.34}$$

Appendix B

Laser Calibration Rotation Sensitivity Analysis

This appendix shows the analysis of the partial derivatives of $\mathbf{p}^n$ with respect to the rotation angles of the sensor position $\mathbf{\Phi}_s^b$, for the example configuration in Chapter 4. The laser sensor geometry of Equation 3.1 and the ground plane constraint of Equation 4.14 are substituted into the partial derivatives from Appendix A:

$$\begin{aligned}\frac{\partial p_x^n}{\partial \phi_{s,x}^b} &= +\cos(\psi_{b,z}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,x}^b)r\sin(\theta)\\ &+\cos(\psi_{b,z}^n)\cos(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\cos(\phi_{s,x}^b)r\sin(\theta)\\ &+\sin(\psi_{b,z}^n)\cos(\phi_{s,z}^b)\sin(\phi_{s,x}^b)r\sin(\theta)\\ &-\sin(\psi_{b,z}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\cos(\phi_{s,x}^b)r\sin(\theta)\end{aligned} \tag{B.1}$$

$$\begin{aligned}\frac{\partial p_y^n}{\partial \phi_{s,x}^b} &= +\sin(\psi_{b,z}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,x}^b)r\sin(\theta) \\ &\quad +\sin(\psi_{b,z}^n)\cos(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\cos(\phi_{s,x}^b)r\sin(\theta) \\ &\quad -\cos(\psi_{b,z}^n)\cos(\phi_{s,z}^b)\sin(\phi_{s,x}^b)r\sin(\theta) \\ &\quad +\cos(\psi_{b,z}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\cos(\phi_{s,x}^b)r\sin(\theta) \qquad \text{(B.2)}\end{aligned}$$

$$\frac{\partial p_z^n}{\partial \phi_{s,x}^b} = \cos(\phi_{s,y}^b)\cos(\phi_{s,x}^b)r\sin(\theta) \qquad \text{(B.3)}$$

$$\begin{aligned}\frac{\partial p_x^n}{\partial \phi_{s,y}^b} &= -\cos(\psi_{b,z}^n)\cos(\phi_{s,z}^b)\sin(\phi_{s,y}^b)r\cos(\theta) \\ &\quad +\cos(\psi_{b,z}^n)\cos(\phi_{s,z}^b)\cos(\phi_{s,y}^b)\sin(\phi_{s,x}^b)r\sin(\theta) \\ &\quad +\sin(\psi_{b,z}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,y}^b)r\cos(\theta) \\ &\quad -\sin(\psi_{b,z}^n)\sin(\phi_{s,z}^b)\cos(\phi_{s,y}^b)\sin(\phi_{s,x}^b)r\sin(\theta) \qquad \text{(B.4)}\end{aligned}$$

$$\begin{aligned}\frac{\partial p_y^n}{\partial \phi_{s,y}^b} &= -\sin(\psi_{b,z}^n)\cos(\phi_{s,z}^b)\sin(\phi_{s,y}^b)r\cos(\theta) \\ &\quad +\sin(\psi_{b,z}^n)\cos(\phi_{s,z}^b)\cos(\phi_{s,y}^b)\sin(\phi_{s,x}^b)r\sin(\theta) \\ &\quad -\cos(\psi_{b,z}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,y}^b)r\cos(\theta) \\ &\quad +\cos(\psi_{b,z}^n)\sin(\phi_{s,z}^b)\cos(\phi_{s,y}^b)\sin(\phi_{s,x}^b)r\sin(\theta) \qquad \text{(B.5)}\end{aligned}$$

$$\begin{aligned}\frac{\partial p_z^n}{\partial \phi_{s,y}^b} &= -\cos(\phi_{s,y}^b)r\cos(\theta) \\ &\quad -\sin(\phi_{s,y}^b)\sin(\phi_{s,x}^b)r\sin(\theta) \qquad \text{(B.6)}\end{aligned}$$

$$\begin{aligned}\frac{\partial p_x^n}{\partial \phi_{s,z}^b} \quad = \quad & -\cos(\psi_{b,z}^n)\sin(\phi_{s,z}^b)\cos(\phi_{s,y}^b)r\cos(\theta) \\ & -\cos(\psi_{b,z}^n)\cos(\phi_{s,z}^b)\cos(\phi_{s,x}^b)r\sin(\theta) \\ & -\cos(\psi_{b,z}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\sin(\phi_{s,x}^b)r\sin(\theta) \\ & -\sin(\psi_{b,z}^n)\cos(\phi_{s,z}^b)\cos(\phi_{s,y}^b)r\cos(\theta) \\ & +\sin(\psi_{b,z}^n)\sin(\phi_{s,z}^b)\cos(\phi_{s,x}^b)r\sin(\theta) \\ & -\sin(\psi_{b,z}^n)\cos(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\sin(\phi_{s,x}^b)r\sin(\theta) \qquad \text{(B.7)}\end{aligned}$$

