MAPPING LARGE, URBAN ENVIRONMENTS WITH GPS-AIDED SLAM

Justin Carlson

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Robotics.

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania

DATE HERE

Committee:
Charles Thorpe (chair)
Brett Browning
Martial Hebert
Frank Dellaert (Georgia Institute of Technology)

Abstract

Simultaneous Localization and Mapping (SLAM) has been an active area of research for several decades, and has become a foundation of indoor mobile robotics. However, although the scale and quality of results have improved markedly in that time period, no current technique can effectively handle city-sized urban areas.

The Global Positioning System (GPS) is an extraordinarily useful source of localization information. Unfortunately, the noise characteristics of the system are complex, arising from a large number of sources, some of which have large autocorrelation. Incorporation of GPS signals into SLAM algorithms requires using low-level system information and explicit models of the underlying system to make appropriate use of the information. The potential benefits of combining GPS and SLAM include increased robustness, increased scalability, and improved accuracy of localization.

This dissertation presents a theoretical background for GPS-SLAM fusion. The presented model balances ease of implementation with correct handling of the highly colored sources of noise in a GPS system.. This utility of the theory is explored and validated in the framework of a simulated Extended Kalman Filter driven by real-world noise.

The model is then extended to Smoothing and Mapping (SAM), which overcomes the linearization and algorithmic complexity limitations of the EKF formulation. This GPS-SAM model is used to generate a probabilistic landmark-based urban map covering an area an order of magnitude larger than previous work.

Contents

Chapter 1

Introduction

1.1 Motivation

Autonomous transportation is a puzzle which, if solved robustly, has the potential to yield immense benefits in safety, ecology, and economy. Yet, despite an immense amount of effort in the field, general autonomous transportation remains an elusive goal.

Although some of the problems of autonomous transportation have been addressed in specific scenarios, such as semi-structured highway scenes, the more generalized problem appears to be less tractable; there are currently no good models that allow robots to navigate in general outdoor environments with anything approaching the efficiency of a natural intelligence.

So what is needed before autonomous transportation can be considered a solved problem, ready to be refined and commercialized?

Autonomous robots cannot yet safely and robustly navigate urban environments. The reasons for this are many: in cities, the simplicity and orthogonality that would be friendly to robot reasoning is trumped by history and geography. At a macroscopic level, cities are like organisms; they evolve in response to an astounding range of stimuli. The result is complex, and doesn't typically map well to the kinds of simplifications and assumptions of which roboticists are fond. Additionally, cities are full of pesky humans who significantly complicate the requirements of an autonomous system by being simultaneously the least predictable and the most consequential actors in a given

scene.

There are two obvious ways to approach the problem of autonomy in such an environment. One is to computationally generate enough semantic understanding of environments to enable reasoning about proper courses of action. In effect, this approach seeks to create a robot that serves as a drop-in artificial replacement for a human driver. This is, in some respects, the ideal solution; such a robot would be flexible and general-purpose without the need for specific domain- (or city-) specific prior knowledge.

I believe that this is a hopeless task in the near term. No robot has come close to displaying the level of cognition necessary to deal with the unstructured and dynamic environments encountered in an urban setting. Perhaps projects in the same spirit as the DARPA Urban Challenge will push this envelope, but I believe that this is fundamentally the wrong approach to urban autonomy at the present time.

The alternative approach is to limit the necessary amount of semantic understanding as much as possible by injecting domain-specific prior knowledge into the system. This approach implicitly rejects the idea that mimicking humans is the most efficient way to approach autonomous transportation. Instead, the problem is approached by attempting to efficiently decompose the larger challenge into pieces that robots can handle efficiently and robustly.

This work is a piece in the larger puzzle of autonomous urban transportation. By demonstrating a tractable, effective way to build localization maps in very large-scale environments, we limit the need for semantic understanding to a much smaller, and hopefully more tractable, set of problems.

1.2 Integrating GPS with SLAM

approach

There is a great deal of useful work to be done in bringing the fields of SLAM and GPS navigation together. I believe that the combination of the two fields will enable the creation of maps of large, urban-sized areas, which are typically hundreds of square kilometers.

High-accuracy large-scale outdoor mapping is not something that can be accom-

plished with GPS alone. Although high-precision GPS receivers exist, they rely on having a continuous, clear line of sight to multiple satellites to function. In areas without a continuous clear view of a wide part of the sky, the location information available from GPS will be degraded or nonexistent. Dense urban areas represent a particularly challenging environment for GPS operation, yet it is in such environments that accurate localization would be most valuable. Furthermore, the use of more elaborate techniques to improve GPS precision imposes ever more stringent requirements on signal availability. In short, increasing GPS precision comes at a cost of decreased availability. This can be mitigated to some extent by incorporating a self-contained integrating error motion estimate into the system, such as odometry or inertial sensing, but this is not a satisfactory solution; during a period of GPS outage, the unbounded growth of pose uncertainty quickly makes high-precision navigation impossible.[1]

Broadly speaking, we wish to be able to bound our localization error *whether or not GPS is currently available*. Looking to the field of SLAM, we find ideas on how to bound our localization without the benefit of a bounded error pose sensor, and how to propagate high precision information through a map to improve our estimates of both the location of landmarks and our pose when we next traverse the area of high uncertainty.

Scaling SLAM systems to large-scale environments is difficult. Although some promising recent work addresses some of the strictly computational issues of large-scale SLAM in a variety of clever ways, significant problems remain, such as robustly associating landmarks at the end of large loop closures, preventing catastrophic failure due to the inevitable occasional incorrect associations, and dealing with long-term feature management.

In addressing these questions, research in outdoor SLAM largely ignores the existence of GPS, instead trying (with mixed results) to scale algorithms that work well indoors to environments both less regular and larger by several orders of magnitude. There is some amount of perception that SLAM and GPS navigation are discrete fields – if you have GPS, the conventional wisdom goes, add an IMU and a Kalman filter and

[1] Given a sufficiently high-precision (and high-cost) sensor the problem may be delayed, but not indefinitely.

don't bother with mapping; with sufficiently good GPS and a high quality IMU, navigation is a solved problem. However, as evidenced by the lack of widespread autonomous vehicles, the GPS-IMU solution is neither sufficiently robust nor (with reasonably priced hardware) sufficiently accurate to cope with complex unstructured environments.

We can do significantly better by using GPS to augment a SLAM system. The two largest problems of GPS-based navigation are outages and limits in accuracy. SLAM brings to the table methods for continually refining positional estimates and dealing with long periods of error integration. On the other hand, one of the primary difficulties in scaling SLAM is that accurately closing loops becomes an increasingly difficult problem as a robot's localization certainty decreases. Providing a non-integrating source of localization information significantly eases this task.

This work moves towards the unification of GPS and SLAM in urban environments. By demonstrating it is feasible to create high-precision, high-coverage maps large enough to encompass significant urban areas, we get closer to the goal of enabling autonomous, robust urban navigation.

1.3 Document Outline

The rest of this document is organized as follows. Chapter 2 highlights significant related work, particularly in the areas of SLAM and GPS navigation. Chapter 3 analyzes the various sources of error and noise in a GPS system with an eye to how GPS information can be used consistently and appropriately in SLAM systems. Chapter 4 integrates GPS into a classical EKF-SLAM system to demonstrate GPS-SLAM integration on a well-understood model. In chapter 5, an integrated GPS-SAM system is presented to show how GPS can be integrated into scalable probabilistic mapping implementations. Finally in chapter 6 we conclude and discuss future directions of work.

Chapter 2

Related Work

Localization and mapping has been a very active area of research recently. This section summarizes some of the major themes which appear in the literature.

2.1 GPS/INS/Odometry integration

GPS integration with Inertial Navigation Systems (INS) is, in some respects, the classic example of sensor fusion. The two sensing modalities are extremely complimentary. GPS provides bounded error, slow-update positional information with bad noise characteristics in high frequencies, and excellent error characteristics in low frequencies. INS systems provide largely the opposite: unbounded integration error, fast update rate with excellent high frequency error characteristics, and pathological low-frequency errors. In situations where GPS is highly available, this sensing combination can provide extremely high-fidelity localization estimation.

The field is sufficiently mature that several books are dedicated to the topic, such as [Grewal *et al.*, 2001] and [Farrell and Barth, 1999].

The techniques for fusing GPS and IMU data typically are categorized as *tightly* or *loosely* coupled. Speaking generally, a loosely coupled system uses the GPS as a black box which generates positional and velocity information. This information is then fused with the IMU acceleration and integrated velocity and position terms to generate an overall state estimate. A tightly coupled system, in contrast, uses the pseudorange and

pseudorange rate as direct inputs into the filter, and solves for vehicle state and dynamics estimates in an integrated manner.

Unfortunately, this is not a complete solution for most urban environments, wherein GPS availability is typically discontinuous. Although there are inertial systems with enough precision to compensate for long GPS outages without introducing significant error, the cost of such units is prohibitively expensive at this time.

This work is very related to tightly coupled methods; by doing more detailed estimation in SLAM using the individual parts of the GPS system instead of treating positional and velocity fixes as black boxes, we seek to improve and appropriately model the underlying probabilistic systems.

2.2 Simultaneous Localization and Mapping

For the past two decades, SLAM has been a hot topic of research. This work nearly universally assumes a lack of any bounded-error beacons for localization, and uses landmarks to both build a map of a robot's environment and localize the robot within the map.

The primary feature which distinguishes SLAM from odometry augmentation is loop closure. When a robot revisits an area in which it has previously operated, ideally it can then bound the error accumulated over the course of the odometry walk both forward and backwards in time; previously visited points can be corrected to become more consistent with a Euclidean space, and the current uncertainty estimate for the robot's location can be reduced (relative to some starting point). The field was essentially started by [Smith and Cheeseman, 1986], who proposed both the problem and a solution based on simultaneously estimating the robot pose and the position of landmarks in a single extended Kalman filter (EKF). The original solution is amazingly elegant, but does not scale well; the matrix inversion required in updating makes the complexity of the approach $O((m+dn)^3)$, where m is the dimensionality of the pose estimate, d is the dimensionality of landmark estimates, and n is the number of landmarks in the map. In addition to the computational issues, the EKF solution uses linear approximations to nonlinear processes at each timestep. The addition of this linearization error can cause

problems for accuracy and stability of the filter.

Even though it is not immediately applicable to more than very small sized environments, this theoretical framework is the starting point for the bulk of SLAM research which has come since.

Most of the work in the field since Smith and Cheeseman's original paper attempts to address some combination of the linearization issue, the computational complexity issues, or the required data association issues.

Linearization Improvements

Within the linearized frameworks, the linearization process has traditionally used the straightforward method of calculating the Jacobian of the nonlinear process around some point of interest to generate the linearized estimate. The usual justification given for this process is that the Jacobian incorporates the first element of the Taylor series expansion of the nonlinear function. [Julier and Uhlmann, 1997] presented an alternative method for linearization based on nonlinear transformation of a small number of carefully selected sample points from the distribution. This method of linearization yields better results than the Extended Kalman Filter in virtually all scenarios, and does not require that the underlying nonlinear mapping be differentiable. The Unscented Transformation at the heart of this work is both more general and better at capturing nonlinear transformation than the EKF's Jacobian approximations, and is generally a drop-in replacement for the EKF. Although it can significantly improve the linear approximation to nonlinear processes, cumulative linearization errors arising from significant excursions prior to landmark revisitation in large-scale SLAM can still be problematic.

Computational Complexity

In EKF-SLAM, the computational complexity can be thought of as arising from the algorithm being rather obsessive about keeping information about relationships between landmarks. As the EKF solution runs, all landmarks become correlated through the maintained covariance matrix; this means that any observation propagates effects through every landmark interrelationship in the entire map.

The key observation being exploited by most modern methods is that this complete cross-landmark information does not actually contribute to the accuracy of the map. In other words, while we *can* track the relationship between features which are far, far apart, the theoretical gain in accuracy is tiny (or, in the case of linearized approaches, even negative), and the computational cost is extremely high, so we *shouldn't*.

Sparse Extended Information Filters

One way to approach this issue is to look to the dual, equivalent formulation of the EKF which relies on the inverse covariance matrix and a projected state estimate. In this form of the filter, uncertainty is represented by an inverse covariance matrix, usually called an *information matrix*[1]. Mathematically, the filter is equivalent in operation to a Kalman filter. Computationally, it has a number of advantages: the information matrix directly represents landmark and positional relationships, making removal of tenuous relationships a relatively straightforward task. Additionally, sparse matrix methods can be used to greatly reduce computational load. This approach, generally known as *Sparse Extended Information Filtering* (SEIF), has been explored in a number of works, notably [Liu and Thrun, 2003], in which it was proposed, and [Thrun *et al.*, 2004], which showed that, under some constraints, it is possible to run the SEIF algorithm in constant time. [Eustice *et al.*, 2005] explored the basis for and consequences of the sparsification used in these techniques. These techniques do still share the linearization problems of their EKF dual. Additionally, [Walter *et al.*, 2007] showed that the naïve method of sparsification of the information matrix is inherently "overconfident", reducing error estimates inappropriately, and provided an alternative sparsification method which can be shown to be consistently conservative.

These methods retain the problem of linear approximation becoming very inaccurate as loop size increases; incorporation of GPS can keep such approximations bounded to allow for large scale usage.

[1] This is not to be confused with the information theory structure of the same name.

Hierarchical Methods

There is also a large body of work which attempts to address both scalability and linearization errors through the use of imposed hierarchy, or fused topological-metric maps [Bulata and Devy, 1996] [Bosse *et al.*, 2004] [Guivant and Nebot, 2001]. Typically in this work, a linear filter is used to build a map of a relatively small area. This map then becomes the building block of a higher level map, which relates the positions of the submaps. Solving for submap relationships becomes a chained optimization problem, but linearization of the individual links allows for more flexible representation of non-linear relationships. Scalability is improved by limiting the pathological algorithms to small sets of landmarks within a small number of submaps at a time. However, loop closure and cross-map boundaries incur an additional computational cost. Related is the work of [Williams, 2001], in which computational efficiency is much improved by maintaining a local submap that is synchronized to a global map at longer intervals.

Of particular interest is the work of Tim Bailey in [Bailey, 2002]. This work deals explicitly with scaling SLAM up in outdoor environments within a hierarchical EKF framework. In addition to introducing an alternative submap formulation to limit computational complexity and contemplating long-term feature management, the work also delves briefly into augmenting GPS with what is termed *Partial SLAM*. This is an interesting first step in the direction explored here. However, in this work, the SLAM algorithm used simply "forgets" about any landmarks which are not currently in view, turning the SLAM algorithm into a local estimator akin to an IMU or odometry. This work, in contrast, takes full advantage of "true" SLAM.

Also related is the recent work of [Paz *et al.*, 2007], in which a divide-and-conquer approach is used to reduce the algorithmic complexity of computing exact filter updates.

Non-recursive Methods

Another major development of late is the cross-pollination of the SLAM and Structure From Motion (SFM) fields in robotics. In the computer vision community, SFM, which has been studied for much longer than SLAM, can be reformulated as a special case of camera-based SLAM. In the SFM literature, *bundle adjustment*, or *sparse bundle adjustment*,

serve the same general purpose as loop closure in the SLAM literature.

From this departure point, it's possible to look at loop closure as a global optimization problem, with a dual and equivalent formulation in Graph Theory. Instead of being recursive, this formulation is global in nature; the entire trajectory is kept and re-optimized at each time step. The solutions are found via linearization of the problem, but the linearizations do not iteratively accumulate error – they are recalculated at each time step. This has the additional advantage that the history allows data association decisions to be reevaluated. Incorrect loop closures can, in theory, be undone; in recursive formulations, the data do not exist to reevaluate such decisions, making incorrect data association a serious issue.

In GraphSLAM [Thrun and Montemerlo, 2006], a graph of robot poses and landmark observations is obtained. To obtain a global map, landmark observations from multiple poses are refactored into constraints between those poses. GraphSLAM chooses to explicitly marginalize out landmarks to improve the pose estimates. The resulting graph and associated matrix are, under most conditions, very sparse and can be efficiently optimized. Of particular relevance to this dissertation is the addition of the capability of using GPS readings to improve the resulting solutions. The implicit extremely optimistic white-noise assumption in this work is mitigated by only allowing inputs into the filter to be extremely "occasional", allowing process noise to dominate the system in between readings.

In [Paskin, 2002], SLAM is approached as a graph problem where nodes group "related" landmarks and edges contain information matrices and associated vectors. In the natural implementation of such a scheme, the graph would quickly become complete, removing any advantage over the linear filter approaches, but the work cleverly develops an efficient maximum-likelihood edge-removal algorithm to remove weak links between nodes without introducing overconfidence. Using the generated map is akin to querying an inference network. Similar work was done simultaneously in [Frese, 2004]. The latter work has been extended in [Frese, 2007] to very large numbers of landmarks, though the source of data association is oracular and there is no provision made for recovering from association errors.

Closely related is the Square Root Smoothing and Mapping (colloquially known as

$\sqrt{SAM}$) work in [Dellaert and Kaess, 2006]. In this work, it is noted that an information matrix formulation in which past robot motion states are not marginalized out remains naturally sparse. This formulation naturally causes a much faster growth in the size of the information matrix, but the maintained sparseness gives rise to very efficient least square solutions, to the extent that much larger systems can be optimized than can be reasonably handled by classical naïve filter formulations.

The flexibility of this approach comes at the cost of long-term computational efficiency; the computational and memory cost of resolving a map grows without bound. In the typical case, sparsity of landmark observations allows for a computation cost linear in the number of observations and the robot trajectory length, with a very low constant. The authors argue that such non-recursive methods are sufficiently efficient to bootstrap a localization system, postulating that if the system were to run to the point that the lack of recursion is a problem, it would be reasonable to "finalize" the map, converting the problem into one of localization using a fixed map.

More recent work has addressed this weakness in two ways. [Kaess *et al.*, 2007] and [Kaess, 2008] present a method of incrementally factoring the information matrix, greatly reducing the computational load, but not the memory requirements. [Ni *et al.*, 2007] presents an alternative approach using submaps with independent coordinate frames which are periodically merged back into a global estimate. This divided approach does not reduce overall storage requirements, but does bound core memory requirements in a promising way.

The scalable implementation presented in this work builds on the foundation of recent SAM research.

Particle Filters

Other recent work in SLAM has used particle filters. Particle filters tend to be extremely computationally intensive to run, and the simple implementation would require $O(n)$ storage per landmark to sample accurately from the distribution of possible locations of that landmark, where n is the number of particles used, and tends to be quite high to make divergence of the filter unlikely. Recent work made the key observation that, given

the robot pose, landmark positions in the SLAM problem are independent, leading to a representation which factorizes landmark posteriors into independent estimations given samples of the pose posterior. Within the framework of a particle filter, this means that each landmark can be represented by one $n \times n$ Kalman filter per particle, where n is the dimensionality of the mapping space. Loop closure simply becomes a matter of discarding those particles (and associated landmark maps) which are inconsistent with the current sensing data [Montemerlo *et al.*, 2002] [Montemerlo, 2003].

These methods are particularly compelling in that they provide one of the most elegant solutions to coping with potential misassociation errors. Sample-driven methods of this style have provided some of the most interesting results in large-scale SLAM to date.

However, sampling methods can run into difficulties accurately approximating distributions in high-dimensional spaces. As will be seen, integrating GPS into the system directly increases the dimensionality of the problem significantly. Hybrid particle-distribution methods, such as [Montemerlo *et al.*, 2003] may provide a suitable foundation for GPS integration, but that possibility is not explored in this work.

Landmark-Free Methods

Not all SLAM methods represent maps as sets of landmarks. In [Lu and Milios, 1997], a method was proposed to align raw laser scans of an indoor environment in a globally consistent manner. This work was extended by [Gutmann and Konolige, 2000] by improving the computational efficiency and implementing a computer-vision inspired method for detecting closure of large loops.

A landmark-free particle method, in which the state of the world is maintained as a modified occupancy grid map, has been developed in a series of papers [Eliazar and Parr, 2003] [Eliazar and Parr, 2004] [Eliazar and Parr, 2005]. This approach incorporates both the full map posterior and the pose posterior into a unified particle filter. Although the theoretical complexity of the most recent iteration of this approach is linear in the number of particles and the size of a single observation, the number of particles needed for convergence in nontrivial cases is extremely large, limiting utility in large-

scale environments.

2.3 Other Work of Interest

The idea of using GPS information in SLAM systems has come up in the literature from time to time. Of note, the work in [Lee *et al.*, 2007] treats GPS and digital road map information as prior constraints to aid their SLAM algorithm in data association and loop closure. In contrast, this work integrates GPS into the SLAM system directly, providing for better estimates and obviating the need for an artificial separation between a priori knowledge and SLAM inputs.

There are several ongoing efforts to generate 3-D maps of urban areas with varying goals and methodology. Microsoft and Google, among other industry players, are known to be developing automated collection systems, but the work is not currently being published in the publicly available literature.

There has also been some work which uses aerial or satellite imagery to provide post-processing global constraints to a ground-based system. This allows the the construction of very large, visually appealing maps of urban areas, but does not provide probabilistic bounds on the accuracy of those maps for localization purposes [Früh and Zakhor, 2003].

Chapter 3

GPS Errors and Mitigation Strategies

3.1 Introduction

"Global Positioning System" (GPS) is a generic name for a wide array of technologies and techniques used to generate localization information. Multiple systems which are accurately described as GPS may differ in almost every implementation detail, from the satellites used to how ranges are measured to how time is measured and propagated. Additionally, the uses of GPS span an extraordinary range, from navigation to surveying to precise time synchronization.

To effectively integrate GPS into a larger system, an understanding of the sources, magnitudes, and characteristics of systemic errors is critical. In this chapter, we will examine the magnitude and characteristics of GPS errors. We will take advantage of the large body of published reference station data to analyze error sources individually where observability permits. While we will devote some time to a basic overview of core concepts, it is not our intention to give an in-depth exposition on how GPS works; most of the various systems involved are already well-documented. For deeper general GPS information see [Kaplan, 1996], [Parkinson *et al.*, 1996], and [Arinc, 2000] among others.

3.2 Constellations

Without qualifications, GPS usually refers to the United States' NAVSTAR system, which is, at this time, the only fully functional global navigation satellite system system (GNSS). The Russian *GLONASS* system, which achieved global coverage in the 1990's, has deteriorated due to a lack of satellite replenishment, but is now being restored by a joint effort of the Russian and Indian governments. Additionally, China is developing a new system called *Compass*, while the European Union is planning *Galileo*; both *Compass* and *Galileo* are launching test vehicles with the goal of having an operational system in the next decade.

All four systems operate or are anticipated to operate on similar principles. With appropriate hardware, it will be possible to improve accuracy and reliability by using multiple systems. However, because the newer constellations are not yet operational, such hybrid configurations are left as future work. References to GPS within this document should be understood to be limited to the NAVSTAR constellation.

Basic Functionality

GPS is enabled by a constellation of satellites in medium earth orbit. The orbits are designed to provide high availability of 5 or more satellites above the horizon at nonextreme latitudes. Each satellite carries highly accurate cesium and/or rubidium clocks. At present, satellites broadcast on two frequencies: 1.57542 GHz (L1) and 1.2276 GHz (L2). A coarse acquisition (C/A) 1.023 MHz repeating pseudorandom code modulates the L1 frequency, and an encrypted precise (P) 10.23 MHz code modulates both the L1 and L2 frequencies. Although the precise code is not available to civilian receivers, techniques exist to utilize the second band without access to this code. These will be briefly discussed later.

Each satellite also broadcasts Keplerian orbital parameters, which make it possible to calculate the position of the satellite with a high degree of precision at a given time.

Given a sufficiently accurate and synchronized clock in the GPS receiver, time of signal flight from the satellite to receiver could be measured and multiplied by the speed of light to obtain a range from receiver to satellite. Each satellite would provide

a range measurement of the form:

$$\rho^{[i]} = \sqrt{(r_x - s_x^{[i]})^2 + (r_y - s_y^{[i]})^2 + (r_z - s_z^{[i]})^2}$$

where $r = (r_x, r_y, r_z)^T$ is the position of the receiver and $s^{[i]} = \left(s_x^{[i]}, s_y^{[i]}, s_z^{[i]}\right)^T$ is the position of satellite i at the time of transmission. Three such measurements from a non-degenerate geometry would be required to solve for the user's position in three dimensions.

In practice, receivers rarely contain oscillators of sufficient precision to derive range directly. Instead, the offset of the receiver clock is represented as an additional unknown for which a solution is found. Each satellite then provides a measurement known as a *pseudorange*, which is of the form[1]:

$$\rho^{[i]} = \sqrt{(r_x - s_x^{[i]})^2 + (r_y - s_y^{[i]})^2 + (r_z - s_z^{[i]})^2} + c\delta t$$

where δt is the offset of the receiver's clock from the "true" system time.

The addition of δt as an unknown in the system brings the number of satellites required to calculate a solution to four, though additional satellites are typically used to improve the accuracy of a solution through standard least-squares techniques.[2]

Pseudoranges lead to positional fixes. In addition to pseudorange information, GPS receivers commonly measure the Doppler shift of satellites' signals with respect to the carrier frequency.

