Robust Bayesian State Estimation and Mapping

by

Matthew C. Graham

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Aeronautics and Astronautics
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 2015

Author ..... Signature redacted
Department of Aeronautics and Astronautics
May 21, 2015

Certified by.... Signature redacted .....
Jonathan P. How
Richard C. Maclaurin Professor of Aeronautics and Astronautics
Thesis Supervisor

Certified by..... Signature redacted .
John Leonard
Samuel C. Collins Professor of Mechanical and Ocean Engineering

Certified by..... Signature redacted .
Donald Gustafson
Member of the Technical Staff, Draper Laboratory

Certified by..... Signature redacted .....
Marc McConley
Member of the Technical Staff, Draper Laboratory

Accepted by..... Signature redacted .....
Paulo C. Lozano
Associate Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

Robust Bayesian State Estimation and Mapping

by

Matthew C. Graham

Submitted to the Department of Aeronautics and Astronautics
on May 21, 2015, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Aeronautics and Astronautics

Abstract

Virtually all robotic and autonomous systems rely on navigation and mapping algorithms (e.g. the Kalman filter or simultaneous localization and mapping (SLAM)) to determine their location in the world. Unfortunately, these algorithms are not robust to outliers and even a single faulty measurement can cause a catastrophic failure of the navigation system. This thesis proposes several novel robust navigation and SLAM algorithms that produce accurate results when outliers and faulty measurements occur.

The new algorithms address the robustness problem by augmenting the standard models used by filtering and SLAM algorithms with additional latent variables that can be used to infer when outliers have occurred. Solving the augmented problems leads to algorithms that are naturally robust to outliers and are nearly as efficient as their non-robust counterparts. The first major contribution of this thesis is a novel robust filtering algorithm that can compensate for both measurement outliers and state prediction errors using a set of sparse latent variables that can be inferred using an efficient convex optimization.

Next the thesis proposes a batch robust SLAM algorithm that uses the Expectation-Maximization algorithm to infer both the navigation solution and the measurement information matrices. Inferring the information matrices allows the algorithm to reduce the impact of outliers on the SLAM solution while the Expectation-Maximization procedure produces computationally efficient calculations of the information matrix estimates.

While several SLAM algorithms have been proposed that are robust to loop closure errors, to date no SLAM algorithms have been developed that are robust to landmark errors. The final contribution of this thesis is the first SLAM algorithm that is robust to both loop closure and landmark errors (incremental SLAM with consistency checking (ISCC)). ISCC adds integer variables to the SLAM optimization that indicate whether each measurement should be included in the SLAM solution. ISCC then uses an incremental greedy strategy to efficiently determine which measurements should be used to compute the SLAM solution. Evaluation on standard benchmark datasets as well as visual SLAM experiments demonstrate that ISCC is robust to a

large number of loop closure and landmark outliers and that it can provide significantly more accurate solutions than state-of-the-art robust SLAM algorithms when landmark errors occur.

Thesis Supervisor: Jonathan P. How
Title: Richard C. Maclaurin Professor of Aeronautics and Astronautics

Acknowledgments

First and foremost, I'd like to thank my family, friends, and especially my wife Veronica for all of their love and support over the last four years. To my mom and dad, you instilled a love of learning and curiosity in me that has carried on to today. I would have never made it this far if not for all of your encouragement and support along the way. To my wife Veronica, you have been the best partner I could hope for. You give me energy when I feel exhausted, you brighten my day when I'm feeling down, you do so many little things that make my life better than I can't even begin to list them all. This thesis wouldn't exist if it weren't for your daily love and encouragement.

I'd also like to thank my academic colleagues who have made my experience at MIT so fantastic. To my advisor Professor Jon How: I've learned so many lessons about how to conduct and present research from you that have been invaluable. I'm also grateful for all of the time and energy you have invested in my success during my time here at MIT. Thanks also to my thesis committee, Professor John Leonard, Marc McConley and Don Gustafson for all of your encouragement and support. I'd also like to thank all of the members of the Aerospace Controls Lab and Marine Robotics Group. It's been a pleasure working with all of you and I am incredibly lucky to have the opportunity to collaborate and interact with so many talented and intelligent people. I can't wait to see what successes you all have in your professional careers. Finally, I'd like to thank all of my colleagues in the Draper Laboratory Fellows program, especially Ted Steiner, Marissa Galfond, Ben Arneberg and Timothy "Timmy" Galvin. Some of my most enjoyable times at MIT were spent having discussions in our office or scribbling out ideas on the whiteboard.

Finally, I would like to thank the Charles Stark Draper Laboratory who funded this research through their Internal Research and Development Program.

Contents

List of Figures

List of Algorithms

List of Tables

Chapter 1

Introduction

Navigation and mapping systems are integral components of virtually any autonomous or robotic system. Some examples include fusing global positioning system (GPS) and inertial measurement unit (IMU) data to produce a location estimate for an unmanned aerial vehicle, localization and mapping for an autonomous car [4], or creating detailed 3D maps of an environment using camera data [5–7]. In each of these cases, the navigation and mapping solution is typically then passed to a higher-level planning and control algorithm that can determine how to follow a desired trajectory, plan a path through a cluttered environment, or perform higher-level task planning and coordination with other agents.

One of the major challenges for navigation and mapping algorithms arises when the models of the system or the sensor data are incorrect. Given faulty data, standard algorithms for navigation and mapping (i.e., the Kalman filter [8] or graph-based simultaneous localization and mapping [9–11]) will produce inaccurate results and may even fail catastrophically. Moreover if the navigation solutions are invalid, the higher-level planning and control algorithms that use the navigation solutions can fail and cause damage to the system (e.g., colliding with obstacles, crashing the vehicle).

1.1 Motivation: Robust Visual Navigation and Mapping

Vision-aided navigation and mapping systems that merge camera images with other sensor data to form a navigation solution are becoming increasingly common. Computer vision algorithms for navigation are becoming fast and light-weight enough to run on small embedded processors [12]. In addition, sensors such as the Microsoft Kinect [13, 14] have made it possible to compute high accuracy mapping solutions for a fraction of the cost of a comparable laser scan based mapping system. This confluence of improving software and revolutionary hardware has made it possible to design vision-based navigation and mapping systems that would have been impossible only 10 years ago.

One of the major challenges of visual navigation is that a significant number of processing steps must be performed to distill the raw image data into a measurement that can be ingested by a navigation or mapping algorithm. If any of these pre-processing steps fail, the impact on the navigation solution can be severe. For instance, many filter-based vision-aided navigation algorithms rely on visual odometry [15] measurements to estimate relative changes in position and orientation in between periodic updates from absolute positioning sensors such as GPS. But changes in lighting conditions and motion blur can cause the feature tracking that underlies visual odometry to produce erroneous measurements and in turn cause significant estimation errors. Moreover, this thesis demonstrates that standard robust filtering algorithms will often misinterpret the large estimation errors as measurement outliers and as a result discard absolute positioning measurements that could have improved the navigation solution.

Visual simultaneous localization and mapping (SLAM) systems typically depend on visual place recognition [16, 17] to determine when the system has returned to a previously visited location (in SLAM this is referred to as a *loop closure* [1, 18]). But in many man-made areas such as cities and buildings, repeated structures and visual features can cause place recognition systems to return incorrect loop closure

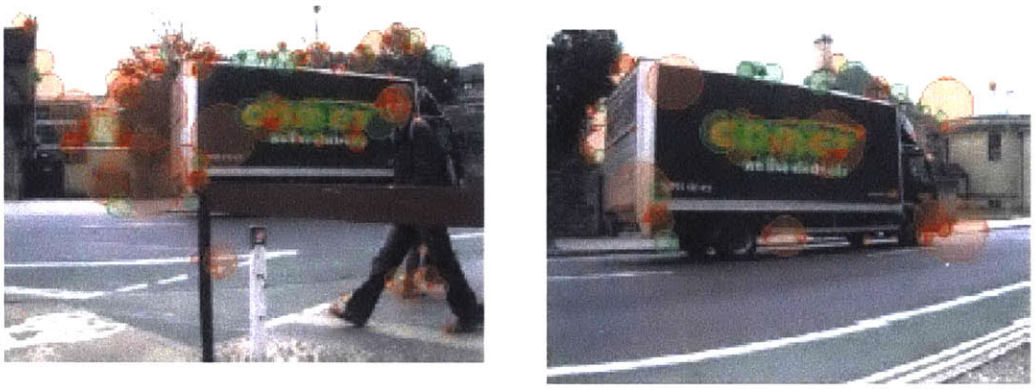


Figure 1-1: Example false loop closure detection caused by visual aliasing [1]

detections (see Figure 1-1). This type of error is known as *visual aliasing* and is a common problem in large-scale visual SLAM datasets [1, 19].

Once an incorrect loop closure has been generated by the place recognition system it can have a disastrous impact on the SLAM solution because standard SLAM algorithms are not robust to outliers [20–22]. Moreover, there is often no way to automatically determine that an incorrect loop closure detection has occurred based solely on the camera data.

A standard approach [1, 17] for preventing false loop closure detections is to set the place recognition detection thresholds to a large value to decrease the chance of false positives. But this approach has two major downsides. First, setting the thresholds to a large value does not guarantee that false loop closure detections will not occur. In fact, the visual aliasing example shown in Figure 1-1 was declared a loop closure with probability 0.969 [1]. Second, by setting the threshold so high, a large number of correct loop closures are potentially being ignored when they could be used to improve the accuracy of the SLAM solution. A receiver operating characteristic (ROC) curve [23], which shows the probability of detection and probability of false alarm for a detection algorithm as a function of the detector threshold (see Figure 1-2), is one way to visualize this loss of detections. Given a standard SLAM solver, current place recognition systems can only operate in the part of the ROC curve that lies close to the y-axis (i.e., the section of the curve that produces very few false

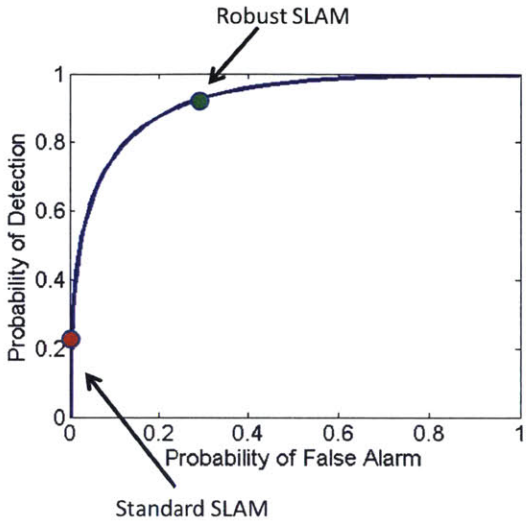


Figure 1-2: Notional ROC Curve for a loop closure detector. The red • shows the operating point for standard SLAM algorithms and the green • shows an operating point for a robust SLAM algorithm. The robust SLAM operating point is more desirable because it can significantly improve the accuracy of the solution using the additional loop closure measurements while rejecting the false loop closures.

alarms). But usually those regions also have a low probability of detection meaning that numerous correct loop closures are missed. Developing a SLAM algorithm that is robust to false loop closures would allow a place recognition system to operate in a regime of the ROC curve (for example the green dot on the curve in Figure 1-2) where significantly more true detections would occur and as a result lead to a more accurate mapping solution.

1.2 Solution Approach

State estimation and mapping algorithms typically assume that the system dynamics and measurements are accurately modeled and that any sources of noise or uncertainty are characterized correctly. In addition, many algorithms assume that the noise in the system is Gaussian. However, in practice the system models are often only known approximately and the noise in the system can be non-Gaussian. When the modeling assumptions are violated, standard state estimation and mapping algorithms have no means to correct for the modeling errors and often the result is a significant

degradation in accuracy and ultimately failure of the navigation system.

One reason the algorithms fail is because the models are inflexible and can not be changed online. The standard algorithms also lack the ability to determine whether the current estimates are consistent with each other and with the current set of measurements.

To address these issues this thesis proposes two approaches for robust state estimation and mapping. The first approach involves adding extra variables to the estimation problems that are used to characterize uncertainty in the dynamics and measurement models of the system (model augmentation). By estimating these auxiliary and state variables simultaneously, the new algorithms can compensate for outliers and modeling errors and provide robust estimates across a wide range of operating conditions.

The second approach is focused specifically on developing robust SLAM solutions. One major challenge for SLAM solvers is determining what measurements are incorrect when the solution is not consistent. This thesis develops an online SLAM algorithm that can efficiently search the measurements and determine which are most likely to be incorrect and remove them from the solution.

1.3 Literature Review

This section provides an overview of related work in the robust state estimation and robust SLAM literature.

1.3.1 Robust State Estimation

Initial efforts in robust state estimation focused on making modifications to the Kalman filter to make it more robust to heavy-tailed noise [24–28]. Often these approaches use influence functions (also known as robust kernels) [29], a concept from robust statistics [30], to modify the Kalman gain for measurements with large residuals (i.e., those that were most likely to be corrupted with outliers from the heavy-tailed noise). The influence function approach works by reducing the gain ap-

plied to measurements with large residuals so that the impact of outliers on the state estimates is reduced. These methods generally assume that the heavy-tailed noise only impacts the measurements or the state predictions, but not both. Recently, Gandhi and Mili have developed a robust statistics based filtering algorithm that relaxes these assumptions and can provide robust state estimates when heavy-tailed noise is present in both the state dynamics and measurements [31].

The main issue with these methods is the selection of the influence function used to modify the Kalman gain. Although there are some basic properties that the influence functions must satisfy in order to ensure that the state estimates are robust, in practice there are an infinite number of candidate influence functions and it is unclear how to choose between them. Additionally for any practical application, the parameters of the influence function must be tuned to the data being considered and this often leads to ad-hoc filter implementations.

Some robust filtering algorithms have been developed specifically to handle errors in the system dynamics models. The most prominent approach is James-Stein filtering [32]. These filters use a variant of the James-Stein estimator [33] to calculate state estimates that are not sensitive to errors in the state dynamics models. However, to guarantee robustness, the James-Stein filter requires that the measurements are not corrupted by outliers and thus it trades off robustness to measurement outliers for robustness to system modeling errors.

Another method for robust state estimation that was an outgrowth of robust control research is the H_∞ filter [34–37]. These algorithms assume that the state predictions and measurements are subject to unknown but bounded errors and then seek to minimize the worst-case estimation error given the unknown errors. Although the H_∞ filter does guarantee bounded state estimation error, often the state estimates are overly conservative. In fact, under nominal noise conditions (i.e., white, zero-mean and Gaussian), the H_∞ filter estimation error can in fact be significantly worse (in a least-squares sense) than the Kalman filter estimation error [37]. Additionally, the H_∞ filter estimates are sensitive to noise that is generated from heavy-tailed distributions [31].

Robust particle filters have also been developed to address the impact of measurement outliers [38–40]. While particle filters are often well suited to state estimation for nonlinear and non-Gaussian systems, they are susceptible [38–40] to modeling errors and outliers in the measurements. When an outlier corrupts the measurements, the measurement likelihood for any given particle will often be zero or very near zero and, as a result, the approximate posterior distribution will be represented by only a few particles. This situation is called "sample impoverishment" and can lead to divergence in the particle filter estimates [41]. Robust particle filters usually address this issues by trying to identify and reject measurements corrupted by outliers before they can impact the state estimates [38–40]. The outliers are detected using hypothesis tests based on statistics calculated from the current set of particles and measurements. In a sense these approaches are extensions of the classical failure detection tests for linear filtering (such as χ^2 residual tests [42]) to particle filters. So far these methods have not tried to address the issue of unmodeled uncertainty in the state propagation models for particle filters.

Recently several robust filtering approaches have been developed that try to explicitly estimate unknown errors by solving a convex optimization problem [43–46]. Mattingley and Boyd developed a filter that assumes that a subset of the measurements being processed are corrupted by errors that do not follow the nominal noise distribution [43]. By representing the errors as *sparse*, the errors can be estimated using a constrained l_1-norm minimization which is convex and can be solved efficiently [47, 48]. Farahmand and Giannakis independently developed a filtering algorithm that uses a similar approach to solve robust filtering problems when the sparse errors are time correlated [49]. Recently, Farahmand et al. have developed a set of convex optimization based algorithms for solving the smoothing problem for dynamics systems with uncertain errors in the state dynamics models and the measurements [50].

Finally, a number of robust state estimation algorithms have been developed that use variational Bayesian techniques. The basic approach for the variational Bayesian filtering algorithms is to introduce extra uncertainty into the system model by mod-

eling the noise parameters (typically the covariances) as random variables. The algorithms then form a variational approximation of the joint log-likelihood of the noise parameters and states and use the Expectation-Maximization algorithm [51] to iteratively solve for the state and noise parameter values.

All of the current variational Bayesian approaches have focused on calculating robust state estimates in the presence of uncertainty in the measurement noise while ignoring uncertainty in the process noise. Initial work by Ting et al. [52–54], as well as Särkkä and Nummenmaa [55], assumed that the measurement noise terms were independent of each other. More recently, Agamennoni et al. [56, 57] have developed robust variational Bayesian filters and smoothers that relax the assumption of independent noise terms by using a structured mean-field variational approximation.

1.3.2 Gaps in Robust Filtering Literature

In general, there is one major gap in the current robust filtering literature. While algorithms exist that are robust to state dynamics modeling errors (e.g., James-Stein filters) or to measurement outliers, none can guarantee accurate solutions when both types of errors occur. The first major contribution of this thesis is a novel robust filtering algorithm, the l_1-norm filter, that is robust to both state prediction errors and measurement outliers. It extends prior work by Mohiuddin et al. [45] that augmented the standard state dynamics and measurement models to include sparse errors in both the state predictions and measurements. These sparse error terms can be jointly estimated with the states to calculate robust estimates when outliers occur in the state predictions and measurements. While Mohiuddin et al. [45] proposed the extended model, the l_1-norm filter was never evaluated with data that contained both state prediction and measurement outliers. This thesis provides both theoretical and empirical analysis of the l_1-norm filter that demonstrates that it can provide significantly more accurate solutions than any other type of robust filtering algorithm in the presence of both types of outliers.

1.3.3 Robust SLAM

Robust SLAM algorithms can be divided into three main categories. The first type of algorithms apply robust kernels to the optimization problem to reduce the impact of outliers on the SLAM solution. The second category approach the problem by adding additional latent variables to the SLAM problem formulation to account for outliers in the measurements (augmented model approaches). The third set of algorithms focus on choosing sets of measurements that lead to a consistent SLAM solution.

Robust Kernel Approaches

In recent years there have been a number of approaches that have been proposed to improve the robustness of SLAM solvers to outliers and incorrect loop closures. The simplest technique involves substituting a Huber cost function [30] for the least squares cost function in the SLAM problem. Huber cost functions are quadratic functions of the error near zero and then transition to a linear function of the error past a given threshold. Including the linear region reduces the impact of incorrect measurements (which would likely have a large error associated with them) on the least-squares solution. The major drawback of using Huber cost functions is that they only reduce the impact of erroneous measurements instead of excising them from the solution completely. In practice, this means that SLAM algorithms using Huber cost functions are still prone to converging to a poor solution when measurement outliers are present in the data [20].

Augmented Model Approaches

Sünderhauf et al. [20, 58, 59] introduced additional variables into the SLAM problem formulation, called *switch variables*, that can take any value in the interval [0 1] and are used as weights for each of the loop closure measurements. The switch variables provide robustness by determining whether to accept or reject a potential loop closure measurement. When a switch variable is equal to zero its corresponding loop closure measurement will not have any impact on the SLAM solution. Agarwal

et al. presented a generalization of switch variables called *dynamic covariance scaling* (DCS) that provides a closed form update for the switch variables and results in faster convergence [60].

Another approach developed by Olson and Agarwal [21, 61], *max-mixtures*, modifies the conditional probability distribution of the measurements so that the noise is represented by a Gaussian mixture instead of a single Gaussian. The algorithm then selects the most likely mixture component before each pose update. Finally, several robust algorithms have been proposed that add additional variables to the robust SLAM problem and solve the augmented problem by applying the Expectation-Maximization (EM) algorithm [62, 63].

Consistency-Based Approaches

An alternative set of robust SLAM algorithms attempt to remove incorrect measurements from the SLAM solution by performing consistency checks on subsets of the measurements.

The RRR algorithm applies a series of χ^2 tests to determine whether loop closure measurements are correct and if they are consistent with each other [22, 64, 65]. Initially loop closures are clustered with other loop closures "near" them spatially. For each cluster, the subset of poses are optimized and a χ^2 test is applied to determine if any loop closures are incorrect. After removing any loop closures that fail the initial test, a χ^2 test is applied to the entire cluster to determine if the solution is consistent. Finally, groups of clusters are jointly optimized and evaluated again using χ^2 tests. Those clusters that pass the final χ^2 test are then used to generate the final SLAM solution. One of the main drawbacks of this approach is that once loop closures fail a χ^2 test, they can not be added back into the solution later, which means that in many cases good loop closure measurements that could improve the solution quality are left out.

Recently, Carlone et al. [66] posed the robust SLAM problem as an optimization with the goal of selecting the largest subset of measurements that produces a consistent SLAM solution. The authors focus on 2D SLAM scenarios where an approximate

solution to the optimization can be calculated using linear programming. While their algorithm is fast and effective, it relies heavily on assumptions that only apply to 2D SLAM problems and currently there is no clear way to extend the algorithm to 3D SLAM datasets.

1.3.4 Gaps in Robust SLAM Literature

One of the major drawbacks of augmented model approaches for robust SLAM is their reliance on tuning parameters that are sensitive to the measurement noise. Often it is difficult to accurately characterize the noise and as a result these tuning parameters can be difficult to select *a priori*. This thesis addresses these issues by developing an augmented model approach to robust SLAM that explicitly estimates the noise parameters of the measurements concurrently with the map estimates. As a result, the algorithm is both robust to outliers and requires no tuning parameters.

To date, most robust SLAM algorithms have focused exclusively on problems with loop closure errors. But many SLAM systems rely on landmark measurements to correct for odometry drift. In addition, previous results [60] have shown that augmented model approaches while robust to loop closure errors are only robust to a small number of incorrect landmark measurements. This indicates that there is something fundamentally different about landmark measurements that needs to be addressed in order to calculate a robust solution. We will discuss some of those differences and how they can be addressed later in this thesis.

A problem with augmented model approaches is that they only focus on local consistency (by determining whether each measurement is correct independently of the others) rather than global consistency. This strategy works well for loop closures because typically only one loop closure measurement exists between any two robot poses. Thus, if the measurement is not locally consistent with its associated pose estimates, the loop closure is likely incorrect and can be ignored.

In contrast, a single landmark will typically have a number of measurements associated with it. By myopically evaluating each landmark measurement, augmented model algorithms miss the opportunity to exploit the additional information avail-

able from the other measurements. For instance, an incorrect landmark measurement could appear consistent given only the landmark and pose estimates associated with it, but not agree with the other landmark measurements.

To address these gaps, this thesis proposes a new robust SLAM cost function that explicitly requires that the SLAM solution is both locally and globally consistent and includes as many measurements as possible. An obvious approach to solving this formulation of the robust SLAM problem is to search over every possible combination of measurements to find the largest set that produces a consistent solution. But performing that search would require searching over a combinatorial set of potential measurements which would cause an exponential increase in computational complexity as the number of measurements increased. Therefore this thesis also proposes a novel robust SLAM algorithm that approximately solves the new robust SLAM cost function using an efficient greedy measurement selection strategy. The new algorithm is a consistency-based approach that has several key innovations over previous algorithms. First, it can be applied to datasets with both landmark and loop closure errors, which was not possible with previous robust SLAM algorithms. In addition, the new algorithm uses a more general set of consistency constraints than the approach proposed by Carlone et al. [66] and as a result it can be applied to both 2D and 3D datasets.

1.4 Thesis Contributions

Overall, this thesis focuses on improving existing techniques for navigation and mapping either by augmenting the underlying system models to capture structure in the problem that can account for outliers and errors in the data or by proposing new cost functions that promote solutions that are more robust to outliers.

The first contribution of this thesis is a novel filtering algorithm that is robust to simultaneous outliers in both the state predictions and measurements. The key component of the algorithm is an augmented state-space model that includes additional variables to account for large deviations from the assumed system models. Given

estimates of the additional variables, which can be found using an efficient convex optimization, their effect can easily be removed to provide robust state estimates. The major contributions are:

- A new model for state estimation that can account for large deviations from the state dynamics and measurement models using a set of sparse variables. The sparsity assumption is a key component of the model because it provides a means of distinguishing between measurement errors and state prediction errors, which is not possible with existing robust filtering methods.

- A robust filtering algorithm, the l_1-norm filter, for the proposed state space model. The algorithm has two major components. First, an efficient convex optimization that estimates the sparse variables to identify when outliers have occurred. Second, an update procedure similar to the Kalman filter update equations that mitigates the impact of the outliers on the state estimates. As a result of these two components, the l_1-norm filter produces solutions for datasets with outliers that are significantly more accurate than non-robust filtering solutions while maintaining computational efficiency.

- Theoretical analysis of the l_1-norm filter that provides additional insight into the filter performance. This analysis shows that unlike other convex optimization based robust filtering algorithms the l_1-norm filter estimates are unbiased. In addition, the analysis shows that the l_1-norm filter performance is not significantly affected when outliers are not correctly detected by the convex optimization step.

- Experimental validation of the l_1-norm filter using both Monte Carlo simulations and vision-aided navigation data collected in urban areas. In the vision-aided navigation data, measurement outliers occur frequently due to GPS multipath while state prediction outliers occur due to the build-up of visual odometric drift during GPS outages. The Monte Carlo trials demonstrated that the l_1-norm filter provided at least an order of magnitude improvement in error performance

relative to other robust filtering algorithms when state prediction and measurement outliers occur simultaneously. For the experimental results, the l_1-norm filter provided a 20% improvement over non-robust filtering algorithms and also outperformed all other robust filtering algorithms.

To date, most robust SLAM algorithms have addressed outliers by adding additional variables to the SLAM optimization. This thesis takes a similar approach with one major difference, instead of using an auxiliary variable that has no relationship to the original underlying problem, a robust SLAM algorithm is proposed that estimates both the measurement information matrices and the map estimates simultaneously. By estimating the information matrices, not only can the new algorithm provide robust estimates, it can also adapt the measurement information matrices to better match the true noise if the assumed noise models are incorrect. The major contributions are:

- A novel robust SLAM algorithm, information matrix SLAM (IM-SLAM), that simultaneously estimates the measurement information matrices and poses using the Expectation-Maximization (EM) algorithm. By estimating the information matrices, the algorithm can dynamically modify the weights for each measurement in the SLAM solution and in the process reduces the impact of outliers on the mapping solution. Moreover, the information matrix updates can be expressed in closed form meaning that IM-SLAM can provide significantly more robust estimates for virtually the same computational cost as a non-robust SLAM algorithm.

- Analysis that proves that IM-SLAM will converge. These proofs leverage the convergence properties of the EM algorithm to show that IM-SLAM will converge to at least a local minimum. This work provides the first convergence proof for an augmented model based robust SLAM algorithm.

- Analysis that provides an alternative interpretation of IM-SLAM as a robust kernel. Robust kernels are often used in maximum likelihood estimation problems because they have desirable robustness characteristics. This interpretation

provides additional insight into why IM-SLAM is able to provide robust solutions.

- Simulated and experimental results are presented that demonstrate that IM-SLAM provides significantly more robust solutions than standard SLAM algorithms for a negligible increase in computational cost.

Although IM-SLAM demonstrates robust performance for a number of robust SLAM datasets, it can only be applied in batch to datasets that contain loop closure measurements. A general purpose robust SLAM solver should be able to be applied to datasets with either loop closure or landmark measurements. Moreover, for many applications it is desirable to provide an online SLAM solution that can be used while collecting data rather than post-processing. Given these limitations for robust SLAM, the final set of thesis contributions are:

- A new formulation of the robust SLAM problem that seeks to optimize the number of measurements included in the solution while ensuring that the solution is consistent (i.e., all of the measurements agree). The consistency constraint ensures that the solution does not include gross outliers, while maximizing the number of measurements produces the solution with the least uncertainty.

- A novel online robust SLAM algorithm, incremental SLAM with consistency checking (ISCC), that is robust to both landmark and loop closure errors. ISCC approximately solves the new robust SLAM problem using an incremental greedy optimization approach that removes as few measurements as possible to produce a consistent solution. The incremental greedy approach is also much more efficient than a brute force approach to the robust SLAM problem which would require searching over a combinatorial set of potential measurements.

- Simulated and experimental results demonstrating that ISCC can significantly outperform state-of-the-art robust SLAM algorithms on datasets with landmark errors and can match the performance of the state-of-the-art algorithms when loop closure errors occur.

1.5 Thesis Outline

The thesis proceeds as follows. Chapter 2 provides background material on state estimation and robotic mapping techniques as well as a brief discussion of robust estimation techniques. Chapter 3 presents a novel Bayesian filtering technique (the l_1-norm filter) that is robust to simultaneous errors in both the state dynamics model and the measurements. In Chapter 4, an augmented formulation of the SLAM problem is proposed and solved leading to a new robust SLAM algorithm, IM-SLAM. Chapter 5 develops a new formulation of the robust SLAM problem and then proposes an online algorithm, ISCC, to solve it. Finally, Chapter 6 summarizes the main results in the thesis and discusses potential areas of future work.

Chapter 2

Background Material

This chapter will provide a review of relevant background material in state estimation, simultaneous localization and mapping (SLAM) and common techniques for robust estimation.

2.1 Bayesian State Estimation

Stochastic state space models are often used in estimation problems for dynamical systems because they provide compact representations of the system that are convenient for analysis and algorithm development [67–69]. The models consist of a dynamics model for the latent state, $\mathbf{x}_k$, that is driven by random process noise, $\mathbf{w}_k$, and a model of the measurements of the state vector, $\mathbf{y}_k$, that are corrupted with noise, $\mathbf{v}_k$. A nonlinear discrete-time stochastic state space model can be expressed as

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{w}_k) \tag{2.1}$$

$$\mathbf{y}_k = h(\mathbf{x}_k, \mathbf{v}_k) \tag{2.2}$$

It is typically assumed that the process and measurement noise are white and mutually uncorrelated, the state sequence is a Markov process, and that each measurement is conditionally independent of the other measurements given the current state.

Given these assumptions, the state estimation problem can be cast as a Bayesian

inference problem. The goal is to calculate the posterior distribution of the state at time k given a set of measurements from time 1 to T. In other words

$$\begin{aligned} \text{Find:} \quad & p(\mathbf{x}_k|\mathbf{y}_1, \cdots, \mathbf{y}_T) \\ \text{Given:} \quad & \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{w}_k) \\ & \mathbf{y}_k \quad = h(\mathbf{x}_k, \mathbf{v}_k) \end{aligned} \tag{2.3}$$

Note that if $T = k$, this problem statement corresponds to a Bayesian filtering problem, while if $T > k$ then the problem statement corresponds to a Bayesian smoothing problem.

2.2 Bayesian Filtering Algorithms

For Bayesian filtering problems, the calculation of the posterior state distribution can be broken down into two steps that are repeated recursively [41]

1. **State Propagation**

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) d\mathbf{x}_{k-1}$$

2. **Measurement Update (Bayes Rule)**

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{y}_{1:k-1})}{\int p(\mathbf{x}_k, \mathbf{y}_k) d\mathbf{x}_k}$$

Given the posterior distribution, the state estimate at time k, $\hat{\mathbf{x}}_k$, can be calculated in several ways. A *maximum a posteriori* (MAP) estimate of the state can be obtained using

$$\hat{\mathbf{x}}_k^{MAP} = \underset{\mathbf{x}_k}{\text{argmax}}\, p(\mathbf{x}_k|\mathbf{y}_1, \cdots, \mathbf{y}_T). \tag{2.4}$$

Alternatively, the state estimate can be calculating by solving for the mean value of $\mathbf{x}_k$ given the posterior distribution. Both of these approaches work best when the posterior distribution is unimodal and greater care should be taken in generating

state estimates when the posterior distribution is multimodal [41]. If the measurement noise is Gaussian and the system is linear, then these two solutions coincide.

Several different classes of filtering algorithms have been developed and are typically specialized for the properties of the underlying dynamical system (e.g., linearity, Gaussian noise). The following subsections will cover the major Bayesian filtering algorithms that are relevant for the thesis.