$$\begin{aligned}\frac{\partial p_y^n}{\partial \phi_{s,z}^b} \quad = \quad & -\sin(\psi_{b,z}^n)\sin(\phi_{s,z}^b)\cos(\phi_{s,y}^b)r\cos(\theta) \\ & -\sin(\psi_{b,z}^n)\cos(\phi_{s,z}^b)\cos(\phi_{s,x}^b)r\sin(\theta) \\ & -\sin(\psi_{b,z}^n)\sin(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\sin(\phi_{s,x}^b)r\sin(\theta) \\ & +\cos(\psi_{b,z}^n)\cos(\phi_{s,z}^b)\cos(\phi_{s,y}^b)r\cos(\theta) \\ & -\cos(\psi_{b,z}^n)\sin(\phi_{s,z}^b)\cos(\phi_{s,x}^b)r\sin(\theta) \\ & +\cos(\psi_{b,z}^n)\cos(\phi_{s,z}^b)\sin(\phi_{s,y}^b)\sin(\phi_{s,x}^b)r\sin(\theta) \qquad \text{(B.8)}\end{aligned}$$

$$\frac{\partial p_z^n}{\partial \phi_{s,z}^b} = 0 \qquad \text{(B.9)}$$

The partial derivatives for all of the sensor angles $\frac{\partial \mathbf{p}^n}{\partial \mathbf{\Phi}_s^b}$ depend on $\psi_{b,z}^n$ and θ, so viewing the environment from different headings and different scan angles will provide non-constant partial derivatives as required by Equation 4.16, as long as the derivatives are not equal to zero. In this appendix, the specific configurations of $\mathbf{\Phi}_s^b$ that yield zero partial derivatives are tabulated, by the method used in Section 4.1. The derivatives of p_x^n and p_y^n are considered independently of p_z^n, to show that the only failing configuration for these is when the sensor is parallel to the ground (with zero roll and pitch angles). In this case, p_z^n is irrelevant, because no ground plane data can be gathered.

Equations B.1 and B.2 form the requirements shown in Tables B.1, B.2 and B.3.

Table B.1: Sensor configuration required for calibration of sensor roll

$\phi^b_{s,x}$	$\phi^b_{s,y}$	$\phi^b_{s,z}$
$\neq 0$	any	$\neq 0$
$\neq \pm 90$	$\neq 0$	$\neq \pm 90$
$\neq 0$	any	$\neq \pm 90$
$\neq \pm 90$	$\neq 0$	$\neq 0$

Table B.2: Condensed requirement for calibration of sensor roll

$\phi^b_{s,x}$	$\phi^b_{s,y}$	$\phi^b_{s,z}$
$\neq 0$	any	any
$\neq \pm 90$	$\neq 0$	any

Table B.3: Unique failing configuration for calibration of sensor roll

$\phi^b_{s,x}$	$\phi^b_{s,y}$	$\phi^b_{s,z}$
$= 0$	$= 0$	any

Equations B.4 and B.5 form the requirements shown in Tables B.4, B.5 and B.6, as seen in Section 4.1.

Table B.4: Sensor configuration required for calibration of sensor pitch

$\phi^b_{s,x}$	$\phi^b_{s,y}$	$\phi^b_{s,z}$
$\neq 0$	$\neq \pm 90$	$\neq 0$
$\neq 0$	$\neq \pm 90$	$\neq \pm 90$
any	$\neq 0$	$\neq \pm 90$
any	$\neq 0$	$\neq 0$

Table B.5: Condensed requirement for calibration of sensor pitch

$\phi^b_{s,x}$	$\phi^b_{s,y}$	$\phi^b_{s,z}$
$\neq 0$	$\neq \pm 90$	any
any	$\neq 0$	any

Table B.6: Unique failing configuration for calibration of sensor pitch

$\phi^b_{s,x}$	$\phi^b_{s,y}$	$\phi^b_{s,z}$
$= 0$	$= 0$	any

Finally, Equations B.7 and B.8 form the requirements in Tables B.7, B.8 and B.9.

Table B.7: Sensor configuration required for calibration of sensor yaw

$\phi^b_{s,x}$	$\phi^b_{s,y}$	$\phi^b_{s,z}$
any	$\neq \pm 90$	$\neq 0$
$\neq \pm 90$	any	$\neq \pm 90$
$\neq 0$	$\neq 0$	$\neq 0$
any	$\neq \pm 90$	$\neq \pm 90$
$\neq \pm 90$	any	$\neq 0$
$\neq 0$	$\neq 0$	$\neq \pm 90$

Table B.8: Condensed requirement for calibration of sensor yaw

$\phi^b_{s,x}$	$\phi^b_{s,y}$	$\phi^b_{s,z}$
$\neq 0$	$\neq 0$	any
any	$\neq \pm 90$	any
$\neq \pm 90$	any	any

Table B.9: Further condensing of requirement for calibration of sensor yaw

$\phi^b_{s,x}$	$\phi^b_{s,y}$	$\phi^b_{s,z}$
any	any	any

Appendix C

Laplace Approximation for Covariance Estimation

In Section 4.1, a sensor calibration procedure was presented in order to determine the most likely position of the sensor with respect to the vehicle body frame. Once the position is determined, it is also desirable to calculate the uncertainty of the location estimate, given the calibration data. Section 4.1.2 discussed a Monte Carlo (MC) importance sampling technique and the Laplace approximation as potential methods to calculate the covariance matrix of the sensor pose, but the MC method was chosen due to the potential for numerical instability of the Laplace method. This section shows the application of the Laplace approximation to the single sensor calibration in Section 4.1.4.