With the frequency known, the shift is a measurement of the dot product of the relative velocity of the satellite and receiver with the unit-normalized vector which is the direction of the satellite:

$$d^{[i]} = \frac{s^{[i]} - r}{|s^{[i]} - r|} \cdot (\dot{s}^{[i]} - \dot{r}) + v$$

[1] C/A pseudorange is a derived value. The C/A signal repeats every 1 ms, which means the raw data we have for the pseudorange is the true value modulo .001 light-seconds. Resolving this ambiguity requires either searching for positional solutions with a "reasonable" amount of residual error or having some very rough (˜300km) estimate of the position of the receiver to start with. This presents no particular difficulty, and so this ambiguity will be ignored.

[2] See [Kaplan, 1996] pp. 43–47

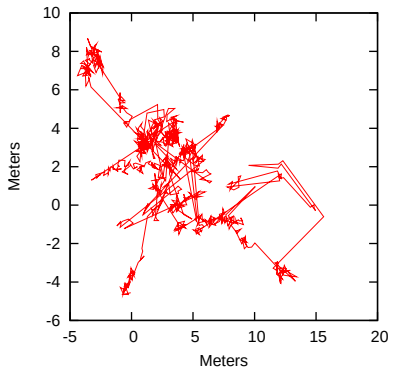

Figure 3.3.1: East-North-Up coordinate frame fixes every 30 seconds over an 8-hour window from using a base station with a fixed antenna. Modeling the outputs of this system is a decidedly nontrivial task.

where $d^{[i]}$ is the measured Doppler shift from satellite i, $s^{[i]}$ is the position of the satellite, r is the position of the receiver, and v is noise.

3.3 Error Sources and Characteristics

GPS is complex; errors arise from a wide variety of sources with differing dependencies and characteristics. This complex error model is the largest barrier to accurate integration of GPS with other systems.

Using position fixes directly is problematic due to the noise characteristics of the processed receiver output. In addition to being highly non-white, the noise in the processed output is dependent on the set of satellites used in calculations. Efforts to formally model positional offsets as drifts are hampered by offsets which suffer "jumps" at unpredictable times when the constellation of used satellites changes.

Figure 3.3 illustrates the complicated nature of using the position as a "black box" output of a GPS subsystem. Instead of attempting to use positional fixes directly, we need to move to a tighter level of integration which exposes the underlying sources of error directly.

Satellite Orbits and Clocks

There are two common ways to calculate the position of a satellite. When in an online application, satellite positions are calculated using orbital parameters which are collectively referred to as the ephemeris. This ephemeris is broadcast by the satellite itself every 30 seconds. Ephemeris parameters are predictive, rather than measured, values, and are generated using higher order models and long-term ground station observations of satellite positions. Predictions from given set of ephemeris parameters become increasingly inaccurate over time, and become unacceptably large in a matter of hours. Error bounds for ephemerides have been improving over time due to improvements in measurement and modeling. [Crum *et al.*, 1997]

The rubidium and/or cesium beam clocks carried by satellites are highly stable by most metrics, but still may drift tens of nanoseconds per day. Thus, the satellite clock offset must also be taken into account when calculating pseudoranges. The broadcast ephemeris includes a second order polynomial model of the clock offset. However, the residual error after correcting with the ephemeris model is still significant.

If online operation is not necessary, satellite locations can be determined by fitting both past and future satellite positional readings to a long-term orbital model. The National Geological Survey (NGS) makes these "precise" orbits available for download. The precise orbits carry a stated positional 1-σ error of approximately 2.5 cm, and a 1-σ clock error of 60 ps, or approximately 2 cm. [Kouba, 2009]

Using the precise orbits as a baseline, we can estimate the broadcast ephemeris positional and clock errors. Figure 2.1 shows these offsets for a single satellite over a 24-hour period. The total offset is the magnitude of the full positional error. Some of this error—any error which maintains the range between satellite and receiver—does not affect the accuracy of a receiver's readings. The radial offset shows the total broadcast offset dotted with a unit vector pointing towards the center of the earth, and represents an approximation to the error that a receiver would perceive. The clock offset is also shown.

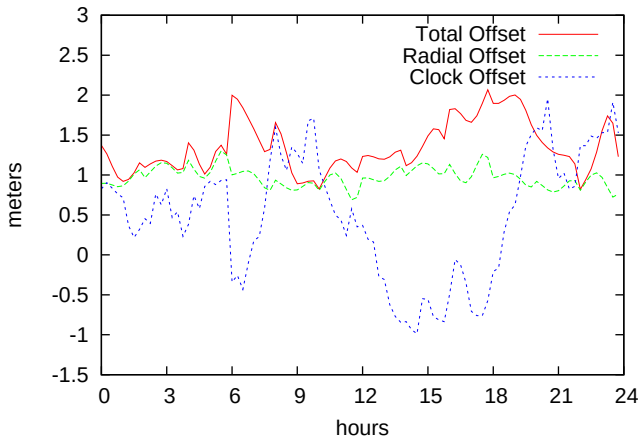

Figure 3.3.2: Offset between ephemeris-derived and precise orbits and clock offsets for a single satellite over a 24-hour period

Atmospheric Effects

During the roughly .15 light-second journey from a satellite to a receiver, the signal is refracted by the atmosphere. The dominant source of this refraction is the charged particles of the ionosphere.

Refractive angles are frequency-dependent. A receiver capable of decrypting and tracking the P-codes on the L1 and L2 frequencies can use the difference in measurements to estimate refractive effects and compensate.[3] Unfortunately, decrypting the P-codes is not an option for civilian users. Without access to the unencrypted P-codes, it is still possible to estimate the ionospheric delay using a dual-frequency receiver. The encrypted P codes on the L1 and L2 frequencies are identical. It is therefore theoretically possible (albeit technically difficult) to determine the propagation delay offset between L2 and L1 by using the correlation of the (unknown, but identical) encrypted P codes. Implementations of this approach are called codeless tracking.

In practice, the data derived from codeless tracking are of greatly diminished quality. Figure 3.3.3 uses data gathered from several codeless tracking reference stations to estimate ionospheric delay over a long period of time. A true dual-frequency receiver

[3]See the GPS ICD section 20.3.3.3.3 for the first-order correction terms.

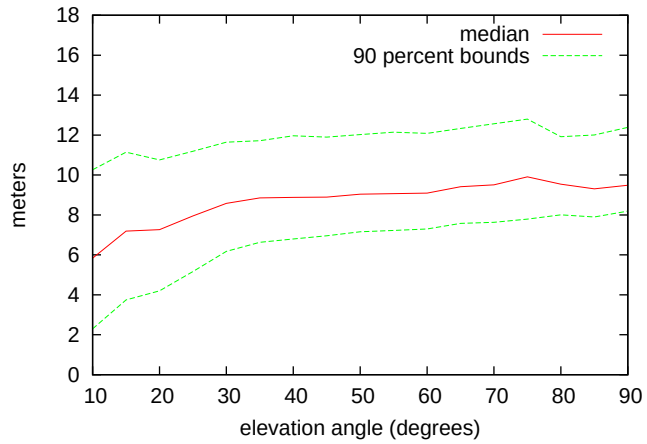

Figure 3.3.3: Measured ionospheric disturbance vs. satellite elevation angle measured from several codeless-tracking base stations.

shows a significant increase in estimated refractive error as the elevation angle of the satellite decreases. As can be seen in the plot, codeless trackers have a tendency towards increased uncertainty at low elevation angles, but do not show an increase in median refractive error.

In addition to refraction, the signal is slowed by passing through atmospheric gases, inducing a perceived delay in signal arrival by the receiver. The troposphere contains the bulk of the atmospheric mass. The amount of delay in a signal depends primarily on atmospheric thickness (approximated by latitude), local temperature, atmospheric pressure, and humidity. The largest delays occur when the transmitting satellite is low on the horizon, causing signals to pass through larger portions of the troposphere. Global models for the state of the troposphere parameterized by altitude, latitude, time of day, and season have been developed, such as [Herring and Shimada, 2001]. Local models using weather information can also be used to estimate tropospheric effects.

Selective Availability

In the past, the United States government intentionally dithered the broadcast signal, adding an additional, slowly varying error on the order of 150 meters per satellite. This

system degradation was called "selective availability", and was disabled in 2000. There do not appear to be plans to reactivate selective availability, but the capability remains in the system.

Velocity

In contrast with receiver position fixes, velocity fixes are much more straightforward to use in a probabilistic model.

Consider the doppler measurement from a single satellite, as discussed in 3.2:

$$d^{[i]} = \frac{s^{[i]} - r}{|s^{[i]} - r|} \cdot (\dot{s}^{[i]} - \dot{r}) + v \tag{3.3.1}$$

Consider the significant sources of colored noise in a GPS system: ionospheric refraction, receiver clock drift, multipath, tropospheric delay, and satellite ephemeris error. Ionospheric refraction and ephemeris error are similar in that both are the result of the signal taking a different path from satellite to receiver than is accounted for in the solution. In particular, the satellite and receiver positions are only used to determine a signal direction. Given that the satellite is, at a minimum, thousands of kilometers away from the receiver, small errors in the calculated position of the satellite (or receiver) tend to be negligible for the purposes of calculating velocity fixes. Tropospheric delay affects the signal time of flight, but not the perceived frequency, and so can be neglected.

Multipath errors are more worrisome, as a reflected signal will have a perceived Doppler which is entirely incorrect. This can be mitigated through use of an appropriate antenna, and many modern receivers actively discard questionable Doppler readings when the system is overconstrained.

In the receiver, unmodeled clock drift rate errors would lead to biases in Doppler measurements, but over short windows the second order drift of commercial oscillators, being driven primarily by temperature shifts, varies quite slowly. This allows us to actively estimate the drift rate of the receiver with sufficient accuracy as to make biases introduced into the Doppler measurement negligible.

The uncertainty of a velocity fix is not readily available, and is largely dependent on the constellation of satellites used to generate the fix. However, in applications needing

a simple implementation, a conservative, diagonal Gaussian is a reasonable choice to incorporate the information into a consistent probabilistic model.

Satellite-Associated Errors

In a high-accuracy system, the errors in the satellite position parameters cannot be neglected. The magnitude of this error has decreased over time due to improvements in the underlying dynamic models and measurement, and varies from satellite to satellite depending on the vehicle capabilities. According to [Warren and Raquet, 2003] the combined RMS positional and clock offset error from ephemeris estimates is approximately .8 m. Velocity errors are several orders of magnitude smaller.

If immediate results are not required, these errors can be greatly diminished through post-processing. Additional ground station observations can be used to generate more accurate orbital estimates over a longer period. There is a basic latency-accuracy tradeoff in satellite parameters; the initial ephemeris predictions are immediately available. At the limit, a "final" orbital solution for each satellite, averaging readings from multiple stations over a long period, is published by the International GNSS service with a latency of 14 days. These final orbital positional solutions carry a RMS error of under 1 cm.

3.4 Differential Techniques

There are a number of techniques used to improve the accuracy of a GPS system though removal of some of the significant sources of error.

Local Area Differential GPS

Many of the significant error sources, including atmospheric effects and ephemeris error, are highly dependent on the location of the receiver. Local area differential GPS uses a second GPS receiver in a known location close to the primary receiver. The second receiver uses its known location to generate a correction to the satellites in view.

If

$$r_{b,i} = \sqrt{(x_b - x_i)^2 + (y_b - y_i)^2 + (z_b - z_i)^2}$$

is the true range from satellite i to the base station, and can be calculated because (x_b, y_b, z_b) is known, then

$$\rho_{b,i} = r_{b,i} + \epsilon_{b,geo} + \epsilon_{b,local} + c\delta t_b$$

is the pseudorange to satellite i measured by the base station. Here we group the many spatially dependent sources of error, such as atmospheric effects and satellite ephemeris error $\epsilon_{b,geo}$, and other sources of error, such as receiver noise, in $\epsilon_{b,local}$. Simultaneously,

$$\rho_{u,i} = r_{u,i} + \epsilon_{u,geo} + \epsilon_{u,local} + c\delta t_u$$

is the pseudorange to satellite i measured by the user. Using the two pseudoranges, we find

$$\rho_{u,i} - (\rho_{b,i} - r_{b,i}) = r_{u,i} + \epsilon_{u,geo} - \epsilon_{b,geo} + \epsilon_{u,local} - \epsilon_{b,local} + c\delta t_u - c\delta t_b \tag{3.4.1}$$

Because, assuming that the distance between user and base station is small,

$$\epsilon_{u,geo} \approx \epsilon_{b,geo}$$

we find that

$$\rho_{u,i} - (\rho_{b,i} - r_{b,i}) \approx r_{u,i} + \epsilon_{u,local} - \epsilon_{b,local} + c\delta t_u - c\delta t_b$$

In general, $\epsilon_{geo} \gg \epsilon_{local}$. While ϵ_{geo}, being caused by conditions which slowly vary over time, tends to be extremely highly autocorrelated, ϵ_{local} removes much of the autocorrelated error.

Wide Area Differential GPS

Wide area differential systems attempt to estimate individual error terms over a large area through interpolation of data from multiple base stations that can be significantly further from the user. Corrections are broadcast via satellite to the receiver. WADGPS systems can, in the best case, bring the single-reading variance down to 1-2 meters, but

error autocorrelation tends to remain high. Various WADGPS systems, both commercial and governmental, exist.

Carrier Phase GPS

In civilian GPS receivers, range is determined using a phase offset of a 1.023 MHz C/A signal modulating a 1575.42 MHz carrier signal. When using a LADGPS system, accuracy can be further improved by calculating the phase offset of the higher frequency carrier signal, and using a differential correction from a fixed base station also tracking carrier phase. This is a tricky proposition; the C/A signal is designed to have a low autocorrelation at incorrect phase offsets, while the carrier signal has a strong carrier-frequency ambiguity which must be resolved. Various techniques exist to lock onto the carrier phase given communication between receiver and base station and an uninterrupted line of sight to the satellite. However, such locks are fragile, depending on continual signal reception; interruption of the line of sight requires a costly reaquisition of the lock. When Carrier Phase GPS is available, it can drive extremely accurate readings. It is very useful for aviation and other work in open spaces, but ill-suited for urban environments, in which satellite line-of-sight is frequently interrupted.

3.5 Nondifferential Error Characterization

Unfortunately, many of the error sources in GPS have a high degree of autocorrelation, making them unsuitable as direct inputs into systems that require noise sources to be reasonably white. Figure 3.5.1 shows the raw error in clock-bias-adjusted pseudorange for a single satellite over the course of several hours.

The sources of error which are most highly autocorrelated are related to the location of the receiver. When using an LADGPS system, the ϵ_{local} noise is "close enough" to white that it is reasonable to incorporate pseudorange readings into filters as a direct observation of the vehicle position and receiver clock bias:

$$\rho_k^i = h(x_k) + \underbrace{v_k}_{\sim N(0,R_k)}$$

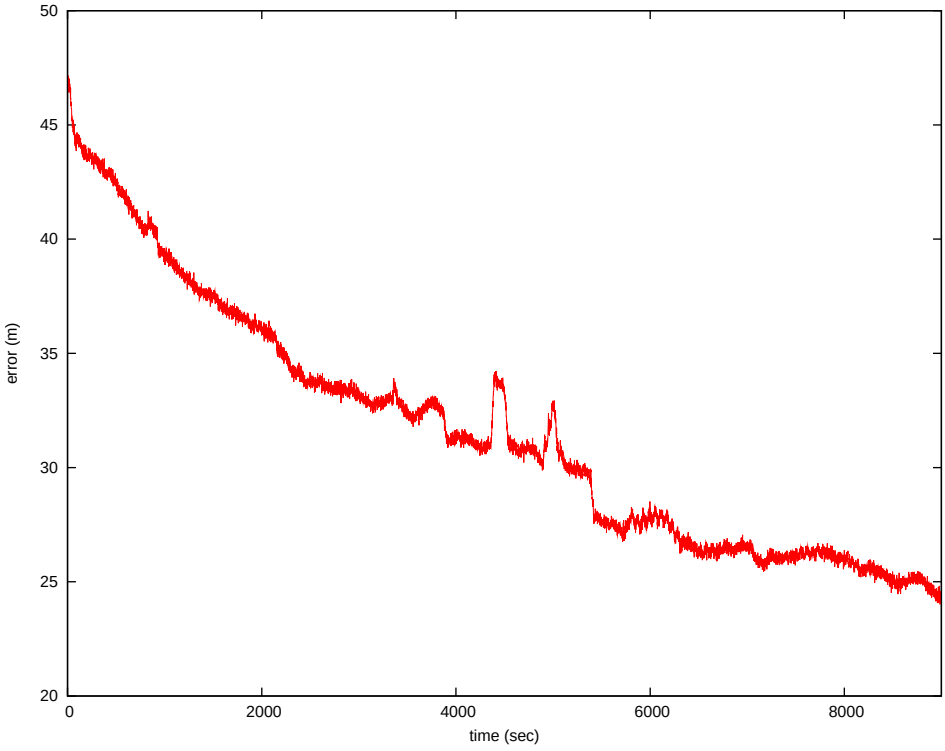

Figure 3.5.1: Raw error of a single satellite range measurement

Furthermore, due to the large distance of the satellite from the receiver, as long as we have even a very approximate location estimate, discarding the nonlinear portion of h results in extremely minimal distortion of the resulting function:

$$\rho_k^i \approx Hx_k + \underbrace{v_k}_{\sim N(0,R_k)}$$

Coverage with even the relatively modest requirements of an LADGPS system is intermittent in urban areas; most of the time some mix of differentially and either non-differentially or wide-area-differentially corrected signals are available. A robust system needs to be able to account explicitly for the autocorrelation characteristics of the various types of errors in such systems. Doing so requires explicitly incorporating the current error into the system model.[4]

To find the desired characteristics of the system, we look to the autocorrelation[5] of the error, and find that the Power Spectrum Density (PSD) of the filter is well modeled by an exponential decay. The PSD and fit function are shown in figure 3.5.2.

Since pseudorange measurements are processed at regular intervals, we'll use a difference model for the system. Consider ϵ_k, the error of a pseudorange measurement at time k. Given the autocorrelation function characteristics, it seems reasonable to model the system as the Gauss-Markov process:

$$\epsilon_{k+1} = A\epsilon_k + v_k$$

with autocorrelation function

$$R_v(\tau) = 30.5e^{-.0005|\tau|}$$

[4]This following derivation skips many details, but is similar to derivations which can be found in [Maybeck, 1979], [Bar-Shalom and Li, 1993], and [Gelb, 1974].

[5]Autocorrelation here is used in the Signal Processing (e.g. unnormalized) sense.

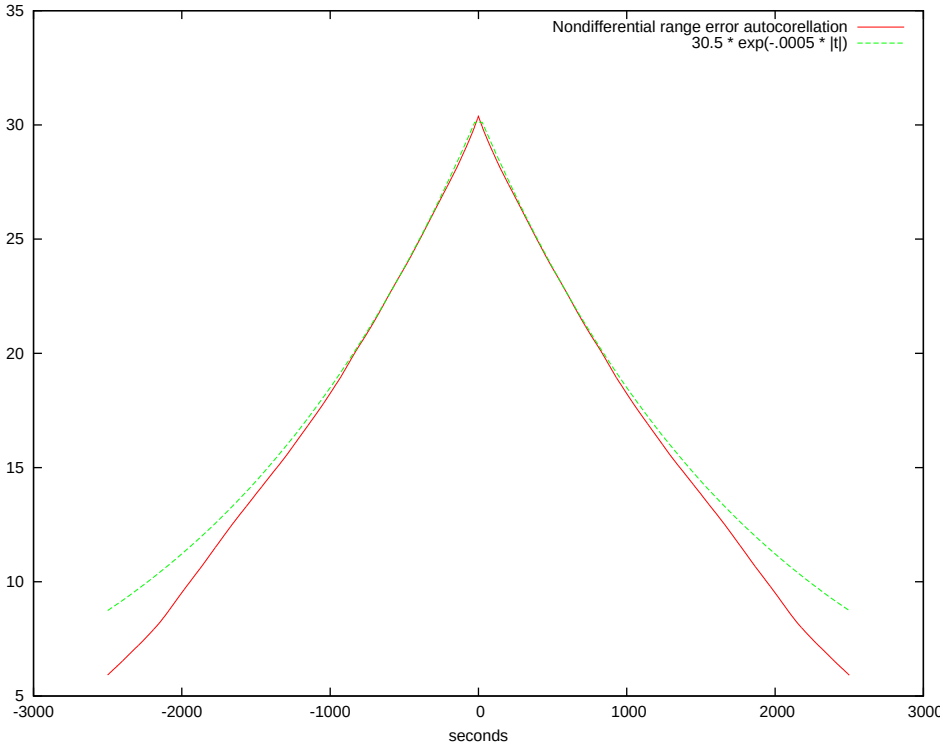

Figure 3.5.2: Autocorrelation of raw satellite pseudorange error

The power spectrum density of this process can be factored:

$$\begin{aligned} PSD_v(j\omega) &= \int_{-\infty}^{-\infty} 30.5e^{-.0005|\tau|}e^{-j\omega\tau}d\tau \\ &= \int_0^{\infty} 30.5e^{-(.0005+j\omega)\tau}d\tau + \int_{-\infty}^{0} 30.5e^{(.0005-j\omega)\tau}d\tau \\ &= \frac{30.5}{.0005+j\omega} + \frac{30.5}{.0005-j\omega} \\ &= \frac{.0305}{(.0005)^2+\omega^2} \\ &= \underbrace{.0305}_{PSD_{ww}(j\omega)} \underbrace{\left(\frac{1}{.0005+j\omega}\right)}_{H(j\omega)} \underbrace{\left(\frac{1}{.0005-j\omega}\right)}_{H^*(j\omega)} \end{aligned} \tag{3.5.1}$$

From linear systems analysis, we recognize $H(j\omega)$ from (3.5.1) as a stable causal system, which leads to this state model:

$$\epsilon_{k+1} = e^{-.0005\tau}\epsilon_k + \underbrace{v_k}_{\sim N(0,.0305\tau)} \tag{3.5.2}$$

This model implies the addition of a bias variable for each satellite to the states being estimated by the underlying system, and modification of the observation model to incorporate this "hidden" state variable.

3.6 Conclusion

GPS provides an extremely useful source of information for outdoor mapping. However, if we wish to use this information in mathematically rigorous models, we incur a significant cost in additional complexity. In this chapter, we have outlined the major sources of errors corrupting GPS receiver observations, and presented a simple-to-implement model which we believe is both sufficiently detailed to accurately represent GPS information in a fusion system. In the next chapter, we will derive and test a sample realization of the model in a fused GPS-SLAM system.

Chapter 4

Sample Extended Kalman Filter Implementation and Analysis

4.1 Introduction

Having looked at the characteristics and proposed some ways to model the characteristics of GPS signals, we now present an model implementation using an Extended Kalman Filter (EKF).

This choice of model may be surprising at first; the EKF has been improved upon in virtually every way by a wide variety of algorithms, and it has been some time since it could be termed state-of-the-art. However, the EKF is arguably the most well-understood (and easiest to understand) approach to the SLAM problem. This makes it ideal for our purpose—demonstration and analysis—and so we set the practical matter of scalability aside for a moment. We will present a solution more suitable for practical implementations in chapter 5.

The EKF model will provide us a platform on which we can develop some intuition about how we can expect a GPS-augmented SLAM system to behave. We will also leverage the simulation as a source of ground-truth for validating the correctness of the approach.

4.2 Simulation Model

Consider a 4-wheel Ackermann steered robot, with control parameterized as a commanded velocity and steering angle, pose parameterized by an $s := (x, y, \phi)$ tuple, and landmark positions parameterized as (x, y, θ) three-dimensional locations. In EKF-SLAM, estimates of the current pose of the robot and positions of all the landmarks the robot knows about are comingled in a single state vector:

$$x_t = \begin{pmatrix} s_t \\ \Theta \end{pmatrix} = \begin{pmatrix} s_{x,t} \\ s_{y,t} \\ s_{\phi,t} \\ \theta_{x,1} \\ \theta_{y,1} \\ \vdots \\ \theta_{x,n} \\ \theta_{y,n} \end{pmatrix} \tag{4.2.1}$$

The process model for our system is:

$$\begin{aligned} x_{t+1} &= f(x_t, \underbrace{\begin{pmatrix} u_{v,t} \\ u_{\gamma,t} \end{pmatrix}}_{u_t}, \underbrace{\begin{pmatrix} w_{v,t} \\ w_{\gamma,t} \end{pmatrix}}_{w_t \sim N(0,Q_t)}) \\ &= x_t + \begin{pmatrix} (w_{v,t} + u_{v,t}) \cos(w_{\gamma,t} + u_{\gamma,t} + s_{\phi,t})\delta t \\ (w_{v,t} + u_{v,t}) \sin(w_{\gamma,t} + u_{\gamma,t} + s_{\phi,t})\delta t \\ \frac{(w_{v,t}+u_{v,t}) \sin(w_{\gamma,t}+u_{\gamma,t})}{k_{wb}} \delta t \\ 0 \\ \vdots \\ 0 \end{pmatrix} \end{aligned}$$

where k_{wb} is the (constant) wheelbase of the robot, $u_{v,t}$ and $u_{\gamma,t}$ are the commanded speed and steering angle of the vehicle at time t, respectively, and w_t is white noise.