2.2.1 Linear Gaussian Systems: Kalman Filter

The Kalman filter is a Bayesian filter for linear dynamical systems with additive, white, Gaussian noise [8]. In other words, systems that take the form

$$\mathbf{x}_{k+1} = F_k \mathbf{x}_k + \mathbf{w}_k \tag{2.5}$$

$$\mathbf{y}_{k+1} = H_{k+1} \mathbf{x}_{k+1} + \mathbf{v}_{k+1} \tag{2.6}$$

$$\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, P_0) \tag{2.7}$$

where F_k is the state transition matrix, H_{k+1} is the measurement matrix, $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, Q_k)$ and $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, R_k)$[1]. Given the linearity and Gaussian assumptions, both $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ and $p(\mathbf{y}_k|\mathbf{x}_k)$ are Gaussian distributions. Moreover this implies that $p(\mathbf{x}_k|\mathbf{y}_1, \cdots, \mathbf{y}_k)$ is a Gaussian distribution as well.

Since the posterior state distribution is Gaussian, it can be completely characterized in terms of its mean, $\hat{\mathbf{x}}_{k|k}$, and covariance, $P_{k|k}$. In addition, the mean and covariance can be calculated in closed form for both the state propagation and measurement update steps. These closed form expressions are the Kalman filter state propagation and update equations. Given $\hat{\mathbf{x}}_{k|k}$, $P_{k|k}$ the Kalman filter propagation equations are [70]

$$\hat{\mathbf{x}}_{k+1|k} = F_k \hat{\mathbf{x}}_{k|k} \tag{2.8}$$

$$P_{k+1|k} = F_k P_{k|k} F_k^T + Q_k \tag{2.9}$$

[1]Throughout the thesis the notation, $\mathcal{N}(\mu, \Sigma)$ indicates a multivariate Gaussian random variable with mean μ and covariance Σ.

The Kalman filter measurement update equations are

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + K_{k+1}\left(\mathbf{y}_{k+1} - H_{k+1}\hat{\mathbf{x}}_{k+1|k}\right) \tag{2.10}$$

$$P_{k+1|k+1} = (I - K_{k+1}H_{k+1})P_{k+1|k} \tag{2.11}$$

where $K_{k+1} = P_{k+1|k}H_{k+1}^T\left(H_{k+1}P_{k+1|k}H_{k+1}^T + R_{k+1}\right)^{-1}$ is the Kalman gain.

The Kalman filter update equations can also be interpreted as a weighted least-squares update. To see this consider the following linear least-squares problem

$$\min_{\mathbf{x}} \frac{1}{2}(\mathbf{y}_{k+1} - H_{k+1}\mathbf{x})^T R_{k+1}^{-1}(\mathbf{y}_{k+1} - H_{k+1}\mathbf{x}) + \frac{1}{2}(\mathbf{x} - \hat{\mathbf{x}}_{k+1|k})^T P_{k+1|k}^{-1}(\mathbf{x} - \hat{\mathbf{x}}_{k+1|k}) \tag{2.12}$$

which can be interpreted as a weighted least-squares problem with a prior estimate $\hat{\mathbf{x}}_{k+1|k}$ that must be factored into the solution along with the measurements $\mathbf{y}_{k+1}$.

The optimal solution can be found by solving the corresponding normal equations

$$\left(H_{k+1}^T R^{-1} H_{k+1} + P_{k+1|k}^{-1}\right)\mathbf{x} = H^T R^{-1}\mathbf{y}_{k+1} + P_{k+1|k}^{-1}\hat{\mathbf{x}}_{k+1|k} \tag{2.13}$$

the solution of which is

$$\begin{aligned}\hat{\mathbf{x}}_{k+1|k+1} = &\left(H_{k+1}^T R^{-1} H_{k+1} + P_{k+1|k}^{-1}\right)^{-1} H^T R^{-1}\mathbf{y}_{k+1} \\ &+ \left(H_{k+1}^T R^{-1} H_{k+1} + P_{k+1|k}^{-1}\right)^{-1} P_{k+1|k}^{-1}\hat{\mathbf{x}}_{k+1|k}\end{aligned}$$

After applying a Schur identity to the first term, the solution can be rewritten as

$$\begin{aligned}\hat{\mathbf{x}}_{k+1|k+1} = &P_{k+1|k}H_{k+1}^T\left(H_{k+1}P_{k+1|k}H_{k+1}^T + R_{k+1}\right)^{-1}\mathbf{y}_{k+1} \\ &+ \left(H_{k+1}P_{k+1|k}H_{k+1}^T + R_{k+1}\right)^{-1} P_{k+1|k}^{-1}\hat{\mathbf{x}}_{k+1|k}\end{aligned}$$

Finally applying the matrix inversion lemma to the second term and combining terms leads to

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + P_{k+1|k}H_{k+1}^T\left(H_{k+1}P_{k+1|k}H_{k+1}^T + R_{k+1}\right)^{-1}\left(\mathbf{y}_{k+1} - H_{k+1}\hat{\mathbf{x}}_{k+1|k}\right)$$

$$= \hat{\mathbf{x}}_{k+1|k} + K_{k+1}(\mathbf{y}_{k+1} - H_{k+1}\hat{\mathbf{x}}_{k+1|k})$$

which is the Kalman filter update equation. In Section 2.4.1 it will be shown that least-squares estimators are not robust to outliers which in turn means that the Kalman filter is also not robust to outliers.

2.2.2 Nonlinear Gaussian Systems: Extended Kalman Filter

For systems with nonlinear state transition and measurement functions such as

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k) + \mathbf{w}_k \tag{2.14}$$

$$\mathbf{y}_{k+1} = h(\mathbf{x}_{k+1}) + \mathbf{v}_{k+1} \tag{2.15}$$

$$\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, P_0) \tag{2.16}$$

where $f(\mathbf{x})$ is the state transition function, $h(\mathbf{x})$ is the measurement function, $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, Q_k)$ and $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, R_k)$; one of the standard estimation algorithms is the extended Kalman filter (EKF) [71]. The EKF approaches the nonlinear filtering problem by approximating the posterior distribution using a linearization of the nonlinear state transition and measurement functions.

In general, the transformation of a Gaussian random variable ($\mathbf{x} \sim \mathcal{N}(\mathbf{0}, P)$) through a nonlinear function ($g(\mathbf{x})$) can be expressed using a Taylor series expansion:

$$\mathbf{z} = g(\mathbf{x}) = g(\bar{\mathbf{x}}) + \nabla g(\bar{\mathbf{x}})\delta\mathbf{x} + \frac{1}{2}\nabla g^2(\bar{\mathbf{x}})\delta\mathbf{x}^2 + \frac{1}{3!}\nabla g^3(\bar{\mathbf{x}})\delta\mathbf{x}^3 + \cdots \tag{2.17}$$

where $\bar{\mathbf{x}} = E[\mathbf{x}]$ and $\delta\mathbf{x} \sim \mathcal{N}(\mathbf{0}, P)$. Given, Eq. 2.17 the mean ($\bar{\mathbf{z}}$) and covariance (P_z) of $\mathbf{z}$ can be expressed as:

$$\bar{\mathbf{z}} = g(\bar{\mathbf{x}}) + \frac{1}{2}\nabla g^2(\bar{\mathbf{x}})P + \frac{1}{4!}\nabla g^4(\bar{\mathbf{x}})E\left[\delta\mathbf{x}^4\right] + \cdots \tag{2.18}$$

$$P_z = E\left[(\mathbf{z} - \bar{\mathbf{z}})(\mathbf{z} - \bar{\mathbf{z}})^T\right]$$

$$= \nabla g P (\nabla g(\bar{\mathbf{x}}))^T + \frac{1}{2 \times 4!} \nabla g^2(\bar{\mathbf{x}}) \left(E\left[\delta \mathbf{x}^4\right] - E\left[\delta \mathbf{x}^2 P\right] - E\left[P \delta \mathbf{x}^2\right] + P^2 \right) + \cdots \tag{2.19}$$

After truncating the series to first-order the mean and covariance of the transformed variables can be approximated as

$$\bar{\mathbf{z}} \approx g(\bar{\mathbf{x}}) \tag{2.20}$$

$$P_z \approx \nabla g P (\nabla g(\bar{\mathbf{x}}))^T \tag{2.21}$$

The EKF uses the linear approximations in Eqs. 2.20 and 2.21 to approximate both $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ and $p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1})$ during the state propagation and measurement update steps. After applying the approximations, the EKF state propagation equations are given by [70]

$$\hat{\mathbf{x}}_{k+1|k} = f(\hat{\mathbf{x}}_{k|k}) \tag{2.22}$$

$$P_{k+1|k} = F_k P_k F_k^T + Q_k \tag{2.23}$$

where

$$F_k = \left. \frac{df(\mathbf{x})}{d\mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k}} \tag{2.24}$$

Similarly the EKF measurement update equations are [70]

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + K_{k+1} \left(\mathbf{y}_{k+1} - h(\hat{\mathbf{x}}_{k+1|k}) \right) \tag{2.25}$$

$$P_{k+1|k+1} = (I - K_{k+1} H_{k+1}) P_{k+1|k} \tag{2.26}$$

where $K_{k+1} = P_{k+1|k} H_{k+1}^T \left(H_{k+1} P_{k+1|k} H_{k+1}^T + R_{k+1} \right)^{-1}$ is the Kalman gain and

$$H_{k+1} = \left. \frac{dh(\mathbf{x})}{d\mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k+1|k}} \tag{2.27}$$

2.3 Simultaneous Localization and Mapping

Virtually any autonomous robotic system must have an accurate map of its surroundings in order to perform its tasks. In the absence of a pre-defined map, a robotic system must be able to form a map of its environment as well as determine where it is within the map. The task of constructing the map and localizing the robot is called simultaneous localization and mapping (SLAM) and is a fundamental problem in robotics [72, 73].

This section provides an overview of the mathematics underlying SLAM as well as a discussion of optimization approaches for solving the SLAM problem.

2.3.1 Problem Formulation and Solution

Typically SLAM systems rely on two types of external measurements to calculate the solution: odometry measurements and loop closure measurements. Odometry measurements provide a measurement of the relative change in position and orientation (*pose*) of the robot between measurement intervals. Odometry measurements can be generated using wheel encoders, IMUs or can be derived from images of the environment using visual odometry [15]. Odometry alone is not sufficient to provide a solution to the SLAM problem because random errors accumulated from each measurement will cause the solution to drift over time. Loop closure measurements, which are generated when the robot returns to a previously visited location, provide a means of correcting for odometric drift.

Loop closures are modeled in two different ways depending on the form that the SLAM problem takes. In pose-based SLAM, loop closures represent relative constraints between non-consecutive poses. Alternatively, in landmark-based SLAM, loop closures occur when a landmark is re-observed and is represented by a pose to landmark constraint. Pose-to-pose loop closure constraints can be generated from landmark to pose constraints by marginalizing the landmarks from the SLAM solution. To avoid confusion between the pose-based and landmark-based SLAM problems, this thesis will only refer to pose-to-pose constraints as loop closures and will

refer to pose-to-landmark constraints as landmark measurements.

The SLAM problem is generally posed as a Bayesian inference problem. In the Bayesian formulation, the goal is to solve for the maximum likelihood estimates of the robot poses ($\mathbf{x} = \{\mathbf{x}_0, \cdots, \mathbf{x}_n\}$) and landmark positions (($\mathbf{l} = \{\mathbf{l}_0, \cdots, \mathbf{l}_m\}$) given a set of odometry ($\{\mathbf{y}_1^o, \cdots, \mathbf{y}_{n_o}^o\}$), landmark ($\{\mathbf{y}_1^l, \cdots, \mathbf{y}_{n_l}^l\}$) and loop closure measurements ($\{\mathbf{y}_1^{lc}, \cdots, \mathbf{y}_{n_{lc}}^{lc}\}$):

$$\begin{aligned}\left(\hat{\mathbf{x}}, \hat{\mathbf{l}}\right) &= \underset{\mathbf{x},\mathbf{l}}{\operatorname{argmax}}\, p(\mathbf{x}, \mathbf{l}|\mathbf{y}^o, \mathbf{y}^{lc}, \mathbf{y}^l) \\ &= \underset{\mathbf{x},\mathbf{l}}{\operatorname{argmax}}\, \frac{p(\mathbf{y}^o, \mathbf{y}^{lc}, \mathbf{y}^l|\mathbf{x}, \mathbf{l})}{p(\mathbf{y}^o, \mathbf{y}^{lc}, \mathbf{y}^l)}\end{aligned} \tag{2.28}$$

Typically, it is assumed that the measurements are conditionally independent so that $p(\mathbf{y}^o, \mathbf{y}^{lc}, \mathbf{y}^l|\mathbf{x}, \mathbf{l})$ can be factored as

$$p(\mathbf{y}^o, \mathbf{y}^{lc}, \mathbf{y}^l|\mathbf{x}, \mathbf{l}) = \prod_{i=1}^{n_o} p(\mathbf{y}_i^o|\mathbf{x}) \prod_{j=1}^{n_{lc}} p(\mathbf{y}_j^l|\mathbf{x}) \prod_{k=1}^{n_l} p(\mathbf{y}_k^l|\mathbf{x}, \mathbf{l}) \tag{2.29}$$

It should be noted that this formulation of the problem can be interpreted as a non-linear Bayesian smoothing problem. The major distinction between this formulation and the filtering problems discussed earlier is that none of the latent variables are marginalized from the problem in the smoothing problem whereas prior state variables are marginalized at each time step in a filtering problem.

Additionally it is typically assumed that the measurements are corrupted by additive zero-mean Gaussian noise so that the conditional probability distributions are given by

$$p(\mathbf{y}_i^o|\mathbf{x}) \propto e^{-\frac{1}{2}(\mathbf{y}_i^o - h_i(\mathbf{x}))^T \mathbf{\Lambda}_i^o (\mathbf{y}_i^o - h_i(\mathbf{x}))} \tag{2.30}$$

$$p(\mathbf{y}_j^{lc}|\mathbf{x}) \propto e^{-\frac{1}{2}(\mathbf{y}_j^l - h_j(\mathbf{x}))^T \mathbf{\Lambda}_j^{lc} (\mathbf{y}_j^{lc} - h_j(\mathbf{x}))} \tag{2.31}$$

$$p(\mathbf{y}_k^l|\mathbf{x}, \mathbf{l}) \propto e^{-\frac{1}{2}(\mathbf{y}_k^l - h_k(\mathbf{x},\mathbf{l}))^T \mathbf{\Lambda}_k^l (\mathbf{y}_k^l - h_k(\mathbf{x},\mathbf{l}))} \tag{2.32}$$

where $\mathbf{\Lambda}_i^o$, $\mathbf{\Lambda}_j^{lc}$ and $\mathbf{\Lambda}_k^l$ are the information matrices for the odometry, loop closure and landmark measurements respectively.

After applying these assumptions the SLAM solution can be expressed as

$$\begin{aligned}
\left(\hat{\mathbf{x}}, \hat{\mathbf{l}}\right) &= \underset{\mathbf{x},\mathbf{l}}{\operatorname{argmax}}\, p(\mathbf{x}, \mathbf{l}|\mathbf{y}^o, \mathbf{y}^{lc}, \mathbf{y}^l) \\
&= \underset{\mathbf{x},\mathbf{l}}{\operatorname{argmax}} \sum_{i=1}^{n_o} \log p(\mathbf{y}_i^o|\mathbf{x}) + \sum_{j=1}^{n_l} \log p(\mathbf{y}_j^l|\mathbf{x}, \mathbf{l}) \\
&= \underset{\mathbf{x},\mathbf{l}}{\operatorname{argmin}} - \sum_{i=1}^{n_o} \log p(\mathbf{y}_i^o|\mathbf{x}) - \sum_{j=1}^{n_l} \log p(\mathbf{y}_j^l|\mathbf{x}, \mathbf{l}) \\
&= \underset{\mathbf{x},\mathbf{l}}{\operatorname{argmin}} \sum_{i=1}^{n_o} \frac{1}{2}(\mathbf{y}_i^o - h_i(\mathbf{x}))^T \mathbf{\Lambda}_i^o (\mathbf{y}_i^o - h_i(\mathbf{x})) \\
&\quad + \sum_{j=1}^{n_{lc}} \frac{1}{2}(\mathbf{y}_j^l - h_j(\mathbf{x}))^T \mathbf{\Lambda}_j^{lc} (\mathbf{y}_j^l - h_j(\mathbf{x})) \\
&\quad + \sum_{k=1}^{n_l} \frac{1}{2}(\mathbf{y}_k^l - h_k(\mathbf{x}, \mathbf{l}))^T \mathbf{\Lambda}_k^l (\mathbf{y}_k^l - h_k(\mathbf{x}, \mathbf{l}))
\end{aligned} \tag{2.33}$$

which is a nonlinear least-squares (NLS) problem.

One standard approach to solving NLS problems is the Gauss-Newton method [74]. The Gauss-Newton method can be applied to SLAM problems by choosing an initial estimate for the poses and landmark positions, $\left(\hat{\mathbf{x}}^0, \hat{\mathbf{l}}^0\right)$, and then updating the pose and landmark estimates $\left(\hat{\mathbf{x}}^k, \hat{\mathbf{l}}^k\right)$ until they have converged using the following update rule

$$\begin{bmatrix} \hat{\mathbf{x}}^{k+1} \\ \hat{\mathbf{l}}^{k+1} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{x}}^k \\ \hat{\mathbf{l}}^k \end{bmatrix} - (J^T \Lambda J)^{-1} J^T \Lambda \mathbf{r} \tag{2.34}$$

where J is a matrix that contains the Jacobians of the measurement functions $h_i(\mathbf{x})$, $h_j(\mathbf{x})$ and $h_k(\mathbf{x}, \mathbf{l})$, Λ is a block diagonal matrix containing the corresponding measurement information matrices, and $\mathbf{r}$ is a vector containing the measurement residuals $\mathbf{y}_i^o - h_i(\mathbf{x})$, $\mathbf{y}_j^l - h_j(\mathbf{x})$, and $\mathbf{y}_k^l - h_k(\mathbf{x}, \mathbf{l})$. In the context of the nonlinear least squares problem, the information matrices act as weights for each measurement so that "larger" information matrices correspond to those measurements having more impact on the nonlinear least squares solution.

Along with the Gauss-Newton method, there are several related optimization techniques that are often applied to solve the SLAM problem including Levenberg-

Marquardt [75, 76] and Powell's Dog-Leg [77–79]. The main difference between these approaches and the Gauss-Newton method is the computation of the step size and direction for each update (i.e., , the $(J^T\Lambda J)^{-1}J^T\Lambda\mathbf{r}$ term in Eq. 2.34). However, the updates for these methods still contain the posterior information matrix ($J^T\Lambda J$) and as a result the interpretation of the information matrix as a weighting term for each measurement still applies.

2.3.2 Graphical Representations of the SLAM Problem

In the last ten years, factor graph representations of the SLAM problem have led to significant advances in the efficiency and scalability of SLAM solvers [9–11]. Factor graphs are a general framework for modeling functions that can be represented as products of factors (i.e., $f(\mathbf{x}) = \prod_i f_i(\mathbf{x}_i)$) [80].

Factor graphs arise in SLAM as a means of modeling the posterior pdf of the poses and landmarks (Eq. 2.29). The posterior is clearly a product of measurement likelihoods and thus can be modeled by a factor graph. An example of a SLAM factor graph is shown in Figure 2-1. In this context each factor, f_i in the graph corresponds to a measurement likelihood while each node in the graph corresponds to a latent variable in the model (either a pose or landmark).

Beyond providing a compact visual representation of the SLAM problem, factor graphs also provide additional insight into the structure of the SLAM problem that can be exploited to produce faster and more scalable algorithms. The major benefit of factor graph models for SLAM is that they expose the underlying sparsity that is inherent in SLAM problems [81, 82]. Specifically, it has been shown that SLAM factor graphs have only a sparse set of edges connecting nodes in the graph. Moreover, the sparsity pattern of the edges coincides with the sparsity pattern of the posterior information matrix, ($J^T\Lambda J$), that appears in the NLS SLAM update. In practice, this means that the matrix inversion operation in each NLS step can be replaced by a significantly more efficient sparse matrix inverse. Thus, factor graphs provide a significant and important connection between the Bayesian model of the SLAM problem and the linear algebra tools that are used to solve it [9–11].

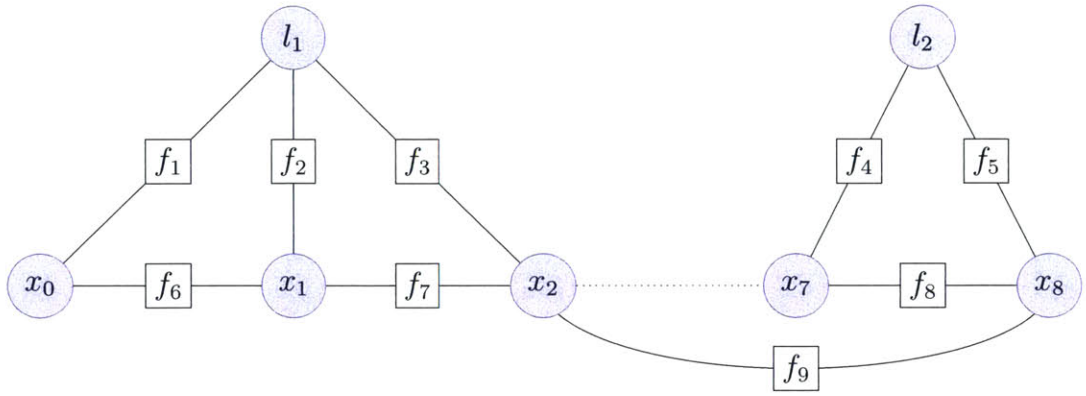


Figure 2-1: Example of a SLAM factor graph. $\mathbf{x}_i$ are poses, $\mathbf{l}_j$ are landmarks and f_k are the factors corresponding to odometry, landmark and loop closure measurements.

2.4 Robustness Issues in State Estimation and SLAM

Although Bayesian filtering and SLAM algorithms provide a powerful set of tools, they are also subject to significant robustness challenges. These robustness issues arise because often the underlying model of the state dynamics and measurements are approximate or incorrect or because the assumptions about the noise (i.e., zero-mean, Gaussian) are incorrect. When these errors occur they can cause catastrophic failures in the standard filtering and SLAM algorithms. The reason these algorithms fail is because they are based on least-squares estimation techniques that are inherently not robust to outliers and gross errors [29, 30].

The rest of this section will provide a brief discussion of the underlying causes of robustness issues in least-squares estimation as well as a discussion of some existing techniques for outlier rejection in least-squares estimation.

2.4.1 Non-Robustness of Least-Squares Estimators

It is a well known fact that least-squares estimators are not robust to outliers [29, 30]. To see this consider that in general a weighted linear least-squares estimator can be expressed as

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{i=1}^{n} (\mathbf{y}_i - H\mathbf{x})^T W_i (\mathbf{y}_i - H\mathbf{x}) \tag{2.35}$$

where $\mathbf{y}_i$ are measurement vectors and $W_i \succ 0$ is a matrix of weights for each measurement. The least-squares estimate $\mathbf{x}^*$ is given by

$$\mathbf{x}^* = (H^T W H)^{-1} H^T W (\mathbf{y} - H\mathbf{x}) \tag{2.36}$$

where

$$W = \begin{bmatrix} W_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & W_2 & \cdots & \mathbf{0} \\ \vdots & & \ddots & \\ \mathbf{0} & \cdots & & W_n \end{bmatrix} \quad H = \begin{bmatrix} H_1 \\ \vdots \\ H_n \end{bmatrix}$$

Note from Eq. 2.36 that the major influences on the least-squares estimate are 1) the weights W_i and 2) the magnitude of the residuals $(\mathbf{y} - H\mathbf{x})$.

Least-squares estimates are not robust because a measurement with a large residual will have an outsized influence on the resulting solution. Therefore, even a single outlier can cause a significant error in the least-squares estimate. Formally, this sensitivity to gross outliers can be captured by the *breakdown point* of an estimator [30]. The breakdown point is a measure of the fraction of arbitrarily incorrect measurements that can be processed by an estimator before the estimates become arbitrarily large. Given the least-squares estimate in Eq. 2.36, it is straight-forward to prove that the least-squares estimator has a breakdown point of 0.

Lemma 1. *The least-squares estimator* $\mathbf{x}^* = (H^T W H)^{-1} H^T W (\mathbf{y} - H\mathbf{x})$ *has a breakdown point of 0.*

Proof. Let the magnitude of the i^{th} measurement, $\|y_i\| \to \infty$. After examining Eq. 2.36 it is clear that as $\|y_i\| \to \infty$, $\|\mathbf{x}^*\| \to \infty$. Therefore, the least-squares estimator can not process any arbitrarily large measurements, and thus has a breakdown point of 0. □

Since the Kalman filter, extended Kalman filter and SLAM solvers all involve computations that are mathematically related to least-squares estimation this result also establishes that they are not robust to outliers and gross errors and have a

breakdown point of zero. The rest of this section discusses some common techniques for outlier mitigation in least-squares estimation.

2.4.2 Outlier Mitigation in Least-Squares Problems

There are two major outlier mitigation methods that are applied to least-squares problems: outlier detection and robust cost functions [42, 83]. Outlier detection techniques attempt to identify measurement outliers using statistical tests and then remove the measurements from the solution. Robust cost function approaches modify the standard least-squares cost function to reduce the impact of outliers on the estimates (in other words increase the breakdown point of the estimator). While this discussion will focus on least-squares estimation, the techniques can be extended and applied to state estimation and SLAM as well.

χ^2 Tests for Outlier Detection

Weighted least-squares problems can often be interpreted as maximum likelihood estimation using measurements with Gaussian noise. Given this interpretation, the weights correspond to the inverse covariance matrices, or information matrices, of the measurement noise.

Many outlier detection techniques for least-squares estimation rely on χ^2 tests because of the connection between Gaussian maximum likelihood estimation and least-squares. The reason χ^2 tests are often used has to do with the connection between normally distributed random variables and the χ^2 distribution. Given a set of m i.i.d. samples, $\{z_1, \cdots, z_m\}$ from a standard normal distribution, it can be shown that

$$\sum_{i}^{m} z_i^2 \sim \chi_m^2 \tag{2.37}$$

where χ_m^2 denotes the χ^2 distribution with m degrees of freedom.

Given the relationship between Gaussian random variables and the χ^2 distribution, a simple residual test can be applied to identify potential outliers in the measurements. Assume there is a prior estimate, $\hat{\mathbf{x}}$, with covariance P and a measurement vector $\mathbf{y}$

with an additive measurement noise covariance, R. Then the measurement residuals $(\mathbf{y} - H\hat{\mathbf{x}}) \sim \mathcal{N}(\mathbf{0}, HPH^T + R)$. Moreover, using Eq. 2.37 the standardized residuals should be distributed as

$$(\mathbf{y} - H\hat{\mathbf{x}})^T (HPH^T + R)^{-1} (\mathbf{y} - H\hat{\mathbf{x}}) \sim \chi^2_{n_y} \tag{2.38}$$

where n_y is the dimension of $\mathbf{y}$. Given the relationship between the residuals and the χ^2 distribution, a hypothesis test can be applied to determine whether the residuals are consistent with the expected Gaussian noise:

$$(\mathbf{y} - H\hat{\mathbf{x}})^T (HPH^T + R)^{-1} (\mathbf{y} - H\hat{\mathbf{x}}) \leq \chi^2_{inv}(p, n_{dof}) \tag{2.39}$$

where $\chi^2_{inv}(p, n_y)$ is the inverse χ-squared cdf with n_y degrees of freedom evaluated at p. If the inequality is satisfied, then the noise models are correct with probability p. If not, then there are likely outliers in $\mathbf{y}$ and the measurements should not be used to update $\hat{\mathbf{x}}$. Typically, p is set to a value close to 1 (e.g., 0.95 or 0.99) to avoid erroneously discarding measurement due to false positive outlier detections.

The one drawback of χ^2 tests and other hypothesis testing approaches to outlier detection is that they do not account for other potential sources of error that could explain the residuals. Other causes of large deviations in the residuals could be a poor initial estimate, incorrect measurement models or overly optimistic measurement noise models. While tests could be applied to detect those errors in isolation, it is difficult to design a hypothesis test that can weigh all of those factors and determine which are most likely. In essence, a χ^2 test can indicate that something has gone wrong, but can not necessarily determine what went wrong.

Robust Cost Functions

Another common method for addressing outliers in maximum likelihood estimation problems is to substitute an alternative cost function for the standard least-squares cost function. In most cases, these alternative cost functions are designed so that

outliers have minimal impact on the final solution. Typically these cost functions are applied as kernels in the standard least-squares cost function so that the robust function can be expressed as:

$$f_{robust}(\mathbf{x}) = \sum_{i=1}^{n} K\left((\mathbf{y}_i - H\mathbf{x})^T W_i (\mathbf{y}_i - H\mathbf{x})\right) \tag{2.40}$$

where $K(\cdot)$ is the robust kernel function. Robust kernel functions are designed so that as the residuals for a specific measurement $(\mathbf{y}_i - H\mathbf{x})$ become larger (i.e., in cases where outliers are more likely to be present) the contribution of that measurement to the solution decreases. In other words, the robust kernels reduce the weights of measurements corrupted by outliers in the least-squares solution.

Some of the most common kernel functions are the Huber kernel [30]

$$K_{Huber}(\delta) = \begin{cases} \delta^2 & |\delta| < b \\ 2b\,|\delta| - b^2 & \end{cases} \tag{2.41}$$

the Cauchy kernel

$$K_{Cauchy}(\delta) = b^2 \log\left(1 + \frac{\delta^2}{b^2}\right) \tag{2.42}$$

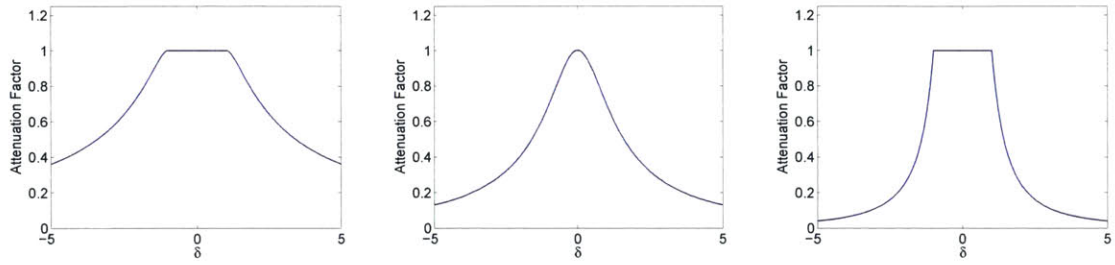
and the Blake-Zisserman kernel [83, 84]

$$K_{BZ}(\delta) = -\log\left(e^{-\delta^2} + e^{-b^2}\right) \tag{2.43}$$

where b is a positive constant. The attenuation factor of a robust kernel provides a means of visualizing how each kernel reduces the weights of measurements as the residuals increase. The attenuation factor of a robust kernel $K(\cdot)$ is defined as [83]

$$AF(\delta) = \frac{K(\delta)}{\delta^2} \tag{2.44}$$

and represents the ratio of the robust kernel and the standard least-squares cost as function of the magnitude of the residuals (δ). The attenuation factors for the Huber, Cauchy and Blake-Zisserman kernels are shown in Figure 2-2. For each kernel, it is



(a) Huber (b) Cauchy (c) Blake-Zisserman

Figure 2-2: Attenuation factors of common robust kernels

clear that as the residuals become larger, the weight of the measurement relative to the least-squares weight decreases and asymptotically the weights converge to zero. By decreasing the weights, each of these robust kernels achieves a break-down point larger than zero and in fact, the Huber kernel can be shown to have a break-down point of 0.5 [30].

2.5 Summary

This chapter provided background for Bayesian state estimation algorithms and the mathematics and solution of the SLAM problem. It also provided an overview of robustness issues associated with least-squares estimation and how those issues relate to the poor performance of standard filtering and SLAM algorithms when outliers occur. The next chapter will develop and analyze a robust filtering algorithm that addresses the robustness issues for state estimation algorithms by augmenting the standard dynamical system models with additional latent variables that capture the impact of outliers on both the state dynamics and measurements.

Chapter 3

Robust Filtering with Sparse Outliers

In state estimation problems, it is often assumed that the process and measurement noise in the system are Gaussian distributed [69, 70]. However, for many practical problems the Gaussian assumption is violated by difficult to model errors (i.e. multipath [85], state prediction errors in target tracking [69]) that can be interpreted as outliers relative to the nominal Gaussian noise distribution. Moreover, algorithms such as the Kalman filter (KF) and extended Kalman filter (EKF) are not robust to outliers and the accuracy of their state estimates significantly degrades when the Gaussian noise assumption does not hold [24].