The Laplace approximation fits a Gaussian about a local maximum of a function, by fitting a quadratic to the log of the function. A general description of the technique can be found in [44, §27], [45, §4.4]. However, problems can arise due to numerical instability when calculating the required Hessian matrix [48]. In this appendix, the Laplace approximation is calculated for the six sensor location parameters of $\{\mathbf{r}_s^b, \mathbf{\Phi}_s^b\}$ *individually*, rather than producing the entire joint Gaussian estimate. As such, numerical instability can be avoided by performing adequate sampling of the function domain, yet the results are not as useful due to the invalid assumption that the six

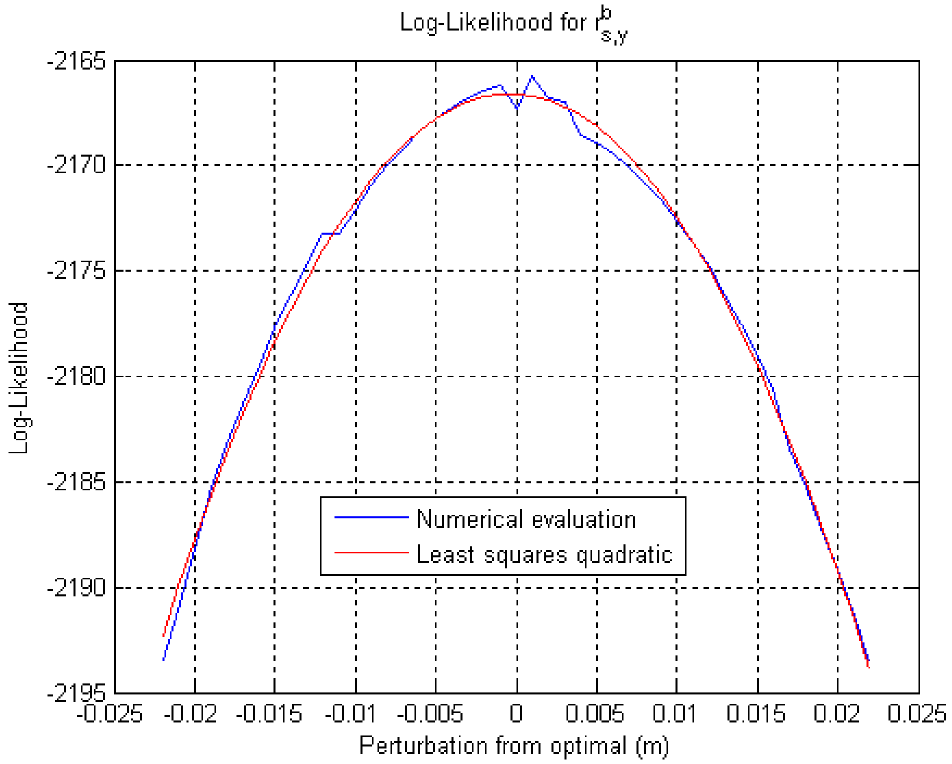


Figure C.1: The log likelihood for the Y sensor offset, for varying amounts of perturbation in metres. The quadratic provides a good approximation to the data.

parameters are independent.

The optimal solution for the single sensor calibration is given from Section 4.1.4. For each parameter in isolation, the solution is perturbed, and the likelihood function $\Lambda(\mathbf{x})$ is evaluated using Equation 4.24.

Figures C.1 and C.2 show the log-likelihood and likelihood for the Y offset of the sensor $r^b_{s,y}$. In Figure C.1, a quadratic is fitted to the log-likelihood and differentiated twice to yield the curvature. This greatly reduces the numerical instability of the Laplace method, but is only possible for small dimensions (in this case one) due to the large number of samples that it requires. In one dimension, the variance of the Gaussian likelihood approximation is given by:

$$\sigma^2 = -1/2a \tag{C.1}$$

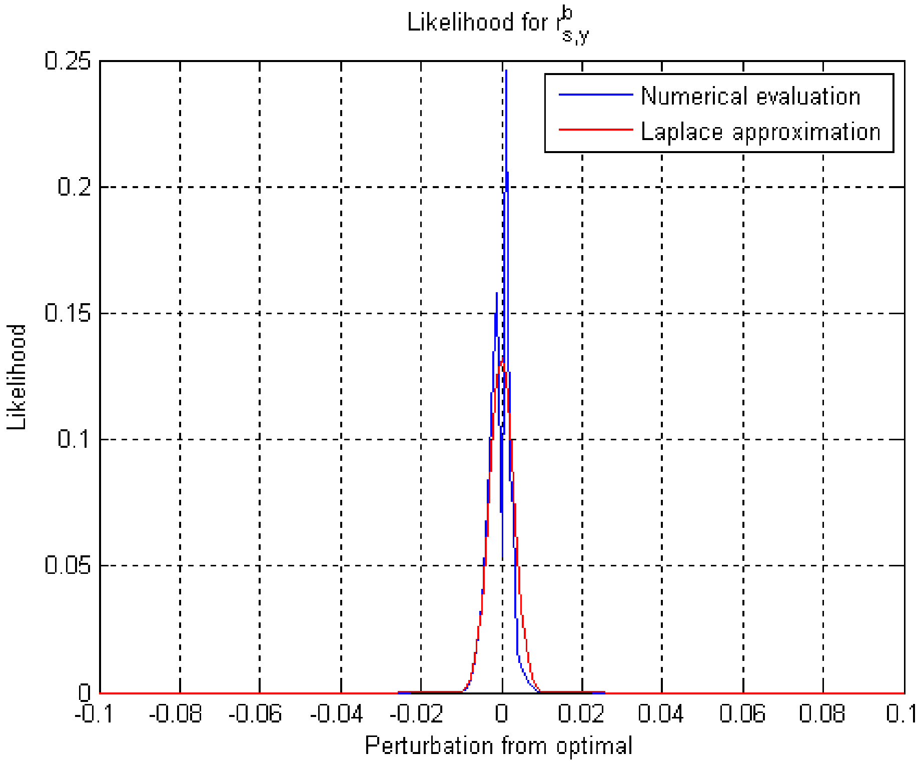


Figure C.2: The likelihood for the Y sensor offset, for varying amounts of perturbation in metres. The Gaussian is a good approximation to the data.