The vehicle is equipped with a sensor that supplies vehicle-relative ranges and bear-

ings to landmarks, again corrupted by white noise:

$$\begin{aligned} z_t &:= \begin{pmatrix} z_{d,t} \\ z_{\psi,t} \end{pmatrix} \\ &= h(x_t, \underbrace{\begin{pmatrix} v_{d,t} \\ v_{\psi,t} \end{pmatrix}}_{v_t \sim N(0,R_t)}) \\ &= \begin{pmatrix} \sqrt{(s_{x,t} - \theta_{x,i})^2 + (s_{y,t} - \theta_{y,i})^2} \\ tan^{-1}(\frac{s_{y,t} - \theta_{y,i}}{s_{x,t} - \theta_{x,i}}) \end{pmatrix} + v_t \end{aligned}$$

where i is the identifier of the landmark being observed.[1]

Using this process model and observation model, our EKF prediction step is:

$$\begin{aligned} \hat{x}_{t+1} &= f(\hat{x}_t, \hat{u}_t, 0) \\ P_{t+1} &= P_t + W_t Q_t W_t^T \end{aligned} \tag{4.2.2}$$

and our update step is:

$$\begin{aligned} K_t &:= P_t^- H_t^T (H_t P_t^- H_t^T + R_t)^{-1} \\ \hat{x}_t &= \hat{x}_t^- + K_t(z_t - h(\hat{x}_t^-, 0)) \\ P_t &= (I - K_t H_t) P_t^- \end{aligned} \tag{4.2.3}$$

where H_t is the Jacobian matrix[2] of h with respect to $\hat{x}_t$:

$$\begin{aligned} \hat{d}_t &:= (\hat{s}_{x,t} - \hat{\theta}_{x,i})^2 + (\hat{s}_{y,t} - \hat{\theta}_{y,i})^2 \\ H_t &= \begin{pmatrix} \frac{\hat{\theta}_{x,i} - \hat{s}_{x,t}}{\hat{d}_t} & \frac{\hat{\theta}_{y,i} - \hat{s}_{y,t}}{\hat{d}_t} & 0 & \cdots & \frac{\hat{s}_{x,t} - \hat{\theta}_{x,i}}{\hat{d}_t} & \frac{\hat{s}_{y,t} - \hat{\theta}_{y,i}}{\hat{d}_t} & \cdots \\ \frac{\hat{s}_{y,t} - \hat{\theta}_{y,i}}{(\hat{d}_t)^2} & \frac{\hat{\theta}_{x,i} - \hat{s}_{x,t}}{(\hat{d}_t)^2} & -1 & \cdots & \frac{\hat{\theta}_{y,i} - \hat{s}_{y,t}}{(\hat{d}_t)^2} & \frac{\hat{s}_{x,t} - \hat{\theta}_{x,i}}{(\hat{d}_t)^2} & \cdots \end{pmatrix} \end{aligned}$$

[1] The data associations (n^t in our SLAM notation) come from an oracle for this example; the vehicle always knows the true mapping from observations to landmarks.

[2] Although this example uses Jacobian matrices as approximate linearizations of the underlying uncertainty, the Unscented Transform [Julier and Uhlmann, 1997] is almost certainly a better method if one is optimizing for anything other than clarity of derivation.

and W_t is the Jacobian matrix of f with respect to w_t:

$$W_t = \begin{pmatrix} \cos(u_{\gamma,t} + \hat{s}_{\phi,t})\delta t & -u_{v,t}\sin(u_{\gamma,t} + \hat{s}_{\phi,t})\delta t \\ \sin(u_{\gamma,t} + \hat{s}_{\phi,t})\delta t & -u_{v,t}\cos(u_{\gamma,t} + \hat{s}_{\phi,t})\delta t \\ \frac{\sin(u_{\gamma,t})}{k_{wb}}\delta t & \frac{u_{v,t}\cos(u_{\gamma,t})}{k_{wb}}\delta t \end{pmatrix} \tag{4.2.4}$$

This gives us our "standard" EKF-SLAM formulation, capable of closing loops and theoretically capable of reducing the uncertainty of landmarks. The limits of the theoretical map accuracy attainable are bounded by the initial uncertainty of the vehicle position.

Next, we add satellites to the simulation. At the time of data collection, there were 31 satellite vehicles active in the GPS constellation. In our simulation, 31 virtual satellites were assigned fixed positions evenly spaced on a circle 1,000 km in radius.

Each satellite provides an observation of the distance from the satellite to the vehicle. However, this particular measurement is corrupted by exponentially correlated noise of the type discussed in section 3.3. To accommodate this noise, four "bias" terms are added to the state vector:

$$x_t = \begin{pmatrix} s_t \\ b_t \\ \Theta \end{pmatrix} \tag{4.2.5}$$

where

$$b_t = \begin{pmatrix} b_{1,t} \\ b_{2,t} \\ b_{3,t} \\ b_{4,t} \end{pmatrix} \tag{4.2.6}$$

is the current bias of the four beacons. To a first order approximation, these terms represent the sum of the errors due to geographic effects, such as unmodeled ionospheric refraction, tropospheric effects, etc.

The process model must be modified as well to reflect the new bias terms:

$$x_{t+1} = \underbrace{\begin{pmatrix} I & 0 & 0 \\ 0 & (e^{k_b \delta t})I & 0 \\ 0 & 0 & I \end{pmatrix}}_{A_t} x_t + \begin{pmatrix} (w_{v,t} + u_{v,t}) \cos(w_{\gamma,t} + u_{\gamma,t} + s_{\phi,t})\delta t \\ (w_{v,t} + u_{v,t}) \sin(w_{\gamma,t} + u_{\gamma,t} + s_{\phi,t})\delta t \\ \frac{(w_{v,t}+u_{v,t}) \sin(w_{\gamma,t}+u_{\gamma,t})}{k_{wb}} \\ w_{b1,t} \\ w_{b2,t} \\ w_{b3,t} \\ w_{b4,t} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{4.2.7}$$

where the constant k_b and the variance of the $w_{bi,t}$ terms come from the power spectrum density analysis of 3.5.2.

Within the augmented model, range observations can now be attributed to the sum of the beacon bias with the beacon-vehicle distance:

$$\begin{aligned} z_t &= \left|\beta_i - s^t\right| + b_{i,t} + v_{\beta i,t} \\ &\approx \underbrace{\begin{pmatrix} \frac{-\beta_{x,i} - \hat{s}_{x,t}}{|\beta_i - \hat{s}_t|} \\ \frac{-\beta_{y,i} - \hat{s}_{y,t}}{|\beta_i - \hat{s}_t|} \\ 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}}_{H_t} x_t + v_{\beta i,t} \end{aligned}$$

where β_i is the pose of the beacon and $v_{bi,t}$ is zero-mean white noise.

Our modified process model means we need a slightly more complicated predict

step than was presented in 4.2.2:

$$\hat{x}_{t+1} = f(\hat{x}_t, \hat{u}_t, 0) \tag{4.2.8}$$
$$P_{t+1} = \mathbf{A_t} P_t \mathbf{A_t^T} + W_t Q_t W_t^T$$

We now have three different types of observations: landmark observations, orientation observations, and global distance observations. This poses no particular difficulty; each type of update has well defined R_t and H_t matrices, which makes multiple applications of update step 4.2.3 straightforward.

To understand the completed system, it is useful to discuss the function of the mechanisms added to the filter.

During the predict stage of the filter, the A_t matrix pulls the current biases towards 0. Without this pull, the movement of the biases would be a simple Markov walk. Additionally, the A_t matrix decays the entries of P_t covariance that track the correlation between the bias terms and landmarks; if we don't observe a landmark over a long period of time, the estimate of that landmark's position becomes decorrelated from the current bias estimate.

If our localization estimate is dominated by beacon data, then repeated observations of a landmark in a short period of time cause the landmarks' localization to become correlated with the bias terms of the beacons in use, mitigating the overconfidence that would result from oversimplifying the system with a simple white noise assumption about beacon data. However, if the same landmark is observed again after a significant time lapse, the beacon bias becomes decorrelated from the landmark position, allowing for a further reduction in uncertainty.

4.3 Pseudorange Noise Simulation

How we simulate noisy measurements is of critical importance to the validity of the simulation. If we attempt to generate pseudorange noise using the derived model for that noise, any results presented would not give any useful data to support or refute the

appropriateness of the model. Instead, we formulate a method to "record" and "replay" the noise from a real receiver in the simulated model.

The National Geodetic Survey (NGS) publishes, in downloadable form, data from a volunteer-run set of continuously operating reference stations (CORS) all over the globe. The location of a given CORS, calculated by integrating GPS readings over a long time series, is typically known with sub-centimeter level precision. Raw code phase data from tracked satellites is published at a rate of 1 Hz.

If we lump all sources of pseudorange error into a single term, we have:

$$\rho_k^i = \left| s_k^i - r_k \right| + \gamma_k + \epsilon_k \tag{4.3.1}$$

where ρ^i is the pseudorange, s^i is the location of the i^{th} satellite at the time of transmission, r is the antenna location, γ is the receiver clock offset, and ϵ is the residual error.

In the case of the CORS, the unknown quantities in 4.3.1 are γ and ϵ. At each epoch, we can generate an estimate of γ, and can use these as observations driving a model of the oscillator state.

This stripping of the receiver clock offset from the residual has a side effect of shifting $E[\epsilon]$ towards zero. The sources of the errors which comprise ϵ are not naturally zero-mean; the dominant components consist of delays in the signal propagation from satellite to receiver. However, if *all* of the pseudoranges in an epoch are biased in the same direction, then the calculated solution for the clock offset will subsume the common bias.

Simulated pseudorange observations were generated by adding the measured ϵ residual to the true simulated satellite-vehicle range.

This model is far from perfect; it ignores, among other things, satellite error correlations due to constellation geometry and receiver clock modeling errors, but it does have the significant advantage of being data-driven.

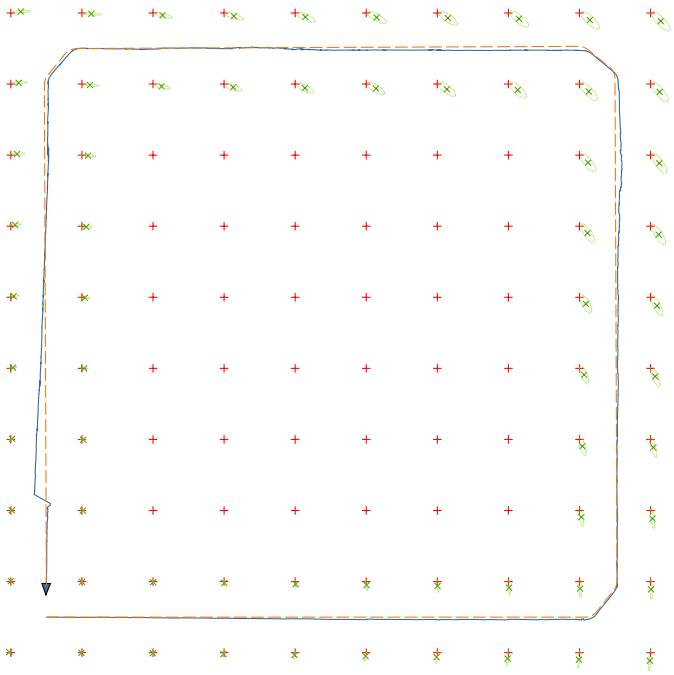

Figure 4.4.1: A classic SLAM loop closure with an EKF. The robot moves counterclockwise around the edge of the presented area.

4.4 Correlation Discussion

SLAM provides answers to the questions "Where am I?" and "Where are things of interest in my environment?". When looking at SLAM-generated maps, the most intuitive and visible results are global positions and uncertainties of poses and landmarks. Results such as a classic EKF loop shown in 4.4.1 make this information very clear. This kind of visual representation, while useful, elides the preponderance of information accrued in the model.

The most significant piece of information which is hidden from view is knowledge about the relative positions of landmarks. In the common case, observations in quick succession of multiple landmarks give much more precise information about their positions in relation to each other than in relation to whatever larger frame underlies the generated map.

In the EKF model, this relative positioning information is maintained in the form of

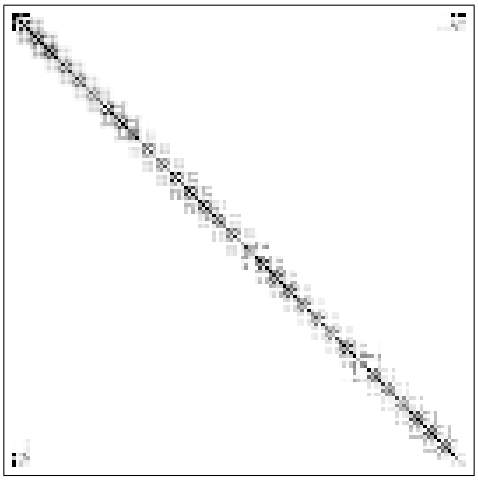

Figure 4.4.2: Inverse covariance for the single loop EKF example

covariances between landmarks. This hidden information is exposed at times of loop closure. In figure 4.4.1, the knowledge of the global position of the most recently seen landmarks was quite imprecise until the robot closed the loop. At loop closure, the relative positioning information already in the model allowed for the chained collapse of uncertainty backwards along the robot's path.

As as been noted in several papers, such as [Liu and Thrun, 2003], and [Thrun and Montemerlo, 2006], and [Eustice *et al.*, 2005], landmark-landmark relative information comes into the model by way of the pose estimate. Observations of landmarks correlate them with the pose estimate at the time of observation. Pose updates transfer pose-landmark correlations to inter-landmark correlations.

The relative information in the model can also be revealed by looking at the inverse covariance, or information matrix. Figure 4.4.2 shows the inverse of the landmarks covariance from the single-loop EKF example.[3] Landmarks are added to the model as they are observed. The resulting information matrix is strongly diagonal; landmarks tend to be highly correlated with other landmarks which were added to the model at about the same time. The off-diagonal patch in the matrix is the loop closure, which relates the positions of the first and last landmarks added to the model.

[3]The whiteness of most of this figure should not be taken to mean it is actually sparse, it is only *nearly* sparse. Although no values in this matrix are zero, several are within machine precision of being so.

A crucial insight into GPS-SLAM is that *whitening filters also introduce inter-landmark correlations*. Counterintuitively, GPS provides far better relative positioning information than absolute positioning. It is for this reason that high precision GPS localization techniques are differential: even though the system provides globally bounded estimates for positions without any differential reference, the bulk of the error sources confounding such an estimate are unobservable in short time spans.

The process through which whitening filters cause dependencies is illustrated in figure 4.4.3. When a GPS reading is taken, the pose of the robot becomes dependent on the state of the whitening filter at the time of the GPS observation. The state of the whitening filter is independent of the landmark positions given the pose at the time of the GPS reading.

The robot moves, and the previous pose is marginalized out of the EKF. This marginalization induces direct dependences between previously observed landmarks and the state of the whitening filter. However, the inter-landmark dependencies at this point are attributable to the robot motion. There are no GPS-induced direct inter-landmark dependencies yet because the whitening filter state at the time of observation is still in the model, and d-separates the landmarks with respect to GPS information.

Finally, the robot receives another GPS reading, and so updates the state of the whitening filter. The previous whitening filter state is marginalized out, pushing GPS information into the inter-landmark dependencies.

The importance of this distinction between motion- and GPS-induced inter-landmark dependencies lies in the different ways the pose and whitening filters are updated: the accuracy of a pose prediction tends to be a function of distance, whereas the accuracy of a filter state prediction is a function of time. Consider a GPS-equipped robot which sees landmark a, then sometime later sees landmark b. Broadly speaking, the robot motion contribution to an a–b dependency will be determined by how far the robot has traveled between the observations, whereas the GPS contribution will be governed by the elapsed time.

Consider, then, what this means for landmark estimates in a SLAM system. If the robot observes two landmarks in rapid succession while receiving GPS information, the nonwhite sources of error (ionosphere, troposphere, etc) will be almost identical for

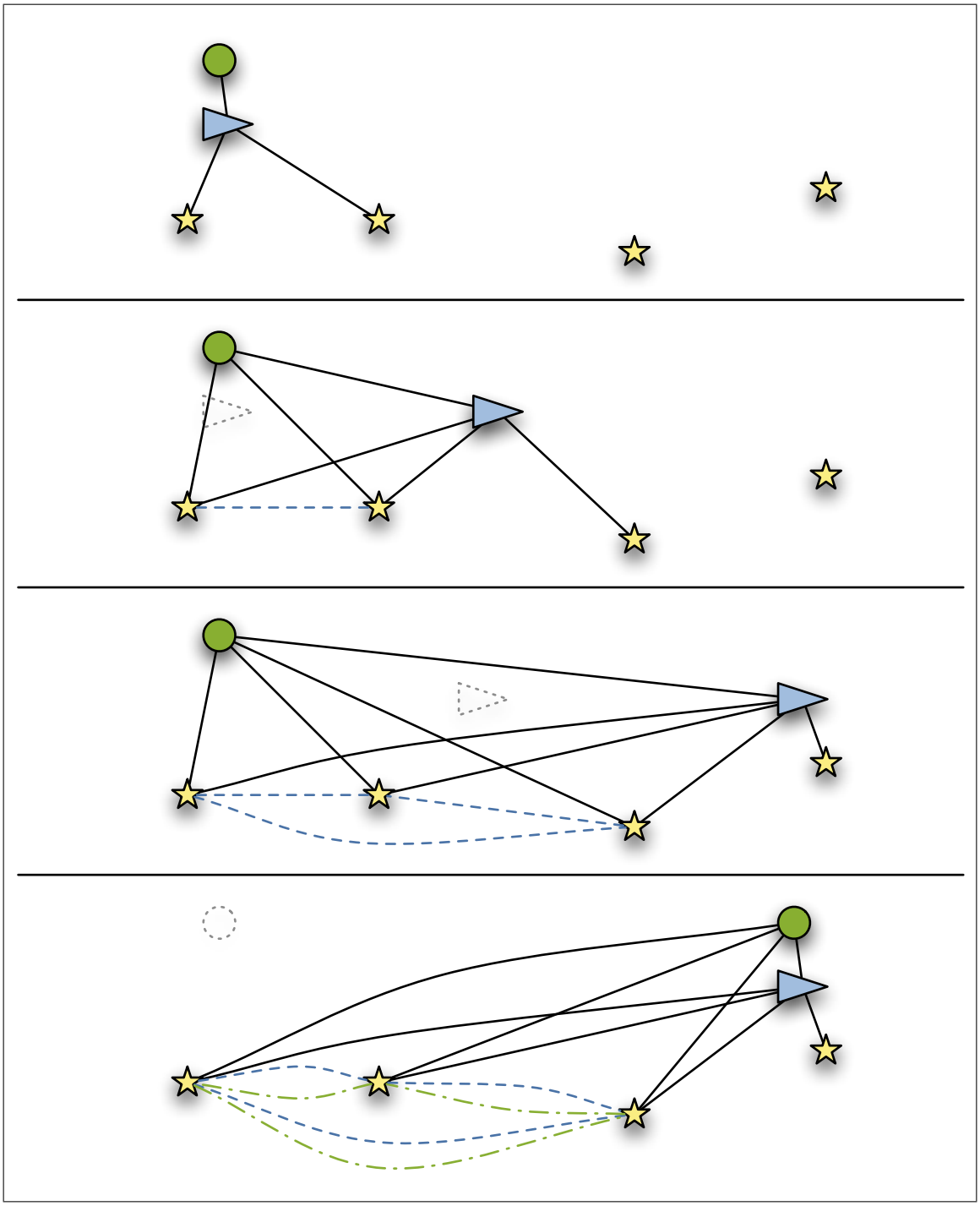

Figure 4.4.3: How dependencies enter the EKF model. The circles, triangles, and stars represent whitening filters, poses, and landmarks, respectively. Blue dashed lines are dependencies created solely by pose updates, while green dot-dashed lines show tightened dependencies created by updates of the whitening filters.

both landmarks. The independent noise sources are typically on the order of centimeters instead of meters.

If at some point in the future, we suddenly gain an excellent global estimate of the positions of one of the landmarks, then we have implicitly gained information about the state of these slow-varying non-white noise sources at the time of landmark observations. This in turn allows us to collapse the uncertainty of landmark positions seen by the robot soon before or after the oracular landmark.

Without very high-quality (and high-cost) odometry or inertial measurement, these landmark relationships borne of GPS information can be significant over a far wider area than those arising from motion.

Within our EKF model, we can illustrate this effect in a number of ways. Figure 4.4.4 revisits the simple loop example but makes two modifications: the initial pose of the robot is not known, and GPS readings arrive at 1 Hz. The autocorrelated error terms become apparent in the associated scatter plot of the landmark estimate errors. The lower map and plot show how the filter reacts when the system is conditionalized on the true value of the lower left landmark. The collapse in uncertainty propagates throughout the entire trajectory, and the plot of errors is centered on the true value.

We can also observe the difference in the information matrix of the GPS-enabled system, as can be seen in figure 4.4.5. The strong diagonal band which also appeared in 4.4.2 is still present. However, the off-diagonal entries now show a non-negligible dependency mesh of all landmarks in the model.

4.5 Model Validation

This EKF-GPS-SLAM model generates results which map well to intuitions of how a probabilistic mapping system which integrates GPS should behave, but thus far we have no real quantitative evidence that the presented model is "correct".

"Correctness" in the context of a probabilistic model is, ironically, a rather fuzzy proposition. The approximations used in the foundation of this formulation remove any straightforward way to show optimality under useful criteria.

One very important question for which we *can* provide a robust answer is how the

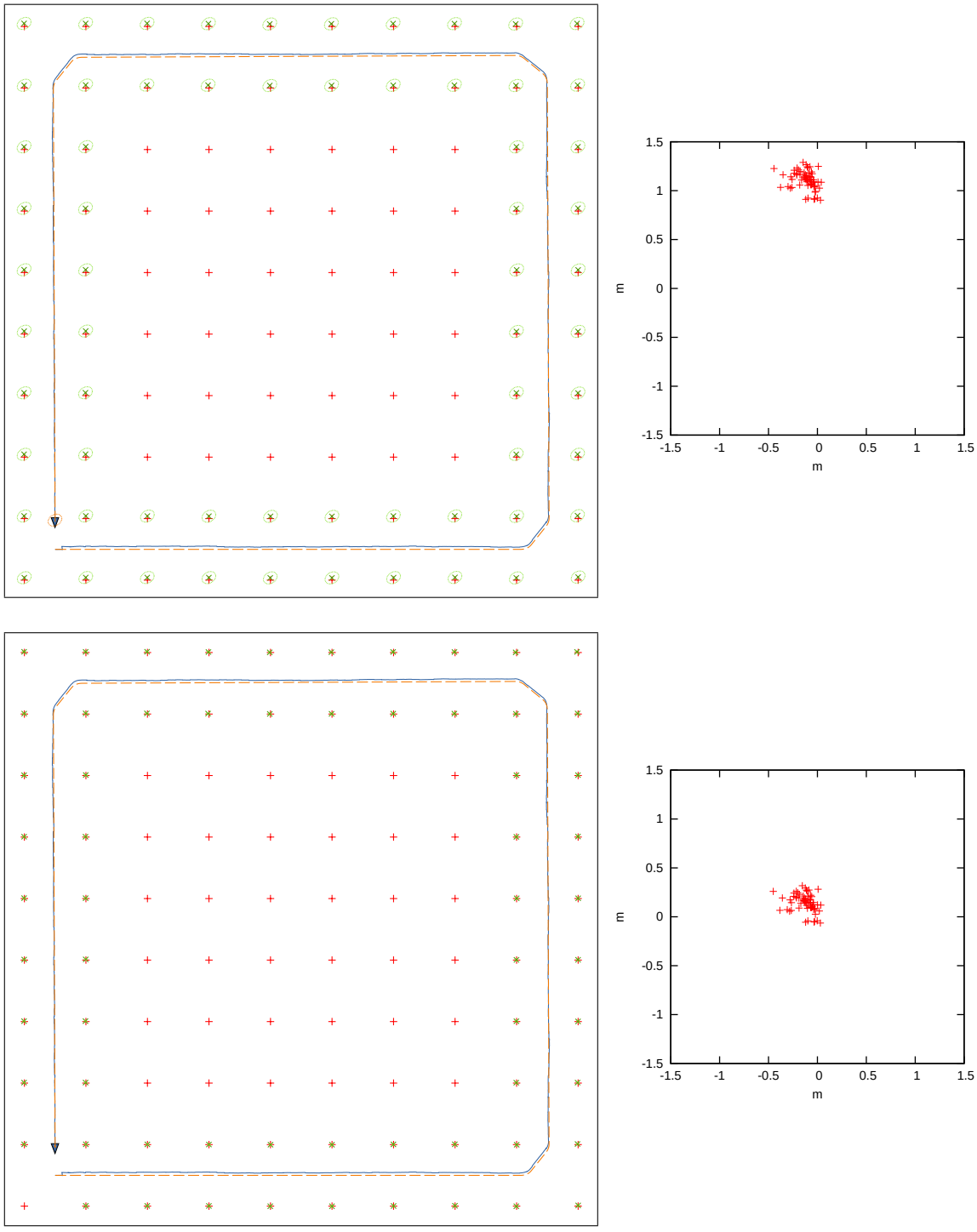

Figure 4.4.4: Top row: single loop with GPS availability, and scatter plot of resulting landmark estimate error. Bottom row: same data after conditionalizing the filter on the true position of the lower left landmark.