A number of robust state estimation algorithms have been developed to mitigate the impact of outliers. Typically these algorithms focus on determining when measurements are corrupted with outliers and either ignoring them entirely [42] or reducing their effect on the updated state estimates [24, 27, 52, 55, 57]. Unfortunately, by focusing solely on measurement errors, these algorithms can not guarantee good performance when there are also large errors in the state predictions. In those cases, the algorithms incorrectly detect outliers in the measurements and end up ignoring information that could help correct the erroneous state estimates [46].

Furthermore, state propagation outliers are a significant problem. Many navigation systems rely on IMU dead-reckoning in lieu of a state prediction model. If there

are any IMU faults in these systems, standard robust filtering algorithms can not detect them. Similarly in SLAM systems where odometry is often used as a proxy for state prediction, the issue of faulty wheel odometry and the "slip or grip" problem [61] has proven to be difficult to address using standard outlier rejection techniques. Finally, target tracking algorithms often have coarse prediction models that are not accurate. Thus, if state prediction errors occur due to model mismatch, there is no means for standard robust filtering algorithms to correct the errors.

The main contribution of this chapter is a robust recursive filtering algorithm, the l_1-norm filter, that can provide accurate state estimates in the presence of both state prediction and measurement outliers. The l_1-norm filter detects the presence of outliers using the solution of a convex program. Given that information, the filter updates the state estimates by jointly estimating the detected errors and the states using the information filter [86]. The algorithm is computationally efficient as it combines a convex optimization with standard recursive filtering steps.

3.1 Robust State Estimation Using the l_1-norm Filter

This section develops the l_1-norm filter algorithm which consists of two parts:

1. Identification of outliers in the state propagation and measurements by solving a convex optimization problem

2. Updating the state estimates given the results of the error identification step

3.1.1 System Models and Problem Statement

It will be assumed that the state dynamics and measurements are linear and corrupted by both additive white Gaussian noise as well as additive sparse errors. Sparse in this context means that at least some components of the errors are equal to zero. Given these assumptions, the state propagation and measurement models are assumed to

take the form:

$$\mathbf{x}_{k+1} = F_k \mathbf{x}_k + \mathbf{w}_k + \mathbf{e}_k^p \tag{3.1}$$

$$\mathbf{y}_{k+1} = H_{k+1} \mathbf{x}_{k+1} + \mathbf{v}_{k+1} + \mathbf{e}_{k+1}^m \tag{3.2}$$

where F_k is the state transition matrix, H_{k+1} is the measurement matrix, $\mathbf{w}_k$ and $\mathbf{v}_{k+1}$ are the Gaussian process and measurement noise, respectively, and $\mathbf{e}_k^p$ and $\mathbf{e}_{k+1}^m$ represent the sparse errors. Note that without the errors, $\mathbf{e}_k^p$ and $\mathbf{e}_{k+1}^m$, these equations are in the standard form of the KF state propagation and measurement equations. The rest of the assumptions for the state prediction and measurement models are as follows:

1. $\mathbf{w}_k$ and $\mathbf{v}_{k+1}$ are white and zero-mean with covariances Q_k and R_{k+1} respectively
2. $\mathbf{w}_k$ and $\mathbf{v}_{k+1}$ are mutually uncorrelated (i.e. $E[\mathbf{v}_{k+1}\mathbf{w}_k^T] = 0, \forall\ k$)
3. The number of combined non-zero components of $\mathbf{e}_k^p$ and $\mathbf{e}_{k+1}^m$ is less than or equal to the number of measurements

The first two assumptions are standard for the KF. The final assumption about the sparse errors is required to ensure a valid state estimate using the l_1-norm filter. The third assumption is discussed in more detail during the state-update portion of this section.

The objective of the state estimation problem is to calculate a state estimate, $\hat{\mathbf{x}}_{k+1|k+1}$, that minimizes the mean squared state estimation error, $E[(\mathbf{x}_{k+1|k+1} - \hat{\mathbf{x}}_{k+1|k+1})^T(\mathbf{x}_{k+1|k+1} - \hat{\mathbf{x}}_{k+1|k+1})]$ given an estimate of the state at time k, $\hat{\mathbf{x}}_{k|k}$, and a set of measurements up to time $k+1$. It will be assumed that the estimation error at time k, $\tilde{\mathbf{x}}_{k|k} = \mathbf{x}_k - \hat{\mathbf{x}}_{k|k}$, is zero-mean and Gaussian distributed with covariance $P_{k|k}$.

For a system with state dynamics and measurements governed by Eqs. 3.1 and 3.2, solving for $\hat{\mathbf{x}}_{k+1|k+1}$ also entails solving for $\hat{\mathbf{e}}_{k|k+1}^p$ and $\hat{\mathbf{e}}_{k+1|k+1}^m$. It should be noted that without the sparsity assumption, this estimation problem is potentially ill-posed and could have multiple solutions. The sparsity assumption acts as a regularizer for

the estimation problem that favors "simpler" explanations of the measurement data when outliers occur.

3.1.2 Error Detection

This section formulates an optimization problem that can approximately solve for the sparse errors. The output of the optimization will be used to determine which components of $\mathbf{e}_k^p$ and $\mathbf{e}_{k+1}^m$ are non-zero. Discussion of how the states and errors are estimated given this information will be covered in the next subsection.

Before the optimization problem can be defined, the measurement residuals need to be expressed in terms of the sparse errors, $\mathbf{e}_k^p$ and $\mathbf{e}_{k+1}^m$, and the *a priori* state estimate. The *a priori* measurement residuals at time $k+1$ can be expressed as:

$$\begin{aligned}
\tilde{\mathbf{y}}_{k+1} &= \mathbf{y}_{k+1} - H_{k+1} F_k \hat{\mathbf{x}}_{k|k} \\
&= H_{k+1}\left(F_k \mathbf{x}_k + \mathbf{w}_k + \mathbf{e}_k^p - \hat{\mathbf{x}}_{k+1|k}\right) + \mathbf{v}_{k+1} + \mathbf{e}_{k+1}^m \qquad (3.3) \\
&= H_{k+1}\left(F_k \tilde{\mathbf{x}}_{k|k} + \mathbf{w}_k + \mathbf{e}_k^p\right) + \mathbf{v}_{k+1} + \mathbf{e}_{k+1}^m \qquad (3.4)
\end{aligned}$$

After rearranging terms in Eq. 3.4 and defining

$$\mathbf{e}_{k+1} \equiv \begin{bmatrix} \mathbf{e}_k^p \\ \mathbf{e}_{k+1}^m \end{bmatrix}$$

and

$$\mathbf{u}_{k+1} \equiv H_{k+1}\left(F_k \tilde{\mathbf{x}}_{k|k} + \mathbf{w}_k\right) + \mathbf{v}_{k+1},$$

the residuals can be related to the error terms by

$$\tilde{\mathbf{y}}_{k+1} = \begin{bmatrix} H_{k+1} & I \end{bmatrix} \mathbf{e}_{k+1} + \mathbf{u}_{k+1} \qquad (3.5)$$

The errors could be estimated from the under-determined system of equations in Eq. 3.5 by solving for the minimum l_2-norm vector that corresponds to the measurement residuals (using a pseudo-inverse least squares solution [86]). However, this approach

is not suitable for estimating sparse vectors such as $\mathbf{e}_{k+1}$ because it tends to allocate signal energy to all of the components of the vector being estimated instead of concentrating it on a few components, thus returning a non-sparse estimate of a sparse vector.

Based on the sparsity assumption, the estimates for $\mathbf{e}_{k+1}$ should have as few non-zero entries as possible. Additionally, if the error estimates are equal to the true error values (i.e. $\hat{\mathbf{e}}_{k+1} = \mathbf{e}_{k+1}$) then the corrected measurement residuals, $\tilde{\mathbf{y}} = \tilde{\mathbf{y}}_{k+1} - \begin{bmatrix} H_{k+1} & I \end{bmatrix} \hat{\mathbf{e}}_{k+1}$, will be equal to $\mathbf{u}_{k+1}$. Note that $\mathbf{u}_{k+1}$ is a zero-mean normally distributed random variable with covariance

$$\Sigma = H_{k+1} \left(F_k P_{k|k} F_k^T + Q_k \right) H_{k+1}^T + R_{k+1}$$

For a normally distributed random variable $\mathbf{p} \in \mathcal{R}^n$ with covariance, W, the weighted inner product $\mathbf{p}^T W^{-1} \mathbf{p}$ is χ^2 distributed with n degrees of freedom. Given these observations, one way to obtain a good estimate of $\mathbf{e}_{k+1}$ is to minimize the number of non-zero entries while ensuring that $\tilde{\mathbf{y}}^T \Sigma^{-1} \tilde{\mathbf{y}} \leq \tau$, where τ is set based on the χ^2 c.d.f. [87]. Mathematically this optimization can be expressed as

$$\begin{aligned} \min_{\hat{\mathbf{e}}_{k+1}} \quad & \|\hat{\mathbf{e}}_{k+1}\|_0 \\ \text{subject to} \quad & \tilde{\mathbf{y}}^T \Sigma^{-1} \tilde{\mathbf{y}} \leq \tau \end{aligned} \tag{3.6}$$

where $\|\cdot\|_0$ is a shorthand expression for the number of non-zero components of a vector [88]. Because this optimization involves searching over a combinatorial set of sparse vectors, it is computationally intractable in general [88]. Fortunately, a tractable approximate solution to Eq. 3.6, can be found by solving the convex optimization [87]

$$\begin{aligned} \min_{\hat{\mathbf{e}}_{k+1}} \quad & \|\hat{\mathbf{e}}_{k+1}\|_1 \\ \text{subject to} \quad & \tilde{\mathbf{y}}^T \Sigma^{-1} \tilde{\mathbf{y}} \leq \tau \end{aligned} \tag{3.7}$$

The optimization in Eq. 3.7 can be recast as a second-order cone program for which a number of efficient algorithms have been developed [47, 48].

In practice, the optimization posed in Eq. 3.7 is acting as a consistency check between the measurements and the *a priori* state estimate generated by the nominal state propagation model, $\hat{\mathbf{x}}_{k+1} = F_k\hat{\mathbf{x}}_k$. If there is an inconsistency, then the l_1 minimization can both detect and attribute it to specific error sources in the measurements and state propagation in one computationally efficient step. If there are no errors present, then the residuals should satisfy the inequality constraint with high probability and the error estimates will be equal to zero.

Although the l_1-minimization step tends to return a sparse estimate of the errors, the estimate often has small spurious non-zero components that are a result of measurement noise. To ensure that the error estimates are sufficiently sparse, the solution returned by the l_1-minimization is thresholded based on the expected noise level. Any elements of the l_1-optimal error estimates that are smaller than the expected noise level (as determined by a χ^2-test) are set to zero. This step ensures that only errors that are inconsistent with the Gaussian process and measurement noise are considered in the state update update portion of the algorithm. Sparse estimates of the errors could also be obtained by applying the reweighted l_1-norm minimization (RWL1) approach proposed by Candes et al. [89]. However, since RWL1 requires iteratively solving an l_1-minimization multiple times, it remains to be seen, if the solution can be generated at the high rate needed for navigation systems.

It should also be noted that, while there is extensive evidence in the compressed sensing literature that l_1-norm minimization encourages sparse solutions [89, 90], the solution to Equation 3.7 is not guaranteed to coincide with the solution to Equation 3.6. The impact of missed detections and false alarms in the error detection procedure will be discussed in more detail in Section 3.1.5.

3.1.3 State and Error Estimation

After performing the error detection, the state estimates are updated by augmenting the state vector with the non-zero error terms and then jointly estimating the errors

and states using the information filter. The combination of thresholding the l_1-optimal solution followed by re-estimation is a common procedure in sparse signal estimation usually referred to as debiasing [87, 91], because in practice the l_1-optimal solutions are biased [87].

The information filter is a recursive filter that is algebraically equivalent to the KF [92], but performs operations on the information matrix, $\Lambda_{k|k}$, and information state, $\hat{\mathbf{d}}_{k|k}$ instead of the state and covariance. Given a state estimate, $\hat{\mathbf{x}}_{k|k}$, and covariance, $P_{k|k}$, the information matrix and state are defined as:

$$\Lambda_{k|k} = \left(P_{k|k}\right)^{-1} \tag{3.8}$$

$$\hat{\mathbf{d}}_{k|k} = \Lambda_{k|k}\hat{\mathbf{x}}_{k|k} \tag{3.9}$$

The information filter is particularly useful for situations where some of the states have uninformative prior estimates (such as the non-zero terms of $\mathbf{e}_k^p$ and $\mathbf{e}_{k+1}^m$).

The *a priori* measurement residuals in Eq. 3.3 will be used to derive the information filter update for the state and error estimates. First, define the augmented state vector $\mathbf{z}_{k+1}$ as

$$\mathbf{z}_{k+1} = \begin{bmatrix} \bar{\mathbf{x}}_{k+1} \\ \mathbf{e}_k^{p,nz} \\ \mathbf{e}_{k+1}^{m,nz} \end{bmatrix} \tag{3.10}$$

where $\bar{\mathbf{x}}_{k+1} = F_k\mathbf{x}_k + \mathbf{w}_k$ and the superscript nz denotes only the non-zero components (as determined by the l_1-norm minimization) of the respective errors. After substituting in the definition of $\mathbf{z}_{k+1}$, the measurements can be expressed as

$$\begin{aligned} \mathbf{y}_{k+1} &= \begin{bmatrix} H_{k+1} & H_p & I_m \end{bmatrix} \mathbf{z}_{k+1} + \mathbf{v}_{k+1} \\ &= \bar{H}_{k+1}\mathbf{z}_{k+1} + \mathbf{v}_{k+1} \end{aligned} \tag{3.11}$$

where H_p is equal to the columns of H_{k+1} corresponding to the non-zero terms in $\hat{\mathbf{e}}_k^p$ and I_m is equal to the columns of the identity matrix corresponding to non-zero entries in $\hat{\mathbf{e}}_{k+1}^m$.

The prior estimate of $\bar{\mathbf{x}}_{k+1|k}$ can be expressed as

$$\hat{\bar{\mathbf{x}}}_{k+1|k} = F_k \hat{\mathbf{x}}_{k|k}$$

and the associated covariance is $P_{\bar{\mathbf{x}}} = F_k P_{k|k} F_k^T + Q_k$. Since the prior estimates of the errors are assumed to be uninformative, the information matrix $\hat{\mathbf{z}}_{k+1|k}$ will be

$$\Lambda_{k+1|k} = \begin{bmatrix} P_{\bar{\mathbf{x}}}^{-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{3.12}$$

with the information state, $\hat{\mathbf{d}}_{k+1|k}$ given by Eq. 3.9. After calculating the information matrix and state, they can be updated as follows [86]

$$\hat{\mathbf{d}}_{k+1|k+1} = \hat{\mathbf{d}}_{k+1|k} + \bar{H}_{k+1}^T R_{k+1}^{-1} \mathbf{y}_{k+1} \tag{3.13}$$

$$\Lambda_{k+1|k+1} = \Lambda_{k+1|k} + \bar{H}_{k+1}^T R_{k+1}^{-1} \bar{H}_{k+1} \tag{3.14}$$

After updating $\hat{\mathbf{d}}_{k+1|k+1}$ and $\Lambda_{k+1|k+1}$, the covariance $P_{k+1|k+1}^z$ and state estimate $\hat{\mathbf{z}}_{k+1|k+1}$ can be calculated from Eqs. 3.8 and 3.9, respectively.

Recall that the total number of non-zero entries in $\mathbf{e}_k^p$ and $\mathbf{e}_{k+1}^m$ was assumed to be less than or equal to the number of measurements. The update procedure in Eqs. 3.13 and 3.14 sets the upper bound on the allowable sparsity of the unmodeled errors. Note that the number of combined non-zero components of $\mathbf{e}_{k+1}^m$ and $\mathbf{e}_k^p$ must be less than or equal to the number of measurements in order to ensure that $\Lambda_{k+1|k+1}$ is full rank and can be inverted. If $\Lambda_{k+1|k+1}$ is singular then it can not be inverted and $\hat{\mathbf{z}}_{k+1|k+1}$ can not be calculated.

In practice, the upper bound on the sparsity of $\mathbf{e}_{k+1}^m$ and $\mathbf{e}_k^p$ could be relaxed by assuming a diffuse prior (i.e. a prior with a large covariance) instead of an uninformative prior for the sparse errors. In this case the *a priori* information matrix from

Algorithm 1 l_1-norm Filter

Require: $\hat{\mathbf{x}}_{k|k}$, $P_{k|k}$, $\mathbf{y}_{k+1}$

1. Solve l_1 minimization problem in Eq. 3.7 for $\hat{\mathbf{e}}_{k+1}$
2. Apply χ^2-test to determine non-zero components of $\hat{\mathbf{e}}_{k+1}$
3. Form information state ($\hat{\mathbf{d}}_{k+1|k}$) and matrix ($\Lambda_{k+1|k}$) for augmented state vector $\hat{\mathbf{z}}_{k+1|k}$
4. Update $\hat{\mathbf{d}}_{k+1|k}$, $\Lambda_{k+1|k}$ with Eqs. 3.13–3.14
5. Calculate $P^z_{k+1|k+1}$, $\hat{\mathbf{z}}_{k+1|k+1}$ with Eqs. 3.8–3.9
6. Calculate $\hat{\mathbf{x}}_{k+1|k+1}$, $P_{k+1|k+1}$ using Eqs. 3.15–3.16

return $\hat{\mathbf{x}}_{k+1|k+1}$, $P_{k+1|k+1}$

Eq. 3.12 will be

$$\Lambda_{k+1|k} = \begin{bmatrix} P_{\bar{\mathbf{x}}}^{-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \epsilon I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \epsilon I \end{bmatrix}$$

where ϵ is a small (i.e., $\ll 1$) positive constant. Using the diffuse prior, $\Lambda_{k+1|k+1}$ will be positive definite and thus invertible regardless of the number of non-zero elements in $\mathbf{e}^m_{k+1}$ and $\mathbf{e}^p_k$.

After calculating $\hat{\mathbf{z}}_{k+1|k+1}$, the posterior state estimate, $\hat{\mathbf{x}}_{k+1|k+1}$, corrected for the sparse errors, is

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\bar{\mathbf{x}}}_{k+1|k+1} + \hat{\mathbf{e}}^p_{k|k+1} \tag{3.15}$$

with covariance

$$P_{k+1|k+1} = P^{\bar{\mathbf{x}}}_{k+1|k+1} + P^{\mathbf{e}^p}_{k+1|k+1} + P_{\bar{x}e} + P_{e\bar{x}} \tag{3.16}$$

where $P^{\bar{\mathbf{x}}}_{k+1|k+1}$ is the covariance of $\hat{\bar{\mathbf{x}}}_{k+1|k+1}$, $P^{\mathbf{e}^p}_{k+1|k+1}$ is the covariance of $\hat{\mathbf{e}}^p_{k|k+1}$, and $P_{\bar{x}e}$ and $P_{e\bar{x}}$ are the cross covariance matrices of $\hat{\bar{\mathbf{x}}}_{k+1|k+1}$ and $\hat{\mathbf{e}}^p_{k|k+1}$, all of which can be obtained from $P^z_{k+1|k+1}$:

$$P^z_{k+1|k+1} = \begin{bmatrix} P^{\bar{\mathbf{x}}}_{k+1|k+1} & P_{\bar{x}e} & \cdot \\ P_{e\bar{x}} & P^{\mathbf{e}^p}_{k+1|k+1} & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix}$$

3.1.4 Algorithm Summary

The l_1-norm filter is summarized in Algorithm 1. There are two main components of the algorithm: outlier detection and a state update based on the outlier detection. Steps 1 and 2 encompass the outlier detection portion of the algorithm, where a constrained l_1-norm optimization is used to estimate the sparse vectors $\mathbf{e}_k^p$ and $\mathbf{e}_{k+1}^m$. A χ^2-test is applied to the error estimates calculated by the l_1-norm optimization to determine which non-zero components of $\hat{\mathbf{e}}_k^p$ or $\hat{\mathbf{e}}_{k+1}^m$ are too large to be explained by the Gaussian process and measurement noise. The large non-zero components of $\hat{\mathbf{e}}_k^p$ and $\hat{\mathbf{e}}_{k+1}^m$ are then re-estimated in the state update step to calculate the robust state estimates.

The state update phase of the algorithm occurs in steps 3–6. The states and non-zero components of the errors are solved for by augmenting the state vector and then processing the measurements using the information filter. This portion of the algorithm is closely related to the KF. Calculating the information state and information matrix (step 3) requires applying the KF state propagation equations, while the state update equations for the KF are analogous to steps 4–6.

3.1.5 Algorithm Analysis

This section will derive closed form expressions for the posterior state estimates and covariance using the l_1-norm filter. These expressions will provide additional insight into the l_1-norm filter and allow analysis of the l_1-norm filter when the errors detected by the l_1-norm minimization are incorrect.

To simplify the derivations that follow, it will be assumed without loss of generality that the states and measurements have been ordered so that they can be partitioned into subsets that are impacted by $\mathbf{e}_k^p$ and $\mathbf{e}_{k+1}^m$. After ordering the states and measurements, $\mathbf{x}_{k+1}$, $\mathbf{y}_{k+1}$, H_{k+1} and R_{k+1} can be partitioned as

$$\mathbf{x}_{k+1} = \begin{bmatrix} \mathbf{x}_{\bar{p}} \\ \mathbf{x}_p \end{bmatrix}, \quad \mathbf{y}_{k+1} = \begin{bmatrix} \mathbf{y}_u \\ \mathbf{y}_c \end{bmatrix}, \quad H_{k+1} = \begin{bmatrix} H_{\bar{p}u} & H_{pu} \\ H_{\bar{p}c} & H_{pc} \end{bmatrix}, \quad R_{k+1} = \begin{bmatrix} R_u & \mathbf{0} \\ \mathbf{0} & R_c \end{bmatrix}$$

where the subscripts u and c denoted corrupted and uncorrupted measurements respectively, and the subscripts $\bar{p}$ and p indicate state variables that are uncorrupted and corrupted by $\mathbf{e}_k^p$ respectively. Similarly, the *a priori* state information matrix and covariance matrix can be partitioned as

$$\Lambda_{k+1|k} = \begin{bmatrix} \Lambda_{\bar{p}} & \Lambda_{\bar{p}p} \\ \Lambda_{p\bar{p}} & \Lambda_p \end{bmatrix} \text{ and } P_{k+1|k} = \begin{bmatrix} P_{\bar{p}} & P_{\bar{p}p} \\ P_{p\bar{p}} & P_p \end{bmatrix}$$

The posterior covariance of the states and errors can be calculated by inverting the posterior information matrix in Eq. 3.14:

$$P^z_{k+1|k+1} = \begin{bmatrix} P^{-1}_{k+1|k} + H^T R^{-1} H & H^T R^{-1} H_p & H^T R^{-1} I_m \\ H_p^T R^{-1} H & H_p^T R^{-1} H_p & H_p^T R^{-1} I_m \\ I_m^T R^{-1} H & I_m^T R^{-1} H_p & I_m^T R^{-1} I_m \end{bmatrix}^{-1}$$

Note that the general form for a blockwise inverse of a matrix is

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} (A - BD^{-1}C)^{-1} & -(A - BD^{-1}C)^{-1}BD^{-1} \\ -D^{-1}C(A - BD^{-1}C)^{-1} & D^{-1} + D^{-1}C(A - BD^{-1}C)^{-1}BD^{-1} \end{bmatrix} \tag{3.17}$$

The derivation of the update formulas will proceed by applying blockwise inversion using $A = P^{-1}_{k+1|k} + H^T R^{-1} H$. After selecting A, D^{-1} can also be calculated by blockwise inversion:

$$D^{-1} = \begin{bmatrix} (H_{pu}^T R_u^{-1} H_{pu})^{-1} & -(H_{pu}^T R_u^{-1} H_{pu})^{-1} H_{pc}^T \\ H_{pc}(H_{pu}^T R_u^{-1} H_{pu})^{-1} & R_c + H_{pc}(H_{pu}^T R_u^{-1} H_{pu})^{-1} H_{pc}^T \end{bmatrix} \tag{3.18}$$

Given these definitions, it can be shown that the covariance term $P^{\bar{x}}_{k+1|k+1}$ in Eq. 3.16 can be described using an update formula similar to a KF update:

Lemma 2. $P^{\bar{x}}_{k+1|k+1} = (I - \overline{KH}) P^{\bar{x}}_{k+1|k}$ *where*

$$\overline{KH} = \begin{bmatrix} \bar{K} H_{\bar{p}u} & \mathbf{0} \\ -\Lambda_p^{-1} \Lambda_{p\bar{p}} \bar{K} H_{\bar{p}u} & \mathbf{0} \end{bmatrix}, \quad \bar{K} = P_{\bar{p}} H_{\bar{p}u}^T \left(\bar{R}_u + H_{\bar{p}u} P_{\bar{p}} H_{\bar{p}u}^T \right)^{-1}$$

$$\bar{R}_u = (R_u^{-1} - R_u^{-1} H_{pu} (H_{pu}^T R_u^{-1} H_{pu})^{-1} H_{pu}^T R_u^{-1})^{-1}$$

Proof. Note that $P_{k+1|k+1}^{\bar{x}}$ is equivalent to the the top left entry in $P_{k+1|k+1}^{z}$. Therefore it can be expressed as

$$\begin{aligned} P_{k+1|k+1}^{\bar{x}} &= (P_{k+1|k}^{-1} + H^T R^{-1} H - B D^{-1} C)^{-1} \\ &= \begin{bmatrix} \Lambda_{\bar{p}} + H_{\bar{p}u}^T \bar{R}_u^{-1} H_{\bar{p}u} & \Lambda_{\bar{p}p} \\ \Lambda_{p\bar{p}} & \Lambda_p \end{bmatrix}^{-1} \end{aligned}$$

$P_{k+1|k+1}^{\bar{x}}$ can now be calculated using blockwise inversion, but first note that $P_{\bar{p}} = (\Lambda_{\bar{p}} - \Lambda_{\bar{p}p} \Lambda_p^{-1} \Lambda_{p\bar{p}})^{-1}$. Using this fact, the upper left hand term of $P_{k+1|k+1}^{\bar{x}}$ is

$$\begin{aligned} P_{\bar{p}}^{\bar{x}} &= (\Lambda_{\bar{p}} - \Lambda_{\bar{p}p} \Lambda_p^{-1} \Lambda_{p\bar{p}} + H_{\bar{p}u}^T \bar{R}_u^{-1} H_{\bar{p}u})^{-1} \\ &= (P_{\bar{p}} + H_{\bar{p}u}^T \bar{R}_u^{-1} H_{\bar{p}u})^{-1} \\ &= P_{\bar{p}} + P_{\bar{p}} H_{\bar{p}u}^T \left(\bar{R}_u + H_{\bar{p}u} P_{\bar{p}} H_{\bar{p}u}^T \right)^{-1} P_{\bar{p}} \\ &= (I - \bar{K} H_{\bar{p}u}) P_{\bar{p}} \end{aligned}$$

where the third equality follows from the matrix inversion lemma [93].

Applying the rest of the blockwise inverse formula leads to

$$\begin{aligned} P_{k+1|k+1}^{\bar{x}} &= \begin{bmatrix} (I - \bar{K} H_{\bar{p}u}) P_{\bar{p}} & -(I - \bar{K} H_{\bar{p}u}) P_{\bar{p}} \Lambda_{\bar{p}p} \Lambda_p^{-1} \\ -\Lambda_p^{-1} \Lambda_{p\bar{p}} (I - \bar{K} H_{\bar{p}u}) P_{\bar{p}} & \Lambda_p^{-1} + \Lambda_p^{-1} \Lambda_{p\bar{p}} (I - \bar{K} H_{\bar{p}u}) P_{\bar{p}} \Lambda_{\bar{p}p} \Lambda_p^{-1} \end{bmatrix} \\ &= \begin{bmatrix} (I - \bar{K} H_{\bar{p}u}) P_{\bar{p}} & P_{\bar{p}p} - \bar{K} H_{\bar{p}u} P_{\bar{p}p} \\ P_{p\bar{p}} + \Lambda_p^{-1} \Lambda_{p\bar{p}} \bar{K} H_{\bar{p}u} P_{\bar{p}} & P_p + \Lambda_p^{-1} \Lambda_{p\bar{p}} \bar{K} H_{\bar{p}u} P_{\bar{p}p} \end{bmatrix} \\ &= (I - \overline{KH}) P_{k+1|k}^{\bar{x}} \end{aligned}$$

□

Given Eq. 3.18 and Lemma 2 the final form of the state estimates and covariance in the l_1-norm filter are given by the following theorem.

Theorem 3. *The l_1-norm filter state estimates in Eq. 3.15 can be expressed as:*

$$\hat{\mathbf{x}}_{k+1|k+1} = \begin{bmatrix} \hat{\mathbf{x}}^{\bar{p}}_{k+1|k} + \bar{K}(\mathbf{y}_u - H_{\bar{p}u}\hat{\mathbf{x}}^{\bar{p}}_{k+1|k}) \\ H^{-L}_{pu}(\mathbf{y}_u - H_{\bar{p}u}\hat{\mathbf{x}}^{\bar{p}}_{k+1|k+1}) \end{bmatrix} \tag{3.19}$$

In addition, the posterior covariance matrix in Eq. 3.16 is given by:

$$P_{k+1|k+1} = \begin{bmatrix} P^{\bar{x}}_{\bar{p}} & -P^{\bar{x}}_{\bar{p}}(H^{-L}_{pu}H_{\bar{p}u})^T \\ -H^{-L}_{pu}H_{\bar{p}u}P^{\bar{x}}_{\bar{p}} & (H^T_{pu}R^{-1}_u H_{pu})^{-1} + H^{-L}_{pu}H_{\bar{p}u}P^{\bar{x}}_{\bar{p}}(H^{-L}_{pu}H_{\bar{p}u})^T \end{bmatrix} \tag{3.20}$$

where

$$H^{-L}_{pu} = \left(H^T_{pu}R^{-1}_u H_{pu}\right)^{-1} H^T_{pu}R^{-1}_u$$

Proof. Using Eqs. 3.9 and 3.13 and the blockwise inversion formula for $P^z_{k+1|k+1}$, the updated state and error estimates are

$$\begin{aligned} \hat{\bar{\mathbf{x}}}_{k+1|k+1} &= P^{\bar{x}}_{k+1|k+1}\left(P^{-1}_{k+1|k}\hat{\mathbf{x}}_{k+1|k} + H^TR^{-1}\mathbf{y} - BD^{-1}\mathbf{i}_e\right) \\ \hat{\mathbf{e}}^{nz}_{k+1|k+1} &= -D^{-1}CP^{\bar{x}}_{k+1|k+1}\left(P^{-1}_{k+1|k}\hat{\mathbf{x}}_{k+1|k} + H^TR^{-1}\mathbf{y} - BD^{-1}\mathbf{i}_e\right) + D^{-1}\mathbf{i}_e \\ &= -D^{-1}C\hat{\bar{\mathbf{x}}}_{k+1|k+1} + D^{-1}\mathbf{i}_e \end{aligned}$$

where

$$\mathbf{i}_e = \begin{bmatrix} H^T_pR^{-1}\mathbf{y} \\ I^T_mR^{-1}\mathbf{y} \end{bmatrix}$$

The matrices BD^{-1} and $D^{-1}C$ arise from the blockwise inversion of $\Lambda_{k+1|k+1}$ and can be shown to be

$$BD^{-1} = \begin{bmatrix} (H^{-L}_{pu}H_{\bar{p}u})^T & H^T_{\bar{p}c} - (H^{-L}_{pu}H_{\bar{p}u})^TH^T_{pc} \\ I & \mathbf{0} \end{bmatrix}, \quad D^{-1}C = \begin{bmatrix} H^{-L}_{pu}H_{\bar{p}u} & I \\ H_{\bar{p}c} - H_{pc}H^{-L}_{pu}H_{\bar{p}u} & \mathbf{0} \end{bmatrix}$$

Note that only $\hat{\mathbf{e}}^p$ appears in Eq. 3.15, thus it is only necessary to calculate $\hat{\mathbf{e}}^{p,nz}$. After substituting the values of BD^{-1}, $D^{-1}C$ and $\mathbf{i}_e$ the estimates of the non-zero $\mathbf{e}^p$ terms are:

$$\hat{\mathbf{e}}^{p,nz} = H^{-L}_{pu}(\mathbf{y}_u - H_{\bar{p}u}\hat{\bar{\mathbf{x}}}^{\bar{p}}_{k+1|k+1}) - \hat{\bar{\mathbf{x}}}^p_{k+1|k+1} \tag{3.21}$$

After substituting BD^{-1} and the value for $P^{\bar{x}}_{k+1|k+1}$ from Lemma 2 into the update equation for $\hat{\hat{\mathbf{x}}}_{k+1|k+1}$ the result is

$$\hat{\hat{\mathbf{x}}}_{k+1|k+1} = \begin{bmatrix} \hat{\hat{\mathbf{x}}}^{\bar{p}}_{k+1|k+1} \\ \hat{\hat{\mathbf{x}}}^{p}_{k+1|k+1} \end{bmatrix} = \begin{bmatrix} (I - \bar{K}H_{\bar{p}u})\hat{\hat{\mathbf{x}}}^{p}_{k+1|k} + (I - \bar{K}H_{\bar{p}u})P_{\bar{p}}H^T_{\bar{p}u}\bar{R}^{-1}_u\mathbf{y}_u \\ \hat{\hat{\mathbf{x}}}^{p}_{k+1|k+1} \end{bmatrix} \tag{3.22}$$

The second term in the $\hat{\hat{\mathbf{x}}}^{\bar{p}}_{k+1|k+1}$ can be simplified using a Schur identity [93] as

$$\begin{aligned} (I - \bar{K}H_{\bar{p}u})P_{\bar{p}}H^T_{\bar{p}u}\bar{R}^{-1}_u\mathbf{y}_u &= (P^{-1}_{\bar{p}} + H^T_{\bar{p}u}\bar{R}^{-1}_u H_{\bar{p}u})^{-1}H^T_{\bar{p}u}\bar{R}^{-1}_u\mathbf{y}_u \\ &= P_{\bar{p}}H^T_{\bar{p}u}(R_u + H_{\bar{p}u}P_{\bar{p}}H^T_{\bar{p}u})^{-1} \\ &= \bar{K}\mathbf{y}_u \end{aligned}$$

Thus the estimate of $\bar{\mathbf{x}}^{\bar{p}}$ is

$$\hat{\hat{\mathbf{x}}}^{\bar{p}}_{k+1|k+1} = \hat{\hat{\mathbf{x}}}^{p}_{k+1|k} + \bar{K}(\mathbf{y}_u - H_{\bar{p}u}\hat{\hat{\mathbf{x}}}^{p}_{k+1|k}) \tag{3.23}$$

Combining the results in Eqs. 3.21 and 3.23 leads to the final form of the l_1-norm filter state estimates given in Eq. 3.15:

$$\begin{aligned} \hat{\mathbf{x}}_{k+1|k+1} &= \begin{bmatrix} \hat{\hat{\mathbf{x}}}^{\bar{p}}_{k+1|k+1} \\ \hat{\hat{\mathbf{x}}}^{p}_{k+1|k+1} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{e}}^{p,nz} \end{bmatrix} \\ &= \begin{bmatrix} \hat{\hat{\mathbf{x}}}^{p}_{k+1|k} + \bar{K}(\mathbf{y}_u - H_{\bar{p}u}\hat{\hat{\mathbf{x}}}^{p}_{k+1|k}) \\ \left(H^T_{pu}R^{-1}_u H_{pu}\right)^{-1} H^T_{pu}R^{-1}_u(\mathbf{y}_u - H_{\bar{p}u}\hat{\mathbf{x}}^{\bar{p}}_{k+1|k+1}) \end{bmatrix} \end{aligned}$$

This proves the state estimate portion of the theorem.