where a is the coefficient of x^2 in the quadratic. As can be seen in Figure C.2 the Gaussian is a reasonable estimate of the likelihood distribution. The same process is applied to each of the six parameters, and the results are compared with the MC method in Table C.1, reproduced below for convenience. The orthogonal standard deviations are very similar to the values produced by the MC method. The Laplace method used two hundred samples for each parameter. If the same sample density was required to estimate the six dimensional joint Gaussian distribution, then 64 trillion samples would be required, which is intractable. There exist numerical Hessian approximation techniques that require far fewer samples, however these can suffer from numerical instability or over-sensitivity as discussed in [48].

Table C.1: Sensor Pose: Optimised and Monte Carlo Mean and Standard Deviation

	$r^b_{s,x}(m)$	$r^b_{s,y}(m)$	$r^b_{s,z}(m)$	$\phi^b_{s,z}(^\circ)$	$\phi^b_{s,y}(^\circ)$	$\phi^b_{s,x}(^\circ)$
$D_{pole} + D_{ground}$	0.053	-0.004	-0.998	-1.741	-7.066	-0.072
$\hat{\mathbf{x}}$ (MC)	0.054	-0.003	-0.990	-1.742	-7.063	-0.071
$\boldsymbol{\sigma}$ (MC)	0.0031	0.0038	0.0673	0.0353	0.0232	0.0085
$\boldsymbol{\sigma}$ (Laplace)	0.0029	0.0030	0.0673	0.0258	0.0202	0.0062

The weighted mean sensor location $\hat{\mathbf{x}}$ and standard deviations $\boldsymbol{\sigma}$ from the Monte Carlo (MC) sampling process in Equation 4.8 and the standard deviations from the Laplace method in Appendix C. The optimal pose from Table 4.4 is repeated here for comparison.

Appendix D

Sampling Order Invariance of Bayes Updates

Theorem D.1. *When sequentially applying Bayes rule to form a posterior state distrubtion from a prior and multiple independent observations, the resulting posterior is invariant to the order with which the samples are applied.*

Proof. Let the state be x, the prior distribution $P(x)$, and three independent observations be z_1, z_2 and z_3.

The posterior after applying Bayes rule with z_1 is given by:

$$P(x|z_1) = \frac{P(z_1|x)P(x)}{P(z_1)} \tag{D.1}$$

The posterior $P(x|z_1)$ is then considered to be the prior for the subsequent application of Bayes rule with the observation z_2, yielding the posterior:

$$\begin{aligned} P(x|z_1, z_2) &= \frac{P(z_2|x)P(x|z_1)}{P(z_2)} \\ P(x|z_1, z_2) &= \frac{P(z_2|x)P(z_1|x)P(x)}{P(z_2)P(z_1)} \end{aligned} \tag{D.2}$$

by substitution of Equation D.1.

Similarly, the application of the third observation z_3, yields the final posterior:

$$P(x|z_1, z_2, z_3) = \frac{P(z_3|x)P(z_2|x)P(z_1|x)P(x)}{P(z_3)P(z_2)P(z_1)} \tag{D.3}$$

It is clear from the commutativity of Equation D.3 that any permutation of $z1, z2, z3$ will result in the same posterior. □

The same method was used to show order ambivalence for the specific case of Bayes updates for binary states in [68, §E.5.1].

Appendix E

Finding Maximally Informative Joint Sensor Subspaces

In Section 5.2.2, mutual information (MI) was shown to be a useful metric for fault detection using analytical redundancy. The MI between data from a thermal infrared camera and a colour video camera was calculated, and it was shown that although the quantity of MI was low (largely due to imprecise sensor alignment and calibration) it could still be used for fault detection. This was achieved by considering the strong relationship between the temporal alignment of the sensors and the corresponding amount of mutual information in the data. When the MI peak is found to align to the known temporal offset, this shows that the data from both sensors are correlated by the environment that they perceive, indicating that the sensors are functioning. However, the assumption that the environment will correlate the sensor information is not always valid. For example, if an IR and colour camera are viewing a plane white surface, the MI in the data will be low. In this appendix, an efficient method of calculating highly mutually informative sub-regions within multi-modal sensor information streams is applied to thermal infrared and colour video camera data, to show that this may be feasible for fault detection in scenarios where only a partial region of the data streams is correlated by the environment.

An efficient method for "learning maximally informative joint subspaces to find crossmodal correspondences" is presented in [78], where the technique is applied to find correlated sections of a video and an audio stream. The method is shown to be capable of identifying the speaker's lips in the video stream as the section most correlated to the audio. In this appendix, the method is applied directly to the thermal IR and colour video data that were seen in Section 5.2.1.