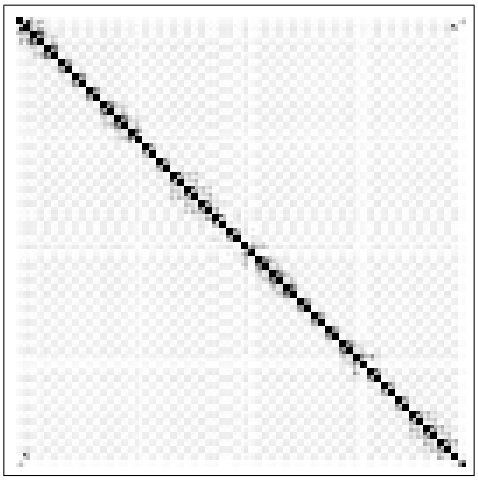

Figure 4.4.5: Inverse covariance of a large loop with GPS.

uncertainty estimate of the filter compares to the actual errors in estimate.

The EKF is, at heart, a single Gaussian distribution of high dimensionality. In simulation we have access to the true global state, so at any timestep we can calculate the squared Mahalanobis distance of the estimate from the true value, e.g.:

$$(\hat{x} - x)^{\mathrm{T}} P^{-1} (\hat{x} - x) \tag{4.5.1}$$

From statistics, we know that if we take samples from a Gaussian distribution, the resulting distribution of squared Mahalanobis distances of samples from the mean follows a chi-square distribution:

$$P(x) = \frac{x^{(k/2-1)} e^{-x/2}}{2^{k/2} \Gamma(k/2)} \tag{4.5.2}$$

where k is the number of degrees of freedom, and Γ is a Gamma function.

This provides the basis for validating our model. If the model is "correct", then the squared Mahalanobis distance of errors should be drawn from the appropriate chi-squared distribution.

Ideally, we would like to be able to use the full error vector and covariance to calculate distances, but the size of the state vector is dependent on the number of unique

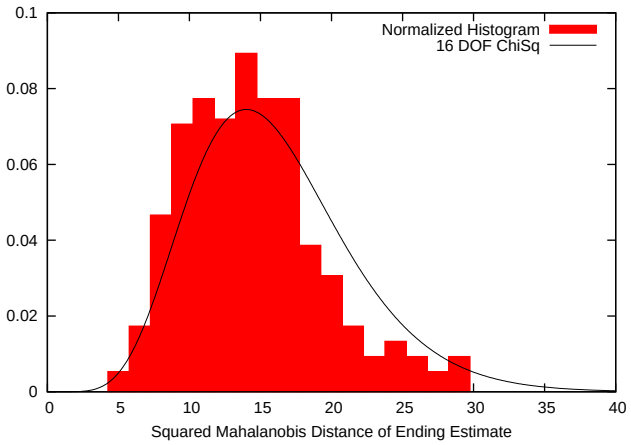

Figure 4.5.1: No time lapse between successive paths

landmarks observed. This prevents us from holding the degrees of freedom constant. Instead, we take a marginal distribution of 8 random landmarks from the model as the basis for our squared Mahalanobis calculation.

The results from three scenarios are presented. All scenarios use the grid landmark set shown in 4.4.1. Each scenario consists of 500 trials, each of which is made up of 20 short paths driven by the robot. A path consists of a random starting point, a random waypoint, and finally a random ending point. At the beginning of each path, the robot has no prior estimate of its pose. At the end of each trial, the squared Mahalanobis distance between the estimate and true values of 8 landmarks under the relevant covariance marginal are recorded.

In each scenario, a different amount of time elapses between path runs. The first scenario involves no time lapse; the robot effectively "teleports" between locations and continues to operate. In the second scenario, an hour passes between the end of one path and the start of the next. In the third scenario, 5 hours passes between path runs.

Figures 4.5.1, 4.5.2, and 4.5.3 show the results of the trials. Statistically, if the error models were exact and Gaussian, the recorded Mahalanobis distances would be drawn from a 16 degree-of-freedom chi-square distribution. As can be seen from the histograms, in all three cases, the Mahalanobis distance histogram is shifted towards zero with respect to the expected distribution. This means our uncertainty estimates

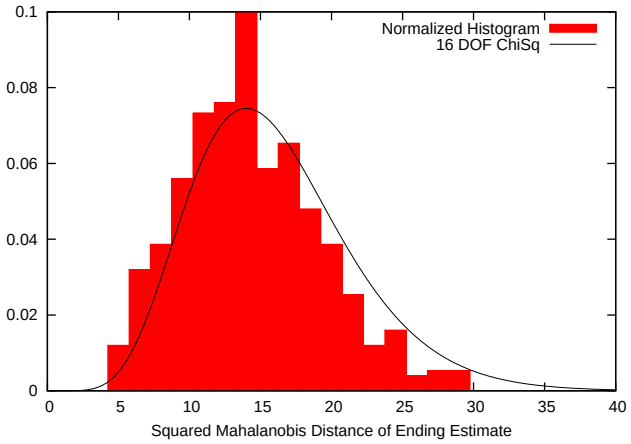

Figure 4.5.2: 1 hour time lapse between successive paths

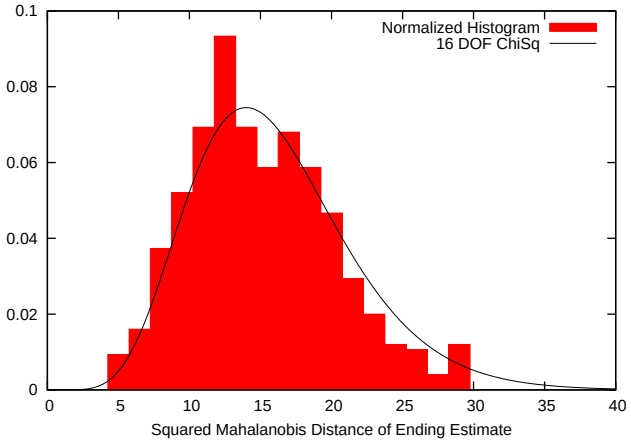

Figure 4.5.3: 5 hour time lapse between successive paths

are conservative, which should help to keep the system stable.

4.6 Conclusion

In this chapter, having developed a model for GPS noise whitening, we have derived and implemented a fused EKF-GPS-SLAM system, thus providing a concrete example of how we can integrate GPS signals and their associated whitening filters into a probabilistic mapping system. This model has been used to discuss how GPS signals affect information propagation in probabilistic maps. We have also taken advantage of simulation's ability to provide absolute ground truth values to generate quantitative metrics of how well the fused system functions.

All of this work was done in the context of the EKF, which has well-known draconian limitations in scalability. In the next chapter, we will show how to apply the same noise models to a very different system which greatly improves our scalability.

Chapter 5

Integrating GPS into a Smoothing and Mapping System

5.1 Introduction

Chapter 4 showed how to integrate GPS into a classical EKF-based SLAM solution, and showed how intermittently available GPS can make the linearization assumptions underlying the method more consistently valid. Unfortunately, GPS does nothing to address the underlying algorithmic complexity issues which prevent the method from scaling to very large environments.

The preponderance of novel mapping approaches in recent years have capitalized on the observation that inter-landmark dependence is the root cause of computational inefficiency. This is most easily seen in the context of Kalman-filter based approaches. At the moment of observation of new landmarks, the positions of those landmarks are conditionally independent given the pose. However, when the robot moves, that independence-preserving pose is marginalized out, causing all landmarks to be dependent not only on landmarks observed from the same pose, but on *all* previously observed landmarks.

Smoothing and mapping (SAM) takes what may be the most straightforward approach to the problem of maintaining these conditional independences: it uses the full trajectory of the robot in estimating the map. In this way marginalization-induced de-

pendencies are prevented from entering the model. In a sense, SAM trades size for independence; SAM estimates many more parameters than a filter-based approach, but in doing so it gains a structure which allows it to estimate the variables much faster. Figure 5.1.1 illustrates the tradeoff: marginalization decreases the number of variables, but increases interconnect.

The use of GPS in a mapping system brings with it a problem which parallels the pose-marginalization problem. In the GPS-mapping problem, landmark positions are conditionally independent given the pose *and the hidden parameters used to whiten the noise of the GPS measurements*. The parallel problem can be solved using a parallel solution to the SAM pose problem; we can maintain sparsity in the estimation problem if we estimate the state of the hidden variables at all relevant timesteps throughout the robot's operation.

In this chapter, we provide a detailed example of how the ideas of this thesis can be realized to enable large-scale urban mapping. We first review the basic probabilistic formulation for batch SAM. The formulation is then extended to accommodate GPS readings over multiple trajectories.

The basic motivation for SAM comes from reasoning about the conditional independences in the mapping problem. As shown in figure 5.1.1, landmark estimates are conditionally independent given the poses at the time of observations. If poses are not marginalized out of the model, then landmarks remain independent.

The surprising result is that the much larger, much sparser full-trajectory-and-map problem can frequently be solved more efficiently than updating a filtering system with dense inter-landmark dependencies.

5.2 Vanilla SAM

For reference purposes, a brief review of the SAM least-squares formulation will be presented, which I will refer to as "Vanilla SAM". For a more thorough discourse, refer to [Dellaert and Kaess, 2006].

A mobile robot's process model, which provides the relationship between successive

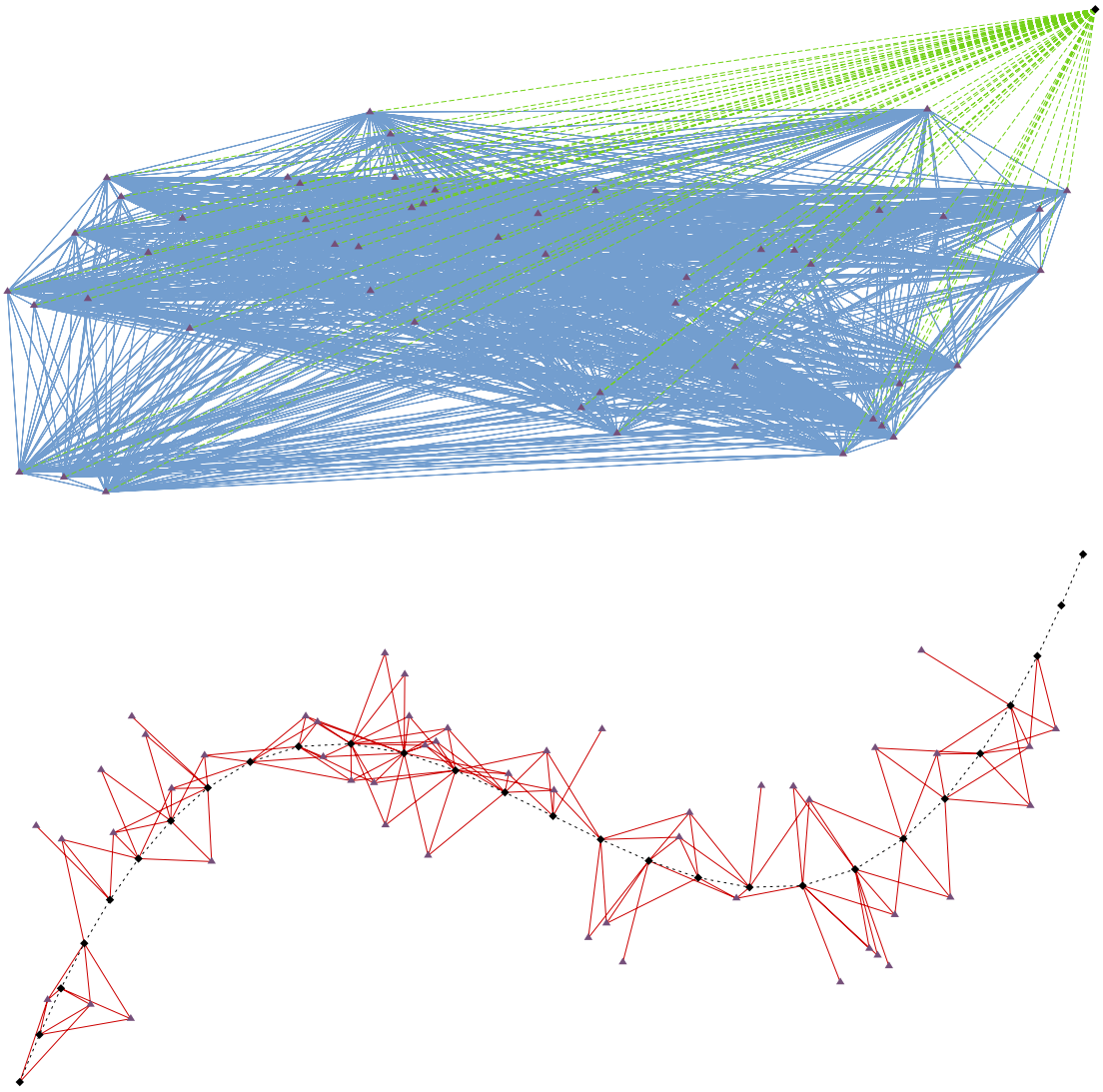

Figure 5.1.1: The effects of marginalization on the underlying Markov random field. Triangles represent landmarks; diamonds represent vehicle poses. In the top diagram, previous poses are continually marginalized out. This reduces the number of parameters being estimated, but increases interconnect. In the bottom diagram, the full trajectory remains in the system. This increases the number of parameters, but vastly diminishes interconnect; nodes are only connected by process model constraints and observations.

poses, can be written abstractly as:

$$x_i = f(x_{i-1}, u_i) + \underbrace{w_i}_{\sim N(0,\Lambda_i)} \tag{5.2.1}$$

where x_i is the pose at time i, u_i is the control input, and w_i is noise[1]

Observations, which provide the relationship between a given pose and a landmark, can be abstractly defined as:

$$z_k = h(x_i, l_k) + \underbrace{v_k}_{\sim N(0,\Sigma_k)} \tag{5.2.2}$$

where x_i is the pose at the time the observation is taken, l_k is the landmark being observed, and v_k is noise.

Making use of the strong assumption that w_i and v_k are Gaussian leads to the probability distributions:

$$\begin{aligned} P(x_i|x_{i-1}, u_i) &= \frac{1}{\sqrt{2\pi}||\Lambda_i||^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x_i - f(x_{i-1}, u_i))^{\mathrm{T}} \Lambda_i^{-1} (x_i - f(x_{i-1}, u_i))\right) \\ &\propto \exp\left(-\frac{1}{2}||f(x_{i-1}, u_i) - x_i||^2_{\Lambda_i}\right) \end{aligned} \tag{5.2.3}$$

$$\begin{aligned} P\left(z_k|x_{i_k}, l_j\right) &= \frac{1}{\sqrt{2\pi}||\Sigma_k||^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\left(h\left(x_{i_k}, l_j\right) - z_k\right)^{\mathrm{T}} \Sigma_k^{-1} \left(h\left(x_{i_k}, l_j\right) - z_k\right)\right) \\ &\propto \exp\left(-\frac{1}{2}||h\left(x_{i_k}, l_j\right) - z_k||^2_{\Sigma_k}\right) \end{aligned} \tag{5.2.4}$$

where $||e||^2_\Sigma$ is used to denote the squared Mahalanobis distance of vector e under covariance matrix Σ.

[1]There is a subtle, but important, difference between this process model and that which is typically used in Kalman filtering and related techniques. SAM requires that the noise be in the same space (and of the same dimension) as the pose. In contrast, variants on Kalman filtering allow the noise to be in a different space (typically that of the control inputs) and transformed into the pose space without requiring the resulting pose space covariance matrix to be full-rank.

To facilitate discussion of a full smoothing problem, let

$$X \triangleq \{x_0 \dots x_M\} \qquad U \triangleq \{u_1 \dots u_M\} \qquad L \triangleq \{l_0 \dots l_N\} \qquad Z \triangleq \{z_0 \dots z_K\} \tag{5.2.5}$$

be the sets of all poses, landmark positions, control inputs, and observations of interest, respectively.

We wish to find the a posteriori estimate maximizing:

$$P(X, L, Z, U) = P(x_0) \prod_{i=1}^{M} P(x_i|x_{i-1}, u_i) \prod_{k=1}^{K} P(z_k|x_{i_k}, l_j) \tag{5.2.6}$$

In 5.2.6, the observations Z and the control inputs U are known. The poses X and the landmark locations L are unknown. We group together the parameters to be estimated as $\Theta \triangleq \{X, L\}$, and note:

$$\begin{aligned} \Theta^* &\triangleq \underset{\Theta}{\operatorname{argmax}}\, P(\Theta|Z, U) \\ &= \underset{\Theta}{\operatorname{argmax}}\, P(X, L, Z, U) \end{aligned} \tag{5.2.7}$$

Using the standard log-likelihood transformation, we arrive at a minimization problem:

$$\begin{aligned} \Theta^* &= \underset{\Theta}{\operatorname{argmax}}\, P(X, L, Z, U) \\ &= \underset{\Theta}{\operatorname{argmin}} - \log P(X, L, Z, U) \\ &= \underset{\Theta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{M} ||f(x_{i-1}, u_i) - x_i||^2_{\Lambda_i} + \sum_{k=1}^{K} ||h(x_{i_k}, l_j) - z_k||^2_{\Sigma_k} \right\} \end{aligned} \tag{5.2.8}$$

Note that 5.2.8 is similar to, but is not quite, a least-squares problem; the summed terms are squared Mahalanobis distances. We set this quibble aside for the moment and continue as if we were solving a standard least-squares problem using a series of linear approximations. We will resolve the Mahalanobis distance issue later.

To proceed we need points around which to linearize, which means we require some

prior estimate for X and L. We will write these as:

$$X^0 = \{x_0^0 \dots x_M^0\} \quad L^0 = \{l_0^0 \dots l_N^0\} \tag{5.2.9}$$

The initial estimate for these values typically comes from odometry.

We now linearize f and h around X^0 and L^0 with respect to the quantities being estimated:

$$F_i^{i-1} \triangleq \frac{\partial f_i(x_{i-1},u_i)}{\partial x_{i-1}}\Big|_{x_{i-1}^0} \quad H_k^i \triangleq \frac{\partial h_k(x_i,l_j)}{\partial x_i}\Big|_{x_i^0,l_j^0} \quad J_k^j \triangleq \frac{\partial h_k(x_i,l_j)}{\partial l_j}\Big|_{x_i^0,l_j^0} \tag{5.2.10}$$

Next, we define

$$\delta x_i = x_i - x_i^0 \tag{5.2.11}$$

to be the offset between the i^{th} pose estimate and the true value. This leads to an approximation of the process model:

$$f_i(x_{i-1}, u_i) \approx f_i(x_{i-1}^0, u_i) + F_i^{i-1}\delta x_{i-1} \tag{5.2.12}$$

And now, manipulating the process terms of 5.2.8, we find:

$$\begin{aligned} f_i(x_{i-1}, u_i) - x_i &\approx \{f_i(x_{i-1}^0, u_i) + F_i^{i-1}\delta x_{i-1}\} - \{x_i^0 + \delta x_i\} \\ &= F_i^{i-1}\delta x_{i-1} - \delta x_i - \underbrace{\left(x_i^0 - f_i(x_{i-1}, u_i)\right)}_{\triangleq a_i} \end{aligned} \tag{5.2.13}$$

Note that we've consolidated some terms into a new variable, a_i, which is the difference between a pose estimate and the prediction of that pose estimate using the previous pose estimate and control input.

Using the definition

$$\delta l_j = l_j - l_j^0 \tag{5.2.14}$$

as the error in the j^{th} landmark estimate, the observation model is approximated as

$$h_k(x_i, l_j) \approx h_k(x_i^0, l_j^0) + H_k\delta x_i + J_k\delta l_j \tag{5.2.15}$$

From the observation terms of 5.2.8 we find:

$$\begin{aligned} h_k(x_i, l_j) - z_k &\approx \{h_k(x_i^0, l_j^0) + H_k \delta x_i + J_k \delta l_j\} - z_k \\ &= H_k \delta x_i + J_k \delta l_j - \underbrace{(z_k - h_k(x_i, l_j))}_{\triangleq c_k} \end{aligned} \tag{5.2.16}$$

where again we have consolidated some terms. c_k is what is sometimes called the *innovation*; it is the difference between our actual observation and what we expect that observation to be given our landmark estimate and pose estimate.

Substituting approximations 5.2.13 and 5.2.16 into 5.2.8, we find a linearized version of 5.2.8:

$$\delta^* = \underset{\delta}{\operatorname{argmin}} \{ \sum_{i=1}^{M} \left|\left| F_i^{i-1} \delta x_{i-1} - \delta x_i - a_i \right|\right|_{\Lambda_i}^2 + \sum_{k=1}^{K} \left|\left| H_k \delta x_i + J_k \delta l_j - c_k \right|\right|_{\Sigma_k}^2 \} \tag{5.2.17}$$

The only thing keeping 5.2.17 from being a standard least-squares problem is still that squared Mahalanobis distance. Recall that the squared Mahalanobis distance is defined as

$$||v||_\Sigma^2 \triangleq v^{\mathrm{T}} \Sigma^{-1} v \tag{5.2.18}$$

Σ, being a covariance matrix, is symmetric positive definite. This implies that Σ^{-1} is also symmetric positive definite, and can be factorized via Cholesky decomposition as[2]

$$\Sigma^{-1} = \Sigma^{-\frac{1}{2}} \Sigma^{-\frac{\mathrm{T}}{2}} \tag{5.2.19}$$

From 5.2.18:

$$\begin{aligned} ||v||_\Sigma^2 &\triangleq v^{\mathrm{T}} \Sigma^{-\frac{1}{2}} \Sigma^{-\frac{\mathrm{T}}{2}} v \\ &= \Sigma^{-\frac{1}{2}} v \Sigma^{-\frac{\mathrm{T}}{2}} v \\ &= \left|\left| \Sigma^{-\frac{\mathrm{T}}{2}} v \right|\right|_2^2 \end{aligned} \tag{5.2.20}$$

This means we can "normalize" the troublesome squared Mahalanobis distance terms

[2]The shorthand $A^{-\frac{\mathrm{T}}{2}}$, used here to keep notation from becoming unwieldy, should be interpreted as $\left(A^{-\frac{1}{2}}\right)^{\mathrm{T}}$

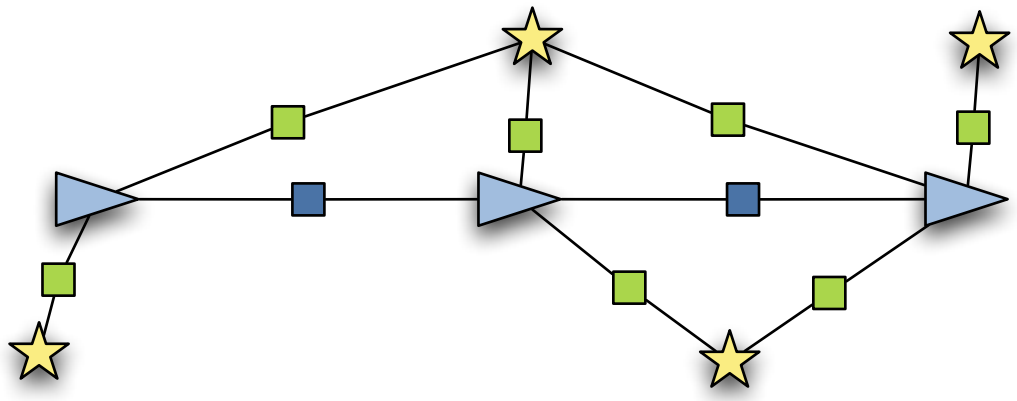

Figure 5.2.1: Factor graph for the scenario in 5.2.22. The system involves 3 poses (triangles), 4 landmarks (stars), 7 landmark observations (green boxes) and 2 motion updates (blue boxes).

in 5.2.17, turning them into squared L2-norms by premultiplying by the square root of the relevant inverse covariance.