The matrices $P^{e^p}_{k+1|k+1}$, $P_{\bar{x}e}$, and $P_{e\bar{x}}$ in Eq. 3.16 can be extracted from $P^{z}_{k+1|k+1}$ using the blockwise inversion formula. After substituting in the values of BD^{-1}, $D^{-1}C$ and D^{-1} the sum of $P^{e^p}_{k+1|k+1}$, $P_{\bar{x}e}$, and $P_{e\bar{x}}$ is

$$P^{\mathbf{e}^p}_{k+1|k+1} + P_{\bar{x}e} + P_{e\bar{x}} = \begin{bmatrix} \mathbf{0} & -P^{\bar{x}}_{\bar{p}p} - P^{\bar{x}}_{\bar{p}}(H^{-L}_{pu}H_{\bar{p}u})^T \\ -P^{\bar{x}}_{p\bar{p}} - H^{-L}_{pu}H_{\bar{p}u}P^{\bar{x}}_{\bar{p}} & -P^{\bar{x}}_{p} + (H^T_{pu}R^{-1}_u H_{pu})^{-1} + H^{-L}_{pu}H_{\bar{p}u}P^{\bar{x}}_{\bar{p}}(H^{-L}_{pu}H_{\bar{p}u})^T \end{bmatrix} \tag{3.24}$$

Substituting Eq. 3.24 into Eq. 3.16 gives the expression for $P_{k+1|k+1}$ in the theorem:

$$\begin{aligned} P_{k+1|k+1} &= \begin{bmatrix} P_{\bar{p}}^{\bar{x}} & P_{\bar{p}p}^{\bar{x}} \\ P_{p\bar{p}}^{\bar{x}} & P_{p}^{\bar{x}} \end{bmatrix} + P_{k+1|k+1}^{\mathbf{e}^p} + P_{\bar{x}e} + P_{e\bar{x}} \\ &= \begin{bmatrix} P_{\bar{p}}^{\bar{x}} & -P_{\bar{p}}^{\bar{x}}(H_{pu}^{-L}H_{\bar{p}u})^T \\ -H_{pu}^{-L}H_{\bar{p}u}P_{\bar{p}}^{\bar{x}} & (H_{pu}^T R_u^{-1} H_{pu})^{-1} + H_{pu}^{-L}H_{\bar{p}u}P_{\bar{p}}^{\bar{x}}(H_{pu}^{-L}H_{\bar{p}u})^T \end{bmatrix} \end{aligned}$$

□

There are several conclusions about the behavior and performance of the l_1-norm filter that can be drawn from Theorem 3. First, notice that the estimate of $\mathbf{x}_p$ in Eq. 3.19 is in fact a least-squares estimate given $\hat{\mathbf{x}}_{k+1|k+1}^{\bar{p}}$ and the uncorrupted measurements, $\mathbf{y}_u$. In other words, the l_1-norm filter is re-initializing the estimate of $\mathbf{x}_p$ using the current set of uncorrupted measurements. Additionally, note that the estimates and covariance do not depend on the measurements corrupted by $\mathbf{e}_{k+1}^m$. This can be seen by observing that the updates do not include any terms that involve $\mathbf{y}_c$, H_c, and R_c. Thus, the same estimates can be reached by discarding the measurements that correspond to non-zero $\mathbf{e}_{k+1}^m$ detections before the joint state and error estimation step. These observations also indicate that the performance of the l_1-norm filter, for the case when only measurement outliers are present, should be comparable to a KF that discards measurements with residuals that exceed a χ^2 threshold. This behavior is verified using Monte Carlo simulations in Section 3.3.

In addition, when all sparse errors are correctly detected, the l_1-norm filter estimates are unbiased. The proof of this result will require the following lemma.

Lemma 4. $\bar{K}H_{pu} = \mathbf{0}$

Proof. First note that

$$\begin{aligned} \bar{R}_u^{-1}H_{pu} &= R_u^{-1}H_{pu} - R_u^{-1}H_{pu}(H_{pu}^T R_u^{-1} H_{pu})^{-1} H_{pu}^T R_u^{-1} H_{pu} \\ &= R_u^{-1}H_{pu} - R_u^{-1}H_{pu} = \mathbf{0} \end{aligned}$$

Applying the matrix inversion lemma [93] shows that

$$\left(\bar{R}_u + H_{\bar{p}u} P_{\bar{p}} H_{\bar{p}u}^T\right)^{-1} = \bar{R}_u^{-1} - \bar{R}_u^{-1} H_{\bar{p}u} (P_{\bar{p}}^{-1} + H_{\bar{p}u} R_u^{-1} H_{\bar{p}u}^T)^{-1} H_{\bar{p}u}^T \bar{R}_u^{-1}$$

Combining these results shows that

$$\begin{aligned}
\bar{K} H_{pu} &= P_{\bar{p}} H_{\bar{p}u}^T \left(\bar{R}_u + H_{\bar{p}u} P_{\bar{p}} H_{\bar{p}u}^T\right)^{-1} H_{pu} \\
&= P_{\bar{p}} H_{\bar{p}u}^T (\bar{R}_u^{-1} H_{pu} - \bar{R}_u^{-1} H_{\bar{p}u} (P_{\bar{p}}^{-1} + H_{\bar{p}u} R_u^{-1} H_{\bar{p}u}^T)^{-1} H_{\bar{p}u}^T \bar{R}_u^{-1} H_{pu}) = \mathbf{0}
\end{aligned}$$

□

Theorem 5. *If Step 2 of Algorithm 1 detects all non-zero components of* $\mathbf{e}_k^p$ *and* $\mathbf{e}_{k+1}^m$ *then* $\hat{\mathbf{x}}_{k+1|k+1}$ *is unbiased.*

Proof. Using Eq. 3.19, the posterior state estimation error is

$$\begin{aligned}
\tilde{\mathbf{x}}_{k+1|k+1} &= \begin{bmatrix} \tilde{\mathbf{x}}_{k+1|k+1}^{\bar{p}} \\ \tilde{\mathbf{x}}_{k+1|k+1}^{p} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{x}_{k+1}^{\bar{p}} - \hat{\mathbf{x}}_{k+1|k}^{\bar{p}} + \bar{K}(\mathbf{y}_u - H_{\bar{p}u} \hat{\mathbf{x}}_{k+1|k}^{\bar{p}}) \\ \mathbf{x}_{k+1}^{p} - \left(H_{pu}^T R_u^{-1} H_{pu}\right)^{-1} H_{pu}^T R_u^{-1} (\mathbf{y}_u - H_{\bar{p}u} \hat{\mathbf{x}}_{k+1|k+1}^{\bar{p}}) \end{bmatrix} \\
&= \begin{bmatrix} (I - \bar{K} H_{\bar{p}u}) \tilde{\mathbf{x}}_{k+1|k}^{\bar{p}} + \bar{K}(H_{pu} \mathbf{x}_{k+1}^{p} + \mathbf{v}_{k+1}^{u}) \\ -\left(H_{pu}^T R_u^{-1} H_{pu}\right)^{-1} H_{pu}^T R_u^{-1} (H_{pu} \tilde{\mathbf{x}}_{k+1|k+1}^{\bar{p}} + \mathbf{v}_{k+1}^{u}) \end{bmatrix}
\end{aligned} \tag{3.25}$$

After applying Lemma 4, $\tilde{\mathbf{x}}_{k+1|k+1}^{\bar{p}}$ can be shown to be

$$\tilde{\mathbf{x}}_{k+1|k+1}^{\bar{p}} = (I - \bar{K} H_{\bar{p}u}) \tilde{\mathbf{x}}_{k+1|k}^{\bar{p}} + \bar{K} \mathbf{v}_{k+1}^{u}$$

and thus $E\left[\tilde{\mathbf{x}}_{k+1|k+1}^{\bar{p}}\right] = \mathbf{0}$. Moreover this implies that $E\left[\tilde{\mathbf{x}}_{k+1|k+1}^{p}\right] = \mathbf{0}$. □

In the case where the error detection works perfectly, these results indicate that the l_1-norm filter performs as desired: it ignores faulty measurements that could negatively impact the state estimates and it corrects erroneous state estimates. But since the l_1-norm solution is not guaranteed to correctly detect $\mathbf{e}_{k+1}^m$ and $\mathbf{e}_k^p$, it is also

important to evaluate the impact of incorrect error detection, either false alarms or missed detections.

Mathematically, both false alarms and missed detections can be defined in terms of the support, $supp(\cdot)$, of $\mathbf{e}_k^p$ and $\mathbf{e}_{k+1}^m$, where $supp(\mathbf{x})$ is the set of non-zero components of $\mathbf{x}$. If $supp(\hat{\mathbf{e}}_k^p) \supset supp(\mathbf{e}_k^p)$ **or** $supp(\hat{\mathbf{e}}_{k+1}^m) \supset supp(\mathbf{e}_{k+1}^m)$, then false alarms have occurred and if $supp(\hat{\mathbf{e}}_k^p) \subset supp(\mathbf{e}_k^p)$ **or** $supp(\hat{\mathbf{e}}_{k+1}^m) \subset supp(\mathbf{e}_{k+1}^m)$, then there are missed detections.

In the case of false alarms (i.e. incorrectly detecting an error when it is not present), the l_1-norm estimates will still be unbiased, with the only penalty for the incorrect detection being a potential increase in the posterior state covariance.

Theorem 6 (False Alarm Case). *If* $supp(\hat{\mathbf{e}}_k^p) \supset supp(\mathbf{e}_k^p)$ ***or*** $supp(\hat{\mathbf{e}}_{k+1}^m) \supset supp(\mathbf{e}_{k+1}^m)$ *then* $\hat{\mathbf{x}}_{k+1|k+1}$ *will be unbiased. If* $supp(\hat{\mathbf{e}}_k^p) = supp(\mathbf{e}_k^p)$ ***and*** $supp(\hat{\mathbf{e}}_{k+1}^m) \supset supp(\mathbf{e}_{k+1}^m)$, *it can also be shown that* $P_{k+1|k+1} > P_{k+1|k+1}^{nfa}$, *where* $P_{k+1|k+1}^{nfa}$ *is the posterior covariance if no false alarms had occurred. In general however, the relative size of the posterior covariance can not be established, i.e., whether* $P_{k+1|k+1} \lesseqgtr P_{k+1|k+1}^{nfa}$.

Proof. If there are only false alarms, the state estimates will take the same form as Eq. 3.19. Moreover this also means that the residuals will take the same form as Eq. 3.25 and thus the estimates will remain unbiased. In other words, because the true errors were also detected, they can not have any impact on the state estimates in the form of a bias.

To simplify the covariance portion of the proof, $\mathbf{e}_{k+1}^m$ and $\mathbf{e}_k^p$ false alarms will be handled separately. If $supp(\hat{\mathbf{e}}_{k+1}^m) \supset supp(\mathbf{e}_{k+1}^m)$, the information matrix without false alarms and the l_1-norm filter information matrix will take the following forms:

$$\Lambda_{k+1|k+1}^{nfa} = \Lambda_{k+1|k} + H_u^T R_u^{-1} H_u + H_c^T R_c^{-1} H_c$$
$$\Lambda_{k+1|k+1}^{l_1} = \Lambda_{k+1|k} + H_u^T R_u^{-1} H_u$$

Taking the difference of the two shows that

$$\Lambda_{k+1|k+1}^{nfa} - \Lambda_{k+1|k+1}^{l_1} = H_c^T R_c^{-1} H_c > 0 \tag{3.26}$$

Which implies that $\Lambda^{nfa}_{k+1|k+1} > \Lambda^{l_1}_{k+1|k+1}$ which in turn implies that $P^{nfa}_{k+1|k+1} < P^{l_1}_{k+1|k+1}$.

If $supp(\hat{\mathbf{e}}^p_k) \supset supp(\mathbf{e}^p_k)$, the information matrix without false alarms and the l_1-norm filter information matrix will take the following forms:

$$\Lambda^{nfa}_{k+1|k+1} = \begin{bmatrix} \Lambda_{\bar{p}} + H^T_{\bar{p}} R^{-1} H_{\bar{p}} & \Lambda_{\bar{p}p} + H^T_{\bar{p}} R^{-1} H_p \\ \Lambda_{p\bar{p}} + H^T_p R^{-1} H_{\bar{p}} & \Lambda_p + H^T_p R^{-1} H_p \end{bmatrix}$$

$$\Lambda^{l_1}_{k+1|k+1} = \begin{bmatrix} \Lambda_{\bar{p}} + H^T_{\bar{p}} R^{-1} H_{\bar{p}} & H^T_{\bar{p}} R^{-1} H_p \\ H^T_p R^{-1} H_{\bar{p}} & H^T_p R^{-1} H_p \end{bmatrix}$$

Taking the difference of the two shows that

$$\Delta\Lambda = \Lambda^{nfa}_{k+1|k+1} - \Lambda^{l_1}_{k+1|k+1} = \begin{bmatrix} \mathbf{0} & \Lambda_{\bar{p}p} \\ \Lambda_{p\bar{p}} & \Lambda_p \end{bmatrix} \tag{3.27}$$

The matrix in Eq. 3.27 is indefinite (unless $\Lambda_{\bar{p}p} = \Lambda_{p\bar{p}} = \mathbf{0}$, then it is positive semi-definite) and thus it can not be established whether $\Lambda^{nfa}_{k+1|k+1} \lesseqgtr \Lambda^{l_1}_{k+1|k+1}$. Moreover this implies that it can not be determined whether $P^{nfa}_{k+1|k+1} \lesseqgtr P^{l_1}_{k+1|k+1}$. □

Another interpretation of this theorem is that $\mathbf{e}^m_{k+1}$ false alarms in the error detection step cause a loss of information, in an information theoretic sense. Since Eq. 3.26 shows that $\Delta\Lambda = \Lambda^{nfa}_{k+1|k+1} - \Lambda^{l_1}_{k+1|k+1}$ is positive definite, this demonstrates that, when false alarms occur, the l_1-norm filter loses some information about the states. Specifically, the loss of information is given by $H^T_c R^{-1}_c H_c$, the state information from the measurements that the l_1-norm filter determined were incorrect.

If there are missed error detections, the following theorem demonstrates that the l_1-norm filter estimates will be biased.

Theorem 7 (Missed Detection Case). *If* $supp(\hat{\mathbf{e}}^p_k) \subset supp(\mathbf{e}^p_k)$ ***or*** $supp(\hat{\mathbf{e}}^m_{k+1}) \subset supp(\mathbf{e}^m_{k+1})$, *then* $\hat{\mathbf{x}}^{k+1|k+1}$ *will be biased. When no outliers are detected (i.e.* $supp(\hat{\mathbf{e}}^p_k) = supp(\hat{\mathbf{e}}^m_{k+1}) = \emptyset$*), the bias will be equal to* $\mathbf{b}^{KF}$, *which is the bias of the Kalman filter estimates.*

Proof. Let $\mathbf{e}_{\bar{p}}^{p}$ and $\mathbf{e}_{u}^{m}$ be the undetected state prediction and measurement outliers respectively. Then the posterior state estimation error will be

$$\tilde{\mathbf{x}}_{k+1|k+1} = \begin{bmatrix} (I - \bar{K}H_{\bar{p}u})(\tilde{\mathbf{x}}_{k+1|k}^{\bar{p}} + \mathbf{e}_{\bar{p}}^{p}) + \bar{K}(\mathbf{v}_{k+1}^{u} + \mathbf{e}_{u}^{m}) \\ -\left(H_{pu}^{T}R_{u}^{-1}H_{pu}\right)^{-1} H_{pu}^{T}R_{u}^{-1}(H_{pu}\tilde{\mathbf{x}}_{k+1|k+1}^{\bar{p}} + \mathbf{v}_{k+1}^{u} + \mathbf{e}_{u}^{m}) \end{bmatrix} \tag{3.28}$$

Taking the expected value of $\tilde{\mathbf{x}}_{k+1|k+1}$ shows that

$$E\left[\tilde{\mathbf{x}}_{k+1|k+1}\right] = \begin{bmatrix} (I - \bar{K}H_{\bar{p}u})\mathbf{e}_{\bar{p}}^{p} + \bar{K}\mathbf{e}_{u}^{m} \\ \left(H_{pu}^{T}R_{u}^{-1}H_{pu}\right)^{-1} H_{pu}^{T}R_{u}^{-1}(H_{\bar{p}u})(I - \bar{K}H_{\bar{p}u})\mathbf{e}_{\bar{p}}^{p} + (I + H_{\bar{p}u}\bar{K})\mathbf{e}_{u}^{m} \end{bmatrix} \neq \mathbf{0} \tag{3.29}$$

thus the estimates are biased. If no outliers are detected, then

$$\tilde{\mathbf{x}}_{k+1|k+1} = \tilde{\mathbf{x}}^{\bar{p}}, \quad H_{\bar{p}u} = H_{k+1} \quad \text{and} \quad \bar{K} = P_{k+1|k}H_{k+1}\left(H_{k+1}P_{k+1|k}H_{k+1}^{T} + R_{k+1}\right)^{-1} = K_{KF}$$

where K_{KF} is the Kalman gain. Substituting these values into Eq. 3.29 shows that the bias is:

$$b_{l_1} = (I - K_{KF}H_{k+1})\mathbf{e}_{k}^{p} + K_{KF}\mathbf{e}_{k+1}^{m} \tag{3.30}$$

When $\mathbf{e}_{k}^{p}$ or $\mathbf{e}_{k+1}^{m}$ are non-zero the Kalman filter residuals are

$$\tilde{\mathbf{x}}_{KF} = (I - K_{KF}H_{k+1})(\tilde{\mathbf{x}}_{k+1|k} + \mathbf{e}_{k}^{p}) + K_{KF}(\mathbf{v}_{k+1} + \mathbf{e}_{k+1}^{m}) \tag{3.31}$$

Taking the expected value of Eq. 3.31 shows that the Kalman filter bias is

$$\mathbf{b}_{KF} = (I - K_{KF}H_{k+1})\mathbf{e}_{k}^{p} + K_{KF}\mathbf{e}_{k+1}^{m} = b_{l_1} \tag{3.32}$$

This result indicates that in the worst case scenario for missed detections, the l_1-norm filter bias will be no worse than the Kalman filter bias. □

It should be noted that if missed detections occur the estimation bias can be absorbed into the $\mathbf{e}_{k}^{p}$ term and detected and corrected by the l_1-norm filter at the next measurement update. The state estimation error from Eq. 3.28 can be rewritten

as

$$\tilde{\mathbf{x}}_{k+1|k+1} = \begin{bmatrix} (I - \bar{K}H_{\bar{p}u})\tilde{\mathbf{x}}^{\bar{p}}_{k+1|k} + \bar{K}\mathbf{v}^{u}_{k+1} \\ -H^{-L}_{pu}(H_{\bar{p}u}\tilde{\mathbf{x}}^{\bar{p}}_{k+1|k+1} + \mathbf{v}^{u}_{k+1}) \end{bmatrix} + \begin{bmatrix} (I - \bar{K}H_{\bar{p}u})\mathbf{e}^{p}_{\bar{p}} + \bar{K}\mathbf{e}^{m}_{u} \\ -H^{-L}_{pu}\mathbf{e}^{m}_{u} \end{bmatrix} = \tilde{\mathbf{x}}^{nom}_{k+1|k+1} + b_{l_1} \tag{3.33}$$

At the next measurement update, the measurement residuals will be given by

$$\begin{aligned} \tilde{\mathbf{y}}_{k+2} &= H_{k+2}\left(F_{k+1}\tilde{\mathbf{x}}_{k+1|k+1} + \mathbf{w}_{k+1} + \mathbf{e}^{p}_{k+1}\right) + \mathbf{v}_{k+2} + \mathbf{e}^{m}_{k+2} \\ &= H_{k+2}\left(F_{k+1}\tilde{\mathbf{x}}^{nom}_{k+1|k+1} + \mathbf{b}_{l_1} + \mathbf{w}_{k+1} + \mathbf{e}^{p}_{k+1}\right) + \mathbf{v}_{k+2} + \mathbf{e}^{m}_{k+2} \\ &= H_{k+2}\left(F_{k+1}\tilde{\mathbf{x}}^{nom}_{k+1|k+1} + \mathbf{w}_{k+1} + \bar{\mathbf{e}}^{p}_{k+1}\right) + \mathbf{v}_{k+2} + \mathbf{e}^{m}_{k+2} \end{aligned} \tag{3.34}$$

where $\bar{\mathbf{e}}^{p}_{k+1} = \mathbf{e}^{p}_{k+1} + b_{l_1}$. Note that the residuals in Eq. 3.34 are in the same form as Eq. 3.4 and that the residual covariance will be given by $H_{k+2}P_{k+2|k+1}H^{T}_{k+2} + R_{k+2}$. Given these facts, this means that the sum of the bias and $\mathbf{e}^{p}_{k+1}$ can be detected with the constrained l_1-norm optimization in Steps 1 and 2 of Algorithm 1.

These results also indicate that it is preferable to set the χ^2 threshold τ to a smaller value (i.e. choosing a value that corresponds to a 95% confidence interval rather than a 99% confidence interval) because it will increase the probability of detecting $\mathbf{e}^{p}_{k}$ and $\mathbf{e}^{m}_{k+1}$ thus reducing the likelihood of biasing the state estimates. The only potential downside of setting the threshold this way is that more false alarms will occur and therefore the covariance may be larger than necessary (by Theorem 6). The Monte Carlo results in the next section demonstrate that even when missed detections and false alarms occur, the l_1-norm filter still provides superior estimation performance over state-of-art robust filtering algorithms.

3.2 Extended l_1-norm filter for Nonlinear Systems

The l_1-norm filter can also be extended and applied to nonlinear systems. Since a significant portion of the extended l_1-norm filter derivation mirrors the derivation of the l_1-norm filter only the major differences will be highlighted in this section.

3.2.1 System Models

As with the l_1-norm filter, it will be assumed that the state dynamics and measurements are corrupted by both additive white Gaussian noise and additive sparse errors:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k) + \mathbf{w}_k + \mathbf{e}_k^p \tag{3.35}$$

$$\mathbf{y}_{k+1} = h(\mathbf{x}_{k+1}) + \mathbf{v}_{k+1} + \mathbf{e}_{k+1}^m \tag{3.36}$$

where $f(\mathbf{x}_k)$ is the nonlinear state propagation function and $h(\mathbf{x}_k)$ is the nonlinear function mapping the states to the measurements, $\mathbf{w}_k$ and $\mathbf{v}_{k+1}$ are the process and measurement noise, respectively, and $\mathbf{e}_k^p$ and $\mathbf{e}_{k+1}^m$ represent the sparse errors.

3.2.2 Error Detection

A linear approximation of the measurement residuals can be expressed in terms of the sparse errors, $\mathbf{e}_k^p$ and $\mathbf{e}_{k+1}^m$, and the *a priori* state estimate. Using the Taylor expansions of the state propagation and measurement functions, the *a priori* measurement residuals at time $k+1$ can be approximated to first-order as:

$$\begin{aligned}
\tilde{\mathbf{y}}_{k+1} &= \mathbf{y}_{k+1} - h(f(\hat{\mathbf{x}}_{k|k})) \\
&\approx H_{k+1}\left(f(\mathbf{x}_k) + \mathbf{w}_k + \mathbf{e}_k^p - \hat{\mathbf{x}}_{k+1|k}\right) + \mathbf{v}_{k+1} + \mathbf{e}_{k+1}^m \qquad (3.37) \\
&\approx H_{k+1}\left(F_k\tilde{\mathbf{x}}_{k|k} + \mathbf{w}_k + \mathbf{e}_k^p\right) + \mathbf{v}_{k+1} + \mathbf{e}_{k+1}^m \qquad (3.38)
\end{aligned}$$

where $F_k = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}|_{\hat{\mathbf{x}}_{k|k}}$ and $H_{k+1} = \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}}|_{\hat{\mathbf{x}}_{k+1|k}}$. After rearranging terms in Eq. 3.38 and defining $\mathbf{e}_{k+1} \equiv \begin{bmatrix} \mathbf{e}_k^p \\ \mathbf{e}_{k+1}^m \end{bmatrix}$ and $\mathbf{u}_{k+1} \equiv H_{k+1}\left(F_k\tilde{\mathbf{x}}_{k|k} + \mathbf{w}_k\right) + \mathbf{v}_{k+1}$, the linearized residuals can be related to the error terms by

$$\tilde{\mathbf{y}}_{k+1} \approx \begin{bmatrix} H_{k+1} & I \end{bmatrix} \mathbf{e}_{k+1} + \mathbf{u}_{k+1} \tag{3.39}$$

Next, the optimization to detect the sparse errors can be expressed as

$$\begin{aligned} \min_{\hat{\mathbf{e}}_{k+1}} \quad & \|\hat{\mathbf{e}}_{k+1}\|_1 \\ \text{subject to} \quad & \tilde{\mathbf{y}}^T \Sigma^{-1} \tilde{\mathbf{y}} \leq \tau \end{aligned} \tag{3.40}$$

As with Eq. 3.7, the optimization in Eq. 3.40 can be recast as a second-order cone program for which a number of efficient algorithms have been developed [91, 94–96].

Although the l_1-minimization step tends to return a sparse estimate of the errors, the estimate often has small spurious non-zero components that are a result of measurement noise. In order to ensure that the error estimates are sufficiently sparse, the solution returned by the l_1-minimization is thresholded based on the expected noise level. Any elements of the l_1-optimal error estimates that are smaller than the expected noise level (as determined by a χ^2-test) are set to zero. This step ensures that only errors that are inconsistent with the Gaussian process and measurement noise are considered in the state update portion of the algorithm.

3.2.3 State and Error Estimation

In lieu of an information filter update, the extended l_1-norm filter uses an extended information filter update to estimate the sparse errors and states.

The *a priori* measurement residuals in Eq. 3.37 will be used to derive the extended information filter update for the state and error estimates. First, define the augmented state vector $\mathbf{z}_{k+1}$ as

$$\mathbf{z}_{k+1} = \begin{bmatrix} \bar{\mathbf{x}}_{k+1} \\ \mathbf{e}_k^{p,nz} \\ \mathbf{e}_{k+1}^{m,nz} \end{bmatrix} \tag{3.41}$$

where $\bar{\mathbf{x}}_{k+1} = f(\mathbf{x}_k) + \mathbf{w}_k$ and the superscript nz denotes only the non-zero components (as determined by the l_1-norm minimization) of the respective errors. After moving $H_{k+1}\hat{\mathbf{x}}_{k+1|k}$ to the left hand side of Eq. 3.37 and substituting in the definition

of $\mathbf{z}_{k+1}$,

$$\begin{aligned}\tilde{\mathbf{y}}_{k+1} + H_{k+1}\hat{\mathbf{x}}_{k+1|k} &= [H_{k+1} \;\; H_p \;\; I_m]\,\mathbf{z}_{k+1} + \mathbf{v}_{k+1} \\ &= \bar{H}_{k+1}\mathbf{z}_{k+1} + \mathbf{v}_{k+1} \end{aligned} \tag{3.42}$$

where H_p is equal to the columns of H_{k+1} corresponding to the non-zero terms in $\hat{\mathbf{e}}_k^p$ and I_m is equal to the columns of the identity matrix corresponding to non-zero entries in $\hat{\mathbf{e}}_{k+1}^m$.

The prior estimate of $\bar{\mathbf{x}}_{k+1|k}$ can be expressed as

$$\hat{\bar{\mathbf{x}}}_{k+1|k} = f\left(\hat{\mathbf{x}}_{k|k}\right)$$

and the associated covariance is $P_{\bar{\mathbf{x}}} = F_k P_{k|k} F_k^T + Q_k$. Since the prior estimates of the errors are assumed to be uninformative, the information matrix $\hat{\mathbf{z}}_{k+1|k}$ will be

$$\Lambda_{k+1|k} = \begin{bmatrix} P_{\bar{\mathbf{x}}}^{-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{3.43}$$

with the information state, $\hat{\mathbf{d}}_{k+1|k}$ given by Eq. 3.9. After calculating the information matrix and state, they can be updated as follows [86]

$$\hat{\mathbf{d}}_{k+1|k+1} = \hat{\mathbf{d}}_{k+1|k} + \bar{H}_{k+1}^T R_{k+1}^{-1} \bar{\mathbf{y}}_{k+1} \tag{3.44}$$

$$\Lambda_{k+1|k+1} = \Lambda_{k+1|k} + \bar{H}_{k+1}^T R_{k+1}^{-1} \bar{H}_{k+1} \tag{3.45}$$

where

$$\bar{\mathbf{y}}_{k+1} = \tilde{\mathbf{y}}_{k+1} + H_{k+1}\hat{\mathbf{x}}_{k+1|k} \tag{3.46}$$

After updating $\hat{\mathbf{d}}_{k+1|k+1}$ and $\Lambda_{k+1|k+1}$, the covariance $P_{k+1|k+1}^z$ and state estimate $\hat{\mathbf{z}}_{k+1|k+1}$ can be calculated from Eqs. 3.8 and 3.9, respectively.