This method was used in [79] to identify correlated target tracks from two uncalibrated sensors, so that the tracks can be used to estimate the offsets between the sensors. They discuss the problem that [78] assumes that the features are seen in a constant location within the sensor space; for example, that the features do not move across the camera frame. However, for the application in this appendix, the spatial arrangement of the two sensors is similar. When a target moves across the IR camera, it moves in a similar way across the video frame, so this assumption can be relaxed. The images are not perfectly aligned or calibrated, so the quality will degrade for larger time sequences because the MI is 'smeared' spatially due to the motion.

A sequence of 30 frames is taken from the data that were used in Section 5.2.1. At approximately 15 frames per second, this corresponds to 2 seconds of data. The colour images are converted to grey scale. The first two rows of Figure E.1 show the initial and final colour and IR frames from the time-synchronised video sequence. A car drives along the road in front of the sensors, crossing the field of view of both. The technique presented in [78] is applied to the data, and the result can be seen in the bottom row of the figure. This row shows the maximally informative joint subspaces of the two original data sources, where white pixels are more highly correlated than dark pixels. It can be seen that the appearance and motion of the car provide the highest source of mutual information in this sequence, so these sub-regions can then be used to directly calculate the relationship of MI to the time offset for fault detection.

The results were produced in approximately 14 seconds using a 2GHz computer running Matlab. The processing time could be reduced by an order of magnitude with a more efficient implementation. Given the resolution of the images, (210×280), there are approximately 3.5 billion permutations of pixels pairs, so to directly calculate the MI between all of them to find the maximally informative joint space would require

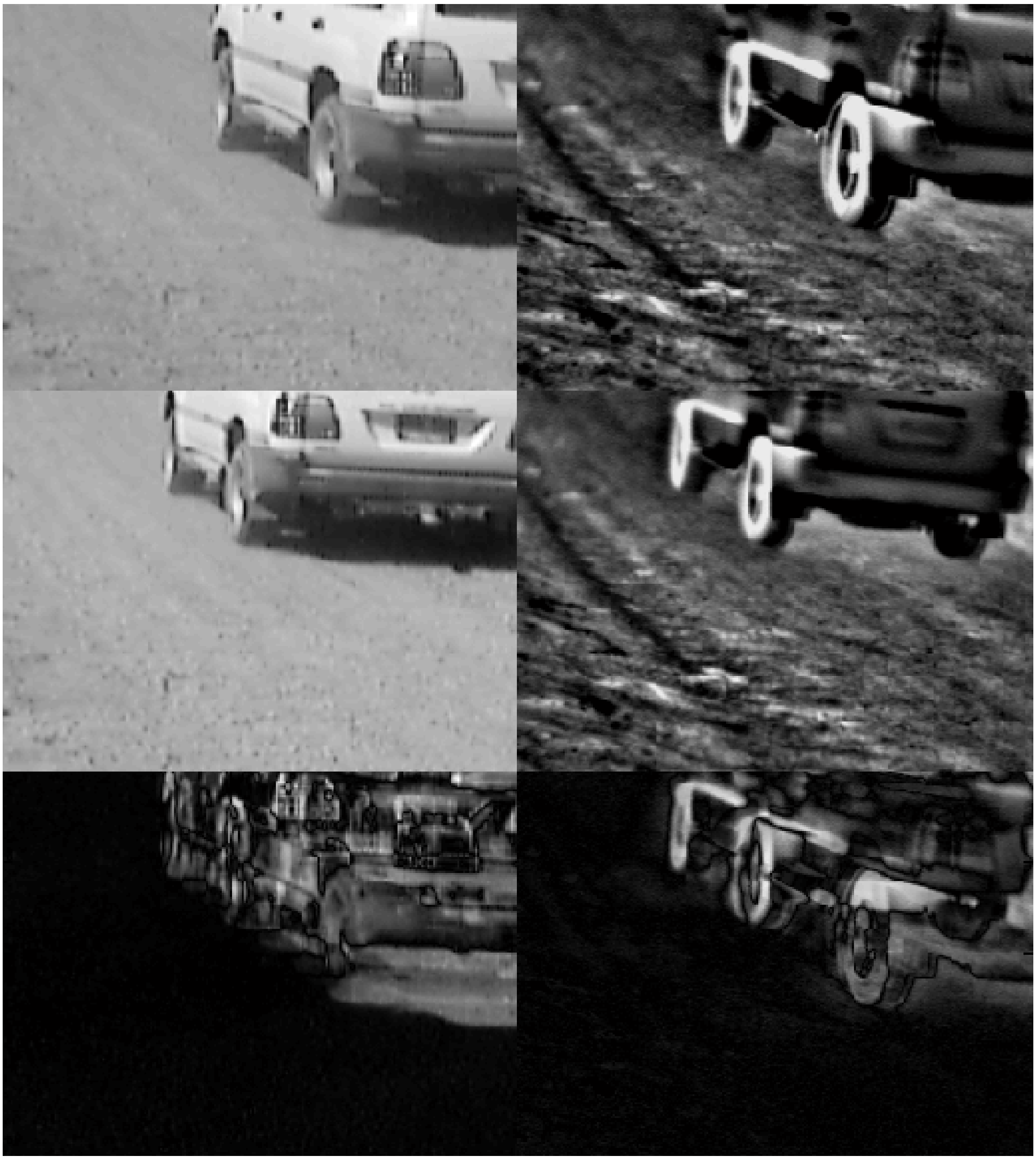


Figure E.1: Colour video and thermal infra-ref video of a car driving along a dirt road. A 2 second segment of the video is selected. The first frame is shown on the first row (colour on the left and IR on the right) and the last frame is shown on the second row. The third row shows the maximally informative joint subspace as calculated by the method in [78], where white pixels are more highly correlated and dark pixels are less correlated.

at least an order of magnitude more time. Furthermore, the dimensionality reduction in this technique means that the processing time increases with the number of whole images in the video sequence, and not significantly with the resolution of each image, so for higher resolution images, the saving in time can be several orders of magnitudes.