For notational purposes, we define some new variables which incorporate this premultiplication:

$$\begin{aligned} \hat{F}_i^{i-1} &\triangleq \Lambda_i^{-\frac{\mathrm{T}}{2}} F_i^{i-1} \\ \hat{a}_i &\triangleq \Lambda_i^{-\frac{\mathrm{T}}{2}} a_i \\ \hat{H}_k &\triangleq \Sigma_k^{-\frac{\mathrm{T}}{2}} H_k \\ \hat{J}_k &\triangleq \Sigma_k^{-\frac{\mathrm{T}}{2}} J_k \\ \hat{c}_k &\triangleq \Sigma_k^{-\frac{\mathrm{T}}{2}} c_k \end{aligned} \tag{5.2.21}$$

We can now gather our constraints into a familiar matrix form. The A matrix of this problem is sparse with block non-zero entries. An example factor graph is shown in figure 5.2.1; the corresponding least squares setup is shown in 5.2.22. The correspondences between the representations can easily be seen: each pose motion factor and

landmark observation factor in the graph correspond to a block row of the A matrix.

$$A = \begin{pmatrix} -\Lambda_1^{-\frac{\mathrm{T}}{2}} & & & & & & \\ \hat{F}_2^1 & -\Lambda_2^{-\frac{\mathrm{T}}{2}} & & & & & \\ & \hat{F}_3^2 & -\Lambda_3^{-\frac{\mathrm{T}}{2}} & & & & \\ \hat{H}_1 & & & \hat{J}_1 & & & \\ \hat{H}_2 & & & & \hat{J}_2 & & \\ & \hat{H}_3 & & & \hat{J}_3 & & \\ & \hat{H}_4 & & & & \hat{J}_4 & \\ & & \hat{H}_5 & & \hat{J}_5 & & \\ & & \hat{H}_6 & & & \hat{J}_6 & \\ & & \hat{H}_7 & & & & \hat{J}_7 \end{pmatrix} \quad \delta = \begin{pmatrix} \delta x_1 \\ \delta x_2 \\ \delta x_3 \\ \delta l_1 \\ \delta l_2 \\ \delta l_3 \\ \delta l_4 \end{pmatrix} \quad b = \begin{pmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{a}_3 \\ \hat{c}_1 \\ \hat{c}_2 \\ \hat{c}_3 \\ \hat{c}_4 \\ \hat{c}_5 \\ \hat{c}_6 \\ \hat{c}_7 \end{pmatrix} \tag{5.2.22}$$

Finally, we use standard least-squares techniques to solve for:

$$\delta^* = \underset{\delta}{\operatorname{argmin}} ||A\delta - b||_2^2 \tag{5.2.23}$$

The solution to 5.2.23 yields the offset terms δ^*, which are used to correct X^0 and L^0. If desired, these corrected poses and landmark locations can be used as the basis of a new iteration of the algorithm.

The efficiency of solving this system depends heavily on maintaining sparsity in the associated matrices. This will be addressed in the context of GPS additions.

5.3 Extending SAM to use GPS

Having now reviewed basic SAM, we now look at what modifications need to be made to incorporate GPS readings. We will assume that the model being used is the bias model developed in 3.5.

Working in a global coordinate system

In its vanilla formulation, SAM makes two assumptions which we now reexamine. First, as is commonly the case with SLAM systems, the original formulation defines the origin of the global coordinate system to be coincident with the first pose. Second, SAM assumes that the set of poses to be smoothed is a single contiguous trajectory.[3]

GPS defines many values with respect to its own global coordinate system. Defining an arbitrary coordinate system which only references the robot trajectory is no longer advantageous; maintaining such a system adds the complexity of estimating a coordinate transform to our problem without significant benefits. The latter assumption, that of a single contiguous trajectory, is rooted in assumptions about data association. If we run a robot for a time, then kidnap it, moving it arbitrarily far from its previous known pose, the robot effectively has infinite uncertainty about its position with respect to the existing landmarks in the system. If, in the robot's new pose, it cannot associate any observations with landmarks from the pre-kidnapped state, it may as well start over with a new map and new coordinate system. However, the availability of a global reference changes the situation. A long-term effort to generate a map of an extensive area can span weeks or more, and it is entirely likely that the robot would, at several points during the mapping effort, be moved while offline, breaking the chain of poses into multiple trajectories.

These two problems have a shared solution; finding a solution for initializing x_0 can be reapplied to any x_n which starts a new trajectory chain.

We can look at this initial-pose problem in two distinct cases: with and without a (Gaussian) prior for x_0.

If

$$x_0 \sim N(\bar{x}_0, \Phi) \tag{5.3.1}$$

[3]Interestingly, these two assumptions will also have to be revisited in the case of multi-robot SAM.

then we can slot this into 5.2.6. Equation 5.2.8 gains an additional term:

$$\Theta^* = \underset{\Theta}{\operatorname{argmin}} \left\{ ||x_0 - \bar{x}_0||^2_\Phi + \sum_{i=1}^{M} ||f(x_{i-1}, u_i) - x_i||^2_{\Lambda_i} + \sum_{k=1}^{K} ||h(x_{i_k}, l_j) - z_k||^2_{\Sigma_k} \right\} \tag{5.3.2}$$

The new term is already linear; we don't need to take any Jacobians to fit within the linear least-squares framework. However, we do need to write the initial pose terms in a form which separates out the residual error we are solving for in one iteration of the algorithm:

$$\begin{aligned} x_0 - \bar{x}_0 &= \left\{ x_0^0 + \delta x_0 \right\} - \bar{x} \\ &= \delta x_0 - \underbrace{(\bar{x} - x_0^0)}_{\triangleq \alpha} \end{aligned} \tag{5.3.3}$$

Using the same premultiplication "trick" from 5.2.20, we can collect the terms into the matrices:

$$A = \begin{pmatrix} \Phi^{-\frac{1}{2}} & & & & \\ \hat{F}_1^0 & -\Lambda_1^{-\frac{\mathrm{T}}{2}} & & & \\ & \hat{F}_2^1 & -\Lambda_2^{-\frac{\mathrm{T}}{2}} & & \vdots \\ & & \hat{F}_3^2 & -\Lambda_3^{-\frac{\mathrm{T}}{2}} & \\ & & \dots & & \end{pmatrix} \quad \delta = \begin{pmatrix} \delta x_0 \\ \delta x_1 \\ \delta x_2 \\ \delta x_3 \\ \vdots \end{pmatrix} \quad b = \begin{pmatrix} \Phi^{-\frac{1}{2}} \alpha \\ \hat{a}_1 \\ \hat{a}_2 \\ \hat{a}_3 \\ \vdots \end{pmatrix} \tag{5.3.4}$$

It may also be the case that we have no prior for x_0. It may be counterintuitive, but this is the common case in GPS-SAM. A typical scenario is illustrated in figure 5.3.1. The robot is operating for a time before it has any information about its global location. In a sense, the start of the trajectory chain is "dangling", constrained in position only by the process model linking it to future poses.

To accommodate this scenario, we need only remove the prior constraint, which is

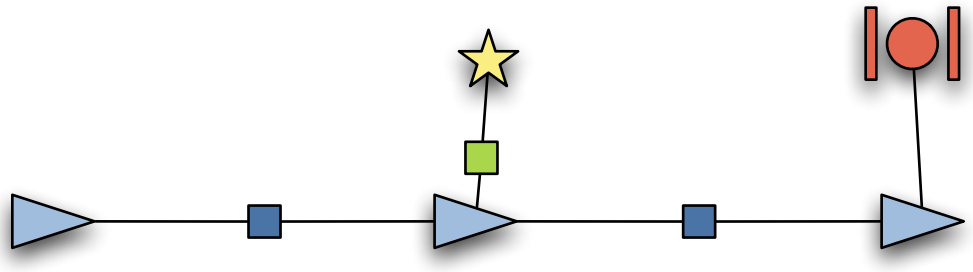

Figure 5.3.1: An example trajectory chain without a prior for the first pose. The robot begins operating, but has no information about its global location. After moving, it observes a landmark, and after moving again it receives GPS information.

represented by the top of the matrix and residual vector in 5.3.4:[4]

$$A = \begin{pmatrix} \hat{F}_1^0 & -\hat{\Lambda}_1^{-\frac{\mathrm{T}}{2}} & & & \\ & \hat{F}_2^1 & -\hat{\Lambda}_2^{-\frac{\mathrm{T}}{2}} & & \\ & & \hat{F}_3^2 & -\hat{\Lambda}_3^{-\frac{\mathrm{T}}{2}} & \vdots \\ & & \cdots & & \end{pmatrix} \quad \delta = \begin{pmatrix} \delta x_0 \\ \delta x_1 \\ \delta x_2 \\ \delta x_3 \\ \vdots \end{pmatrix} \quad b = \begin{pmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{a}_3 \\ \vdots \end{pmatrix} \tag{5.3.5}$$

Note this is not a return to the original form presented in 5.2.22. We are estimating δx_0 in addition to the rest of the trajectory chain. This can easily be seen by looking at a SAM problem which has no observations. In this hypothetical scenario, if we fix the initial pose to the origin and use 5.2.22, A is a square, full-rank matrix. If we do not so constrain the initial pose, then we use 5.3.5; the resulting A matrix has more columns than rows, revealing that the system is underconstrained.

[4]This can also be seen by imagining a prior with infinite variance. If $\Phi^{-\frac{1}{2}}$ is zero, then the terms in the dropped rows are 0.

Bias Estimation

Next let's look at how the whitening filter developed in 3.5 can be incorporated into the SAM system. The fundamental idea behind a whitening filter is the augmentation of estimated state to include nonwhite error sources. Consider an abstract example, represented by the equation

$$a = f(x_i, \ldots) + v_i \tag{5.3.6}$$

where a is some result of interest (typically an observation or subsequent state), x is the system state (which we are trying to estimate), and v is corrupting noise which has undesirable characteristics, such as a non-zero mean or autocorrelation.

Whitening v_i involves decomposing it into white noise and a new parameter to be estimated:

$$v_i = q_i + w_i \tag{5.3.7}$$

where w_i is white and q_i models some underlying phenomenon, such as exponential decay correlation, sinusoidal periodicity correlation, constant offset, or some other characteristic.

Our abstract example equation, then, becomes:

$$a = f(x_i, q_i, \ldots) + w_i \tag{5.3.8}$$

where w_i becomes a parameter which we wish to estimate as a part of the model.

Much like poses, the states of a whitening parameter form a dependency chain in the Markov random field representation of a SAM problem. For our GPS-SAM system, in which pseudorange noise is being whitened using an exponential decay parameter, we have approximately 32 chains of "bias" variables, one for each observed satellite, which are being estimated in the system. The most conceptually straightforward path to incorporating these bias variables is simply to integrate them into the vehicle model and extend the process model to handle the new terms.

This augmented-state model is correct, but a poor match for the problem at hand; by injecting this additional state into the process model, we effectively estimate the state of every bias term at every pose time-step in the system. Our model of the random process

of the noise in the readings is stationary: the autocorrelation of the noise at two points is parameterized only by the time difference. Adding intermediate estimates of this state will only result in interpolation of the bias at those points. In the unlikely event that we want to recover the bias estimates at every pose, then this is easily and more appropriately accomplished by post-processing. The situation is illustrated in figure 5.3; we have no need to know the state of bias term k except at times when we are actually taking a reading from satellite k.

Instead, we define a parallel bias process model which is responsible for updating the bias terms between observations:

$$\beta^k_{j+1} = f^*(\beta^k_i, \delta t) + \underbrace{\upsilon_j}_{\sim N(0,\Upsilon)} \tag{5.3.9}$$

where δt is the time elapsed from the previous observation.

The receiver clock offset also has a process model which propagates it from observation epoch to observation epoch:

$$\eta_{j+1} = f'(\eta_i, \delta t) + \underbrace{\xi_j}_{\sim N(0,\Xi)} \tag{5.3.10}$$

Pseudorange Observations

A GPS reading can be thought of as an observation without an associated landmark. Whereas landmark observations add constraints to the relative position of a particular pose and a landmark, GPS observations add a global constraint to a pose. These observations don't fit into the model defined in 5.2.2, so we define an additional observation model for the GPS readings:

$$\rho_i = g(x_j, \beta^k_l, \eta_m, \gamma_i) + \underbrace{\psi_j}_{\sim N(0,\Psi_j)} \tag{5.3.11}$$

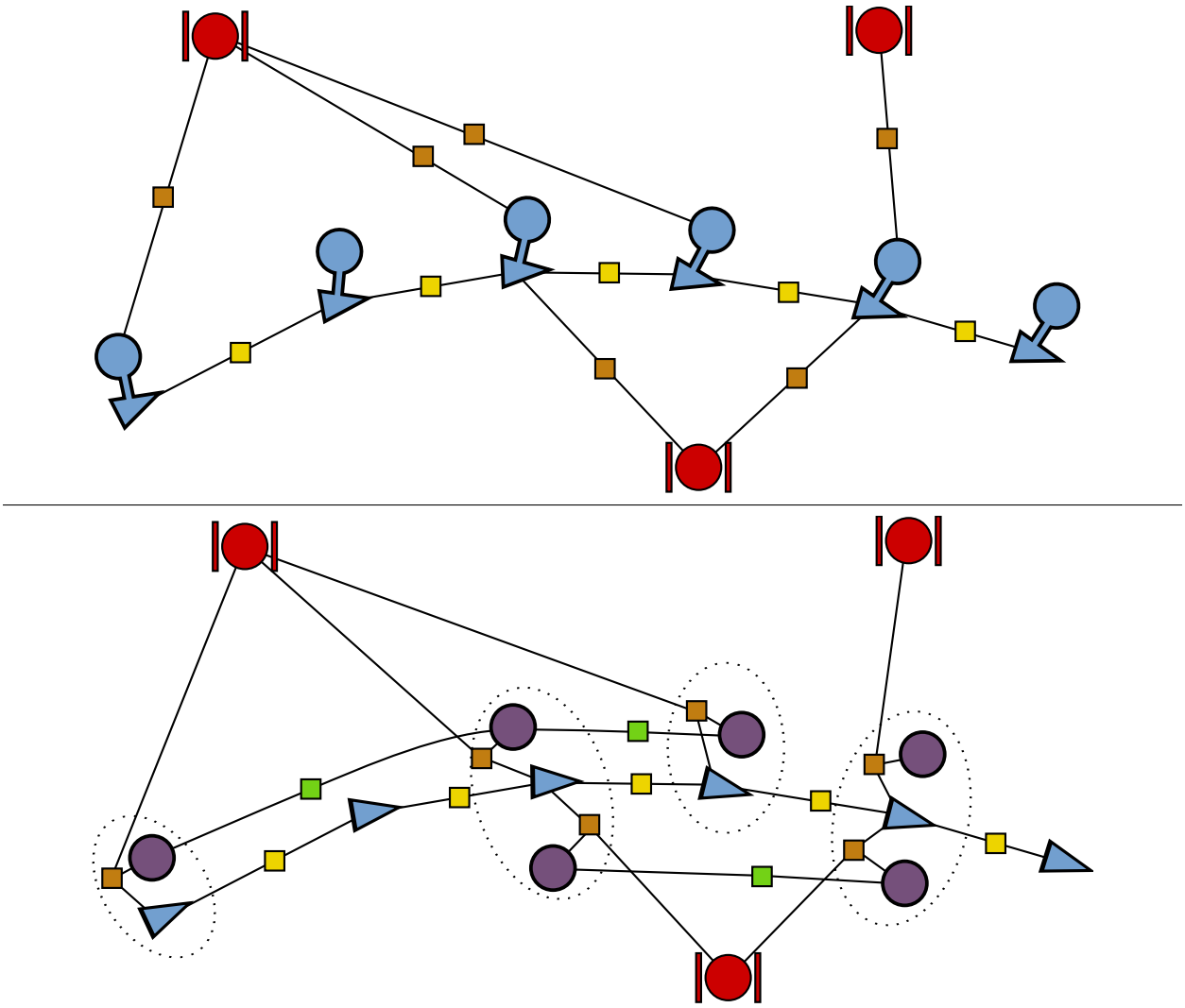

Figure 5.3.2: Factor graphs of two bias augmentation strategies. Constraints from process models or measurements are represented by small squares. The top diagram shows the least invasive way to augment the system mathematically: augmenting the robot state with bias information and propagating this through the existing process model. This has an appealing simplicity but hides significant inefficiency; since the entire set of biases is estimated at every pose step. The lower figure illustrates separating out the bias process model to avoid adding unnecessary terms to the system; biases are estimated only when we take a reading from the associated satellite. The dotted lines group terms which are coincident in time. (Note: landmarks have been omitted for clarity, and satellite representations are included to clarify groupings; they do not represent variables being estimated.)

where ρ_i is the pseudorange, η_m is the receiver clock offset[5], γ is the state of the GPS system, and ψ_j is the (Gaussian) measurement noise. Let

$$r_i \triangleq \begin{pmatrix} r_{x,i} \\ r_{y,i} \\ r_{z,i} \end{pmatrix} \tag{5.3.12}$$

be the global frame position of the GPS receiver antenna at pose x_i. A GPS reading can be written as:

$$g(x_i, \beta_l^k, \eta_m, s_k^t) = ||r_i - s_k^t||_2 + \beta_l^k + \eta_m \tag{5.3.13}$$

where we take γ_i to be the complete state of the GPS system, s_k^t to mean the global frame position of satellite k at the time of transmission, and β_l^k to be the satellite's associated bias offset.

As with 5.2.3 and 5.2.4, this leads directly to the (in this case, 1-dimensional) probability densities:

$$\begin{aligned} P(\beta_i|\beta_{i-1}, \delta t) &= \frac{1}{\sqrt{2\pi\Upsilon_i}} \exp\left(-\frac{(\beta_i - f^*(\beta_{i-1}, \delta t))^2}{2\Upsilon_i}\right) \\ &\propto \exp\left(-\frac{1}{2}||f^*(\beta_{i-1}, \delta t) - \beta_i^\iota||^2_{\Upsilon_i}\right) \end{aligned} \tag{5.3.14}$$

$$\begin{aligned} P(\eta_i|\eta_{i-1}, \delta t) &= \frac{1}{\sqrt{2\pi\Xi_i}} \exp\left(-\frac{(\eta_i - f^*(\eta_{i-1}, \delta t))^2}{2\Xi_i}\right) \\ &\propto \exp\left(-\frac{1}{2}||f^*(\eta_{i-1}, \delta t) - \eta_i^\iota||^2_{\Xi_i}\right) \end{aligned} \tag{5.3.15}$$

[5]GPS clock offsets are typically measured in meters, meaning the time offset multiplied by the speed of light.

$$
\begin{aligned}
P(\rho_j|x_i,\gamma_i) &= \frac{1}{\sqrt{2\pi\Psi_i}}\exp\left(-\frac{\left(\rho_j - g(x_i,\gamma_j)\right)^2}{2\Psi_i}\right) \\
&\propto \exp\left(-\frac{1}{2}||g(x_i,\gamma_j)-\rho_j||^2_{\Psi_i}\right)
\end{aligned}
\tag{5.3.16}
$$

Least Squares Formulation

We first gather our bias terms and observation terms into new vectors:

$$
\begin{aligned}
B &\triangleq \{\beta_0^0 \dots \beta_m^0, \beta_0^1 \dots \beta_n^1, \dots\} \\
Y &\triangleq \{\rho_0 \dots \rho_n\} \\
H &\triangleq \{\eta_0 \dots \eta_n\}
\end{aligned}
\tag{5.3.17}
$$

We can now write a joint probability model for the system:

$$
\begin{aligned}
P(B,H,Y,X,U,Z,L) = \\
&\underbrace{\prod_{i=1}^{k} P(\beta_0^i)}_{\text{Bias priors}} \underbrace{\prod_{i=1}^{m}\prod_{j=1}^{n_i} P(\beta_j^i|\beta_{j-1}^i,\delta t)}_{\text{Bias process model}} \underbrace{\prod_{i=1}^{m} P(\eta_i|\eta_{i-1},\delta t)}_{\text{Clock offset process model}} \underbrace{\prod_{i=1}^{\kappa} P(\rho_i|x_{j_i},\eta_j,\beta_k^s,\gamma)}_{\text{GPS observations}} \\
&\underbrace{\prod_{i=1}^{M} P(x_i|x_{i-1},u_i)}_{\text{Motion process model}} \underbrace{\prod_{i=1}^{K} P(z_i|x_j,l_k)}_{\text{Landmark observations}}
\end{aligned}
\tag{5.3.18}
$$

Θ, the vector of unknowns, is expanded to include the state of satellite biases at observations times:

$$
\begin{aligned}
\Theta &\triangleq (B,H,X,L) \\
\Theta^* &\triangleq \underset{\Theta}{\operatorname{argmin}} -\log P(B,H,Y,X,U,Z,L)
\end{aligned}
\tag{5.3.19}
$$

This begets the non-linear minimization problem:

$$\begin{aligned}\Theta^* = \underset{\Theta}{\operatorname{argmin}} \Bigg\{ &\sum_{i=1}^{k} ||\beta_0^i - E[\beta_0^i]||_{\Gamma}^2 + \sum_{i=1}^{m}\sum_{j=1}^{n_i} ||f^*(\beta_{j-1}^i, \delta t) - \beta_j^i||_{\Upsilon_j}^2 \\ &+ \sum_{i=1}^{m} ||f'(\eta i - 1, \delta t) - \eta^i||_{\Xi_i}^2 + \sum_{i=1}^{\kappa} ||g(x_j, \beta_k^s, \gamma) - \rho_i||_{\Psi_j}^2 \\ &+ \sum_{i=1}^{M} ||f(x_{i-1}, u_i) - x_i||_{\Lambda_i}^2 + \sum_{i=1}^{K} ||h(x_j, l_k) - z_i||_{\Sigma_k}^2 \Bigg\} \end{aligned} \tag{5.3.20}$$

The motion model and landmark observation terms of 5.3.20 have not changed from the vanilla formulation. The bias prior terms need no linearization; we simply write each of them as estimate plus a residual error:

$$\begin{aligned}\beta_0^i - E[\beta_0^i] &= \beta_0^{i,0} + \delta\beta_0^i - E[\beta_0^i] \\ &= \delta\beta_0^i - \underbrace{(E[\beta_0^i] - \beta_0^{i,0})}_{\triangleq \alpha_i}\end{aligned} \tag{5.3.21}$$

The bias and clock offset process models linearize in much the same way as the robot motion model:

$$\begin{aligned} F_i^{s,*} &\triangleq \left.\frac{\partial f^*(\beta_i^s, \delta t)}{\partial \beta_i^s}\right|_{\beta_i^{s,0}} \\ f^*(\beta_{j-1}^i, \delta t) - \beta_j^i &\approx \{f_i^*(\beta_{j-1}^{i,0}, \delta t) + F_i^{*,i-1}\delta\beta_{j-1}^i\} - \{\beta_j^{i,0} + \delta\beta_j^{i,0}\} \\ &= F_i^{*,i-1}\delta\beta_{j-1}^i - \delta\beta_j^i - \underbrace{\left(\beta_j^{i,0} - f^*(\beta_{j-1}^i, \delta t)\right)}_{\triangleq d_i} \end{aligned} \tag{5.3.22}$$

$$
\begin{aligned}
F_i' &\triangleq \frac{\partial f'(\eta_i, \delta t)}{\partial \eta_i}\Big|_{\eta_i} \\
f'(\eta_{j-1}, \delta t) - \eta_j &\approx \{f_i'(\eta_{j-1}, \delta t) + F_i' \delta\eta_{j-1}\} - \{\eta_j + \delta\eta_j\} \\
&= F_i' \delta\eta_{j-1} - \delta\eta_j - \underbrace{(\eta_j - f'(\eta_{j-1}, \delta t))}_{\triangleq \epsilon_i}
\end{aligned} \tag{5.3.23}
$$

Pseudorange observations must be linearized with respect to the relevant pose, bias, and receiver clock offset:

$$
\begin{aligned}
C_i &\triangleq \frac{\partial g(x_j, \beta_k^s, \eta_l, \gamma)}{\partial x_j}\Big|_{x_j^0} \\
D_i &\triangleq \frac{\partial g(x_j, \beta_k^s, \eta_l, \gamma)}{\partial \beta_k^s}\Big|_{x_j^0} \\
\Delta_i &\triangleq \frac{\partial g(x_j, \beta_k^s, \eta_l, \gamma)}{\partial \eta_l}\Big|_{x_j^0} \\
g(x_j, \beta_k^s, \gamma) - \rho_i &\approx \left(g(x_j^0, \beta_k^{s,0}, \gamma) + C_i \delta x_j + D_i \delta\beta_k^s + \Delta_i \delta\eta_l\right) - \rho_i \\
&= C_i \delta x_j + D_i \delta\beta_k^s + \Delta_i \delta\eta_l - \underbrace{e_i}_{\triangleq \rho_i - g(x_j^0, \beta_k^{s,0}, \eta_l, \gamma)}
\end{aligned} \tag{5.3.24}
$$

Pulling all the linearized terms together, we come to a large linear least-squares problem:

$$
\begin{aligned}
\delta^* = \underset{\delta}{\operatorname{argmin}} \Bigg\{ &\sum_{i=1}^{k} \left|\left| \delta\beta_0^i - \alpha_i \right|\right|_{\Gamma}^2 + \sum_{i=1}^{m} \sum_{j=1}^{n_i} \left|\left| F_i^{*,i-1} \delta\beta_{j-1}^i - \delta\beta_j^i - d_j^i \right|\right|_{\Upsilon_j}^2 \\
&+ \sum_{i=1}^{\kappa} ||C_i \delta x_j + D_i \delta\beta_k^s + \Delta_i \delta\eta_l - e_i||_{\Psi_j}^2 \\
&+ \sum_{i=1}^{M} \left|\left| F_i^{i-1} \delta x_{i-1} - \delta x_i - a_i \right|\right|_{\Lambda_i}^2 + \sum_{k=1}^{K} \left|\left| H_k \delta x_i + J_k \delta l_j - c_k \right|\right|_{\Sigma_k}^2 \Bigg\}
\end{aligned} \tag{5.3.25}
$$