After calculating $\hat{\mathbf{z}}_{k+1|k+1}$, the posterior state estimate, $\hat{\mathbf{x}}_{k+1|k+1}$, corrected for the

Algorithm 2 Extended l_1-norm Filter

Require: $\hat{\mathbf{x}}_{k|k}$, $P_{k|k}$, $\mathbf{y}_{k+1}$

1. Solve l_1 minimization problem in Eq. 3.40 for $\hat{\mathbf{e}}_{k+1}$
2. Apply χ^2-test to determine non-zero components of $\hat{\mathbf{e}}_{k+1}$
3. Form information state ($\hat{\mathbf{d}}_{k+1|k}$) and matrix ($\Lambda_{k+1|k}$) for augmented state vector $\hat{\mathbf{z}}_{k+1|k}$
4. Update $\hat{\mathbf{d}}_{k+1|k}$, $\Lambda_{k+1|k}$ with Eqs. 3.44–3.45
5. Calculate $P^z_{k+1|k+1}$, $\hat{\mathbf{z}}_{k+1|k+1}$ with Eqs. 3.8–3.9
6. Calculate $\hat{\mathbf{x}}_{k+1|k+1}$, $P_{k+1|k+1}$ using Eqs. 3.47–3.48

return $\hat{\mathbf{x}}_{k+1|k+1}$, $P_{k+1|k+1}$

non-Gaussian errors, is

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\bar{\mathbf{x}}}_{k+1|k+1} + \hat{\mathbf{e}}^p_{k|k+1} \tag{3.47}$$

with covariance

$$P_{k+1|k+1} = P^{\bar{\mathbf{x}}}_{k+1|k+1} + P^{\mathbf{e}^p}_{k+1|k+1} + P_{\bar{x}e} + P_{e\bar{x}} \tag{3.48}$$

where $P^{\bar{\mathbf{x}}}_{k+1|k+1}$ is the covariance of $\hat{\bar{\mathbf{x}}}_{k+1|k+1}$, $P^{\mathbf{e}^p}_{k+1|k+1}$ is the covariance of $\hat{\mathbf{e}}^p_{k|k+1}$ and $P_{\bar{x}e}$ and $P_{e\bar{x}}$ are the cross covariance matrices of $\hat{\bar{\mathbf{x}}}_{k+1|k+1}$ and $\hat{\mathbf{e}}^p_{k|k+1}$, all of which can be obtained from $P^z_{k+1|k+1}$:

$$P^z_{k+1|k+1} = \begin{bmatrix} P^{\bar{\mathbf{x}}}_{k+1|k+1} & P_{\bar{x}e} & \cdot \\ P_{e\bar{x}} & P^{\mathbf{e}^p}_{k+1|k+1} & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix}$$

3.2.4 Summary and Algorithm Description

The extended l_1-norm filter algorithm is shown in Algorithm 2.

It should also be noted that, as with the EKF, the theoretical guarantees for the extended l_1-norm filter are not as strong. For instance, it can not be guaranteed that the state estimates will be unbiased because of the impact of linearization errors. However, the vision-aided navigation experimental results in this chapter demonstrate that the extended l_1-norm filter can provide superior state estimation performance relative to other state-of-the-art robust filtering algorithms.

3.3 Monte Carlo Simulation Results

A simulated target tracking scenario is used in this section to evaluate the performance of the l_1-norm filter in the presence of unmodeled measurement and state prediction errors. Monte Carlo trials were run for three different cases: scenarios with measurement errors only, scenarios with state prediction errors and finally scenarios with simultaneous measurement and state prediction errors. Several other filtering algorithms were also evaluated in order to demonstrate the improved performance of the l_1-norm filter, especially in cases where both unmodeled state prediction and measurement errors occur.

3.3.1 Simulation Setup

The Monte Carlo trials simulate a 2D single target tracking scenario with position and velocity measurements. The estimated states were the target position and velocity. Four independent sets of position and velocity measurements of the target were simulated with an update rate of 1 Hz. The target dynamics were simulated using a constant velocity model [69] and the total length of each Monte Carlo trial was 30 seconds. The nominal process and measurement noise covariances were

$$Q_k = 0.1 \begin{bmatrix} \Delta t^4/4 & 0 & \Delta t^3/2 & 0 \\ 0 & \Delta t^4/4 & 0 & \Delta t^3/2 \\ \Delta t^3/2 & 0 & \Delta t^2 & 0 \\ 0 & \Delta t^3/2 & 0 & \Delta t^2 \end{bmatrix},$$

$$\sigma^2_{position} = 1m, \quad \sigma^2_{velocity} = 0.1m/s$$

where $\Delta t = 1s$ is the propagation time between sets of measurements.

Measurement errors were simulated by sampling the measurement noise for the position (velocity) measurements from a Gaussian distribution with a mean of 30 meters (1 meters/sec.) instead of the nominal zero mean distribution. State prediction errors were induced by sampling the process noise from a Gaussian distribution

with a larger covariance ($\bar{Q} = 10 \cdot Q_k$) than the nominal process noise model while generating the target trajectory. For the simulations with both state prediction and measurement outliers, the number of measurement outliers at any time step was chosen to ensure that the error sparsity requirements for the l_1-norm filter were met (i.e. the dimension of the non-zero state prediction and measurement errors were less than or equal to the number of measurements).

For each set of Monte Carlo trials, the performance of the l_1-norm filter (L1KF) was compared against the Kalman Filter (KF), unscented Kalman filter (UKF) [97], a robust statistics based Kalman filter (RKF) [24, 27], and a variational Bayes robust filter (VBAKF) [55]. The χ^2 threshold parameter τ for the l_1-norm filter was set to 15.5073, which corresponds to a 95% confidence interval for the χ^2 test. The robust cost function for the RKF was chosen so that it was equivalent to a KF that discards measurements with residuals that fail a χ^2-test. The threshold for the RKF χ^2-test was set to match the χ^2 thresholds used in the l_1-norm filter so that if only measurement errors are present the RKF and l_1-norm filter will identify the same set of corrupted measurements.

3.3.2 Measurement Error Only Results

The first set of simulations focused on assessing the performance of the l_1-norm filter when errors were present in the measurements only. The percentage of measurements that were corrupted with the off-nominal noise was varied from 0 to 100% in increments of 10%. For each percentage level, 100 Monte Carlo trials were performed with the corrupted measurements chosen uniformly at random.

The average position error as a function of the percentage of corrupted measurements is shown in Figure 3-1. Error bars were left off of the UKF and KF results to preserve the clarity of the plot. As the number of measurement outliers increases, the performance of the non-robust filters (KF and UKF) degrades significantly. In contrast, the robust approaches are able to maintain reasonable average positioning errors even as all of the measurements are corrupted with errors. Additionally, these plots empirically verify that the l_1-norm filter and the RKF performance are similar

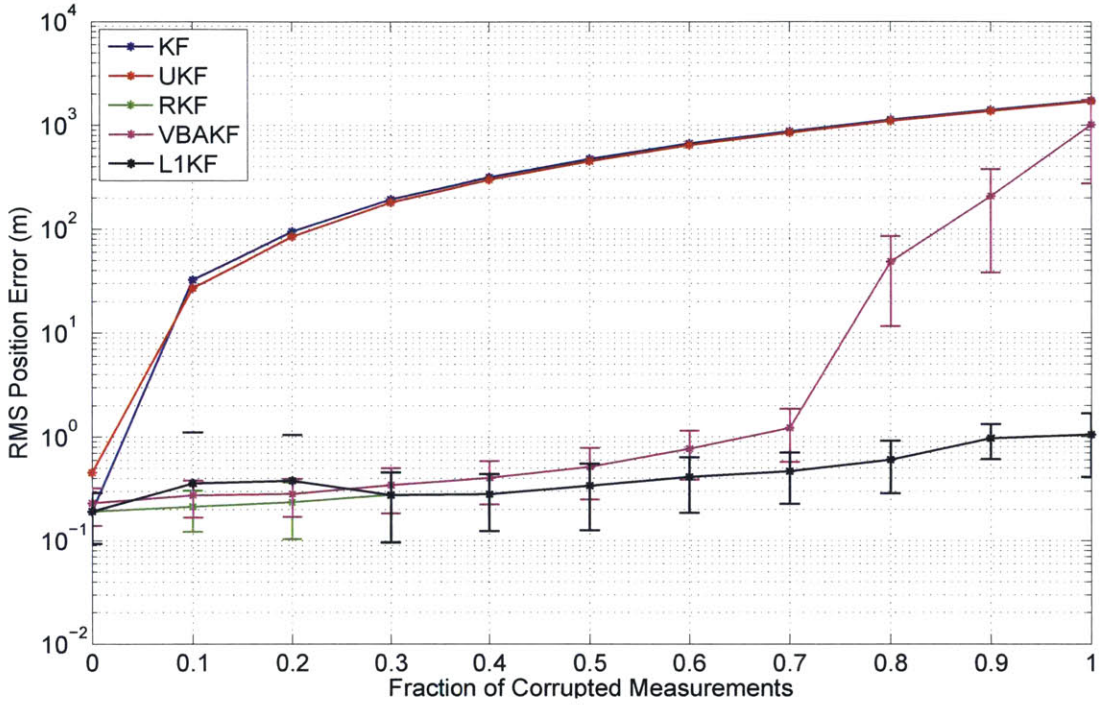


Figure 3-1: Average position error vs. fraction of corrupted measurements

when only measurement outliers are present.

Finally, the average probability of detection (p_d) and probability of false alarm (p_{fa}) for $\mathbf{e}_{k+1}^m$ were 0.9996 and 0.0 respectively. The average p_{fa} for $\mathbf{e}_k^p$ was 0.008 and the majority of the false alarms can be attributed to correcting biases introduced by missed $\mathbf{e}_{k+1}^m$ detections.

3.3.3 Process Error Only Results

The next set of simulations focused on assessing the performance of the l_1-norm filter when errors were present in the state predictions only. The percentage of state updates that were corrupted with the off nominal noise was varied from 0 to 100% in increments of 10%. For each percentage level, 100 Monte Carlo trials were performed with the corrupted state updates chosen uniformly at random.

The average position error as a function of the percentage of process errors is shown in Figure 3-2. Error bars were left off of the RKF and VBAKF results to preserve the clarity of the plot. The l_1-norm filter results and KF results are nearly

identical in this case and correspond to the line at the bottom of the plot. In contrast to the measurement error only results, the KF and UKF outperform all of the robust filters (with the exception of the l_1-norm filter) even when only a small fraction of the state updates are corrupted with additional noise. In this case, the error models for the RKF and VBAKF are not adequate to compensate for the additional noise because neither algorithm accounts for additional errors in the process model beyond the nominal process noise. The l_1-norm filter explicitly models for both process and measurement errors and thus is able to correct for the additional process noise when it is present.

For this example, p_d and p_{fa} for $\mathbf{e}_k^p$ were 0.12 and 0.0 respectively. There were no $\mathbf{e}_{k+1}^m$ false alarms. The low p_d values can in part be attributed to the distribution chosen for $\mathbf{e}_k^p$, which was zero-mean but had a larger covariance than the nominal process noise. At least some of the samples drawn from that distribution would be consistent with the nominal process noise and thus difficult to detect. These results indicate that correcting for the largest state prediction errors (i.e. the ones most likely to be detected) provides a significant performance gain. In addition, these results indicate that when there are missed detections in the l_1-norm filter they often correspond to errors that are small relative to the measurement and process noise and thus will have limited impact on the state estimates.

3.3.4 Combined Measurement and Process Error Results

The final set of simulations focused on assessing the performance of the l_1-norm filter when errors were present in both the state predictions and measurements. In this case, the percentage of state updates that were subject to the off nominal noise and the percentage of measurement errors were varied together (i.e. 10% of measurements were corrupted and 10% of state updates were corrupted for the same set of Monte Carlo trials). For each percentage level, 100 Monte Carlo trials were performed with the corrupted measurement and state updates chosen uniformly at random. The simulations were only run up to 80% error corruption because after that the error sparsity assumption could not be satisfied.

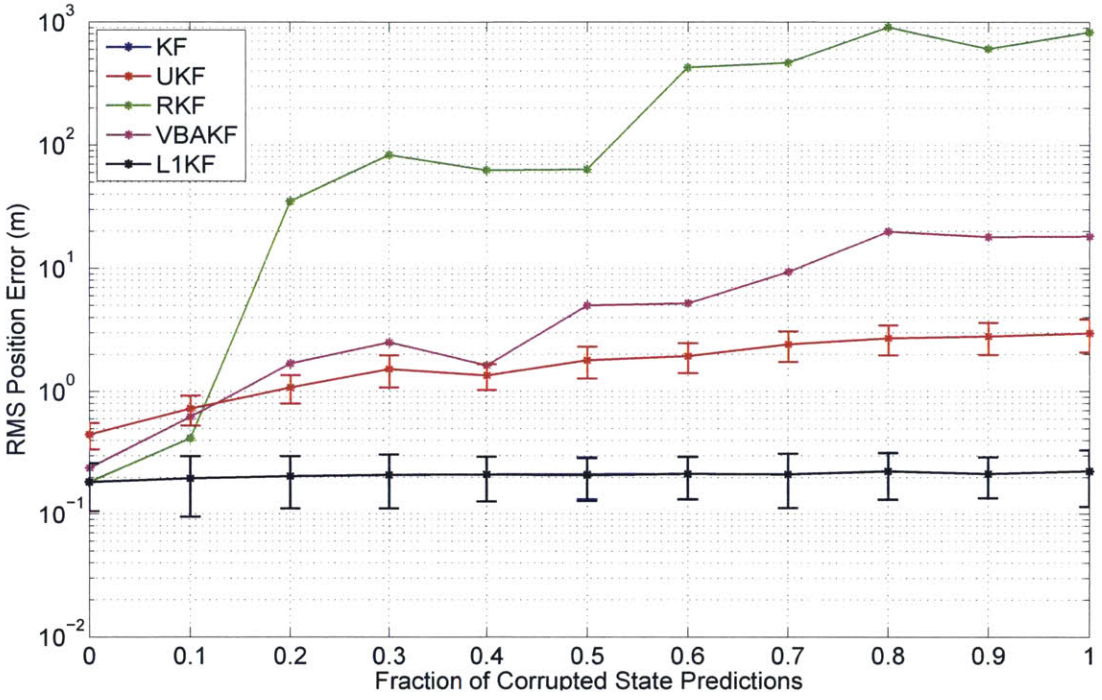


Figure 3-2: Average position error vs. fraction of process errors

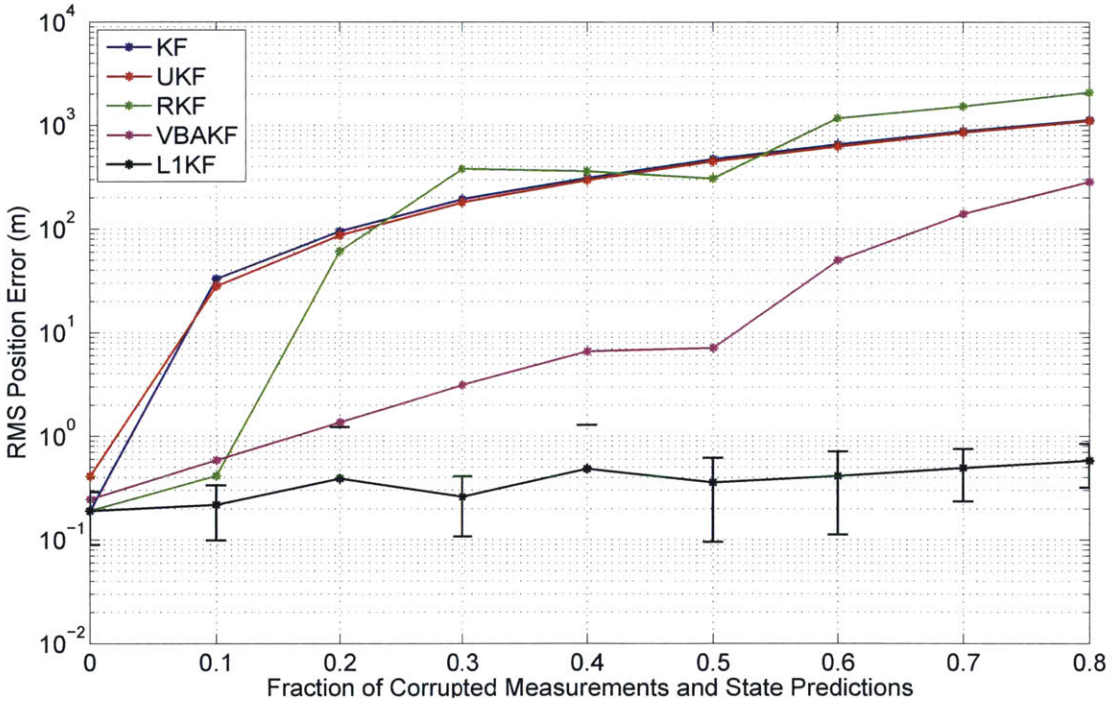


Figure 3-3: Average position error vs. fraction of process and measurement errors

The average position error as a function of the percentage of process errors is shown in Figure 3-3. This set of trials represents a worst case scenario that only the l_1-norm filter can handle. The KF and UKF estimates are not robust to the measurement errors and thus have large state estimation errors while the RKF and VBAKF can not correctly compensate for the process errors. Only the l_1-norm filter is able to correctly compensate for both the state prediction and measurement errors when they occur simultaneously and is able to maintain reasonable performance even when the majority of the state predictions and measurements are incorrect.

For this example, p_d and p_{fa} for $\mathbf{e}_{k+1}^m$ were 0.9996 and 0.0 respectively. p_d and p_{fa} for $\mathbf{e}_k^p$ were 0.15 and 0.005 respectively. As with the measurement error only case, the majority of the $\mathbf{e}_k^p$ false alarms can be attributed to correcting biases introduced by missed $\mathbf{e}_{k+1}^m$ detections.

Overall these Monte Carlo simulations show that the l_1-norm filter can provide robust state estimates over a broader range of conditions than other robust filtering algorithms. In situations where only measurement errors are present, the l_1-norm filter can match the performance of state-of-the-art robust filtering algorithms. For situations with state prediction outliers the l_1-norm filter can provide superior performance to other robust filtering approaches because it explicitly models state predictions errors while the other algorithms do not.

3.4 Vision-Aided Navigation Experimental Results

This section presents experimental results demonstrating the performance of the extended l_1-norm filter applied to vision-aided navigation in an urban area. In the data collected, GPS measurements were corrupted intermittently with multipath, while the state predictions were corrupted by drift from visual odometry measurements. Three other filtering approaches (the EKF, and two robust filtering techniques) are compared to the performance of the l_1-norm filter. This experiment demonstrates that the l_1-norm filter is able to outperform the other algorithms because it can compensate for both the GPS measurement errors and the accumulated state prediction

errors from the visual odometry measurements.

3.4.1 Vision-aided Navigation Background

Vision-aided navigation focuses on how to fuse visual information captured from a camera with other sensors to localize a vehicle in a global coordinate system.

In recent years, vision-aided navigation has been demonstrated on a number of platforms. Often these systems fuse visual odometry with other sensors (IMU, GPS, LiDAR) to generate a global state estimate. Visual odometry has several error sources that can impact the accuracy of these state estimates. Most notably, a bias is introduced by long range features in stereo visual odometry [98, 99]. Scale factor and misalignment errors in the visual odometry data can also occur and cause the navigation solution to drift over time [15].

Although recursive filtering approaches to vision-aided navigation have been developed [100], many current approaches use optimization-based pose graph estimation techniques to generate the navigation solutions [98, 101]. Recent research has shown that optimization based approaches to vision-aided navigation can outperform recursive filtering algorithms for a number of applications [102]. One reason that pose graph optimization tends to perform better than filtering is that previous poses can be updated each time the optimization is solved, thus allowing errors in previous pose estimates to be corrected, leading to a more accurate positioning solution at the current time. In contrast, filtering algorithms can not retroactively change previous state estimates in an efficient way because the estimates are marginalized out at each measurement update. Thus, any state estimation errors made earlier in the filter will propagate forward to future state estimates.

The l_1-norm filter tackles this problem by detecting situations when the current state estimate is inconsistent with the current set of measurements. After detecting these situations, the filter adjusts the state estimates to account for the impact of state estimation error that has been propagated to the current time step. In this way, the l_1-norm filter can adjust its state estimates when drift errors accumulate without having to resolve for any of its previous estimates.

3.4.2 Vision-aided Navigation with the l_1-norm Filter

There were several challenges associated with using both vision and GPS measurements in the l_1-norm filter. First, the GPS and visual odometry data were not being generated at the same rate (10 Hz for the vision vs. 1 Hz for the GPS). In practice, large errors in the visual odometry are not observable unless there is additional information from another measurement such as GPS. Thus, the majority of the visual odometry measurements could not be checked for errors directly by the l_1-norm filter. Additionally, it was found that the errors in the visual odometry data were often below the detection threshold of the l_1-norm filter for any given measurement even when GPS measurements were available. Fortunately, it was determined that the cumulative effects of the visual odometry errors (over several sets of measurements) were large enough and could be detected by the l_1-norm filter as state propagation errors, $\mathbf{e}_k^p$, when GPS measurements were available.

3.4.3 Experimental Setup

The data used for this experiment was collected while driving along roads in the Boston area. Environments driven through varied between dense urban canyons and areas of good GPS coverage along the Charles River. The total time for the experiment took approximately 25 minutes from start to finish and the total distance covered was 7.32 km. Vehicle speeds varied between 0 and 72 km/h. The estimated states were the car's position and velocity in Earth-Centered Earth-Fixed coordinates.

The sensors used for the experiment were a dashboard-mounted stereo vision camera (Point Grey BumbleBee2 with a resolution of 512 x 384 and 43° field of view) and a consumer grade GPS receiver (uBlox EVK-6T). Visual odometry measurements (measuring the change in position of the car between camera frames) and GPS pseudoranges were processed in the navigation filter. Visual odometry measurements were provided at 10 Hz while the GPS receiver reported pseudoranges at 1 Hz when they were available. More details about the system used for the experimental data collection can be found in [103]. A high accuracy GPS positioning solution that was

generated by the receiver was used as ground-truth for the experiment.

The pseudoranges and state predictions were compared against the truth data to verify that the error sparsity assumptions were satisfied. These comparisons indicate that, during the experiment, at most 2 pseudorange measurements were corrupted with multipath at each time step and that, when multipath errors occurred, there were at least 6 total pseudorange measurements available. When large state prediction errors occurred (i.e., $\mathbf{e}^p \neq \mathbf{0}$), at least 6 pseudorange measurements were available. Additionally, the results showed that simultaneous state prediction and multipath errors never occurred. Therefore, during the experiment, the error sparsity requirements of the l_1-norm filter were satisfied because the number of available measurements was always larger than the number of non-zero entries of the sparse errors.

3.4.4 Experimental Results

The experimental data was processed using the l_1-norm filter as well as three other algorithms to compare the performance of each in an urban navigation scenario. The visual odometry measurements were modeled using the stochastic cloning technique proposed by Roumeliotis et al. [104]. The vehicle dynamics were modeled using a constant velocity model [69].

The three other algorithms were an EKF, a VB robust filter called the outlier robust Kalman filter (ORKF) [56, 57], and an EKF that uses robust statistics to reduce the impact of measurement outliers. In the experimental results, the last filter will be referred to as the robust Kalman filter (RKF) and is similar to algorithms presented by Masreliez and Martin [24] and Schick and Mitter [27]. Since the EKF is not a robust estimator and the experimental dataset contains both GPS multipath errors and visual odometry drift and bias errors, the robust filtering algorithms should produce better results than the EKF.

A comparison of the positioning error of the navigation solutions for the four algorithms is shown in Figure 3-4. The RKF solution has a number of instances where the positioning error exceeds all of the other algorithms by a significant amount.

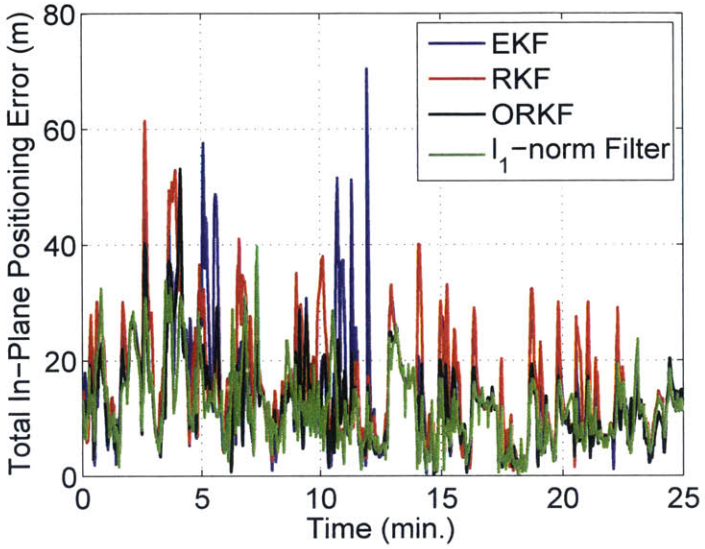


Figure 3-4: Positioning error vs. time for each of the filtering algorithms

These large errors are primarily caused by the fact that the RKF can not distinguish between *a priori* state estimation errors (in this case caused by errors accumulated from the visual odometry measurements) and GPS measurement errors. For instance, the large deviation from truth shown in Figure 3-5 is the result of accumulated visual odometry errors that occurred, when the vehicle turned at the intersection. In this case, the turning motion of the car induced errors in the visual odometry solution because most of the features that were being tracked left the field of view of the camera. The GPS measurement residuals became large and as a result, the RKF significantly downweighted GPS measurements that could have been used to correct for the accumulated visual odometry errors in the state estimates. In the case shown in Figure 3-5, the ORKF also takes more time than the EKF and l_1-norm filter to recover from the visual odometry errors because it can not differentiate between the state propagation errors and GPS measurement errors and also ends up downweighting GPS measurements that could help the filter converge to the correct solution.

In contrast, the l_1-norm filter was able to determine that the large measurement residuals were the results of *a priori* state errors instead of GPS measurement errors and as a result, was able to use the GPS measurements to recover from the visual

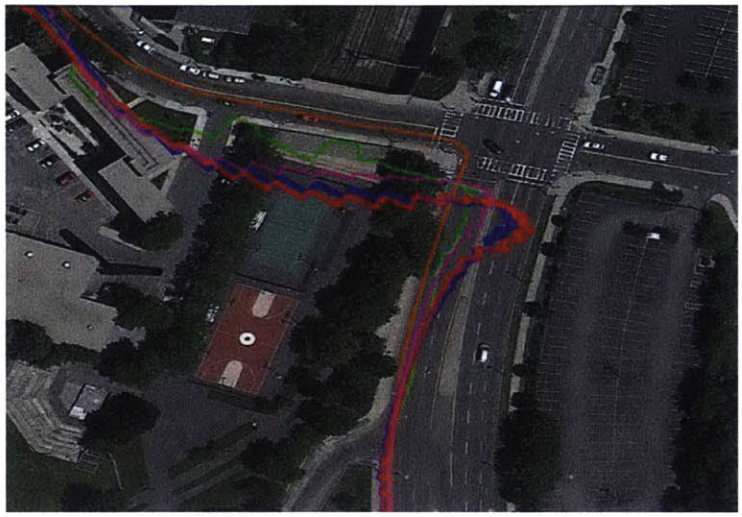


Figure 3-5: Impact of visual odometry errors on the RKF (shown in red) and ORKF (shown in magenta) solutions. In this case, the turning motion of the car at the intersection induced errors in the visual odometry solution because most of the features that were being tracked left the field of view of the camera. The EKF and l_1-norm filter solutions are shown in blue and green respectively. GPS truth in this figure is shown in orange.

Table 3.1: Comparison of Positioning Error Results

Algorithm	Mean Error (m)	σ Error (m)	Max Error (m)	Error Relative to EKF (m)
EKF	15.10	10.16	70.45	0.0
RKF [24, 27]	15.53	9.66	61.47	0.43
ORKF [56, 57]	12.38	7.51	53.10	-2.72
l_1-norm Filter	**12.00**	**6.87**	**39.76**	**-3.10**

odometry errors. The EKF was not significantly affected in these situations because even though visual odometry errors have accumulated in the state estimates, processing the GPS measurements quickly corrects for the impact of the error because the measurement residuals are so large.

Upon examining the EKF results, there are several large positioning errors around 11 minutes into the experiment. These are the result of multipath errors in the GPS pseudorange measurements caused by a large building (see Figure 3-6 for a more detailed view). In this case, all of the robust filters were able to detect and eliminate the impact of the multipath on the navigation solution as expected.

Summary statistics for all of the algorithms are shown in Table 3.1. Based on this dataset, the l_1-norm filter is able to provide the best solution out of the four

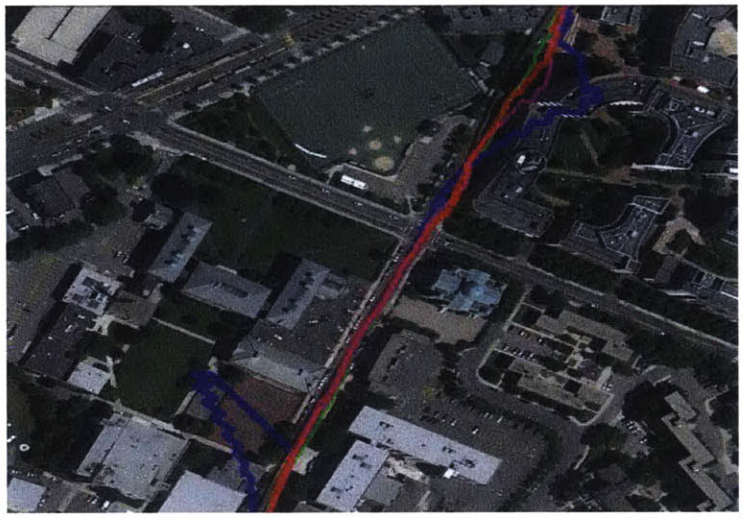


Figure 3-6: Impact of multipath on the EKF navigation solution (shown in blue). The multipath errors are caused by reflections off of the tall buildings near the road. The ORKF (shown in magenta), RKF (shown in red) and l_1-norm filter (shown in green) were able to detect and compensate for the impact of the multipath in this case.

algorithms. It is able to provide accurate state estimates when the GPS measurements are corrupted with multipath and avoids incorrectly ignoring GPS as the ORKF and RKF do when significant visual odometry errors accumulate. Additionally, the l_1-norm filter has the ability to perform state estimation reliably when both of these situations occur simultaneously, which none of the other algorithms can guarantee.

The absolute position errors shown in Table 3.1 are larger than one might expect from a navigation solution based in part on GPS data. In this experiment, additional corrections for errors in the pseudoranges due to ionospheric effects (i.e. corrections generated by the Wide-Area Augmentation System (WAAS)) were unavailable and as a result the pseudoranges were biased. Although errors due to ionospheric delays could have been corrected using WAAS data, localized errors in the pseudoranges such as GPS multipath could not have been compensated for and would still have been present. Additionally, the WAAS corrections would not have had an impact on the visual odometry errors that occurred. Thus, while using WAAS corrections would have reduced the absolute error for all of the algorithms, the reductions in error relative to the EKF (the final column of Table 3.1) would still have occurred because

they are related to the compensation of multipath and visual odometry errors that the WAAS corrections could not fix.

3.5 Summary

This chapter presented a recursive state estimation algorithm, the l_1-norm filter, that is robust to both unmodeled state prediction and measurement errors. The l_1-norm filter detects the presence of unmodeled errors using a convex optimization. Given that information, the filter can then adjust the *a priori* state estimates and measurements accordingly to compensate for the errors. The algorithm is also computationally efficient as it combines a convex optimization with standard recursive filtering steps.

A simulated target tracking scenario was used to evaluate the performance of the l_1-norm filter and compare it to existing state of the art robust state estimation algorithms. The l_1-norm filter was also evaluated on a dataset consisting of visual odometry and GPS data collected in urban areas around Boston. In both cases, the l_1-norm filter was able to outperform state-of-the-art robust state estimation algorithms, because it could compensate for both state prediction and measurement outliers that occurred in the data.

Chapter 4

Robust SLAM via Information Matrix Estimation

One of the major challenges associated with SLAM systems arises because the Gaussian assumption is often a poor approximation of the actual measurement noise. For instance, loop closure measurements can be generated incorrectly because of visual aliasing which can lead to measurement residuals that are much larger than the expected Gaussian noise. Additionally, the measurement noise statistics could be a function of time or location of the robot (i.e. due to multipath effects for sonar or radar) which is difficult to capture with a single Gaussian model. If the measurement noise is assumed to be a fixed Gaussian and these types of errors occur, the SLAM solution can dramatically degrade [20, 61]. In the case of incorrect loop closures, even a single instance can lead to divergence in the estimates of the robot poses [61]. A comparison of the impact of incorrect loop closures on a SLAM algorithm and the algorithm proposed in this chapter, Information Matrix SLAM (IM-SLAM), is shown in Figure 4-1.