Bibliography

[1] N. S. Nise, *Control Systems Engineering*, 4th ed. John Wiley and Sons, 2004.

[2] J. Leonard, J. How, S. Teller, M. Berger, S. Campbell, G. Fiore, L. Fletcher, E. Frazzoli, A. Huang, S. Karaman, O. Koch, Y. Kuwata, D. Moore, E. Olson, S. Peters, J. Teo, R. Truax, M. Walter, D. Barrett, A. Epstein, K. Maheloni, K. Moyer, T. Jones, R. Buckley, M. Antone, R. Galejs, S. Krishnamurthy, and J. Williams, “A perception driven autonomous urban vehicle,” *Journal of Field Robotics*, vol. 25, no. 10, pp. 727–774, September 2008.

[3] J. Rossiter, *Model-Based Predictive Control.* CRC Press, 2003.

[4] A. Kelly, A. Stentz, O. Amidi, M. Bode, D. Bradley, A. Diaz-Calderon, M. Happold, H. Herman, R. Mandelbaum, T. Pilarski, P. Rander, S. Thayer, N. Vallidis, and R. Warner, “Toward reliable off road autonomous vehicles operating in challenging environments,” *International Journal of Robotics Research*, vol. 25, pp. 449–483, May/June 2006.

[5] H. Moravec and A. Elfes, “High resolution maps from wide angle sonar,” *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, vol. 2, pp. 116–121, March 1985.

[6] H. F. Durrant-Whyte, “An Autonomous Guided Vehicle for Cargo Handling Applications,” *The International Journal of Robotics Research*, vol. 15, no. 5, pp. 407–440, 1996.

[7] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” *Transactions of the ASME, Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.

[8] B. Siciliano and O. Khatib, Eds., *Handbook of Robotics.* Springer, 2008.

[9] M. S. Grewal, L. R. Weill, and A. P. Andrews, *Global Position Systems, Inertial Navigation and Integration.* New York: John Wiley & Sons, Inc, 2001.

[10] H. Durrant-Whyte, D. Pagac, B. Rogers, M. Stevens, and G. Nelmes, "Field and service applications - an autonomous straddle carrier for movement of shipping containers - from research to operational autonomous systems," *Robotics & Automation Magazine, IEEE*, vol. 14, no. 3, pp. 14–23, September 2007.

[11] S. Scheding, "High integrity navigation," Ph.D. dissertation, The University of Sydney, December 1997.

[12] (2009) DARPA Urban Challenge 2007. [Online]. Available: http://www.darpa.mil/GRANDCHALLENGE/

[13] A. Bacha, C. Bauman, R. Faruque, M. Fleming, C. Terwelp, C. Reinholtz, D. Hong, A. Wicks, T. Alberi, D. Anderson, S. Cacciola, P. Currier, A. Dalton, J. Farmer, J. Hurdus, S. Kimmel, P. King, A. Taylor, D. V. Covern, and M. Webster, "Odin: Team victortango's entry in the DARPA urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 467–492, 2008.

[14] I. Miller, M. Campbell, D. Huttenlocher, F.-R. Kline, A. Nathan, S. Lupashin, J. Catlin, B. Schimpf, P. Moran, N. Zych, E. Garcia, M. Kurdziel, and H. Fujishima, "Team cornell's skynet: Robust perception and planning in an urban environment," *Journal of Field Robotics*, vol. 25, no. 8, pp. 493–527, 2008.

[15] D. Moore, A. S. Huang, M. Walter, E. Olson, L. Fletcher, J. Leonard, and S. Teller, "Simultaneous local and global state estimation for robotic navigation," *Submitted to the IEEE International Conference on Robotics and Automation*, 2009, under review for publication.

[16] L. Connolly, "Generic architecture design for navigation filtering," Master's thesis, The University of Sydney, December 2008.

[17] L. Connolly and S. Scheding, "Architecture design for multi-sensor fusion," *International Global Navigation Satellite Systems Society IGNSS Symposium*, December 2007.

[18] B. K. H. Soon, L. Connolly, and S. Scheding, "Error analysis of an integrated inertial navigation system and pseudoslam during gps outages," *Proceedings of international global navigation satellite systems society IGNSS symposium*, 2006.

[19] B. K. H. Soon, S. Scheding, H.-K. Lee, H.-K. Lee, and H. Durrant-Whyte, "An approach to aid ins using time-differenced gps carrier phase (TDCP) measurements," *GPS Solutions*, vol. 12, pp. 261–271, January 2008.

[20] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. N. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, M. Gittleman, S. Harbaugh, M. Hebert, T. M. Howard, S. Kolski, A. Kelly, M. Likhachev, M. McNaughton, N. Miller, K. Peterson, B. Pilnick, R. Rajkumar, P. Rybski, B. Salesky, Y.-W. Seo, S. Singh,

J. Snider, A. Stentz, W. R. Whittaker, Z. Wolkowicki, J. Ziglar, H. Bae, T. Brown, D. Demitrish, B. Litkouhi, J. Nickolaou, V. Sadekar, W. Zhang, J. Struble, M. Taylor, M. Darms, and D. Ferguson, "Autonomous driving in urban environments: Boss and the urban challenge," *J. Field Robot.*, vol. 25, no. 8, pp. 425–466, 2008.