Assuming we apply the same premultiplication described in 5.2.20, we can again

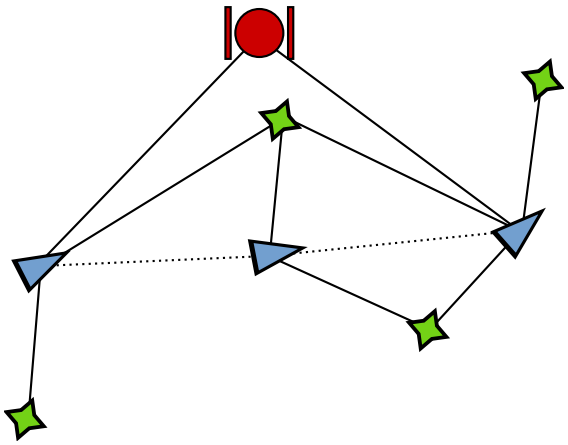

Figure 5.3.3: The scenario for 5.3.26 and 5.3.27

gather terms into a large, sparse matrix. We can revisit the scenario presented in 5.2.1, adding an observation from the same satellite in the first and third pose. The new scenario is shown in figure 5.3.3.

$$A = \begin{pmatrix}
-\Lambda_1^{-\frac{\mathrm{T}}{2}} & & & & & & & & & \\
\hat{F}_2^1 & -\Lambda_2^{-\frac{\mathrm{T}}{2}} & & & & & & & & \\
 & \hat{F}_3^2 & -\Lambda_3^{-\frac{\mathrm{T}}{2}} & & & & & & & \\
 & & & -\Xi_1^{-\frac{\mathrm{T}}{2}} & & & & & & \\
 & & & F_2'^1 & -\Xi_2^{-\frac{\mathrm{T}}{2}} & & & & & \\
 & & & & & -\Upsilon_{0,1}^{-\frac{\mathrm{T}}{2}} & & & & \\
 & & & & & \hat{F}_1^{*,0} & -\Upsilon_{1,1}^{-\frac{\mathrm{T}}{2}} & & & \\
\hat{C}_1 & & & \Delta_1 & & \hat{D}_1 & & & & \\
 & & \hat{C}_2 & & \Delta_2 & & \hat{D}_2 & & & \\
\hat{H}_1 & & & & & & & \hat{J}_1 & & \\
\hat{H}_2 & & & & & & & & \hat{J}_2 & \\
 & \hat{H}_3 & & & & & & & \hat{J}_3 & \\
 & \hat{H}_4 & & & & & & & & \hat{J}_4 \\
 & & \hat{H}_5 & & & & & & \hat{J}_5 & \\
 & & \hat{H}_6 & & & & & & & \hat{J}_6 \\
 & & \hat{H}_7 & & & & & & & \hat{J}_7
\end{pmatrix} \tag{5.3.26}$$

$$\delta = \begin{pmatrix} \delta x_1 \\ \delta x_2 \\ \delta x_3 \\ \delta \beta_0^1 \\ \delta \beta_1^1 \\ \delta \eta_1 \\ \delta \eta_2 \\ \delta l_1 \\ \delta l_2 \\ \delta l_3 \\ \delta l_4 \end{pmatrix} \qquad b = \begin{pmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{a}_3 \\ \hat{\epsilon}_1 \\ \hat{\epsilon}_2 \\ \hat{\alpha}_1 \\ \hat{d}_1^1 \\ \hat{d}_2^1 \\ \hat{c}_1 \\ \hat{c}_2 \\ \hat{c}_3 \\ \hat{c}_4 \\ \hat{c}_5 \\ \hat{c}_6 \\ \hat{c}_7 \end{pmatrix} \tag{5.3.27}$$

In the absence of external information, the linearization points for the satellite bias terms are seeded with 0, which is their expected value. The clock offset can be initialized by interpolating the calculated receiver offset when fixes are available.[6]

Integration of Differential Corrections

As was discussed in 3.4, we can substantially improve the accuracy of our GPS readings by incorporating differential corrections from a fixed base station at a known location. As illustrated by equation 3.4.1, using a differential correction can remove a substantial portion of errors introduced by the local environment. However, use of the differential

[6]Note that receivers take varying approaches to managing receiver clock offset. Some systems discipline their clocks to GPS time, while others allowed the receiver clock to drift freely. Inevitably receivers of the latter type have discontinuities in clock bias estimates; typically the estimate wraps around at $\pm .5\text{ms} \times c$. This presents no particular difficulty, but must be taken into account when calculating offset innovations.

correction introduces the clock offset of the base station as a new parameter in the system. There are several ways we can proceed.

If we always have differential corrections to a given satellite, then there is a clear path forward. In this case, we have no need for the clock offsets of the base station and the roving station to be independently observable; we can treat them as an aggregate parameter to be estimated. In this case, no modifications to the underlying derivation need to be made, although the specific parameters of the bias process model and observation noise would change accordingly.

Ideally, we want to be able to use this differential information opportunistically. Due to differences in sky visibility and hardware limits on the number of satellites a given receiver can track, it is common for readings from a satellite to have corrections available some but not all of the time.

To accommodate this opportunistic usage of information, we must add parameters to the model representing the clock offset of the base receiver at each time differential corrections are used.

The base station receiver clock offset has an independent process model with the same form as 5.3.10. If we take as our differential observations the difference between the measured pseudorange to a satellite and the known range to the satellite, we have the following observation model:

$$g^*(\beta_l^k, \eta'_m) = \beta_l^k + \eta'_m \tag{5.3.28}$$

where η'_m is the clock offset of the base station at time m. By applying the appropriate variance premultiplication, these extra differential terms can be folded directly into equation 5.3.25.

Conceptually, this can be thought of as using the base station differential readings as observations of the bias terms in the model. When we have differential corrections, we can place tighter bounds on the bias terms. When we don't have differential corrections available, we can still use the information with the looser bias model.

5.4 Conclusion

In this chapter, we have presented a thorough derivation of a fused SAM-GPS system. This model intelligently accounts for the nonwhite error models of a GPS system and allows for the incorporation of differential corrections on an opportunistic basis. In the next chapter we will give a real-world example implementation of the framework.

Chapter 6

Application: Integrated GPS-SAM using Navlab 11

6.1 Introduction

In chapter 5 we established a theoretical framework for GPS-SAM integration. In this chapter, we present a concrete implementation of the framework on a real-world robot. The maps built use more than 100 km of robot trajectories. We demonstrate loop closure after arbitrarily large explorations and show how time-correlated GPS data affects landmark estimation via revisitation. The same data is also used to show how differential corrections can also be integrated into the model to improve accuracy and lower convergence rates.

The platform for this test is the Navlab 11 vehicle. Navlab 11, pictured in 6.1.1, is a street-legal Jeep Wrangler which carries a wide variety of sensors.

The sensing used in this work is provided by:

- encoders on each of the 4 wheels.
- a Trimble AgGPS 114 using a roof-mounted choke ring antenna.
- 2 SICK LMS laserscanners providing range returns parallel to the ground plane on the left and right sides of the vehicle.

Figure 6.1.1: The Navlab 11 vehicle

- a strapped-down KVH-5000 fiber optic gyroscope for yaw measurements.
- a strapped-down Crossbow VG400 3-axis inertial measurement unit.

In the configuration used, the vehicle does not provide steering angle information.

6.2 Coordinate Frames

The "native" GPS earth-fixed Cartesian frame could be used as our global frame. However, it will be more convenient to define our earth-fixed frame as one in which the z axis is parallel to the local gravity vector at a point central to our work. This frame is simply a rotation of the canonical ECEF frame, and is defined rather than estimated.

This effectively defines our own projection of the surface of the earth, assuming the world is locally flat. For our purposes, this is fine, but for very large scale applications this would need to be revisited; the alignment of the "correct" frame shifts approximately 1 degree per 100km away from the central point.

The coordinate frame tied to the vehicle is defined with x forward, y right, and z down, with its origin at the midpoint of the rear axle.

6.3 Vehicle State

The state of the vehicle is defined to be:

$$x = (g_x, g_y, g_z, \phi, \theta, \psi)^{\mathrm{T}} \tag{6.3.1}$$

where the first three terms are the position of the vehicle in the global frame, and the latter three terms are the roll, pitch, and yaw of a Tait-Bryan system.

From a given $(\phi, \theta, \psi)^{\mathrm{T}}$, we can create a matrix which will rotate a vehicle-frame vector into the global frame:

$$\begin{aligned} R_v^g &= \begin{pmatrix} \mathrm{c}\psi & -\mathrm{s}\psi & 0 \\ \mathrm{s}\psi & \mathrm{c}\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathrm{c}\theta & 0 & -\mathrm{s}\theta \\ 0 & 1 & 0 \\ \mathrm{s}\theta & 0 & \mathrm{c}\theta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \mathrm{c}\phi & -\mathrm{s}\phi \\ 0 & \mathrm{s}\phi & \mathrm{c}\phi \end{pmatrix} \\ &= \begin{pmatrix} \mathrm{c}\theta\mathrm{c}\psi & \mathrm{c}\psi\mathrm{s}\theta\mathrm{s}\phi - \mathrm{c}\phi\mathrm{s}\psi & \mathrm{c}\phi\mathrm{c}\psi\mathrm{s}\theta + \mathrm{s}\phi\mathrm{s}\psi \\ \mathrm{c}\theta\mathrm{s}\psi & \mathrm{c}\phi\mathrm{c}\psi + \mathrm{s}\theta\mathrm{s}\phi\mathrm{s}\psi & -\mathrm{c}\psi\mathrm{s}\phi + \mathrm{c}\phi\mathrm{s}\theta\mathrm{s}\psi \\ -\mathrm{s}\theta & \mathrm{c}\theta\mathrm{s}\phi & \mathrm{c}\theta\mathrm{c}\phi \end{pmatrix} \end{aligned} \tag{6.3.2}$$

As our inputs to the system, we take the odometry and angular rates provided by the gyroscopes:

$$u = (v, \dot{\omega}_x, \dot{\omega}_y, \dot{\omega}_z)^{\mathrm{T}} \tag{6.3.3}$$

where v is the speed of the vehicle in the vehicle-frame x direction and the $\dot{\omega}$ terms denote angular rates around the vehicle-frame axes. These vehicle-frame angular rates

can be transformed into global-frame roll-pitch-yaw rates using:[1]

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & \mathrm{s}\phi\mathrm{t}\theta & -\mathrm{c}\phi\mathrm{t}\theta \\ 0 & \mathrm{c}\phi & \mathrm{s}\phi \\ 0 & -\frac{\mathrm{s}\phi}{\mathrm{c}\theta} & \frac{\mathrm{c}\phi}{\mathrm{c}\theta} \end{pmatrix}}_{\triangleq \Omega_v^g} \begin{pmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{pmatrix} \tag{6.3.4}$$

This brings us to our motion update rule:

$$\begin{aligned} f(x, u, \delta t) &= \begin{pmatrix} \begin{pmatrix} g_x \\ g_y \\ g_z \end{pmatrix} + R_v^g \begin{pmatrix} v \\ 0 \\ 0 \end{pmatrix} \\ \begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix} + \Omega_v^g \begin{pmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{pmatrix} \delta t \end{pmatrix} \\ &= \begin{pmatrix} g_x + d\mathrm{c}\theta\mathrm{c}\psi \\ g_x + d\mathrm{c}\theta\mathrm{s}\psi \\ g_z - d\mathrm{s}\theta \\ \phi + \dot{\omega}_x - \dot{\omega}_z\mathrm{c}\phi\mathrm{t}\theta + \dot{\omega}_y\mathrm{s}\phi\mathrm{t}\theta \\ \theta + \dot{\omega}_y\mathrm{c}\phi + \dot{\omega}_z\mathrm{s}\phi \\ \psi + \frac{(\dot{\omega}_z\mathrm{c}\phi - \dot{\omega}_y\mathrm{s}\phi)}{\mathrm{c}\theta} \end{pmatrix} \end{aligned} \tag{6.3.5}$$

Recall that SAM uses motion model additive noise in the pose space with covariance Λ. We have control space variance:

$$\Sigma_u = \begin{pmatrix} \sigma_d & 0 & 0 & 0 \\ 0 & \sigma_{\dot{\omega}_x} & 0 & 0 \\ 0 & 0 & \sigma_{\dot{\omega}_y} & 0 \\ 0 & 0 & 0 & \sigma_{\dot{\omega}_z} \end{pmatrix} \tag{6.3.6}$$

We linearize around the current pose estimate to transform this covariance into the

[1] The derivation of this matrix can be found in [Kelly, 1994] pp 33-34.

pose space:

$$W \triangleq \frac{\partial f(x,u)}{\partial u}\Big|_{\hat{x}}$$
$$\Lambda = W\Sigma_u W^{\mathrm{T}} \tag{6.3.7}$$

However, SAM requires an inverse Λ, which means Λ must be full rank. 6.3.7 will result in a matrix which is rank 4 at most. In reality the uncertainty between poses is always going to span the full dimensionality of the pose space.

We can skirt this problem by using several intermediate linearizations of the variance between pose steps.

The frequency of pose estimates is an engineering decision which is constrained by the rate of observations, the linearity of the motion, and the size of the overall system to be smoothed. Clearly, we need a pose estimate at each observation time. Additionally, using smaller time-steps between pose estimates improves our linear approximation to the true nonlinear motion covariances.

However, fewer pose states decrease the size of our model, which is good for computational efficiency. There is also numerical stability to consider. If the uncertainty between two pose states is too small, the inverse covariance used in the least squares system becomes unstable.

After empirical observations, the minimum update frequency was set to 10 Hz, with a maximum update frequency of 20 Hz.

6.4 Landmark observations

Landmarks are identified using two SICK LMS laserscanners mounted on opposite sides of the vehicle. We search for strong "corners" in the plane of the scan, where a corner is defined to be a pair of line segments for which two endpoints are within a threshold distance, and the interior angle is within a threshold range. Interior angles which are too small are discarded as likely noisy reading, and angles which are too large are discarded because they diminish the accuracy with which the location of the corner can

be calculated.

The idea of such a model is well-known; however, several heuristics are used to help generate high-quality landmarks. These heuristics are fairly specific to the environment in which the robot operated.

A "good" landmark is one which is stationary and which we can find again when we revisit the area in the future. In the data presented, by far the largest source of "bad" landmarks is vehicles. Stationary buses, in particular, present very strong, linear corners which are very difficult to distinguish from buildings or other permanent structures.

To reduce the number of vehicles added to the system as landmarks, we implement some basic filters on any landmark candidates. Any landmark which forms an interior angle close to 90 degrees is suspect, and is required to pass through more gates before it is considered to be a true landmark observation. The "suspicious" landmark will be included only if it is large enough that it is unlikely to be a vehicle, or it is concave with respect to the scanner. The landmark decision flow is illustrated in figure 6.4.1. Some common scenarios are illustrated in figure 6.4.2.

Because the laserscanners being used sense only in a 2-dimensional plane, we make as assumption, based on qualitative observations of urban scenes, that corner features extend infinitely in the vertical direction. In other words, our expectation is that the global x–y location of the landmark is constant regardless of the height at which the landmark is observed.

$$l^k = \left(l_x^k, l_y^k\right)^{\mathrm{T}} \tag{6.4.1}$$

It is helpful to think of landmarks as vertical lines in the global space; we expect to find a corner when a sensor observes that line at any height.

An observation i on sensor j is parameterized in Cartesian form in the sensor frame. Note that the discussion of observations is specific to a sensor, but extra subscripting has been dropped for the sake of notational simplicity.

$$z_i = \left(z_{i,x}, z_{i,y}\right)^{\mathrm{T}} \tag{6.4.2}$$

Consider the (homogeneous) sensor matrix for converting from the sensor frame to

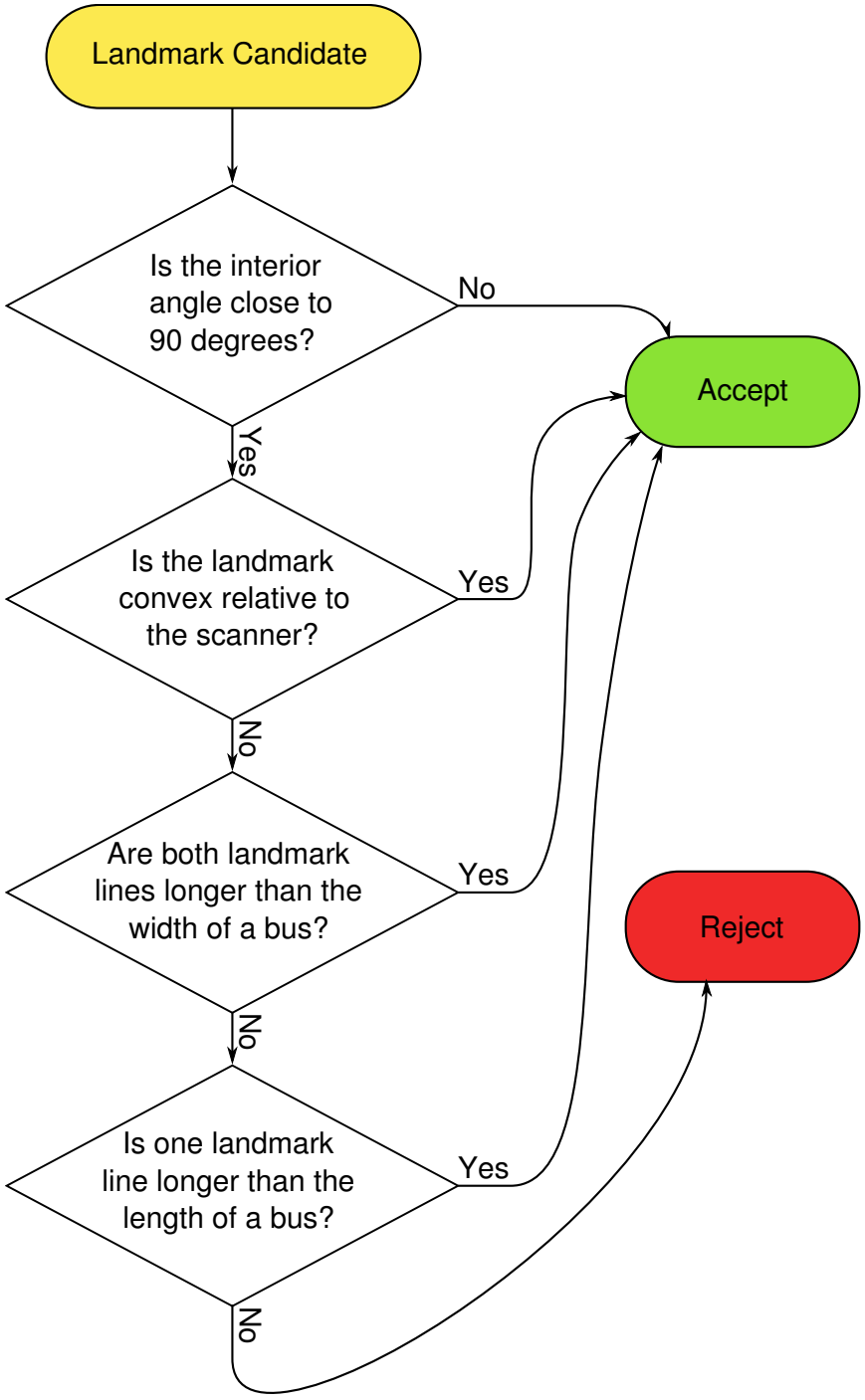

Figure 6.4.1: Landmark filtering. These gates are used to minimize vehicles inappropriately being incorporated into the map as landmarks.

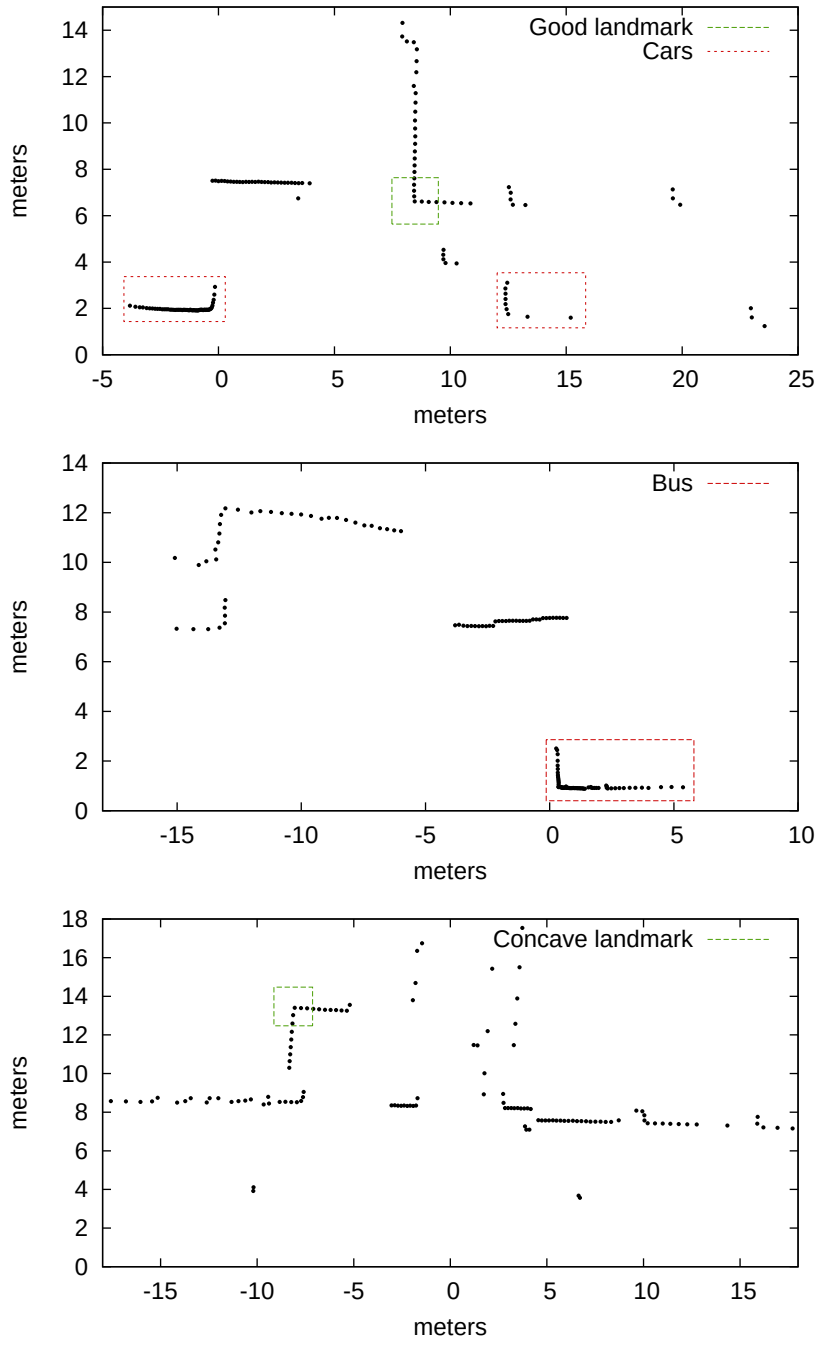

Figure 6.4.2: Common scan scenarios. The scanner is at (0, 0).

the vehicle frame:

$$S = \begin{pmatrix} s_{11} & s_{12} & s_{13} & s_{14} \\ s_{21} & s_{22} & s_{23} & s_{24} \\ s_{31} & s_{32} & s_{33} & s_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{6.4.3}$$

A scan is coincident with the x–y plane of the corresponding sensor frame. Equation 6.3.2 gives the rotation from the vehicle frame to the global frame. From this, we can create the homogeneous transform from the sensor frame to the global frame:

$$T_g^j = \begin{pmatrix} & & & g_x \\ & R_g^v & & g_y \\ & & & g_z \\ 0 & 0 & 0 & 1 \end{pmatrix} S^j \tag{6.4.4}$$

If landmarks were 3-dimensional points, our observation model would come directly from the inverse of this transform. However, because our landmark model is not fully constrained in 3 dimensions, we need to find the intersection of the sensor plane with the landmark "line" in global coordinates.