The major contribution of this chapter is a SLAM algorithm that retains the efficiency of nonlinear least-squares SLAM algorithms while compensating for incorrect loop closure measurements. An additional set of variables, corresponding to the information matrix (equivalent to the inverse of the measurement noise covariance matrix) of the loop closure measurements, are introduced to the SLAM inference problem and

(a) SLAM (b) IM-SLAM

Figure 4-1: Comparison solutions for the Sphere2500 dataset [105] with 500 incorrect loop closures. Non-robust SLAM algorithms (left) are unable to reject the false loop closures and the map quality suffers as a result. The method proposed in this chapter, IM-SLAM, can accurately reject false loop closures and produces the correct solution (right). Truth is shown in gray in both plots.

estimated along with the robot poses. An expectation-maximization (EM) procedure is used to iteratively estimate the robot poses and the information matrices. By estimating and adapting the information matrices, the impact of incorrect measurements on the pose estimates is significantly reduced. Additionally, the EM procedure only involves a closed form update rule for the information matrices and a standard nonlinear least-squares update step for the pose estimates. In practice, this leads to a computationally efficient SLAM algorithm that is significantly more robust to measurement errors than traditional SLAM techniques.

4.1 IM-SLAM

A major component of IM-SLAM is an augmented SLAM factor graph (see Figure 4-2) with an additional set of latent variables (Λ). The extra latent variables, Λ_j, correspond to the information matrices (or equivalently the inverse noise covariance matrices) of each of the loop closure measurements in the factor graph. By inferring these variables, incorrect loop closure measurements can be accounted for in the SLAM solution by increasing their corresponding covariances. In effect, by adapting the covariances the weight of those measurements will be reduced in the SLAM cost

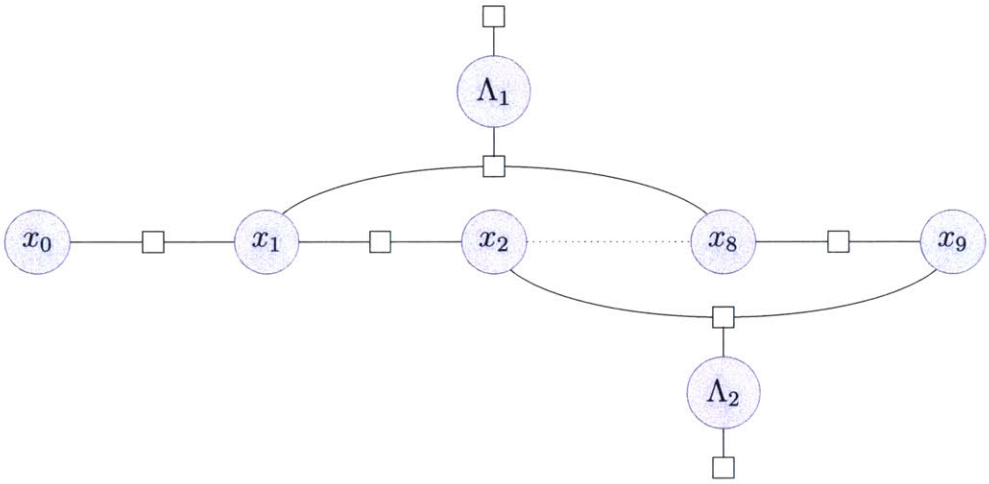


Figure 4-2: Augmented SLAM factor graph model used for IM-SLAM

function.

As with the SLAM problem, it will be assumed that the conditional probability of a loop closure measurement given the poses ($\mathbf{x}$) and its information matrix ($\mathbf{\Lambda}_j$) is Gaussian:

$$p(\mathbf{y}_j^l|\mathbf{x}, \mathbf{\Lambda}_j) \propto \frac{1}{\left|\mathbf{\Lambda}_j^{-1}\right|^{1/2}} e^{-\frac{1}{2}(\mathbf{y}_j^l - h_j(\mathbf{x}))^T \mathbf{\Lambda}_j (\mathbf{y}_j^l - h_j(\mathbf{x}))} \tag{4.1}$$

It will also be assumed that each loop closure measurement has a nominal covariance matrix, Σ_j^{nom} that is known *a priori*. Given these assumptions, the posterior distribution of the poses can be solved for by marginalizing over the joint posterior of the loop closure information matrices and the poses

$$p(\mathbf{x}|\mathbf{y}) \propto \prod_{i=1}^{n_o} p(\mathbf{y}_i^o|\mathbf{x}) \int p(\mathbf{y}^l|\mathbf{x}, \mathbf{\Lambda}) p(\mathbf{\Lambda}) d\mathbf{\Lambda} \tag{4.2}$$

The objective of the modified SLAM problem is to solve for the ML estimate of $\mathbf{x}$ given Eq. 4.2:

$$\hat{\mathbf{x}}^{SLAM} = \underset{\mathbf{x}}{\text{argmax}} \int_{\Lambda} \prod_{i=1}^{n_o} p(\mathbf{y}_i^o|\mathbf{x}) p(\mathbf{y}^l|\mathbf{x}, \mathbf{\Lambda}) p(\mathbf{\Lambda}) d\mathbf{\Lambda} \tag{4.3}$$

Unfortunately, this calculation is computationally intractable in general, so an alternative approximate approach must be used to ensure that an efficient estimate for

$\hat{\mathbf{x}}^{SLAM}$ can be found.

One method for approximately solving this type of estimation problem is to apply the classification expectation maximization (EM) algorithm [51, 106, 107] to iteratively solve for the best estimates of both Λ and $\mathbf{x}$. Each iteration of the algorithm involves updating the current estimates of the poses and the information matrix parameters ($\hat{\mathbf{x}}^k$ and $\hat{\Lambda}^k$) as follows:

$$\begin{aligned}\hat{\Lambda}^{k+1} &= \underset{\Lambda}{\operatorname{argmax}}\, p(\Lambda|\hat{\mathbf{x}}^k, \mathbf{y}^o, \mathbf{y}^l) \\ &= \underset{\Lambda}{\operatorname{argmax}}\, p(\Lambda|\hat{\mathbf{x}}^k, \mathbf{y}^l) \qquad (4.4)\end{aligned}$$

$$\hat{\mathbf{x}}^{k+1} = \underset{\mathbf{x}}{\operatorname{argmax}} \log p(\mathbf{x}|\hat{\Lambda}^{k+1}, \mathbf{y}^o, \mathbf{y}^l) \qquad (4.5)$$

Equations 4.4 and 4.5 are referred to as the E-step and the M-step, respectively. These two update steps are repeated until the pose estimates have converged. This optimization can also be interpreted as a form of block-wise coordinate descent.

4.1.1 E-Step Derivation

In order to derive the E-Step, the cost function in Eq. 4.4 must be investigated in more detail. The probability distribution in the E-step can be rewritten as

$$p(\Lambda|\hat{\mathbf{x}}^k, \mathbf{y}^l) = \frac{p(\mathbf{y}^l|\Lambda, \hat{\mathbf{x}}^k)p(\Lambda)}{p(\mathbf{y}^l|\hat{\mathbf{x}}^k)} \qquad (4.6)$$

Then the optimization in the E-step can be expressed as

$$\begin{aligned}\max_{\Lambda} p(\Lambda|\hat{\mathbf{x}}^k, \mathbf{y}^l) &= \max_{\Lambda} \frac{p(\mathbf{y}^l|\Lambda, \hat{\mathbf{x}}^k)p(\Lambda)}{p(\mathbf{y}^l|\hat{\mathbf{x}}^k)} \\ &= \max_{\Lambda} \log p(\mathbf{y}^l|\Lambda, \hat{\mathbf{x}}^k) + \log p(\Lambda) \\ &= \max_{\Lambda} \log \prod_{j \in S} p(\mathbf{y}_j^l|\hat{\mathbf{x}}^k, \Lambda_j) + \log p(\Lambda) \\ &= \max_{\Lambda} \sum_{j \in S} \log p(\mathbf{y}_j^l|\hat{\mathbf{x}}^k, \Lambda_j) + \log p(\Lambda)\end{aligned}$$

$$= \max_{\Lambda} \frac{1}{2} \sum_{j \in S} -\mathbf{r}_j^T \Lambda_j \mathbf{r}_j - \frac{1}{2} \log \left| \Lambda_j^{-1} \right| + \log p(\Lambda)$$
$$= \min_{\Lambda} \frac{1}{2} \sum_{j \in S} \mathbf{r}_j^T \Lambda_j \mathbf{r}_j + \frac{1}{2} \log \left| \Lambda_j^{-1} \right| - \log p(\Lambda)$$

where $\mathbf{r}_j = \mathbf{y}_j^l - h_j(\hat{\mathbf{x}}^k)$ are the residuals for the jth loop closure measurement.

At this point, the prior distribution for the information matrix parameters, $p(\Lambda)$, still needs to be selected in order to have an explicit update formula. For IM-SLAM, the prior was chosen such that each Λ_i is independent of all other Λ_j, when $i \neq j$, and that each information matrix is Wishart distributed:

$$p(\Lambda_j) \propto |\Lambda_j|^{\frac{\nu - n_y - 1}{2}} e^{-\frac{1}{2} tr\left(V^{-1} \Lambda_j\right)} \tag{4.7}$$

where n_y is the dimension of Λ_j, and ν and V are the parameters of the Wishart prior specified so that $\nu > n_y$ and V is a $n_y \times n_y$ symmetric positive definite matrix. The notation $tr(\cdot)$ denotes the matrix trace. The Wishart prior is often used in information matrix estimation problems because it is the conjugate prior for a Gaussian distribution with an unknown information matrix and with the proper choice of hyperparameters is the least informative prior as well [108]. Additionally, using the Wishart prior produces an E-step cost function that is convex with respect to Λ.

After substituting Eq. 4.7 into Eq. 4.6 the E-step optimization can be expressed as

$$\max_{\Lambda} p(\Lambda | \hat{\mathbf{x}}^k, \mathbf{y}) = \min_{\Lambda} \frac{1}{2} \sum_{j=1}^{n_l} \mathbf{r}_j^T \Lambda_j \mathbf{r}_j - (\nu - n_y) \log(|\Lambda_j|) + tr\left(V^{-1} \Lambda_j\right)$$

Since the optimization problem is a sum of independent terms, each Λ_j can be solved for individually. Moreover, the optimization function is convex in each Λ_j so the optimal solution can be found by solving for the values of Λ_j such that

$$\frac{d}{d\Lambda_j} \left(\mathbf{r}_j^T \Lambda_j \mathbf{r}_j - (\nu - n_y) \log(|\Lambda_j|) + tr\left(V^{-1} \Lambda_j\right) \right) = 0 \tag{4.8}$$

Note that the derivatives of the terms in Eq. 4.8 can be calculated as:

$$\frac{\partial \mathbf{r}_j^T \Lambda_j \mathbf{r}_j}{\partial \Lambda_j} = \mathbf{r}_j \mathbf{r}_j^T$$
$$\frac{\partial \log(|\Lambda_j|)}{\partial \Lambda_j} = \Lambda_j^{-1}$$
$$\frac{\partial tr\left(V^{-1}\Lambda_j\right)}{\partial \Lambda_j} = V^{-1}$$

After substituting in the derivative values and solving Eq. 4.8 the information matrix update formula for the E-step is

$$\Lambda_j^{k+1} = (\nu - n_y)\left(V^{-1} + \mathbf{r}_j \mathbf{r}_j^T\right)^{-1} \tag{4.9}$$

This update rule is intuitive in the sense that as the residuals become larger for a given loop closure measurement the corresponding information matrix terms will decrease meaning that the loop closure will have less weight in the SLAM solution. Moreover if the modeled noise covariance was overly optimistic (i.e. smaller than the true covariance) for a given measurement, the residuals should reflect that and as a result cause an adjustment to the information matrix. Thus, by adapting the information matrix parameters using the update rule in Eq. 4.9 both incorrect loop closures (which should have large residuals associated with them) as well as measurements with incorrect noise covariances can be accounted for. Moreover, the following theorem proves that as $\|\mathbf{r}_j\| \to \infty$ the contribution of the measurement $\mathbf{y}_j$ to the SLAM cost function ($\mathbf{r}_j^T \Lambda_j^{k+1} \mathbf{r}_j$) will converge to a constant and thus the measurement will have a small effect on the solution.

Theorem 8. *In the limit as* $\|\mathbf{r}_j\| \to \infty$, $\mathbf{r}_j^T \Lambda_j^{k+1} \mathbf{r}_j$ *converges to a constant.*

Proof. Without loss of generality, let $\mathbf{r}_j = \alpha \mathbf{d}$ where $\alpha > 0$ and $\|d\| = 1$. Then

$$\begin{aligned}\Lambda_j^{k+1} &= (\nu - n_y)\left(V^{-1} + \alpha^2 \mathbf{d}\mathbf{d}^T\right)^{-1} \\ &= V - V\mathbf{d}\left(\frac{1}{\alpha^2} + \mathbf{d}^T V \mathbf{d}\right)^{-1} \mathbf{d}^T V\end{aligned} \tag{4.10}$$

where the second equality follows from the matrix inversion lemma.

Using Eq. 4.10, the product of Λ_j^{k+1} and $\mathbf{r}_j$ is given by:

$$\Lambda_j^{k+1}\mathbf{r}_j = (\nu - n_y)\,\alpha V\mathbf{d} - \alpha V\mathbf{d}(\frac{1}{\alpha^2} + \mathbf{d}^T V\mathbf{d})^{-1}\mathbf{d}^T V\mathbf{d}$$

Note that taking the limit as $\|\mathbf{r}_j\| \to \infty$ is equivalent to taking the limit as $\alpha \to \infty$ thus

$$\begin{aligned}
\lim_{\alpha\to\infty} \mathbf{r}_j^T \Lambda_j^{k+1}\mathbf{r}_j &= \lim_{\alpha\to\infty} (\nu - n_y)\left(\alpha^2\mathbf{d}^T V\mathbf{d} - \alpha^2\mathbf{d}^T V\mathbf{d}(\frac{1}{\alpha^2} + \mathbf{d}^T V\mathbf{d})^{-1}\mathbf{d}^T V\mathbf{d}\right) \\
&= \lim_{\alpha\to\infty} (\nu - n_y)\left(\alpha^2\mathbf{d}^T V\mathbf{d}\left(1 - \frac{\mathbf{d}^T V\mathbf{d}}{\frac{1}{\alpha^2} + \mathbf{d}^T V\mathbf{d}}\right)\right) \\
&= \lim_{\alpha\to\infty} (\nu - n_y)\left(\alpha^2\mathbf{d}^T V\mathbf{d}\left(\frac{\frac{1}{\alpha^2}}{\frac{1}{\alpha^2} + \mathbf{d}^T V\mathbf{d}}\right)\right) \\
&= \lim_{\alpha\to\infty} (\nu - n_y)\left(\mathbf{d}^T V\mathbf{d}\left(\frac{\alpha^2}{1 + \alpha^2\mathbf{d}^T V\mathbf{d}}\right)\right) \\
&= (\nu - n_y)\,\mathbf{d}^T V\mathbf{d}
\end{aligned}$$

which is a constant and concludes the proof. □

This result demonstrates that IM-SLAM improves the breakdown point of the SLAM optimization. Recall that the breakdown point of an estimator is the fraction of measurements that can be arbitrarily large while the estimates remain bounded. In addition, recall that least-squares estimators and by extension SLAM optimizers have a breakdown point of 0, meaning that even a single unbounded measurement can cause the estimator to return an arbitrarily large estimate. Theorem 8 shows that IM-SLAM has a breakdown point higher than zero because for an unbounded measurement, the cost and thus the estimate will remain finite. This also indicates that the E-step in IM-SLAM could alternatively be interpreted as a robust kernel function similar to the Huber and Cauchy kernels in robust statistics. Plotting the attenuation factor for IM-SLAM (shown in Figure 4-3) shows that IM-SLAM is similar to the Cauchy and Huber kernels because it reduces the weights of measurements relative to the least squares weights as the residuals increase. Next, we will see that

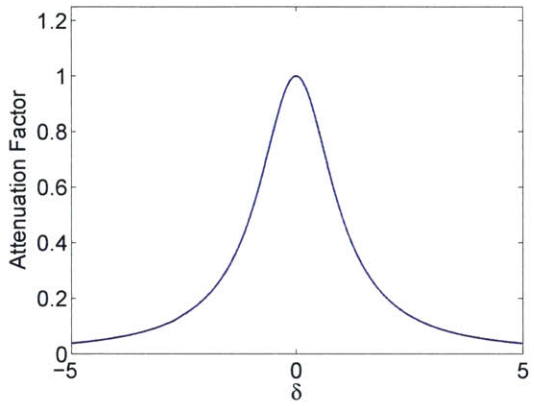


Figure 4-3: Attenuation factor for IM-SLAM. This plot demonstrates that as the residual errors (δ) increase, the ratio of the IM-SLAM cost to the least-squares cost decreases significantly. Thus, if the measurement is corrupted with outliers, the IM-SLAM kernel reduces the impact of the outliers on the solution.

an advantage IM-SLAM has over those kernel functions is the intuitive way that the hyperparameters ν and V can be selected.

The update rule for Λ_j^{k+1} also gives some insight into how to choose the parameters ν and V of the Wishart distribution. Typically, each measurement should have a nominal covariance, $\Sigma_{nom,j}$, associated with it that represents the best *a priori* estimate of the noise covariance. If $\Sigma_{nom,j}$ is chosen optimistically (i.e. it represents the best case noise covariance) then reasonable choices for the hyperparameters are, $\nu = n_y + 1$, and $V = \Sigma_{nom,j}^{-1}$. Given these choices for the parameters, the largest value that Λ_j^{k+1} can take is $\Sigma_{nom,j}^{-1}$, the best case information matrix, while as the residuals grow larger, Λ_j^{k+1} will decrease as desired.

4.1.2 M-Step Derivation

For the M-step the development of the update formula is straight-forward. In this case, the solution of the optimization problem in Eq. 4.5 is equivalent to a nonlinear least-squares problem with the weighting matrix given by the block diagonal matrix $\hat{W}^{k+1} = diag(\mathbf{\Lambda}^o, \hat{\Lambda}^{k+1})$ where $\mathbf{\Lambda}^o$ is a block diagonal matrix composed of the odometry measurement information matrices and $\hat{\Lambda}^{k+1}$ is a block diagonal matrix composed of the estimate loop closure information matrices, Λ_j^{k+1}, from the E-step. Thus, the

M-step can be carried out by solving the SLAM optimization problem with the loop closure information matrices given by $\hat{\Lambda}^{k+1}$.

In practice, solving the nonlinear least-squares problem at each iteration of the EM algorithm significantly increases the computation time and can cause the EM algorithm to get stuck in poor local optima of Eq. 4.3. A better approach is to apply a single nonlinear least-squares update of the poses at each iteration. Thus, the M-step can be carried out by applying the nonlinear least-squares update in Eq. 2.34 to the poses with $\hat{W}^{k+1}$ substituted for W:

$$\hat{\mathbf{x}}^{k+1} = \hat{\mathbf{x}}^{k} - (J^T \hat{W}^{k+1} J)^{-1} J^T \hat{W}^{k+1} \mathbf{r} \tag{4.11}$$

This type of incremental update is commonly used in implementations of the EM algorithm to avoid convergence to poor local optima and to improve runtime efficiency [109].

4.1.3 Algorithm Summary

In summary, the IM-SLAM starts with an initial guess for the poses, $\hat{\mathbf{x}}^0$, and then proceeds to iteratively apply updates to the poses and the information matrices using Eq. 4.9 and Eq. 4.11 until the pose estimates converge. Since the information matrix and pose updates correspond to the EM algorithm, this procedure is guaranteed to converge to a local optimum of the cost function in Eq. 4.3 [106]. The next section will show that in practice this algorithm provides good performance even with large numbers of incorrect loop closure measurements. Moreover, because the information matrix updates can be calculated in closed form, the difference in computation time between the robust SLAM algorithm and a traditional SLAM algorithm is relatively small. The full algorithm description is shown in Algorithm 3.

4.1.4 Convergence Analysis

For iterative algorithms such as IM-SLAM, one major question is whether the iterations will converge to a fixed point. The following theorem proves that indeed the

Algorithm 3 IM-SLAM

Require: Initial pose estimate $\hat{\mathbf{x}}^0$, measurements $\mathbf{y}$
  $k = 0$
  while not converged **do**
    // Expectation Step
    for $j = 1$ to n_l **do**
      Compute Λ_j^{k+1} using Eq. 4.9
    end for
    // Maximization Step
    Compute $\hat{\mathbf{x}}^{k+1}$ using Eq. 4.11
    $k = k + 1$
  end while
  return $\hat{\mathbf{x}}^{SLAM} = \hat{\mathbf{x}}^{k+1}$

IM-SLAM iterations will converge.

Theorem 9. *Assume that* $\mathbf{x} \in X$ *and* $\mathbf{\Lambda} \in S$ *where* X *and* S *are compact sets. Then the following statements are true*

1. $\log p(\mathbf{x}^{k+1}, \Lambda^{k+1}|\mathbf{y})$ *is monotonically non-decreasing with* k

2. $\log p(\mathbf{x}^{k+1}, \Lambda^{k+1}|\mathbf{y})$ *converges to a local maximum as* $k \to \infty$

Proof. To begin, note that

$$\log p(\mathbf{x}, \Lambda|\mathbf{y}) = \log p(\mathbf{x}|\Lambda, \mathbf{y}) + \log p(\Lambda|\mathbf{y}) \tag{4.12}$$

$$= \log p(\Lambda|\mathbf{x}, \mathbf{y}) + \log p(\mathbf{x}|\mathbf{y}) \tag{4.13}$$

The E-step in Algorithm 3 solves the optimization

$$\max_{\Lambda \in S} \log p(\Lambda|\hat{\mathbf{x}}^k, \mathbf{y})$$

and therefore

$$\log p(\Lambda^{k+1}|\mathbf{x}^k, \mathbf{y}) \geq \log p(\Lambda^k|\mathbf{x}^k, \mathbf{y})$$

Moreover, using Equation 4.13 shows that

$$\log p(\mathbf{x}^k, \Lambda^{k+1}|\mathbf{y}) = \log p(\Lambda^{k+1}|\mathbf{x}^k, \mathbf{y}) + \log p(\mathbf{x}^k|\mathbf{y})$$

$$\geq \log p(\Lambda^k|\mathbf{x}^k, \mathbf{y}) + \log p(\mathbf{x}^k|\mathbf{y}) = \log p(\mathbf{x}^k, \Lambda^k|\mathbf{y})$$

The M-step in Algorithm 3 solves the optimization

$$\max_{\mathbf{x}\in X} \log p(\mathbf{x}|\Lambda^{k+1}, \mathbf{y})$$

and therefore

$$\log p(\mathbf{x}^{k+1}|\Lambda^{k+1}, \mathbf{y}) \geq \log p(\mathbf{x}^k|\Lambda^{k+1}, \mathbf{y})$$

Moreover, using Equation 4.12 shows that

$$\begin{aligned}
\log p(\mathbf{x}^{k+1}, \Lambda^{k+1}|\mathbf{y}) &= \log p(\mathbf{x}^{k+1}|\Lambda^{k+1}, \mathbf{y}) + \log p(\Lambda^{k+1}|\mathbf{y}) \\
&\geq \log p(\mathbf{x}^k|\Lambda^{k+1}, \mathbf{y}) + \log p(\Lambda^{k+1}|\mathbf{y}) \\
&= \log p(\mathbf{x}^k, \Lambda^{k+1}|\mathbf{y}) \geq \log p(\mathbf{x}^k, \Lambda^k|\mathbf{y})
\end{aligned}$$

Therefore $\log p(\mathbf{x}^{k+1}, \Lambda^{k+1}|\mathbf{y}) \geq \log p(\mathbf{x}^k, \Lambda^k|\mathbf{y})$ which implies that $\log p(\mathbf{x}^{k+1}, \Lambda^{k+1}|\mathbf{y})$ is monotonically non-decreasing with k.

Because both $\mathbf{x}$ and Λ belong to compact sets and the cost function is non-decreasing with increasing k, this implies that $\log p(\mathbf{x}^{k+1}, \Lambda^{k+1}|\mathbf{y})$ converges to a local maximum as $k \to \infty$. □

4.1.5 Practical Implementation Details

While IM-SLAM is guaranteed to converge, sometimes it converges relatively slowly. To address this issue in practice, an additional test can be applied before each E-step to detect and discard likely incorrect loop closure measurements. Let r_{ji} be the i^{th} component of the loop closure residual vector $\mathbf{r}_j$ and $\sigma^2_{nom,ji}$ be the i^{th} component of the diagonal of $\Sigma_{nom,j}$, then the following comparison is applied to each loop closure measurement

$$\frac{r^2_{ji}}{\sigma^2_{nom,ji}} < \eta^2 \quad \forall i \tag{4.14}$$

where $\eta > 0$, is a user-specified threshold. If the test fails for any component i, then the loop closure information matrix is set to zero.

This test is based on the fact that the E-step update provides an empirical estimate of the measurement covariance. Since the inverse of the information matrix is the covariance matrix, the empirical covariance estimate generated by the E-step is:

$$\hat{\Sigma}^{k+1} = (\Lambda_j^{k+1})^{-1} = \frac{V^{-1} + \mathbf{r}_j \mathbf{r}_j^T}{\nu - n_y} \tag{4.15}$$

If $\nu = n_y + 1$, and $V = \Sigma_{nom,j}^{-1}$ then the i^{th} diagonal element of $\hat{\Sigma}^{k+1}$ can be written as

$$\sigma_i^2 = \sigma_{nom,ji}^2 + r_{ji}^2 \tag{4.16}$$

Incorrect loop closure measurements will typically have large residuals, thus σ_i^2 for an incorrect loop closure should be significantly larger than $\sigma_{nom,ji}^2$. One way to quantify this change is to consider the ratio of σ_i^2 and $\sigma_{nom,ji}^2$:

$$\frac{\sigma_i^2}{\sigma_{nom,ji}^2} = 1 + \frac{r_{ji}^2}{\sigma_{nom,ji}^2}$$

Note that the main driver of the ratio between the expected covariance and the empirical covariance will be the ratio $\frac{r_{ji}^2}{\sigma_{nom,ji}^2}$.

Intuitively, the test in Eq. 4.14 is evaluating how unlikely the residuals are given the nominal covariance. For instance, a three standard deviation residual (i.e. $r_{ji}^2 = 9\sigma_{nom,ji}^2$) would occur less than 0.3% of the time assuming that the measurement noise was Gaussian with covariance $\sigma_{nom,ji}^2$. Thus, selecting a value of η corresponds to selecting a minimum probability that the residuals were generated from a correct loop closure with measurement covariance $\sigma_{nom,ji}^2$.

The modified version of IM-SLAM is shown in Algorithm 4. Unlike the initial version of IM-SLAM, Algorithm 4 is not guaranteed to converge, however in practice the algorithm converged and often did so much faster than the basic version of IM-SLAM. In addition, IM-SLAM with thresholding's sensitivity to the parameter η (evaluated in Section 4.2) is small and there is a large basin of convergence with

Algorithm 4 IM-SLAM with Thresholding

Require: Initial pose estimate $\hat{\mathbf{x}}^0$, measurements $\mathbf{y}$, nominal covariance matrices Σ^{nom}, threshold parameter η

```
k = 0
while not converged do
    // Expectation Step
    for j = 1 to n_l do
        if (∃i | r_ji^2/σ_nom,ji^2 > η^2) then
            Λ_j^{k+1} = 0 // Likely an incorrect loop closure
        else
            Compute Λ_j^{k+1} using Eq. 4.9
        end if
    end for
    // Maximization Step
    Compute x̂^{k+1} using Eq. 4.11
    k = k + 1
end while
return x̂^SLAM = x̂^{k+1}
```

respect to η across all of the datasets that were evaluated.

4.2 Algorithm Evaluation

In order to evaluate IM-SLAM several sets of evaluations were performed. The runtime and accuracy performance of IM-SLAM were compared to a non-robust SLAM algorithm (g^2o [11]) as well as several state-of-the-art robust SLAM algorithms (RRR [22, 64, 65], max-mixtures [61] and DCS [60]). Additionally, the sensitivity of the proposed algorithm to the incorrect loop closure detection threshold was evaluated. All of the algorithms were implemented using the g^2o software package [11] to ensure that there was a consistent baseline for comparing runtime and accuracy performance.

4.2.1 Datasets and Evaluation Set-Up

Several simulated benchmark SLAM datasets were used for the algorithm evaluations and are listed in Table 4.1. Since the simulated datasets considered do not contain incorrect loop closures, additional incorrect loop closures were added to the datasets

Table 4.1: Simulated SLAM Datasets Used For Evaluation

Dataset	Nodes	Edges	Max. Incorrect Loop Closures
Manhattan3500 [11, 105]	3500	5598	500
Intel [111]	943	1776	500
City10000 [9]	10000	20687	500
Sphere2500 [9]	2500	4949	500

artificially. Additional outliers were generated using the random, local, grouped and local grouped approaches proposed in [110]. The random outlier generation strategy chooses two poses uniformly at random and generates a false loop closure measurement between them. Local outliers produce loop closure measurements between nodes that are in close proximity to each other in the graph. The grouped outlier strategy creates clusters of mutually consistent false loop closures. The local grouped strategy combines the local and grouped outlier generation approaches. The number of outliers added to each dataset was varied between 100 and 500 in 100 outlier increments. 30 Monte Carlo trials were performed for each dataset, number of outliers and outlier selection strategy for a total of 600 trials for each dataset.

Along with the simulated data, IM-SLAM and the other algorithms were evaluated using the experimentally collected Bicocca dataset [2, 3]. The Bicocca dataset used stereo cameras for loop closure detection and laser scan matching to calculate odometry measurements. There were numerous opportunities for visual aliasing to occur in the dataset as a number of the hallways were visually similar (see Figure 4-4).

The metrics of performance used for the evaluations were root mean-squared position error (RMSE) of the poses and runtime. RMSE was calculated by aligning the SLAM solution with truth and calculating the position error for each node in the graph.

Four different robust SLAM algorithms were compared: DCS [60], max-mixtures [61], RRR [64, 65], and IM-SLAM. All of the algorithms were implemented using the g2o package [11]. The robust kernel implementation of DCS that is included with g2o was used for all evaluations. The nominal value of $\Phi = 1$ was used for all of the DCS

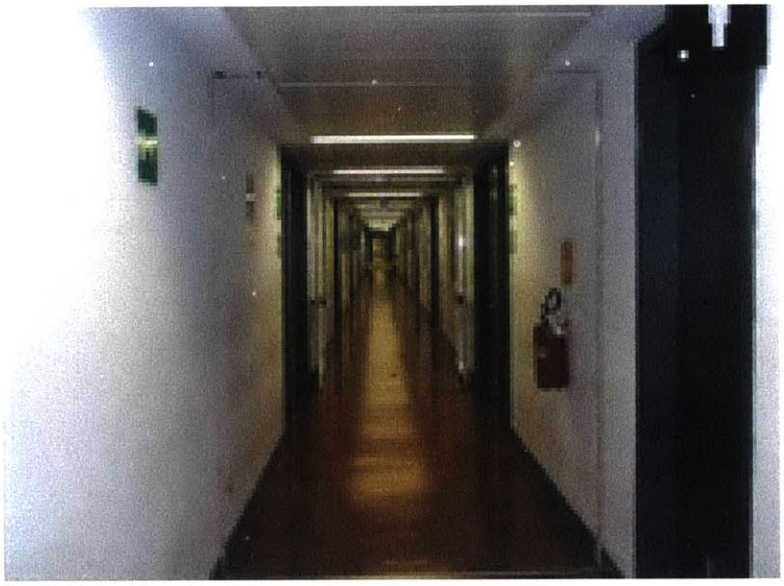


Figure 4-4: Many of the hallways in the Bicocca dataset look similar and cause visual aliasing errors [2, 3]

evaluations except for the Bicocca dataset where $\Phi = 5$ as specified by the authors in their original evaluation [60]. An open source version of max-mixtures was used for the experiments [112]. Two mixture components were used for max-mixtures. The first mixture corresponds to a nominal measurement and had a weight equal to 1. The second mixture corresponds to an outlier and had a weight equal to 0.01 with an information matrix equal to the nominal information matrix scaled by 10^{-6}. Finally, the parameter $\eta = 3$ for IM-SLAM on all of the simulated datasets. Since the measurement noise was not well characterized for the Bicocca dataset, the original version of IM-SLAM without thresholding (Algorithm 3) was applied instead.