[21] J. Bohren, T. Foote, J. Keller, A. Kushleyev, D. Lee, A. Stewart, P. Vernaza, J. Derenick, J. Spletzer, and B. Satterfield, "Little ben: The ben franklin racing team's entry in the 2007 DARPA urban challenge," *Journal of Field Robotics*, vol. 25, no. 9, pp. 598–614, 2008.

[22] Y.-L. Chen, V. Sundareswaran, C. Anderson, A. Broggi, P. Grisleri, P. P. Porta, P. Zani, and J. Beck, "Terramax: Team oshkosh urban robot," *Journal of Field Robotics*, vol. 25, no. 10, pp. 841–860, 2008.

[23] I. Miller and M. Campbell, "A Mixture-Model Based Algorithm for Real-Time Terrain Estimation," *Journal of Field Robotics*, pp. 755–775, September 2006.

[24] C. Ellum and N. El-Sheimy, "Land-based mobile mapping systems," *Photogrammetric Engineering and Remote Sensing*, vol. 68, no. 1, pp. 13–17+28, 2002.

[25] J. C. B. Redfearn, "Transverse Mercator Formulae," *Empire Survey Review*, vol. 69, pp. 318–322, 1948.

[26] (2008) International Commitee on Surveying and Mapping. [Online]. Available: http://www.icsm.gov.au/icsm/gda/gdatm/index.html

[27] R. M. Rogers, *Applied Mathematics in Integrated Navigation Systems*, 2nd ed. 1801 Alexander Bell Drive, Reston, VA 20191-4344: American Institute of Aeronautics and Astronautics, Inc., 2003.

[28] E. Nettleton, "Decentralised Architectures for Tracking and Navigation with Multiple Flight Vehicles," Ph.D. dissertation, The University of Sydney, 2003.

[29] J. Selig, "Active versus passive transformations in robotics," *Robotics and Automation Magazine, IEEE*, vol. 13, no. 1, pp. 79–84, March 2006.

[30] H. C. Corben and P. Stehle, *Classical Mechanics*, 2nd ed. New York: Dover Publications, August 1994.

[31] H. Goldstein, C. P. Poole, and J. L. Safko, *Classical Mechanics*, 3rd ed. San Francisco: Addison Wesley, June 2001.

[32] H. Rabitz, M. Kramer, and D. Dacol, "Sensitivity analysis in chemical kinetics," *Annual Review of Physical Chemistry*, vol. 34, no. 1, pp. 419–461, 1983. [Online]. Available: http://arjournals.annualreviews.org/doi/abs/10.1146/annurev.pc.34.100183.002223

[33] J. Stewart, *Calculus*, 3rd ed. Brooks/Cole Publishing Company, 1995.

[34] L. Kaelbling, M. Littman, and A. Cassandra, “Planning and acting in partially observable stochastic domains,” *Artificial Intelligence*, vol. 101, pp. 99–134, 1998.

[35] J. Gonzalez and A. Stentz, “Planning with uncertainty in position using high-resolution maps,” *Robotics and Automation, 2007 IEEE International Conference on*, pp. 1015–1022, April 2007.

[36] (2008) Network Time Protocol. [Online]. Available: http://www.ntp.org/

[37] D. L. Mills, *Computer Network Time Synchronisation: The Network Time Protocol.* CRC Press, 2006.

[38] S. Julier and J. Uhlmann, “A New Extension of the Kalman Filter to Nonlinear System,” in *Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defence Sensing, Simulation and Controls.* Orlando, FL: International Global Navigation Satellite System Society, April 1997.

[39] J. Underwood, F. Ramos, and S. Scheding, “Real-time map building with uncertainty using colour camera and scanning laser,” *In Proceeding of the Australian Conference on Robotics and Automation (ACRA)*, 2007.

[40] E. Krotkov, “Laser calibration for a walking robot,” *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2568–2573, April 1991.

[41] G.-Q. Wei and G. Hirzinger, “Active self-calibration of hand-mounted laser range finders,” *IEEE Transactions on Robotics and Automation, vol. 14, no. 3*, pp. 493–497, June 1998.

[42] Q. Zhang and R. Pless, “Extrinsic calibration of a camera and laser range finder,” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2301–2306, 2004.

[43] F. Mirzaei and S. Roumeliotis, “A kalman filter-based algorithm for imu-camera calibration,” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2427–2434, 29 2007-Nov. 2 2007.

[44] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms.* Cambridge University Press, 2003.

[45] C. M. Bishop, *Pattern Recognition and Machine Learning.* Springer, August 2006.

[46] O. Bengtsson and A.-J. Baerveldt, “Robot localization based on scan-matching–estimating the covariance matrix for the idc algorithm,” *Robotics and Autonomous Systems*, vol. 44, no. 1, pp. 29–40, July 2003.

[47] A. Censi, “An accurate closed-form estimate of ICP’s covariance,” *Robotics and Automation, 2007 IEEE International Conference on*, pp. 3167–3172, April 2007.

[48] T. Bailey and J. Nieto, “Scan-slam: Recursive mapping and localisation with arbitrary-shaped landmarks,” *Workshop of Robotics Science and Systems (RSS)*, 2008.