If we define $P^* = (l_x, l_y, \lambda)^{\mathrm{T}}$ to be the global-frame point at which the landmark is expected to be observed, then we proceed by solving for λ in terms of the the landmark position and the vehicle pose. The most straightforward way to do this requires a global-frame normal to the sensor plane, and a point on that plane. A global-frame normal can be found with:

$$N = R_g^v \begin{pmatrix} s_{13} \\ s_{23} \\ s_{33} \end{pmatrix} \tag{6.4.5}$$

The most readily available global-frame point on the sensor plane is the origin of the

sensor frame:

$$P = T_g^j \begin{pmatrix} s_{14} \\ s_{24} \\ s_{34} \\ 1 \end{pmatrix} \tag{6.4.6}$$

$P^* - P$ lies in the sensor plane, and N is normal to that plane. Assuming P and P^* are not the same point, it must be true that:

$$N \cdot (P^* - P) = 0 \tag{6.4.7}$$

Expanding 6.4.7 and solving for λ, we find:

$$\begin{aligned} \lambda = [&-s_{13}s_{14} - s_{23}s_{24} - s_{33}s_{34} + s_{13}g_z\mathrm{s}\theta + (s_{23}\mathrm{c}\phi - s_{33}\mathrm{s}\phi)((l_y - g_y)\mathrm{c}\psi + (x - l_x)\mathrm{s}\psi) \\ &+ \mathrm{s}\theta(s_{33}\mathrm{c}\phi + s_{23}\mathrm{s}\phi)((l_x - g_x)\mathrm{c}\psi + (l_y - g_y)\mathrm{s}\psi) \\ &- \mathrm{c}\theta(s_{33}g_z\mathrm{c}\phi + s_{13}(g_x - l_x)\mathrm{c}\psi + s_{23}g_z\mathrm{s}\phi + s_{13}(g_y - l_y)\mathrm{s}\psi)] \\ &/(s_{13}\mathrm{s}\theta - \mathrm{c}\theta(s_{33}\mathrm{c}\phi + s_{23}\mathrm{s}\phi)) \end{aligned} \tag{6.4.8}$$

In the case of the Navlab vehicle, the sensor frames of the relevant laser scanners are aligned in z axis with the vehicle frame. This means that $(s_{13}, s_{23}, s_{33}) = (0, 0, 1)$, which leads to considerable simplification of 6.4.8:

$$\lambda = \frac{s_{34} + \mathrm{s}\phi((l_y - g_y)\mathrm{c}\psi - (l_x - g_x)\mathrm{s}\psi) - \mathrm{s}\theta\mathrm{c}\phi((l_x - g_x)\mathrm{c}\psi + (l_y - g_y)\mathrm{s}\psi) + \mathrm{c}\theta\mathrm{c}\phi g_z}{\mathrm{c}\theta\mathrm{c}\phi} \tag{6.4.9}$$

And at last we have all the pieces to state our landmark observation model:

$$
\begin{aligned}
h(x,l) &\triangleq (T_g^j)^{-1} \begin{pmatrix} l_x \\ l_y \\ \lambda \end{pmatrix} [1:2] \\
&= \begin{pmatrix} \frac{\mathrm{c}\phi((l_x-g_x)\mathrm{c}\psi+(l_y-g_y)\mathrm{s}\psi)-s_{34}\mathrm{s}\theta+\mathrm{s}\theta\mathrm{s}\phi((l_x-g_x)\mathrm{s}\psi-(l_y-g_y)\mathrm{c}\psi)}{\mathrm{c}\phi\mathrm{c}\theta} \\ \frac{((l_y-g_y)\mathrm{c}\psi+s_{34}\mathrm{s}\phi-(l_x-g_x)\mathrm{s}\psi)}{\mathrm{c}\phi} \end{pmatrix}
\end{aligned}
\tag{6.4.10}
$$

Here the $[1:2]$ is taken to mean the first two elements of the resulting vector.

6.5 Data Association

Data association is a critical part of any landmark-based mapping algorithm. When we observe a landmark, the answer to the question "Have I seen this landmark before, and if so, where?" has a dramatic effect on the map estimate. When mapping works well, robots close loops and suddenly the uncertainty of the pose and recently seen landmark locations is sharply reduced. When associations go awry, the results are frequently catastrophic, sometimes irrecoverably so.

For this work, observation associations are divided into two categories: *trivial* and *loop closures*. A trivial association is one in which the observation is associated with a landmark which has been seen "recently". If the robot sees a landmark, moves a small amount, then sees the same landmark again, the relative uncertainty of observation and landmark is extremely low. In our particular case, where the density of landmark observations is low compared to odometry accuracy, these trivial data associations can be implemented by keeping a set of recently observed landmarks and using simple thresholding for association of observations with this set.

Loop closures are more complicated; when the relative position of the vehicle and a landmark is not tightly constrained, it is more difficult to find correct associations. This work implements two strategies to increase data association accuracy: out-of-band information and joint compatibility tests.

Out-of-band information refers to the idea, argued for in [Ho and Newman, 2007], that use of information sources other than positional compatibility to determine associations is good practice in a mapping system. With our corner-based landmarks, we have two candidate external information sources: the interior angle and the global orientation of the corner. Interior corner angle is certainly helpful in disambiguating landmarks. Barring extreme pitching or rolling of the vehicle, we expect that the interior angle of observations of a landmark will remain consistent. Orientation would not be useful as a stand-alone association compatibility metric in a system which has unbounded yaw error; the "expected" orientation observation is compounded by the robot's yaw error.[2] With GPS, however, our expectations vis-a-vis yaw error are no longer unbounded error during exploration; this makes stand-alone use of observation orientation feasible.

Generating robust in-band loop closure information requires a representation of the relative uncertainties. [Golub and Plemmons, 1979] presents some results that are helpful to us in recovering those portions of the covariance needed to do data association. Golub shows that, given a Cholesky factorization of the inverse covariance:

$$\Sigma^{-1} = LL^{\mathrm{T}} \tag{6.5.1}$$

where L is $n \times n$ and lower triangular, then entries of the covariance can be recursively calculated using the following recurrences:

$$\sigma_{ii} = \frac{1}{l_{ii}} \left(\frac{1}{l_{ii}} - \sum_{j=1}^{n} l_{ji} \sigma_{ji} \right) \tag{6.5.2}$$

$$\sigma_{ij} = \frac{1}{l_{ii}} \left(- \sum_{k=i+1}^{j} l_{ki} \sigma_{kj} - \sum_{k=j+1}^{n} l_{ki} \sigma_{jk} \right) \tag{6.5.3}$$

Note that the summation terms pair sparse L entries with covariance entries. This limits the number of covariance entries required to recover a given entry.

As shown in [Kaess, 2008] this recursive recovery can be efficiently implemented using dynamic programming by memoizing values of the covariance matrix as they are calculated. This approach is used to find relevant covariance entries in this work.

[2] One possibility (not explored here) in such a system is to use orientation in a joint compatibility test

In a classical mapping system, the uncertainty of the pose estimate increases without bound until the robot revisits a previously seen landmark. In the limit, the long-wandering robot has effectively useless information about its current global position and must consider the possibility that any new observation could correspond with an arbitrarily large set of previously seen landmarks. We can increase confidence in association by working with sets of observations and landmarks as a unit; this is the basic idea behind techniques such as joint compatibility branch and bound (JCBB) [Neira and Tardos, 2001].

One obvious and immediate benefit of folding GPS into a SAM system is the global bounding of pose uncertainty; no matter how long the robot wanders without revisiting an area, GPS provides pose information from an absolute reference. When using a technique such as JCBB, no modifications need to be made to the algorithm; the thresholds used naturally incorporate the additional GPS information.

However, urban environments have some characteristics which make them particularly prone to occasional misassociation even when using joint compatibility techniques. As illustrated in figure 6.5.1, it is common to find sets of landmarks in repeating configurations, which lead to high joint compatibility assignments, and can lead to association failure.

6.6 Results and Analysis

To demonstrate the scalability of the GPS-SAM fusion system, it was run on a dataset consisting of approximately 6 hours of data gathered primarily from the Shadyside and Friendship neighborhoods in Pittsburgh. The total trajectory length of the vehicle is approximately 115 kilometers.

First, we present some visual representations of the system's output. Figure 6.6.1 shows the entirety of the dataset with raw scan data shown in red, landmarks in green, and the trajectories of the robot in blue. Figure 6.6.2 shows the same data overlaid upon aerial imagery of the mapped region.

Moving closer, figure 6.6.3 shows a detail segment from the larger map. The sources of landmarks become visible at this level of detail, and it is easier to see the trajectories

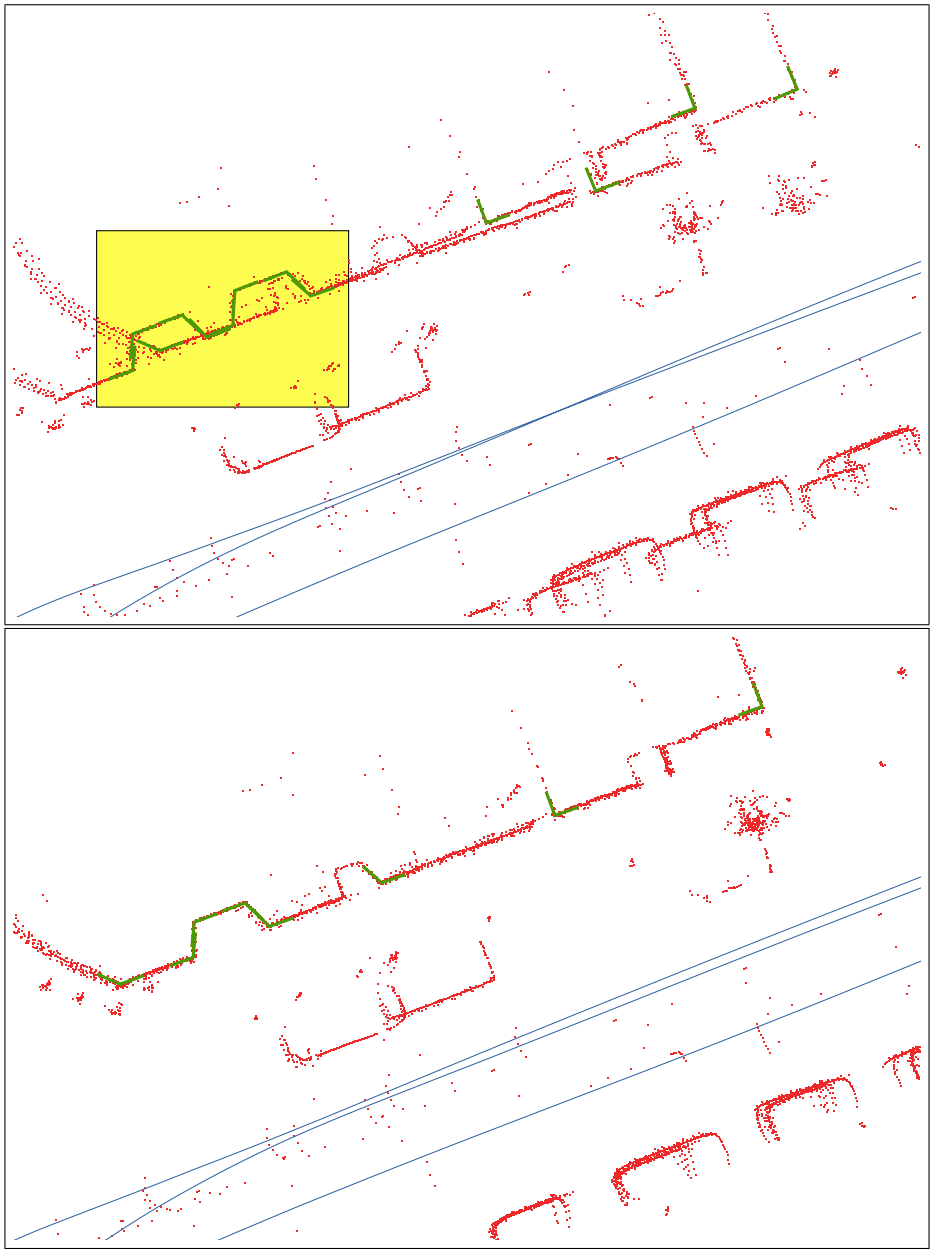

Figure 6.5.1: Association failure. Urban environments can have ambiguously repeating landmark patterns. The highlighted area shows one such area. Even using joint compatibility methods, it is difficult to associate such patterns correctly. The bottom diagram shows the same area with a corrected association.

of the robot.

Convergence Rates

As has been noted, the SAM technique solves a linearized version of the least squares problem 5.3.20. The accuracy of this linear solution depends on linearization points being of sufficient accuracy to make the linear assumption reasonable for the region between the linearization point and the desired maximization.

The addition of more variables increases the nonlinearity of the underlying system. As can be seen in the generated maps, this doesn't cause the incremental linearized solutions to diverge from the nonlinear maximum a posteriori result, but it is reasonable to expect that the additional terms being estimated will slow convergence.

Figure 6.6.4 illustrates the convergence rate of SAM applied to this dataset. Although initial corrections move quickly toward the final value, it can take up to 10 iterations before all estimated variables move less than 1 cm after a smoothing pass.

Maintaining Sparsity

Finding a good variable ordering is critical to keeping the system sparse and efficient. [Dellaert and Kaess, 2006] explores this topic, and suggests using the column approximate minimum degree ordering (*colamd*) described by [Davis *et al.*, 2004] to find a good ordering. In particular, it is noted that calling *colamd* on groups of related parameters (i.e. considering all variables from a given pose as a unit) is a way to use domain knowledge to reduce fill-in in the factored representation.

It is unclear how GPS-related variables should be treated with respect to variable ordering. As such, several experiments were run using different reordering strategies to see the results on the factored representation.

The first case, presented as a baseline, does no grouping of variables; *colamd* is left free to consider all variables individually. In the second case, each pose is considered as a single column, as is each landmark. All receiver clock bias terms and satellite bias terms are left ungrouped. In the third case, in addition to the groupings of the second

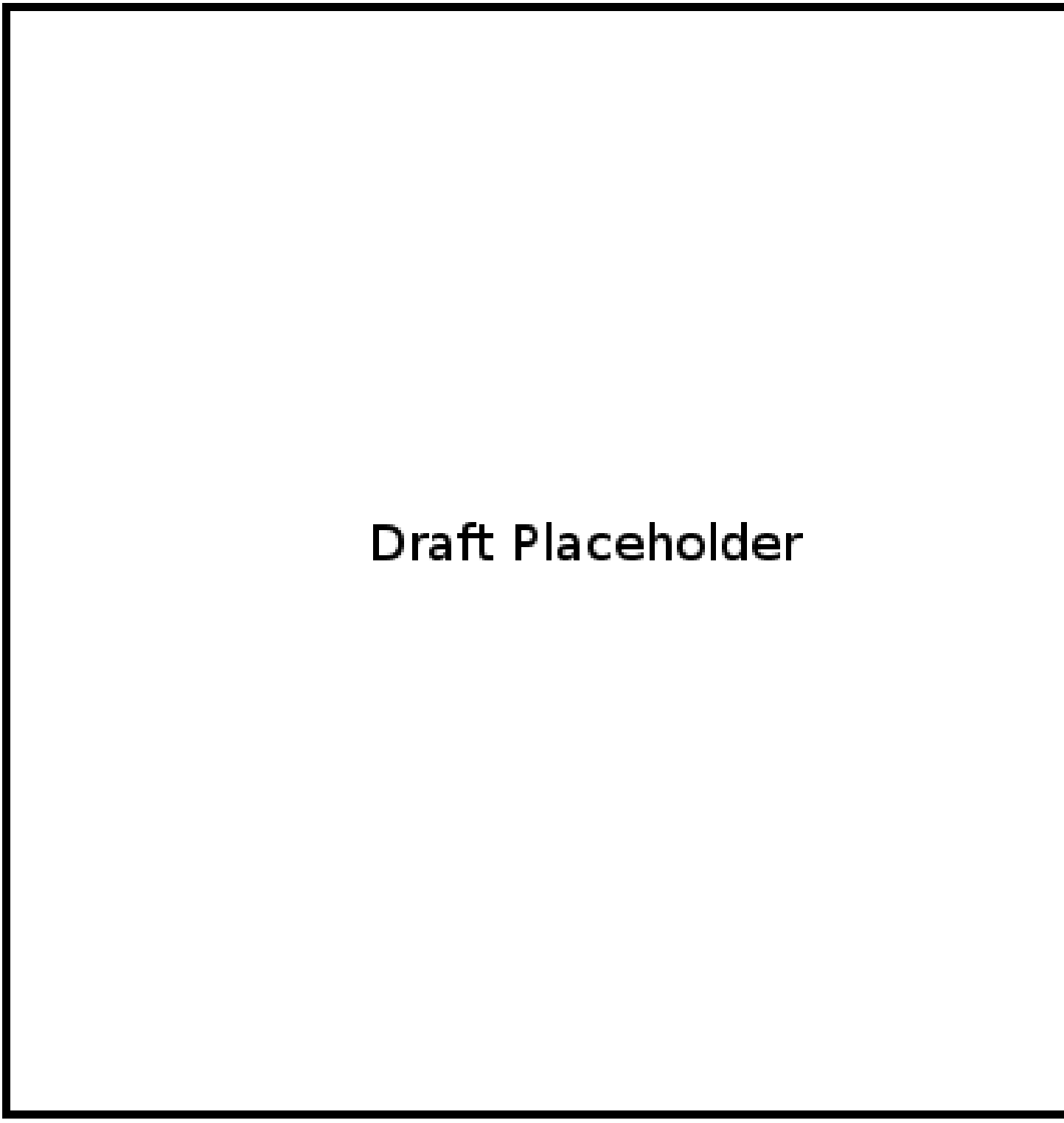

Figure 6.6.1: Entire aligned dataset

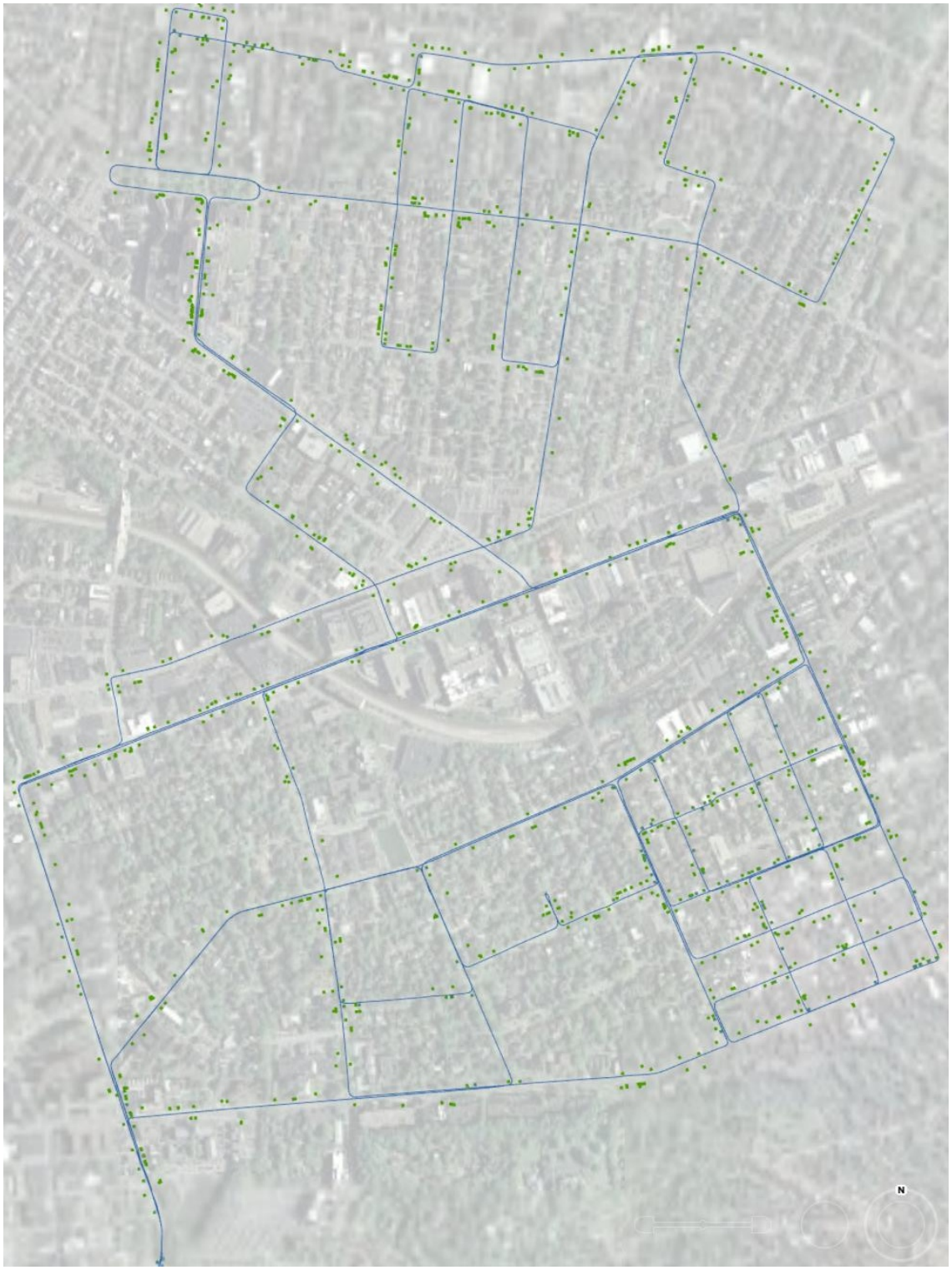

Figure 6.6.2: Satellite imagery overlay

Figure 6.6.3: Map detail overlaid on aerial imagery. North is to the right of the page.

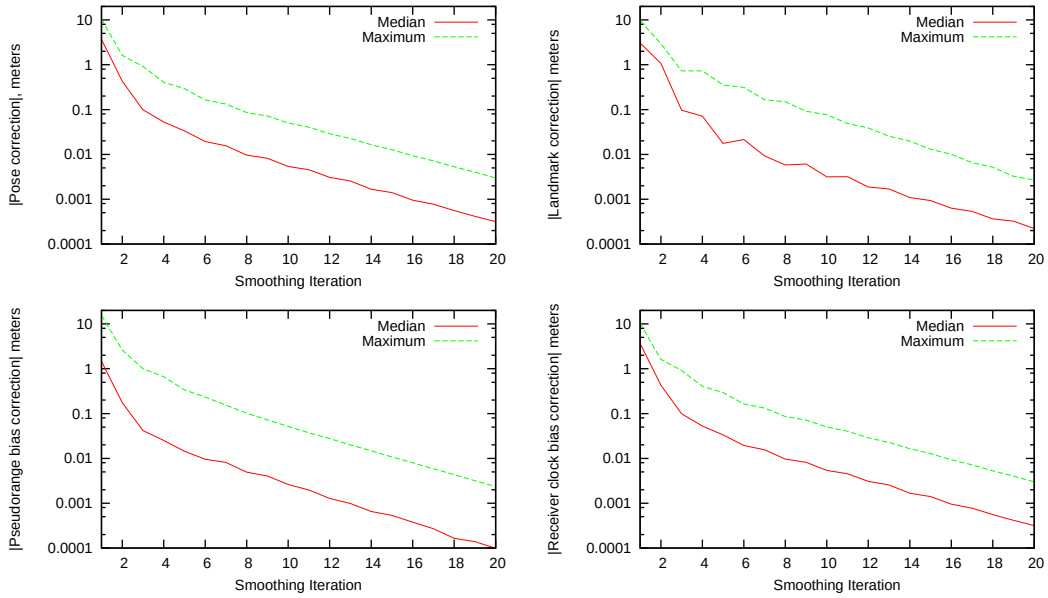

Figure 6.6.4: Convergence rates of the various types of parameters being estimated in the Pittsburgh dataset without differential corrections. Note the log scale.

case, GPS terms from the same epoch are grouped together. In the final case, GPS terms are grouped together with the pose of the same epoch.

The (unpermuted) A matrix of the system can be seen in 6.6.5. The resulting Cholesky factors of the information matrix are shown in figure 6.6.6. Interestingly, using pose and landmark groupings is helpful, though the large number of poses relative to observations makes the 30% reduction in fill-in less dramatic than other published results. Grouping GPS terms by epoch appears to be detrimental to fill-in, though grouping epochs and poses into a single group is significantly better than grouping poses and epochs independently.

6.7 Using Local Area Differential Corrections

The University of Pittsburgh Department of Geology and Planetary Science maintains a continuously operating reference station, named papt, in close proximity to the Pittsburgh datasets. They generously publish make this data freely available for download.

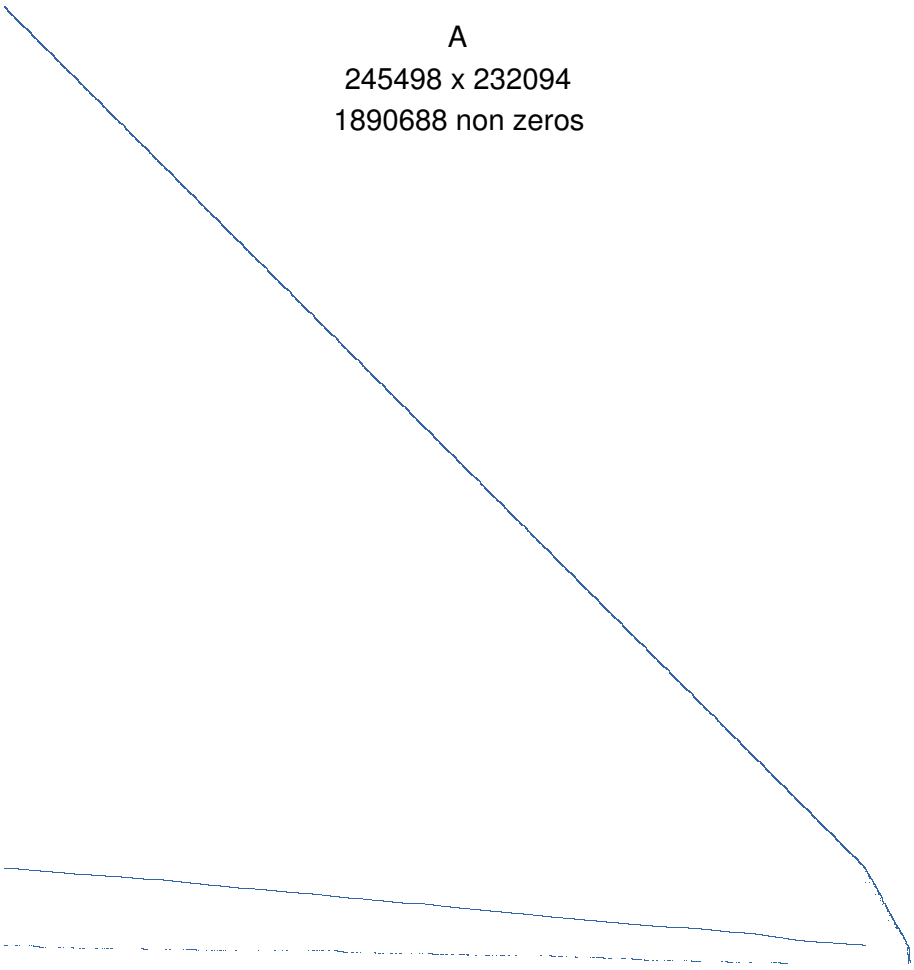

Figure 6.6.5: The A matrix of the full mapping run. The top rows of the matrix are the pose updates. These are followed by the GPS observations and then the landmark observations. Columns are poses, then clock offsets and satellite biases, then landmark positions.