4.2.2 Simulation Results

Representative results for the non-robust SLAM solver and IM-SLAM for each simulated dataset are shown in Figure 4-5. Clearly, even for a large number of loop closures IM-SLAM is able to converge to the correct solution and reject the false loop closures. In addition the average runtime per iteration results in Table 4.2 show that IM-SLAM is able to achieve robustness for virtually the same computational cost as a non-robust SLAM algorithm. This also shows that the information matrix

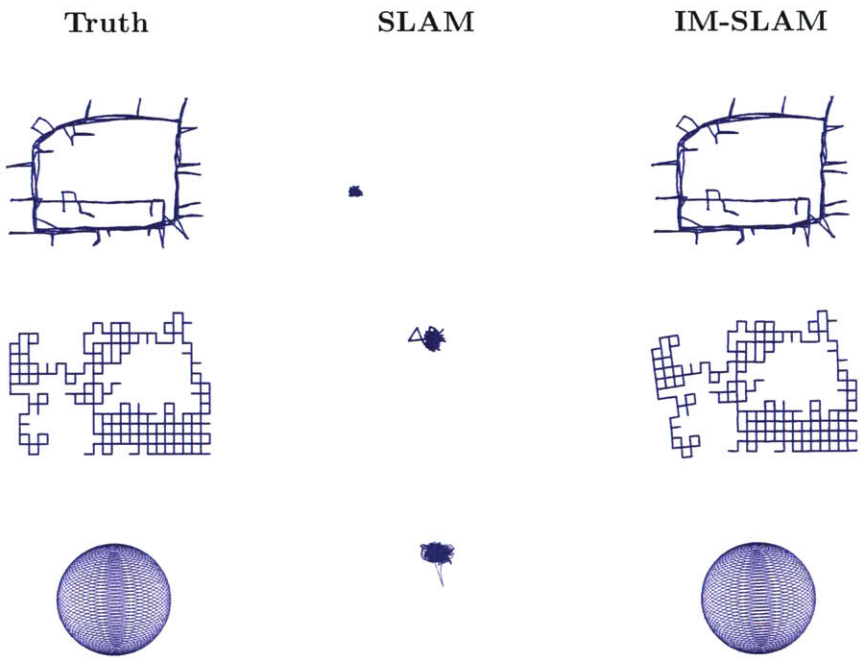


Figure 4-5: Representative results for a non-robust SLAM algorithm and IM-SLAM on the simulated datasets with 500 false loop closures.

Table 4.2: Average Runtime per Iteration

Dataset	SLAM	IM-SLAM
Manhattan3500	0.019 s	0.021 s
Intel	0.012s	0.012s
City10000	0.165 s	0.165 s
Sphere2500	0.685 s	0.688 s

update incurs relatively little computational cost relative to the cost of the nonlinear least-squares update for each iteration of IM-SLAM.

Table 4.3 shows the average root mean squared error (RMSE) for each of the robust algorithms on the simulated datasets. In addition, Table 4.4 average runtimes for each algorithm, respectively. For all of the datasets, the non-robust SLAM algorithm failed to converge so its average runtime results are not shown. The robust SLAM algorithms, with the exception of RRR, achieve comparable error performance. In addition, IM-SLAM was the best performing algorithm on all of the datasets except for Sphere2500. But even for the sphere dataset, IM-SLAM's error performance was comparable to the best performing algorithm.

Table 4.3: Mean RMSE Results for Simulated Datasets

Dataset	DCS [60]	Max-Mixtures [61]	RRR [64]	IM-SLAM
Manhattan3500	0.80	0.81	6.21	0.80
Intel	0.006	0.001	0.716	0.001
City10000	1.00	1.08	2.75	1.00
Sphere2500	0.34	0.33	27.97	**0.35**

Best results shown in green. Bolded results indicate IM-SLAM results that are comparable to the best performing algorithm.

Table 4.4: Average Runtime

Dataset	DCS [60]	Max-Mixtures [61]	RRR [64]	IM-SLAM
Manhattan3500	0.38 s	0.28 s	3.16 s	0.25s
Intel	0.14 s	0.05 s	0.36s	0.29s
City10000	3.80 s	4.23 s	101.19s	3.76 s
Sphere2500	14.85 s	14.48 s	214.54s	**14.66 s**

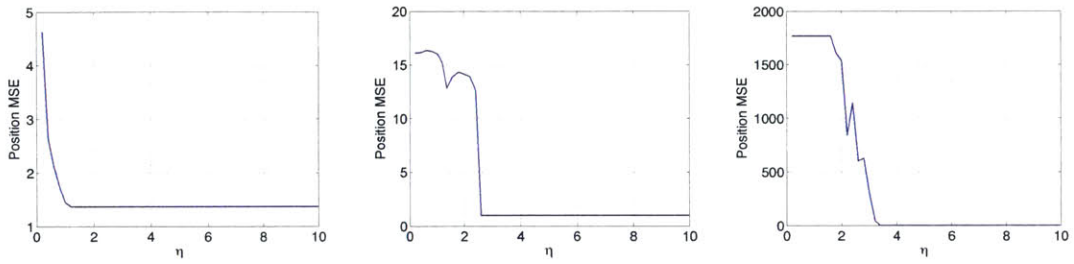
Best results shown in green. Bolded results indicate IM-SLAM results that are comparable to the best performing algorithm.

These experiments have shown that IM-SLAM can match the performance of existing robust SLAM approaches in both runtime performance and accuracy of the solution. In addition, the new algorithm has nearly the same run-time per iteration as a non-robust SLAM algorithm and thus can provide significantly more robust solutions for virtually the same computational cost.

4.2.3 Parameter Sensitivity

Another series of Monte Carlo trials was conducted to evaluate the IM-SLAM's sensitivity to the threshold parameter η. The threshold parameter was varied between 0.1 and 10 and 30 Monte Carlo trials were run for each parameter value. These tests were run on the Manhattan3500, City10000 and Sphere2500 datasets with 500 incorrect loop closures added.

Figure 4-6 shows the average MSE for the simulated datasets as a function of the threshold parameter η. For all of the datasets the performance suffers for small values of η (i.e. $\eta < 1$). In those cases, the incorrect loop closure detection is correctly



(a) Manhattan3500 (b) City10000 (c) Sphere2500

Figure 4-6: Plots of the average MSE for the Manhattan3500, City10000, and Sphere2500 datasets as a function of the threshold parameter η. These results show that IM-SLAM's performance is stable and insensitive to the value of η over a large range of values.

discarding the incorrect loop closures but also ignoring a large number of correct loop closures. In the limit as $\eta \to 0$, all of the loop closure measurements would be ignored and the final solution would be based only on the odometry solution which would contain a substantial amount of drift and cause the MSE to increase as a result. Once η is greater than 3 all of the datasets converge to the correct solution and as a result the MSE is relatively low. Moreover, even as the threshold is increased to much larger values, the solution still converges. This result shows that there is a wide basin of convergence for the parameter η and that in practice values of $\eta \in [3 \quad 5]$ will provide robust solutions for a wide range of problems.

4.2.4 Experimental Results (Bicocca Dataset)

To evaluate the impact of false loop closures in real datasets, the algorithms were also evaluated on the 41 processed Bicocca data files that were released with the RRR package [113]. Each file was generated from the original Bicocca data, using a different loop closure detection threshold α. The threshold α was varied between 0 and 1, and the number of false loop closure detections increased as the threshold was decreased. Figures 4-7–4-9 show the solutions for each algorithm for several different values of α.

Figure 4-10 shows the error performance of each algorithm on the Bicocca dataset as funcion of the threshold parameter α. There are several key takeaways from these

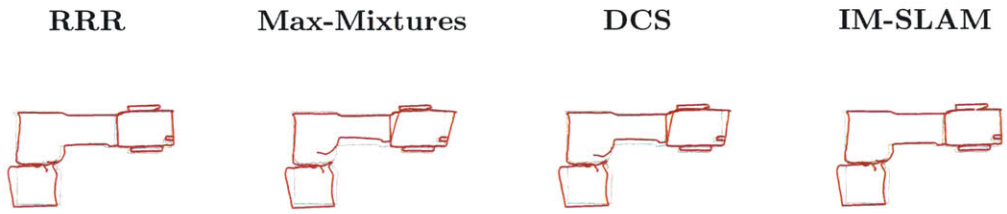


Figure 4-7: Robust SLAM solutions for the Bicocca dataset with $\alpha = 1.0$. Truth is shown in gray.

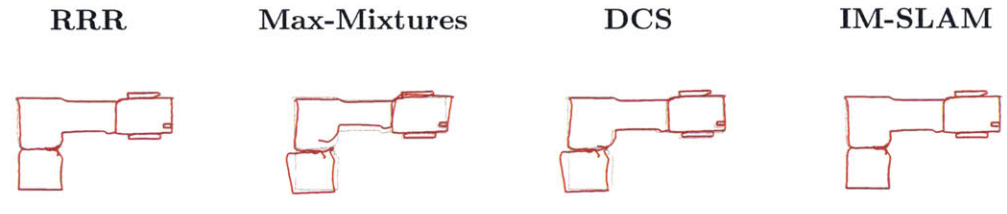


Figure 4-8: Robust SLAM solutions for the Bicocca dataset with $\alpha = 0.5$. Truth is shown in gray.

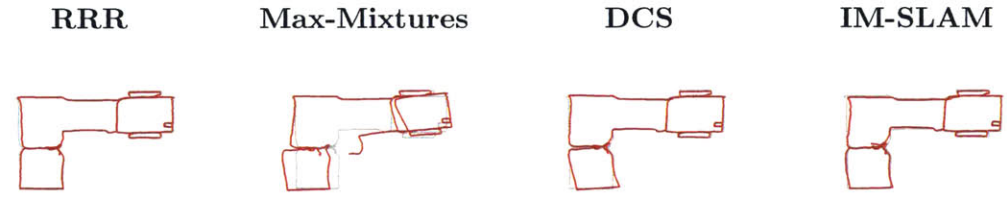


Figure 4-9: Robust SLAM solutions for the Bicocca dataset with $\alpha = 0$. Truth is shown in gray.

results. First, IM-SLAM and RRR are the best performing algorithms and have virtually identical performance for each threshold level. It should be noted that while IM-SLAM and RRR have comparable performance for the Bicocca dataset, IM-SLAM significantly outperformed RRR on the Monte Carlo trials in Section 4.2.2. Second, as the loop closure detection threshold decreases, the RMSE decreases as well for IM-SLAM. This might at first seem like a counterintuitive result because as the threshold decreases more false loop closures are detected. But, as the threshold decreases more correct loop closures that can reduce the RMSE are also detected, meaning that if a robust SLAM algorithm can filter out the incorrect loop closures, it can produce a more accurate solution! This result shows that by using a robust SLAM algorithm the

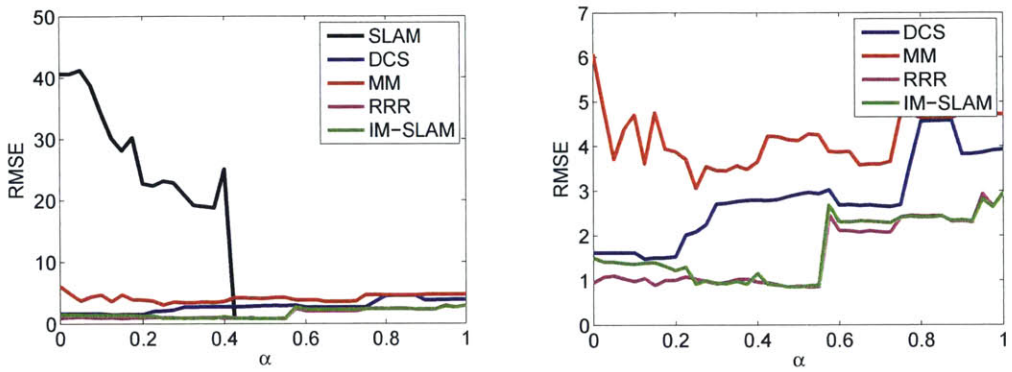


Figure 4-10: Comparison of RMSE for Bicocca dataset as a function of the loop closure detection threshold α. (b) shows the same plot with the non-robust SLAM algorithm removed.

loop closure detection threshold can be set less conservatively because the impact of false loop closures can be mitigated and because the solution can actually be improved by the additional correct loop closures that are detected.

4.3 Summary

One of the major challenges associated with SLAM systems arises because the Gaussian assumption is often a poor approximation of the actual measurement noise. For instance, loop closure measurements can be generated incorrectly because of visual aliasing which can lead to measurement residuals that are much larger than the expected Gaussian noise. Additionally, the measurement noise statistics could be a function of time or location of the robot (i.e. due to multipath effects for sonar or radar) which is difficult to capture with a single Gaussian model. If the measurement noise is assumed to be a fixed Gaussian and these types of errors occur, the SLAM solution can dramatically degrade [20, 61]. In the case of incorrect loop closures, even a single instance can lead to divergence in the estimates of the robot poses [61].

This chapter presented a novel robust SLAM algorithm, IM-SLAM, that retains the efficiency of nonlinear least-squares SLAM algorithms while compensating for both incorrect loop closure measurements and measurements with noise covariances

that do not match the modeled noise. An additional set of variables, corresponding to the information matrices of each measurement, are introduced to the SLAM inference problem and estimated along with the robot poses. An EM procedure is used to iteratively estimate the robot poses and the information matrices. By estimating the information matrices, the impact of incorrect measurements and noise covariances on the pose estimates is significantly reduced. Additionally, the EM procedure only involves a closed form update rule for the noise parameters and a standard nonlinear least-squares update step for the pose estimates. In practice, this leads to a computationally efficient SLAM algorithm that is significantly more robust to measurement errors than traditional SLAM techniques.

Monte Carlo simulations and experimental results demonstrated that IM-SLAM provides significantly more accurate estimates than non-robust SLAM solvers when incorrect loop closures are present and can match the performance of state-of-the-art robust SLAM algorithms. Moreover, the computation time for IM-SLAM is on par with non-robust SLAM techniques because the update rule for the information matrix parameters can be calculated in closed form. In addition, IM-SLAM was shown to match the error and run-time performance of alternative state-of-the-art robust SLAM techniques. Finally, a sensitivity study demonstrated that IM-SLAM has a wide basin of convergence with respect to its tuning parameter and moreover the tuning parameter can easily be specified in terms of the measurement covariances.

Chapter 5

Robust Incremental SLAM with Consistency Checking

5.1 Introduction

While several robust SLAM algorithms have been proposed in the literature [20, 60–64, 66], most have focused on robustness to loop closure outliers rather than the problem of landmark measurement outliers. Previously [60], it has been demonstrated that robust SLAM algorithms can fail when applied to problems with landmark measurement outliers (see Figure 5-1). This failure occurs because most current robust algorithms only focus on ensuring local map consistency by evaluating whether each measurement, independent of the others, is an outlier. Ideally, a robust SLAM algorithm should verify that the map is both locally and globally consistent.

In addition, most robust SLAM algorithms are designed for batch processing rather than incremental (online) processing (with the notable exceptions of max-mixtures [21] and iRRR [65]). Online SLAM is a case where robust algorithms are needed most because a robotic system applying online SLAM will typically use the mapping solution for planning and control. Given an inconsistent map corrupted by measurement outliers the planning and control systems will likely provide suboptimal results because the estimates do not reflect the true structure of the environment.

The contributions of this chapter are (1) a new formulation of the robust SLAM

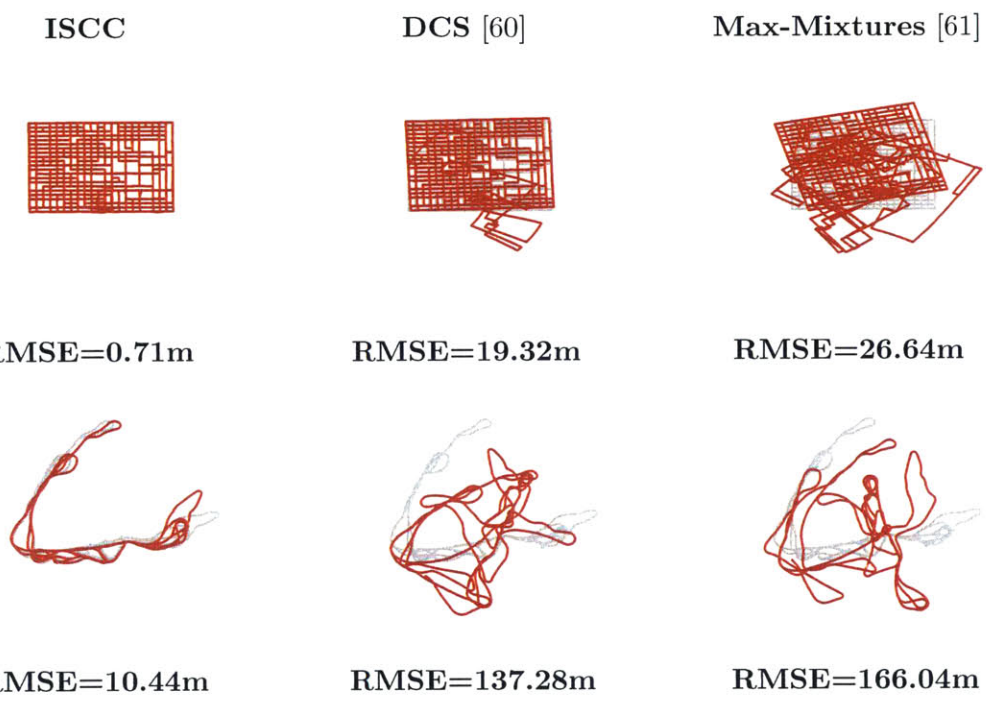


Figure 5-1: Typical optimization results for CityTrees10000 (top row) and Victoria Park (bottom row) with 1000 incorrect landmark measurements. Root mean-squared error values for each solution are shown below each figure. Both max-mixtures and dynamic covariance scaling fail to correctly identify the incorrect measurements and the SLAM solution suffers as a result. New ISCC method can correctly detect the incorrect measurements and provides substantially better results. Truth in each figure is shown in gray.

problem and (2) a novel incremental SLAM algorithm called incremental SLAM with consistency-checking (ISCC) that approximately solves the robust SLAM problem. ISCC solves the robust SLAM problem by removing as few measurements as possible from the SLAM solution to make it locally and globally consistent. We demonstrate in the chapter that ISCC significantly outperforms existing robust SLAM algorithms when landmark measurement outliers occur and can match the performance of state-of-the-art algorithms when loop closure errors occur.

The outline of the rest of the chapter is as follows. Section 5.2 presents the new formulation of the robust SLAM problem. Section 5.3 develops ISCC. Section 5.4 presents simulated and experimental results that compares the performance of ISCC to state-of-the-art robust SLAM algorithms. Finally, Section 5.6 provides a brief summary.

5.2 Robust SLAM Problem Formulation

There are two primary criteria for robust SLAM:

1. Generate a solution that is *consistent* both locally and globally,

2. Generate a solution that is as accurate as possible.

In this context, consistent means that the solution generated by the SLAM algorithm agrees with the measurements used in the solution. Consistency is defined more formally in the next section.

The approach taken in this chapter is to decide which measurements should be included in the factor graph and then solve the SLAM problem on that graph. However, it should be noted that the two criteria do not necessarily align. For instance, a consistent solution could be generated by only processing the odometry measurements, but it would not be metrically accurate due to the accumulation of odometry drift. Therefore consistency is not sufficient to guarantee an accurate solution. To generate the most accurate solution, ideally the SLAM solution would include as many of the measurements as possible so long as the solution remained consistent.

The rest of this section defines a set of consistency tests and then poses the robust SLAM problem based on the criteria described above.

5.2.1 Consistency Tests

If the factor graph were composed entirely of odometry measurements, a consistent solution could be calculated by setting the pose estimates such that they satisfy the odometry exactly. However, when landmark and loop closure measurements are added to the graph, the SLAM solution will not be able to satisfy all of the measurements exactly because of measurement noise. If all of the measurements are correct, the resulting measurement residuals should still be small. But, if any of the measurements are outliers, we would expect the measurement residuals to be larger because some pose estimates will be set to values that are inconsistent with their respective constraints. Therefore, the sum of measurement residuals provides a tool for determining

whether the graph is consistent.

A straight-forward means of testing whether the graph is globally consistent is to apply a χ^2 test to the weighted sum of measurement residuals. Given pose and landmark estimates, odometry, landmark and loop closure measurements, the weighted sum of measurement residuals of the graph can be expressed as

$$\chi^2_G = \sum_i \chi^2_{o,i} + \sum_{\mathbf{y}^L_{i,j} \in L} \chi^2_{L,ij} + \sum_{\mathbf{y}^{LC}_{i,k} \in LC} \chi^2_{LC,ik} \tag{5.1}$$

where $\chi^2_{o,i}$, $\chi^2_{L,ij}$ and $\chi^2_{LC,ik}$ are given by

$$\chi^2_{o,i} = ((\mathbf{x}_i \oplus \mathbf{y}^{\mathbf{o}}_{\mathbf{i}}) \ominus \mathbf{x}_{i+1})^T \Lambda_o ((\mathbf{x}_i \oplus \mathbf{y}^{\mathbf{o}}_{\mathbf{i}}) \ominus \mathbf{x}_{i+1}) \tag{5.2}$$

$$\chi^2_{L,ij} = ((\mathbf{x}_i \oplus \mathbf{y}^{\mathbf{L}}_{\mathbf{i,j}}) \ominus \mathbf{l}_j)^T \Lambda_L ((\mathbf{x}_i \oplus \mathbf{y}^{\mathbf{L}}_{\mathbf{i,j}}) \ominus \mathbf{l}_j) \tag{5.3}$$

$$\chi^2_{LC,ik} = ((\mathbf{x}_i \oplus \mathbf{y}^{\mathbf{LC}}_{\mathbf{i,k}}) \ominus \mathbf{x}_k)^T \Lambda_{LC} ((\mathbf{x}_i \oplus \mathbf{y}^{\mathbf{LC}}_{\mathbf{i,k}}) \ominus \mathbf{x}_k) \tag{5.4}$$

Λ_o, Λ_L and Λ_{LC} are the information matrices of the odometry, landmark and loop closure measurements respectively, and $\oplus$ and $\ominus$ are standard pose composition operators [72].[1] If the solution is consistent we would expect χ^2_G to satisfy the following inequality with probability p:

$$\chi^2_G \leq \chi^2(p, n_{dof}) \tag{5.5}$$

where $\chi^2(p, n_{dof})$ is the inverse χ-squared cdf with n_{dof} degrees of freedom evaluated at p. By setting p to a value close to 1 (i.e. 0.95) we can verify that the graph is consistent with high probability as long as it satisfies Eq. 5.5.

In addition to being globally consistent, ideally each measurement included in the factor graph should be locally consistent with its associated pose and landmark estimates. Thus, we also define local consistency tests for each landmark and loop closure measurement

$$\chi^2_{L,ij} \leq \chi^2(p, n_{dof,L}) \quad \forall j \tag{5.6}$$

[1] In general, Eqs. 5.2 and 5.4 should also include a logarithmic map that maps the residuals in $SE(2)$ or $SE(3)$ to their respective tangent spaces. For more details see [114, 115].

$$\chi^2_{LC,ik} \leq \chi^2(p, n_{dof,LC}) \quad \forall k \tag{5.7}$$

where $n_{dof,L}$ and $n_{dof,LC}$ are the number degrees of freedom of the landmark and loop closure measurements. If a graph satisfies Eqs. 5.5–5.7 then we declare the graph to be consistent.

5.2.2 Robust SLAM Problem Formulation

With the consistency tests defined, the final step is to formulate a cost function for robust SLAM. We make the standard assumptions that the measurement noise is Gaussian and that the odometry measurements are not corrupted with outliers. We also define a set of binary variables s^L_j and s^{LC}_k that indicate whether the jth landmark measurement and kth loop closure measurement are included ($s^L_j = 1$,$s^{LC}_k = 1$) in the graph. Given the intuition that a robust SLAM solution should include as many measurements as possible while remaining consistent, we define the robust SLAM problem as:

$$\begin{aligned}
\max_{\mathbf{x},\mathbf{l},\mathbf{s}} \quad & \sum_{j=1}^{n_L} s^L_j + \sum_{k=1}^{n_{LC}} s^{LC}_k \\
\text{s.t.} \quad & \sum_{i=1}^{n_o} \chi^2_{o,i} + \sum_{j=1}^{n_L} s^L_j \chi^2_{l,j} + \sum_{k=1}^{n_{LC}} s^{LC}_k \chi^2_{LC,ik} \leq \chi^2(p, n_{dof}) \\
& s^L_j \chi^2_{L,ij} \leq \chi^2(p, n_{dof,L}) \quad \forall j \\
& s^{LC}_k \chi^2_{LC,ik} \leq \chi^2(p, n_{dof,LC}) \quad \forall k \\
& s^L_j \in \{0,1\} \quad \forall j, \quad s^{LC}_k \in \{0,1\} \quad \forall k
\end{aligned} \tag{5.8}$$

where n_{dof} is the number of degrees of freedom in the factor graph. This cost function ensures that any solution will meet the criteria for a robust SLAM solution. Maximizing the sum of the indicator variables encourages a solution that includes as many measurements as possible, which will lead to an accurate solution, and the χ^2 constraints ensure that the solution will be consistent.

While the problem formulation in Eq. 5.8 meets the criteria for a robust SLAM

cost function, it is not practical to optimize it directly. In particular, Eq. 5.8 is a mixed integer nonlinear program and is NP-hard to solve [116]. The remainder of this chapter proposes and evaluates an algorithm that approximately solves Eq. 5.8.

5.3 Robust Incremental SLAM with Consistency Checking

This section proposes an incremental greedy solution to the robust SLAM problem in 5.8 called ISCC. By processing the data incrementally and verifying that the current graph satisfies Eqs. 5.5–5.7 we ensure that a consistent graph is maintained throughout the solution process. Additionally, as new measurements are added to the graph they can be identified as outliers by evaluating whether the updated graph is inconsistent or not. If the graph is inconsistent after adding a new measurement, a greedy search is performed to find the minimum number of measurements that can be removed from the graph in order to make the graph consistent again.

Pseudocode for ISCC is shown in Algorithm 5. Measurements are processed as they are generated by the SLAM front-end. Odometry measurements are assumed to be outlier free and are automatically added to the factor graph. Landmark and loop closure measurement are added to graph and then the graph is checked for consistency using Eqs. 5.5–5.7. If the graph is found to be inconsistent, ISCC performs a greedy search (lines 7–8 of Algorithm 5) that seeks to remove the fewest number of measurements from the graph such that it becomes consistent. This process removes outliers from the graph and ensures that the graph is always consistent after each measurement is processed. The rest of this section describes in detail the greedy search procedure for removing outliers from the graph.

5.3.1 Outlier Identification

If the graph becomes inconsistent after adding a new measurement, $\mathbf{y}_i$, there are two possible explanations: 1) $\mathbf{y}_i$ is an outlier or 2) a previously processed measurement

Algorithm 5 ISCC

Require: Measurement queue $\mathbf{Y}$
1: Initial graph $G = \emptyset$
2: **for** (each $\mathbf{y}_i \in \mathbf{Y}$) **do**
3: Add $\mathbf{y}_i$ to G and update
4: **if** ($\mathbf{y}_i$ is not odometry) **then**
5: Check if G is consistent using Eqs. 5.5–5.7
6: **if** (G is not consistent) **then**
7: $\mathbf{Y}_{test} \leftarrow$ findCandidateOutliers($\mathbf{y}_i$,G)
8: $G \leftarrow$ findConsistentMeasurementSet($\mathbf{y}_i$,$\mathbf{Y}_{test}$,G)
9: **end if**
10: **end if**
11: **end for**
12: **return** Optimized graph G

is an outlier and was erroneously accepted. To make the graph consistent, ISCC first determines which previously accepted measurements are most likely to be outliers (using Algorithm 6) and then removes as few measurements as possible from the graph while ensuring that the resulting graph is consistent (using Algorithm 7).

ISCC applies two different strategies for determining which measurements are potential outliers depending on the measurement type. If $\mathbf{y}_i$ is a landmark measurement, then logically the other measurements of that landmark are the most likely candidates to be outliers. Therefore, if $\mathbf{y}_i$ is a landmark measurement Algorithm 6 simply returns the set of measurements of that landmark (Step 3).

In the case of loop closures, note that if the graph has become inconsistent after adding $\mathbf{y}_i$, any loop closure measurements that are locally inconsistent (i.e. they fail the test in Eq. 5.7) must contain information which is not consistent with $\mathbf{y}_i$. This means that if $\mathbf{y}_i$ is not an outlier, then at least one of the measurements that failed the local consistency tests is an outlier. Using this information, if $\mathbf{y}_i$ is a loop closure measurement, Algorithm 6 returns the set of loop closures in the graph that are locally inconsistent (Step 5).

Algorithm 6 findCandidateOutliers

Require: current measurement $\mathbf{y}$, factor graph G

1: **if** $(\mathbf{y} \in L)$ **then**
2: $\quad j^* \leftarrow$ Index of landmark associated with $\mathbf{y}$
3: $\quad Y_{test} \leftarrow \left\{\mathbf{y}_{i,j}^{L} \in L \mid j = j^*\right\}$
4: **else**
5: $\quad Y_{test} \leftarrow \left\{\mathbf{y}_{i,k}^{LC} \in LC \mid \chi^2_{LC,ik} > \chi^2(p, n_{dof,LC})\right\}$
6: **end if**
7: **return** Set of potential outliers Y_{test}

5.3.2 Greedy Outlier Removal

Given a set of candidate outliers Y_{test}, the next step (Algorithm 7) is to determine which measurements should be removed from the graph. Since ISCC is optimizing Eq. 5.8, it should remove as few measurements as possible to make the graph consistent. Note that since the graph was consistent before adding $\mathbf{y}_i$, an admissible solution to the outlier removal problem is to remove $\mathbf{y}_i$. However that solution may not be unique because $\mathbf{y}_i$ could, in fact, be correct, in which case there should be at least one other measurement that could be removed from the graph to generate a consistent solution. Therefore, in order to remove any potential outliers from the graph, ISCC tests each measurement in Y_{test} to determine if any of them can be removed from the graph to make it consistent (Steps 2–8 of Algorithm 7). If any potential outliers are found in Y_{test}, ISCC removes them from the graph along with $\mathbf{y}_i$ because there is no clear way to decide which of the measurements are the outliers (Step 9 of Algorithm 7). However, if no other measurements can be removed, only $\mathbf{y}_i$ is removed from the graph.

While it is difficult to establish exact complexity bounds for ISCC, we can at least establish bounds on the total number of optimizations required for a given dataset. If there are k non-odometry edges in the current graph, the worst-case number of optimizations for one iteration of ISCC is $k+1$. This case would occur if every single loop closure measurement failed the local consistency test (or all of the landmark measurements were associated with one landmark) and thus all had to be checked by Algorithm 7. Following this logic, the maximum number of optimizations can be

Algorithm 7 findConsistentMeasurementSet

Require: Current Measurement $\mathbf{y}$, test set Y_{test}, factor graph G

```
outliers ← y
for (each y_i ∈ Y_test ) do
    Remove y_i from G, Update G
    if (G is consistent) then
        outliers ← outliers ∪ y_i
    end if
    Add y_i to G
end for
Remove outliers from G, Update G
return Updated factor graph G
```

upper bounded by $\sum_i^N i = (N^2 + N)/2$, where N is the number of loop closure and landmark edges in the graph. So the total number of optimizations is $\mathbb{O}(N^2)$. In practice however, often only a few candidate outliers are found using Algorithm 6 and the actual number of optimizations required to run ISCC tends to be much closer to N. The runtime performance of ISCC will be examined in more detail in the experimental results.

5.4 Simulation Results

Simulated and real-world datasets were used to evaluate ISCC and compare it to state-of-the-art robust SLAM algorithms. The evaluations focused on the accuracy of the solutions, as well as, how accurately the robust SLAM methods identified outliers in the datasets.