[49] T. MathWorks, “Matlab optimisation function: fmincon,” http://www.mathworks.com, 2007, website as at 18/01/07.

[50] J. Underwood, A. Hill, and S. Scheding, “Calibration of Range Sensor Pose on Mobile Platforms,” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007.

[51] “SICK Laser Measurement Systems, LMS-291-S05 Technical Description,” 2008. [Online]. Available: http://www.sick.com/

[52] R. Karp, J. Elson, D. Estrin, and S. Shenker, “Optimal and global time synchronization in sensornets,” 2003. [Online]. Available: citeseer.ist.psu.edu/karp03optimal.html

[53] (2008) Riegl Laser Measurement Systems. [Online]. Available: http://www.riegl.com/

[54] “SICK Laser Measurement Systems, LMS-291-S05 Telegram Listing,” 2008. [Online]. Available: http://www.sick.com/

[55] (2008) QNX Realtime Operating System. [Online]. Available: http://www.qnx.com/

[56] P. Pfaff, R. Triebel, and W. Burgard, “An Efficient Extension to Elevation Maps for Outdoor Terrain Mapping and Loop Closing,” *The International Journal of Robotics Research*, vol. 26, no. 2, pp. 217–230, 2007.

[57] C. Wellington, A. Courville, and A. T. Stentz, “A Generative Model of Terrain for Autonomous Navigation in Vegetation,” *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1287 – 1304, December 2006.

[58] D. J. C. MacKay, “Introduction to Gaussian Processes,” *NATO ASI Series F Computer and Systems Sciences*, vol. 168, pp. 133–166, 1998.

[59] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes in Machine Learning.* The MIT Press, December 2005.

[60] T. Lang, C. Plagemann, and W. Burgard., “Adaptive non-stationay kernel regression for terrain modelling,” *In Proceedings of Robotics Science and Systems (RSS)*, 2007.

[61] A. Okabe, B. N. Boots, and K. Sugihara, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams.* Wiley, 1992.

[62] T. Bailey, B. Upcroft, and H. Durrant-Whyte, "Validation gating for non-linear non-gaussian target tracking," *Information Fusion, 2006 9th International Conference on*, pp. 1–6, July 2006.

[63] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (slam) problem," *Robotics and Automation, IEEE Transactions on*, vol. 17, no. 3, pp. 229–241, Jun 2001.

[64] F. T. Ramos, "Recognising, Representing and Mapping Natural Features in Unstructured Environments," Ph.D. dissertation, The University of Sydney, February 2007.

[65] W. R. Tobler, "A computer movie simulating urban growth in the detroit region," *Economic Geography*, vol. 46, pp. 234–240, June 1970.

[66] Y. Bar-Shalom, T. Kirubarajan, and X. Lin, "Probabilistic data association techniques for target tracking with applications to sonar, radar and eo sensors," *IEEE Aerospace and Electronic Systems Magazine*, vol. 20, no. 8, pp. 37–56, 2005.

[67] I. Rhee, M. Abdel-Hafez, and J. Speyer, "Observability of an integrated gps/ins during maneuvers," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 40, no. 2, pp. 526–535, April 2004.

[68] R. Grover, "A Model for Sensing and Reasoning in Autonomous Systems," Ph.D. dissertation, The University of Sydney, March 2006.

[69] J. Gertler, *Fault detection and diagnosis in engineering systems.* CRC Press, 1998.

[70] N. Bonavita, M. Fazio, and M. Mainini, "Model-based fault detection and isolation techniques: the parameter identification approach within the framework of the topmuss cae-system," *Control Applications, 1994., Proceedings of the Third IEEE Conference on*, pp. 1369–1378 vol.2, Aug 1994.

[71] B. Xie, Z. Xiang, H. Pan, and J. Liu, "Polarization-based water hazards detection for autonomous off-road navigation," *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pp. 3186–3190, 29 2007-Nov. 2 2007.

[72] G. Brooker, *Introduction to Sensors for Ranging and Imaging.* Scitech Publishing Inc., 2009.

[73] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Prentice Hall, 2003.

[74] K. Murphy, “Bayes Net Toolbox for Matlab,” bnt.sourceforge.net, 1997–2002, last updated: 19 October 2007.

[75] A. Stentz, “Optimal and efficient path planning for partially-known environments,” *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pp. 3310–3317 vol.4, May 1994.

[76] T. Allen, J. Underwood, and S. Scheding, “A Planning System for Autonomous Ground Vehicles Operating in Unstructured Dynamic Environments,” *In Proceeding of the Australian Conference on Robotics and Automation (ACRA)*, 2007.

[77] T. Allen, A. Hill, J. Underwood, and S. Scheding, “Dynamic path planning with multi-agent data fusion - the parallel hierarchical replanner system,” *Submitted to the IEEE International Conference on Robotics and Automation*, 2009, under review for publication.

[78] J. W. F. III and T. Darrell, “Speaker association with signal-level audiovisual fusion,” *IEEE Transactions on Multimedia*, vol. 6, no. 3, pp. 406–413, June 2004.

[79] K. Kodagoda, A. Alempijevic, J. Underwood, S. Kumar, and G. Dissanayake, “Sensor registration and calibration using moving targets,” *Control, Automation, Robotics and Vision, 2006. ICARCV ’06. 9th International Conference on*, pp. 1–6, Dec. 2006.