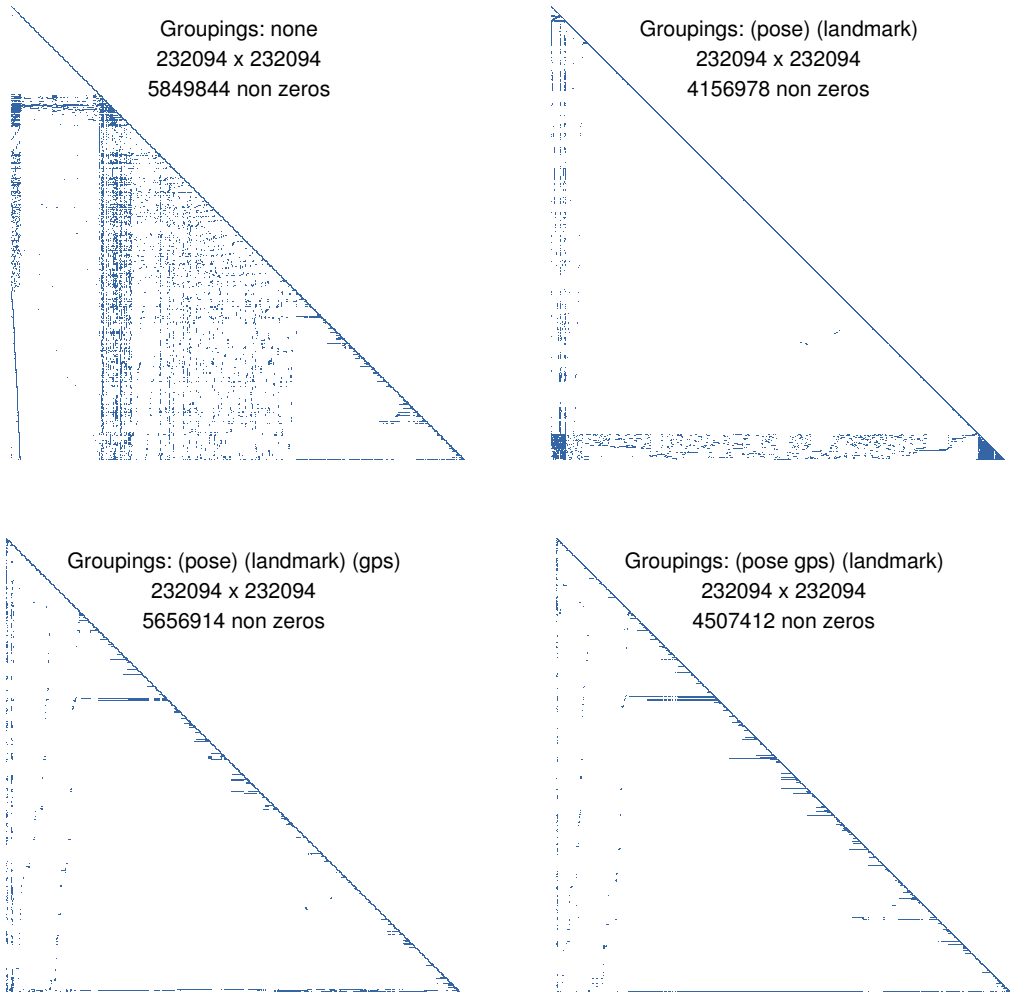

Figure 6.6.6: L factors of $A^{\mathrm{T}}A$ for various variable grouping strategies

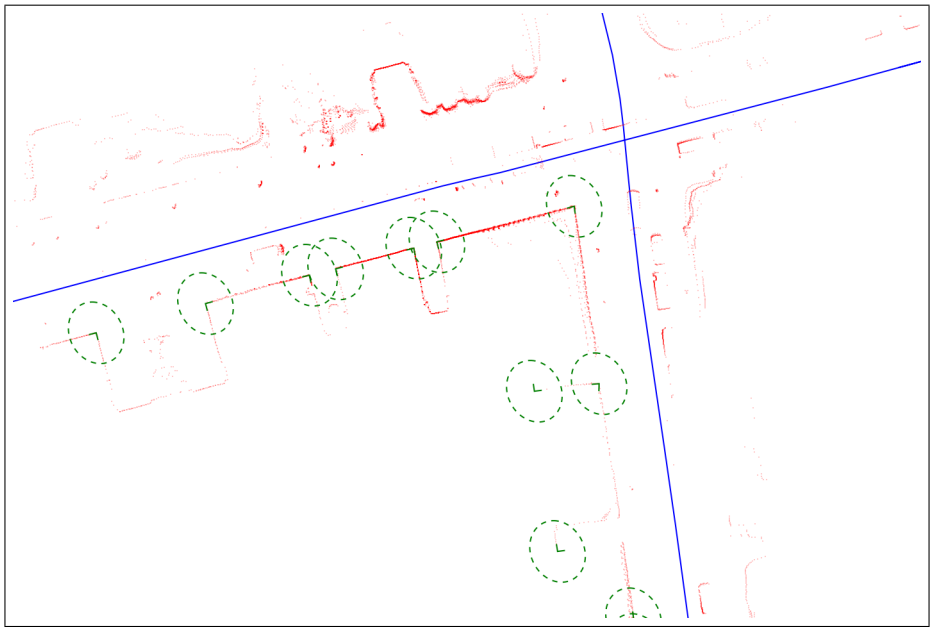

Figure 6.7.1: Map of an infrequently visited neighborhood without differential corrections. 2-σ error ellipses around landmarks are shown.

Using the reference station logs, LADGPS readings can be opportunistically incorporated into the model as described in 5.3. Doing so results in a significant reduction in the uncertainty of our map. Note that no coordination between base station and mobile receiver is required to use the differential corrections; we merely take advantage of any overlaps in satellite selection as they occur.

The reduction in uncertainty only becomes obvious when we add error ellipses to our output, as has been done in figures 6.7.1 and 6.7.2. Generation of these error estimates is made possible by the dynamic programming algorithm presented in [Kaess, 2008].

The base station also can act as a source of ground truth to help validate the output of the algorithm. Ideally, we would have a statistically significant number of our map landmarks surveyed with a high degree of confidence. However, the integrated SAM-GPS model makes no distinction between the estimated per-satellite biases and landmark positions in its estimation process. If we can show that the per-satellite bias estimates are valid, this indirectly improves our confidence in the landmark estimates.

We do not have precise ground truth data from the base station, we do have a

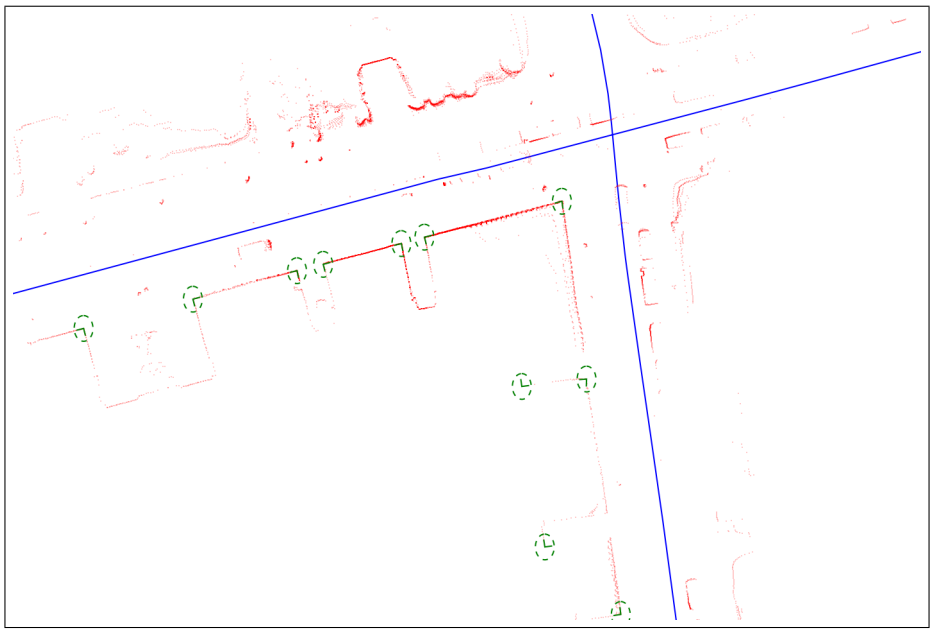

Figure 6.7.2: The same neighborhood as the previous figure, but including 30-second interval local area differential corrections from the papt base station. 2-σ error ellipses around landmarks are shown.

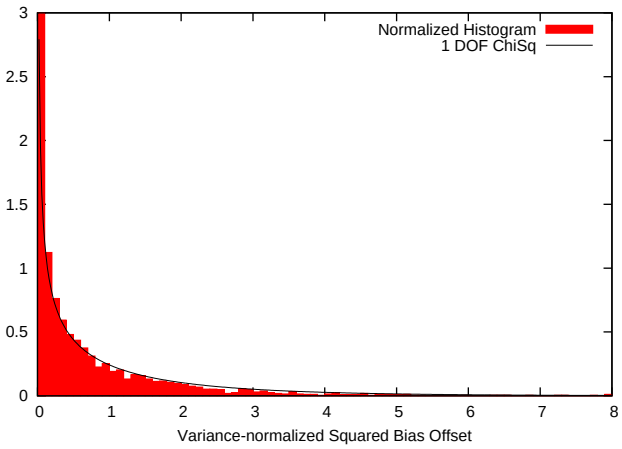

Figure 6.7.3: Normalized histogram of variance-normalized satellite offsets.

source of data with well-characterized error. [Parkinson *et al.*, 1996]. The sources of noise from the base station are not perfectly white, but base station autocorrelation effects are negligible given a sufficiently long time-series such as the one gathered here.

In section 4.5, we took advantage of the ground truth available in simulation to generate a set of squared errors normalized by variance. We can follow a similar process here. We start by smoothing a map without using the papt differential corrections. This generates approximately 130,000 bias estimates at a minimum interval of 1 second.

Of these readings, we have 2,663 readings in common with the 30-second interval papt base station logs. The square of the offset of the base station derived offset and the SAM-estimated offset is normalized by the sum of the SAM-estimated variance and base station variance.

If our model were perfect, then we would expect the resulting distribution of normalized offsets to be drawn from a 1-DOF chi-square distribution. The normalized histogram, shown in figure 6.7.3, shows that the SAM-generated satellite bias information is somewhat conservative, but not excessively so. This in turn gives us confidence that our landmark maps have appropriate uncertainty characteristics.

6.8 Conclusion

In this chapter, we began with the standard batch SAM formulation, and extended the theoretical framework to incorporate GPS readings. The extended framework encompasses the use of multiple runs, global coordinate systems, and nonwhite noise measurements from global sensors. Furthermore, we have demonstrated a successful implementation of the extended system on a real world robot, showing that the combined system is capable of dramatically increasing the scale at which it is feasible to generate probabilistically grounded maps.

Chapter 7

Conclusions

Outdoor mobile robotics is a rapidly maturing field. With ever-increasing amounts of data storage, network bandwidth, and sensing, the development of large geographic databases, both formally and informally, is inevitable. Increasingly relevant questions facing us include how do we process, organize, and use this data?

Probabilistic mapping methods from the SLAM community give us excellent frameworks for organizing data into forms which can answer interesting questions, such as "Where am I?" in a way which allows robots to reason about and perform useful tasks in an uncertain world. GPS, on the other hand, is an engineering marvel which has been extensively used in outdoor robotic navigation, but has not been extensively integrated with SLAM algorithms before now.

The reasons for a lack of GPS-SLAM integration were primarily a lack of good probabilistic error models for integration and a bias in the SLAM literature towards loop-closure as the sole source of uncertainty reductions.

In the case of outdoor robotics, this separation of SLAM and GPS is unwarranted. As has been shown in this dissertation, relatively simple models for per-satellite GPS errors can be used to augment SLAM with GPS readings, improving its accuracy and robustness, and increasing the usefulness of generated maps in long-term applications.

Future Directions

One of the most difficult aspects of this work was resisting the urge to explore the wide variety of problems which are related but not core to GPS-SLAM. We content ourselves with discussing some of the more interesting roads-not-taken as possible directions for future research.

In chapter 5, the underlying system scales well as the size of the map grows, but the growth of computational complexity still requires a presupposition that mapping and navigation are separate tasks; we can spend a lot of time and memory building the map as a fixed up-front cost, and then use the map statically to navigate. [Kaess, 2008] presents an alternative view which does not require full information matrix factorization at each timestep. One path of future investigation is how incremental and amortized variable reordering would mesh with the presented models of GPS noise.

Wide area differential GPS (WADGPS) is left largely untouched by this document. Finding and implementing a reasonably accurate WADGPS error model which is of sufficient mathematical elegance to incorporate into probabilistic mapping would be extraordinarily useful. As the name suggests, WADGPS has a much wider range of applicability than LADGPS; a good framework for WADGPS integration would increase the applicability of GPS-SLAM tremendously.

The problem of long-term map maintenance is an open question. In this work, specific steps are taken to identify good landmarks which we believe will be stable over a long period of time, but no environment is truly static. Even landmarks as solid as buildings and walls are torn down, built, or rebuilt in slightly different configurations given a long enough operation time. Some initial work has looked at this problem in the context of indoor systems, but the problem in the context of long-term outdoor operation remains significant.

Global navigation is evolving rapidly; there are opportunities to examine the effects of new capabilities as they come online. As mentioned in chapter 3, new constellations of global navigation satellites are being put into place by China and the European Space Agency. If current planned launches come to fruition, in the next decade a receiver capable of processing signals from multiple systems may face scenarios in which 20 or

more usable satellites are being tracked at once. Additionally, the NAVSTAR system is slated to begin broadcasting a second civilian coarse acquisition code which can and should be characterized for use in a merged system.

In short, as researchers in outdoor mobile robotics, we have an interesting future with many challenges ahead of us.

Bibliography

[Arinc, 2000] Arinc. Navstar gps space segment/navigation user interfaces (ICD-GPS-200), 2000.

[Bailey, 2002] Tim Bailey. *Mobile Robot Localisation and Mapping in Extensive Outdoor Environments*. PhD thesis, The University of Sydney, 2002.

[Bar-Shalom and Li, 1993] Yaakov Bar-Shalom and Xiao-Rong Li. *Estimation and Tracking: Principles, Techniques and Software*. Artech House, Inc., 1993.

[Bosse *et al.*, 2004] Michael Bosse, Paul Newman, John Leonard, and Seth Teller. SLAM in large-scale cyclic environments using the Atlas framework. 23(12):1113–1139, 2004.

[Bulata and Devy, 1996] Hanna Bulata and Michel Devy. Incremental construction of a landmark-based and topological model of indoor environments by a mobile robot. In *Proceedings of the 1996 IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1054–1060, 1996.

[Crum *et al.*, 1997] Jeffrey D. Crum, Steven T. Hutsell, and Ronals T. Smetek Jr. In *29th Annual Precise Time and Time Interval PTTI Meeting*, December 1997.

[Davis *et al.*, 2004] Timothy A. Davis, John R. Gilbert, Stefan I. Larimore, and Esmond G. Ng. A column approximate minimum degree ordering algorithm. *ACM Trans. Math. Softw.*, 30(3):353–376, 2004.

[Dellaert and Kaess, 2006] Frank Dellaert and Michael Kaess. Square root SAM. *Intl. Journal of Robotics Research*, 2006.

[Dissanayake *et al.*, 2001] M. W. M. Gamini Dissanayake, Paul Newman, Steven Clark, Hugh Durrant-Whyte, and Michael Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, 2001.

[Dow *et al.*, 2005] J. M. Dow, R. E. Neilan, and G. Gendt. The international GPS service (IGS): Celebrating the 10th anniversary and looking to the next decade. *Advanced Space Research*, 36(3):320–326, 2005.

[Eliazar and Parr, 2003] Austin Eliazar and Ronald Parr. DP-SLAM: Fast, robust simultaneous localization and mapping without predetermined landmarks. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)* [2003], pages 1135–1142.

[Eliazar and Parr, 2004] Austin I. Eliazar and Ronald Parr. DP-SLAM 2.0. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA)* [2004], pages 1314–1320.

[Eliazar and Parr, 2005] Austin I. Eliazar and Ronald Parr. Hierarchical linear/constant time SLAM using particle filters for dense maps. In *Advances in Neural Information Processing Systems 18*, 2005.

[Eustice *et al.*, 2005] R. Eustice, M. Walter, and J. Leonard. Sparse extended information filters: Insights into sparsification. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, pages 641–648, August 2005.

[Farrell and Barth, 1999] Jay A. Farrell and Matthew Barth. *The Global Positioning System and Inertial Navigation*. McGraw-Hill, 1999.

[Feng and Zheng, 2005] Yanming Feng and Yi Zheng. Efficient interpolations to GPS orbits for precise wide area applications. *GPS Solutions*, 9(4):273–282, 2005.

[Frese, 2004] Udo Frese. *An $O(log\ n)$ Algorithm for Simulateneous Localization and Mapping of Mobile Robots in Indoor Environments*. PhD thesis, University of Erlangen-Nürnberg, 2004.

[Frese, 2007] Udo Frese. Efficient 6-DOF SLAM with treemap as a generic backend. In *Proceedings of the Internation Conference on Robotics and Automation, Rome*, 2007.

[Früh and Zakhor, 2003] Christian Früh and Avideh Zakhor. Constructing 3D city models by merging ground-based and airborne views. *Computer Vision and Pattern Recognition*, 02(1063-6919):562, 2003.

[Gelb, 1974] A. Gelb, editor. *Applied Optimal Estimation*. MIT Press, 1974.

[Golub and Plemmons, 1979] Gene H. Golub and Robert J. Plemmons. Large scale geodetic least squares adjustment by dissection and orthogonal decomposition. Technical report, 1979.

[Grewal *et al.*, 2001] Mohinder S. Grewal, Lawrence R. Weill, and Angus P. Andrews. *Global Positioning Systems, Inertial Navigation, and Integration*. John Wiley and Sons, 2001.

[Grisetti *et al.*, 2005] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2443–2448, 2005.

[Guivant and Nebot, 2001] J. Guivant and E. Nebot. Optimization of the simultaneous localization and map building algorithm for real time implementation. *IEEE Transactions on Robotics and Automation*, May 2001.

[Gutmann and Konolige, 2000] Jens-Steffen Gutmann and Kurt Konolige. Incremental mapping of large cyclic environments. In *Proceedings of the 2000 IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, 2000.

[Herring and Shimada, 2001] Thomas Herring and Seiichi Shimada. Estimating spatial variations in atmospheric delays using GPS. In *Proceedings of the GPS MET Conference*, 2001.

[Ho and Newman, 2007] Kin Leong Ho and Paul Newman. Detecting loop closure with scene sequences. *International Journal of Computer Vision*, 74(3):261–286, September 2007.

[ICR, 2004] 2004.

[IJC, 2003] Acapulco, Mexico, August 2003.

[IJR,]

[Jefferson *et al.*, 2001] David C. Jefferson, Michael B. Heflin, and Ronald J. Muellerschoen. Examining the C1-P1 pseudorange bias. *GPS Solutions*, 4(4), April 2001.

[Julier and Uhlmann, 1997] Simon J. Julier and Jeffrey K. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Orlando, FL*, 1997.

[Kaess *et al.*, 2007] Michael Kaess, Ananth Ranganathan, and Frank Dellaert. Fast incremental square root information smoothing. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, 2007.

[Kaess, 2008] Michael Kaess. *Incremental Smoothing and Mapping*. PhD thesis, Georgia Institute of Technology, December 2008.

[Kaplan, 1996] Elliot D. Kaplan, editor. *Understanding GPS: Principles and Applications*. Artech House, 1996.

[Kelly, 1994] Alonzo Kelly. Essential kinematics for autonomous vehicles. Technical Report CMU-RI-TR-94-14, Robotics Institute, Pittsburgh, PA, May 1994.

[Kouba, 2009] Jan Kouba. A guide to using international GNSS service (IGS) products, May 2009.

[Kuipers *et al.*, 2004] Benjamin Kuipers, Joseph Modayil, Patrick Beeson, Matt MacMahon, and Francesco Savelli. Local metrical and global topological maps in the hybrid spatial semantic hierarchy. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA)* [2004].

[Lee *et al.*, 2007] Kwang Wee Lee, Sardha Wijesoma, and Javier Ibañez Guzmán. A constrained SLAM approach to robust and accurate localisation of autonomous ground vehicles. *Robotics and Autonomous Systems*, 55(7):527–540, July 2007.

[Liu and Thrun, 2003] Yufeng Liu and Sebastian Thrun. Results for outdoor-SLAM using sparse extended information filters. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1227–1233, Taipei, Taiwan, September 2003.

[Lu and Milios, 1997] Feng Lu and Evangelos Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4(4):333–349, 1997.

[Maybeck, 1979] Peter S. Maybeck. *Stochastic Models, Estimation and Control*, volume 1. Academic Press, 1979.

[Montemerlo *et al.*, 2002] Michael Montemerlo, Sebastian Thrun, Daphne Koller, and Ben Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings, The 18th National Conference on Artificial Intelligence and the 14th Innovative Applications of Artificial Intelligence Conference*, pages 593–598, Edmonton, Canada, July 2002.

[Montemerlo *et al.*, 2003] Michael Montemerlo, Sebastian Thrun, Daphne Koller, and B Wegbreit. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)* [2003].

[Montemerlo, 2003] Michael Montemerlo. *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, July 2003.

[Neira and Tardos, 2001] Jose Neira and Juan D. Tardos. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17(6), December 2001.

[Ni *et al.*, 2007] Kai Ni, Drew Steedly, and Frank Dellaert. Tectonic sam: Exact, out-of-core, submap-based slam. In *Proceedings of the 2007 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1678–1685, 2007.

[Parkinson *et al.*, 1996] Bradford W. Parkinson, James J. Spilker Jr., Penina Axelrad, and Per Enge, editors. *Global Positioning System: Theory and Applications*, volume 1. American Institute of Aeronautics and Astronautics, 1996.

[Paskin, 2002] Mark A. Paskin. Thin junction tree filters for simultaneous localization and mapping. Computer Science Division Technical Report CSD-02-1198, University of California, Berkeley, September 2002.

[Paz *et al.*, 2007] L.M. Paz, P. Jensfelt, J.D. Tardós, and J. Neira. EKF SLAM updates in O(n) with Divide and Conquer SLAM. In *Proceedings of the 2007 IEEE International Conference on Robotics and Automation (ICRA)*, 2007.

[Smith and Cheeseman, 1986] R. C. Smith and Peter Cheeseman. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56–68, 1986.

[Thrun and Montemerlo, 2006] Sebastian Thrun and Michael Montemerlo. The Graph-SLAM algorithm with applications to large-scale mapping of urban structures. In *The International Journal of Robotics Research* [], pages 403–430.

[Thrun *et al.*, 2004] Sebastian Thrun, Yufeng Liu, Daphne Koller, Andrew Y. Ng, Zoubin Ghahramani, and Hugh F. Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *The International Journal of Robotic Research*, 23(7-8):693–716, 2004.

[Walter *et al.*, 2007] Matthew R. Walter, Ryan M. Eustice, and John J. Leonard. Exactly sparse extended information filters for feature-based SLAM. In *The International Journal of Robotics Research* [], pages 335–359.

[Warren and Raquet, 2003] David L. M. Warren and John F. Raquet. Broadcast vs. precise GPS ephemerides: a historical perspective. *GPS Solutions*, 7(3), December 2003.

[Williams, 2001] Stefan Williams. *Efficient Solutions to Autonomous Mapping and Navigation Problems*. PhD thesis, Australian Centre for Field Robotics, Department of Mechanical and Mechatronic Engineering, The University of Sydney, Sydney, Australia, September 2001.