5.4.1 Datasets Used for Evaluation

Several standard benchmark SLAM datasets (Manhattan3500, Intel, City10000, Sphere2500, CityTrees10000, Victoria Park, and Torus2000Points) were used for the algorithm evaluations. Manhattan3500, Intel, City10000, and Sphere2500 contain loop closure measurements while CityTrees10000, Victoria Park and Torus2000Points contain landmark measurements. Sphere2500, City10000, CityTrees10000, Victoria Park

and Torus2000Points are available as part of the iSAM package [10]. The Manhattan3500 and Intel datasets are available as part of the g2o distribution [117]. We also compared the algorithms on the real-world Bicocca dataset using the processed data files released with the RRR package [113]. Table 5.1 provides a summary of the benchmark datasets used for the evaluations.

Table 5.1: Benchmark SLAM Datasets Used For Evaluation

Dataset	Nodes	Edges	Measurement Type	Data Type
Manhattan3500 [11, 105]	3500	5598	Loop Closures	Simulated
Intel [111]	943	1776	Loop Closures	Experimental
City10000 [9]	10000	20687	Loop Closures	Simulated
Sphere2500 [9]	2500	4949	Loop Closures	Simulated
Bicocca [113]	8358	8803	Loop Closures	Experimental
CityTrees1000 [10]	10100	14442	Landmarks	Simulated
Victoria Park [10]	7120	10608	Landmarks	Experimental
Torus2000 [10]	1093	2034	Landmarks	Simulated

5.4.2 Evaluation Procedure

Since the simulated datasets do not contain incorrect measurements, additional outliers were added artificially. For the datasets containing loop closures, additional outliers were generated using the random, local, grouped and local grouped approaches proposed in [110]. The random outlier generation strategy chooses two poses uniformly at random and generates a false loop closure measurement between them. Local outliers produce loop closure measurements between nodes that are in close proximity to each other in the graph. The grouped outlier strategy creates clusters of mutually consistent false loop closures. The local grouped strategy combines the local and grouped outlier generation approaches. Landmark measurement outliers were generated by choosing landmarks and poses at random and generating measurements between them as in the comparisons from [60]. The number of outliers added to each dataset was varied between 200 and 1000 in 200 outlier increments. 30 Monte Carlo trials were performed for each dataset, number of outliers and outlier selection strategy for a total of 600 trials.

The Bicocca dataset was collected experimentally and contains loop closures that were generated by a visual bag-of-words based place recognition system. The false loop closures in the dataset were caused by visual aliasing.

Five different robust SLAM algorithms were compared: DCS [60], max-mixtures [61], RRR [64, 65], Carlone et al.'s linear programming approach (l_1-SLAM)[66], and ISCC. All of the algorithms (with the exception of l_1-SLAM) were implemented using the g2o package [11]. The robust kernel implementation of DCS that is included with g2o was used for all evaluations. The nominal value of $\Phi = 1$ was used for all of the DCS evaluations except for the Bicocca dataset where $\Phi = 5$ as specified by the authors in their original evaluation [60]. An open source version of max-mixtures was used for the experiments [112]. Two mixture components were used for max-mixtures. The first mixture corresponds to a nominal measurement and had a weight equal to 1. The second mixture corresponds to an outlier and had a weight equal to 0.01 with an information matrix equal to the nominal information matrix scaled by 10^{-6}. Finally, the χ^2 probability threshold for ISCC was $p = 0.95$.

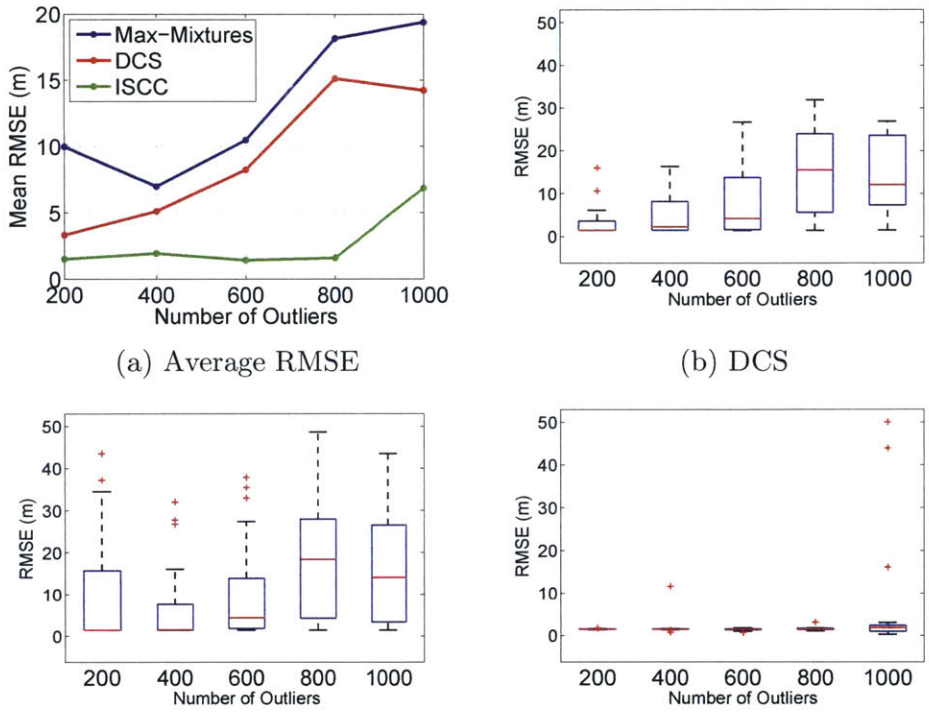
The metrics of performance used for the evaluations were root mean-squared position error (RMSE) of the poses and precision and recall of the measurements. RMSE was calculated by aligning the SLAM solution with truth and calculating the position error for each node in the graph. Precision in this context measures the fraction of measurements included in the final graph that were correct, while recall measures the fraction of the total correct measurements that were included in the graph. Note that an ideal robust SLAM algorithm would achieve precision and recall values of 1. Precision and recall could not be used to evaluate DCS because it does not make binary decisions about whether each measurement in the graph is an outlier.

5.4.3 Landmark Dataset Results

Results for the landmark datasets are shown in Figures 5-2–5-4. Representative examples of typical results are also shown in Figure 5-1. Results for l_1-SLAM and RRR are not shown because they are not designed for landmark-based SLAM datasets and could not be applied. Overall, ISCC significantly outperforms the other robust

SLAM algorithms. ISCC achieved the lowest average RMSE performance across every dataset by a significant margin. Moreover, in the case of the Torus2000 dataset, DCS and max-mixtures diverged for nearly every Monte Carlo trial while ISCC consistently converged to a good solution. These results indicate that searching for solutions that are both globally and locally consistent leads to significantly more robust solutions when landmark errors occur.

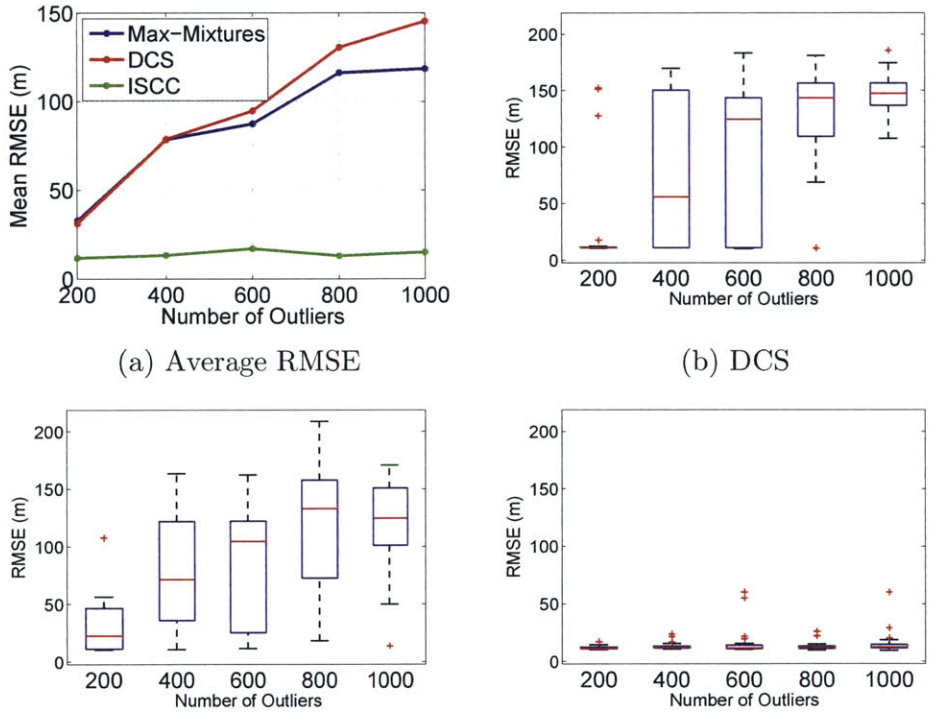
Figure 5-5 shows the precision and recall values for max-mixtures and ISCC applied to the Victoria Park dataset with 1000 incorrect landmark measurements. The precision values indicate that most of the measurements included in the graph by both algorithms are correct, but the recall values indicate that max-mixtures ignores a significant fraction of the correct landmark measurements. Ignoring so many correct measurements can cause the SLAM solution to be more heavily impacted by incorrect measurements and allows for the local build-up of substantial odometry drift errors. Also, note that the optimal solution to Eq. 5.8 should have precision and recall values of 1, so ISCC is near-optimal for most of the Victoria Park Monte Carlo trials. Results for the CityTrees10000 dataset were similar and are shown in Figure 5-6.



(a) Average RMSE (b) DCS

(c) Max-Mixtures (d) ISCC

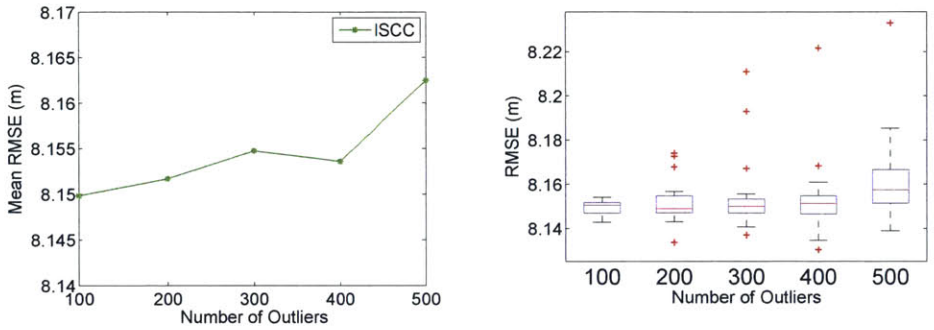
Figure 5-2: Comparison of Monte Carlo Results for the CityTrees10000 dataset.



(a) Average RMSE (b) DCS

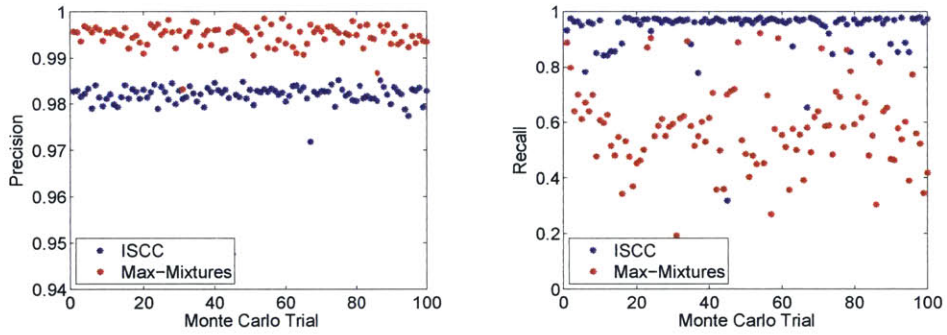
(c) Max-Mixtures (d) ISCC

Figure 5-3: Comparison of Monte Carlo Results for the Victoria Park dataset.



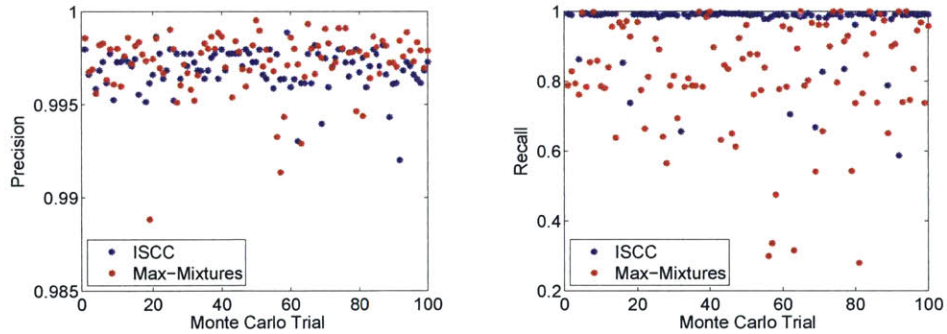
(a) Average RMSE (b) ISCC

Figure 5-4: RMSE Results for the Torus2000Points dataset. Max-mixtures and DCS could not converge to a valid solution on this dataset.



(a) Precision (b) Recall

Figure 5-5: Precision and recall results for the Victoria Park dataset. Max-mixtures shows a significant drop in recall with respect to ISCC. The drop in recall indicates that max-mixtures is incorrectly rejecting a large fraction of the correct landmark measurements.

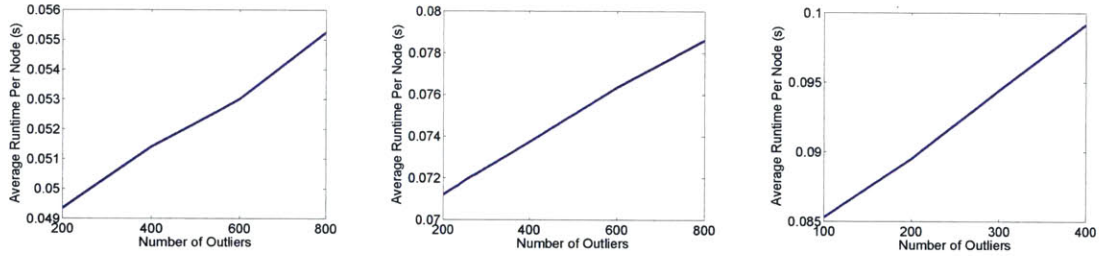


(a) Precision (b) Recall

Figure 5-6: Precision and recall results for the CityTrees10000 dataset. Max-mixtures shows a significant drop in recall with respect to ISCC. The drop in recall indicates that max-mixtures is incorrectly rejecting a large fraction of the correct landmark measurements.

5.4.4 Runtime Performance of ISCC

Figure 5-7 shows the runtime performance of ISCC as a function of the number of outliers. It does not make sense to make a direct runtime comparison between ISCC and the other robust SLAM algorithms because the robust algorithms solve the batch SLAM problem. Batch solvers will inherently have faster total runtimes because they only need to solve a single SLAM problem instance, while incremental methods must solve a SLAM problem at each iteration. Thus, instead of measuring total runtime, average runtime per node is shown to provide insight into how much latency would be required on average between measurements to allow ISCC to run online. Unsurprisingly, the computation time increases as the number of outliers increases because ISCC must perform more optimizations and more consistency checks in order to remove the outliers. However, note that the average runtime per node is still small for each dataset even when there are 1000 outliers in the data.



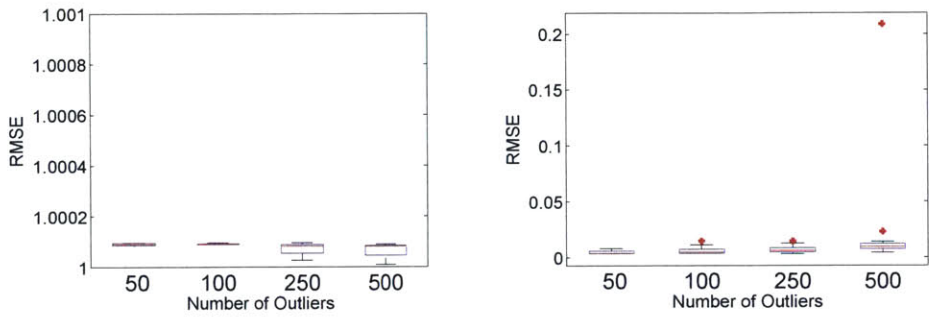
(a) Victoria Park (b) CityTrees10000 (c) Torus

Figure 5-7: Runtime performance for ISCC as a function of the number of outliers.

5.4.5 Loop Closure Dataset Results

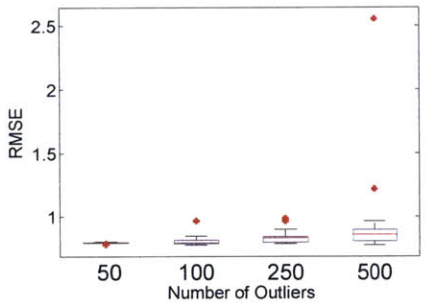
Figures 5-8 and 5-9 show the error performance for ISCC as a function of the number of outliers and the outlier generation strategy, respectively. Overall, ISCC is relatively insensitive to both the number of outliers and the outlier generation strategy used and consistently achieves accurate performance for each dataset.

Average RMSE and precision and recall results for each of the robust SLAM algorithms on the loop closure datasets are shown in Tables 5.2 and 5.3. Results for l_1-SLAM were omitted for Sphere2500 because l_1-SLAM can only be applied to 2D

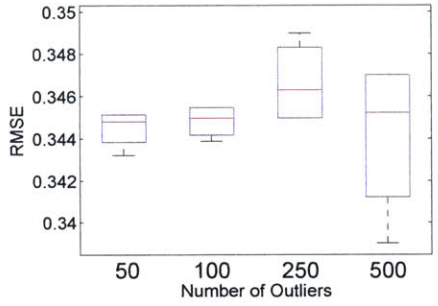


(a) City10000

(b) Intel



(c) Manhattan3500

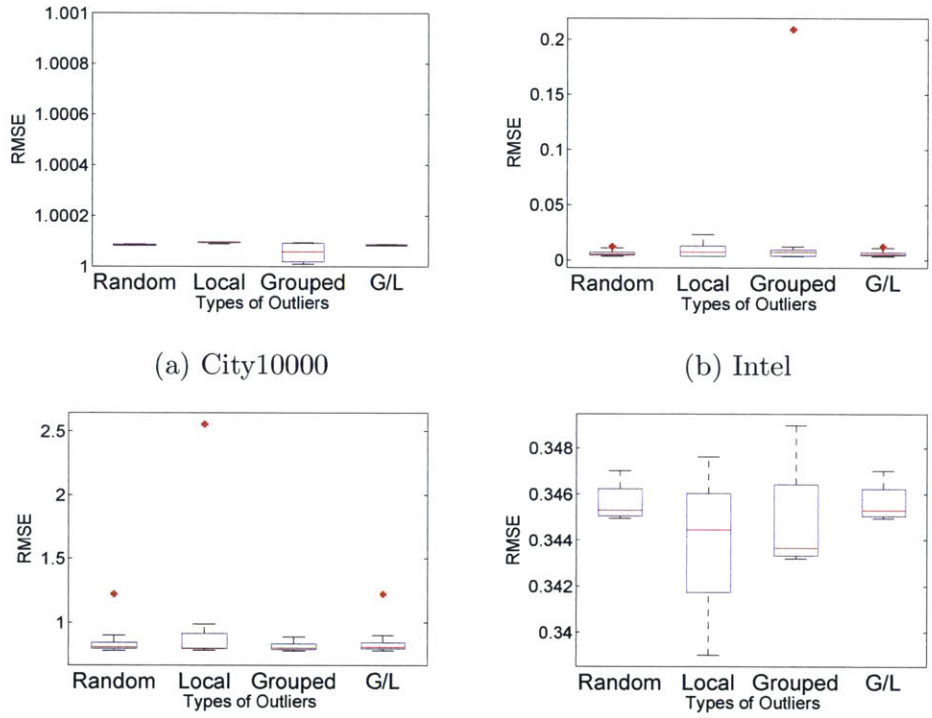


(d) Sphere3500

Figure 5-8: RMSE performance for ISCC on each simulated dataset as a function of the number of outliers.

datasets. Additionally, while processing the City10000 dataset with l_1-SLAM, the computer ran out of memory and as a result a solution could not be generated.

There are several notable findings in the results. First, ISCC's performance is comparable to the best results from the other robust SLAM algorithms on each dataset. Moreover, ISCC significantly outperforms RRR on every dataset except for Bicocca where the results are comparable. The low recall scores for RRR indicate that the difference in RMSE performance between ISCC and RRR can be attributed to RRR rejecting a larger number of correct loop closures. In effect, RRR is finding a consistent solution that is not metrically accurate because it does not explicitly attempt to maximize the number of measurements included in the factor graph. Overall, these results demonstrate that ISCC's outlier rejection strategy can provide comparable performance to existing robust SLAM techniques when loop closure errors occur.



(a) City10000

(b) Intel

(c) Manhattan3500

(d) Sphere3500

Figure 5-9: RMSE performance for ISCC on each simulated dataset as a function of the outlier generation strategy.

Table 5.2: Average RMSE Results for Loop Closure Datasets

Dataset	DCS [60]	Max-Mixtures [61]	RRR [64]	l_1-SLAM [66]	ISCC
Manhattan3500	0.80	0.81	6.21	1.70	**0.85**
Intel	0.006	0.001	0.716	0.010	**0.009**
Bicocca	1.83	4.17	1.59	1.73	1.56
City10000	1.00	1.08	2.75	N/A	1.00
Sphere2500	0.34	0.33	27.97	N/A	**0.35**

Best results shown in green. Bolded results indicate ISCC results that are comparable to the best performing algorithm. Results for l_1-SLAM were omitted for Sphere2500 because the algorithm can not be applied to 3D datasets. Results for l_1-SLAM were omitted for City10000 because the algorithm ran out of memory while attempting to perform the optimization.

Table 5.3: Average Precision/Recall Results for Loop Closure Datasets

Dataset	Max-Mixtures [61] Prec./Rec.	RRR [64] Prec./Rec.	l_1-SLAM [66] Prec./Rec.	ISCC Avg. Prec./Rec.
Manhattan3500	0.999/1.00	0.997/0.894	0.999/0.994	0.999/0.984
Intel	1.00/0.991	1.00/0.770	1.00/0.994	1.00/0.988
City10000	1.00/0.997	0.998/0.994	N/A	1.00/0.999
Sphere2500	0.958/1.00	1.00/0.505	N/A	1.00/0.993

Note: Results for l_1-SLAM were omitted for Sphere2500 because the algorithm can not be applied to 3D datasets. Results for l_1-SLAM were omitted for City10000 because the algorithm ran out of memory while attempting to perform the optimization.

5.5 Experimental Results

To evaluate the algorithms in a more realistic scenario, two experimental datasets were collected. The first dataset (ACL-landmarks) is a landmark-based SLAM dataset that includes repeated landmarks which induce visual aliasing and data association problems. The second dataset (ACL-loops) uses both landmarks and visual loop closure detection to generate the SLAM solution, and contains a number of visual aliasing events that cause false loop closure and landmark detections.

The ACL-landmarks dataset was collected using an Asus Xtion Live Pro RGB-D camera mounted on a cart. Visual odometry measurements were calculated using FOVIS [118]. Twenty-five unique AprilTags [119] were used as visual landmarks in the dataset and detecting using the AprilTags C++ library [120]. In addition, seven duplicate AprilTags were placed in the test area to generate incorrect landmark measurements similar to ones caused by visual aliasing. Placing duplicate tags in the dataset created a more realistic scenario than the Monte Carlo simulations because incorrect landmark detections are clustered spatially rather than randomly distributed throughout the dataset. Approximately 13% of the landmark measurements were generated from the duplicate tags and acted as outliers. Truth was generated using a Vicon motion capture system [121].

The ACL-loops dataset was collected in a laboratory and office space using a Microsoft Kinect RGB-D camera mounted on an Adept Pioneer 3DX robot. Odometry measurements were generated from wheel odometry and IMU measurements. Loop closure detections were generated using the DBoW2 library [17]. Finally, the landmarks used in the dataset were chairs located in the test area. Several of the chairs are identical and thus could produce incorrect data associations and landmark measurements. AprilTags were placed on the chairs to simplify the detection of the objects, however in practice it could be possible to detect the objects directly [122]. Truth data was generated by removing the false loop closures and landmark detections from the data and optimizing the cleaned data using g^2o.

Results for the landmark experimental dataset are shown in Figure 5-10. As

with the Monte Carlo trials, DCS and max-mixtures were unable to converge to the correct solution. In contrast, ISCC was able to recover the correct solution with a small amount of reconstruction error.

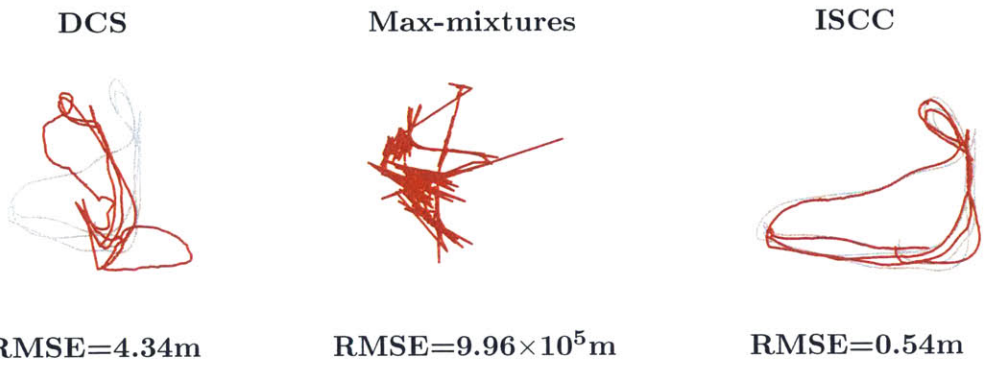


Figure 5-10: Experimental results for the ACL-landmarks dataset. Truth is shown in gray in each figure. DCS fails to converge to the correct solution and max-mixtures diverges. In contrast ISCC is able to accurately reconstruct the solution.

Figure 5-11 shows the DCS, ISCC and Max-mixtures solutions for the dataset with both landmark and loop closure errors. There are several significant deviations from the truth in both the DCS and Max-mixtures solutions. These deviations were caused by visual aliasing between landmarks. In contrast, ISCC was able to generate a solution with a small amount of reconstruction error and is not impacted by the landmark aliasing.

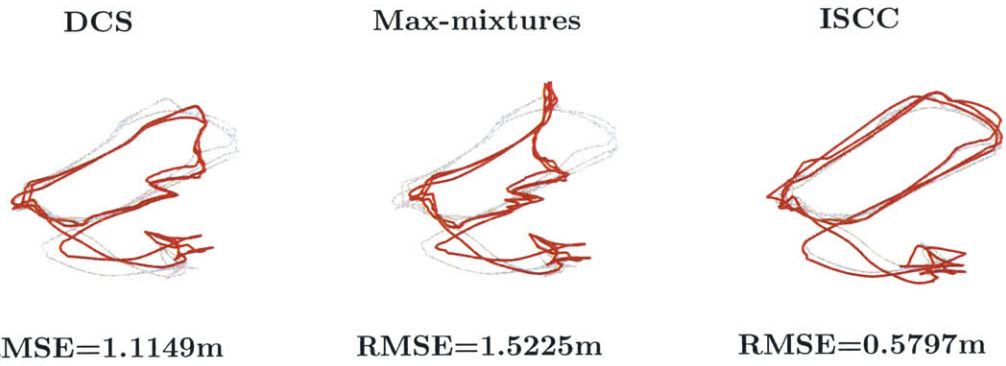


Figure 5-11: Experimental results for the ACL-loops dataset. Truth is shown in gray in each figure.

5.6 Conclusions

This chapter demonstrated that state-of-the-art robust SLAM algorithms can not provide robust solutions for datasets with incorrect landmark measurements. The root cause of these issues is that current robust algorithms focus on ensuring that the SLAM solution is locally consistent but do not require the solution to be globally consistent.

To address this issue, we developed a new formulation of the robust SLAM problem that requires both a globally and locally consistent solution. We also presented a novel incremental SLAM algorithm, ISCC, that can provide robust solutions when errors occur in landmark and loop closures measurements. Simulated and experimental results demonstrated that the new algorithm provides significantly better solutions than current robust SLAM algorithms when incorrect landmark measurements occur and can match the error performance of state-of-the-art algorithms when loop closure errors occur.

Chapter 6

Conclusion

This thesis has presented several novel algorithms for robust state estimation and mapping. These algorithms have demonstrated that robustness can be achieved by augmenting the state dynamics and measurement models with additional variables to model the impact of outliers and uncertainty in the model parameters. In addition, this thesis developed a formulation of the robust SLAM problem that has led to the first robust SLAM algorithm that can be applied to datasets with either loop closure or landmark data association errors.

Chapter 3 presented a novel filtering algorithm, the l_1-norm filter, that is robust to simultaneous outliers in both the state predictions and measurements. The key component of the algorithm is an augmented state-space model that includes additional variables to account for large deviations from the assumed system models. Given estimates of the additional variables, which can be found using an efficient convex optimization, their effect can easily be removed to provide robust state estimates.

Next, chapter 4 developed a robust SLAM algorithm, IM-SLAM, that estimates both the measurement information matrices and the map estimates simultaneously. By estimating the information matrices, not only can IM-SLAM provide robust estimates, it can also adapt the measurement information matrices to better match the true noise if the assumed noise models are incorrect.

Finally, Chapter 5 introduced a new formulation of the robust SLAM cost function and developed an incremental robust SLAM algorithm, ISCC, to solve new problem.

Simulated and experimental results demonstrated that ISCC can provide significantly more robust solutions that existing robust SLAM algorithms when landmark errors occur.

6.1 Future Work

There are still numerous interesting open problems to be addressed in robust navigation and mapping. This section provides a brief overview of some potential application areas and extensions of the work presented in the thesis.

6.1.1 Parameter Estimation for SLAM with non-Gaussian Noise Models

To date, most graph-based SLAM algorithms have exclusively considered Gaussian measurement noise models. However, in real-world applications the noise may not be well modeled by a Gaussian distribution and could be better represented using a different distribution. Recently, Rosen et al. have proposed an extension of standard graph-based SLAM algorithms for non-Gaussian distributions [123]. One challenge that could arise in these problems is parameter uncertainty in the non-Gaussian noise models. Applying the extended factor graph model that underlies IM-SLAM to these problems could provide a principled method of estimating the noise model parameters along with the poses. Moreover, if the non-Gaussian distributions are in the exponential family, selecting a conjugate prior factor for the parameters could lead to an efficient EM solution similar to IM-SLAM.

6.1.2 Global Convergence Guarantees and Verification for Robust SLAM

While a number of robust SLAM algorithms have been proposed, there is still a major research question to be addressed: given the output of a robust SLAM algorithm how can we verify that the solution is correct and optimal? Currently, this question can

only be addressed empirically by evaluating the algorithms on simulated datasets or datasets with known truth as was done in this thesis. But for datasets without truth information, usually a human operator must inspect the resulting SLAM solution to verify that it is correct. To fully enable autonomous SLAM systems, there must be a way to automatically verify that a solution is correct without human intervention.

One potential approach to the verification problem is to apply duality theory from nonlinear optimization [74, 124] to the robust SLAM cost function to calculate lower bounds for the robust SLAM problem. The dual solutions could also be used to calculate feasible solutions to calculate upper bounds on the global optimum of the robust SLAM cost function. If a solution from a robust SLAM algorithm fell outside of the bracket formed by the upper and lower bounds, it would indicate that the robust SLAM algorithm had converged to a local optimum rather than the global optimum. Duality theory has recently been applied to the 2D pose graph optimization, and it has been shown that the upper and lower bounds can be computed efficiently[125]. Additionally, in many cases the bounds are tight meaning that the bounds coincide with the global minimum of the problem.

6.1.3 Text-Spotting for SLAM

Recently researchers have considered how to use text in scenes as landmarks for the SLAM problem [126]. Repeated text (i.e. from exit signs, stop signs, etc.) could cause false positive data associations and ultimately lead to incorrect loop closure and landmark measurements. ISCC could be applied to this problem directly and could also be used as a means of determining when new landmarks that share text with existing landmarks should be declared. In this way, ISCC could be used to improve and augment the data association capabilities of a SLAM system that relies on text-spotting, providing a feedback loop between the SLAM front-end detecting the text and the SLAM back-end optimizing the solution.

Bibliography

[1] M. Cummins and P. Newman, "FAB-MAP: Probabilistic localization and mapping in the space of appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.

[2] A. Bonarini, W. Burgard, G. Fontana, M. Matteucci, D. G. Sorrenti, and J. D. Tardos, "RAWSEEDS: Robotics advancement through web-publishing of sensorial and elaborated extensive data sets," in *In proceedings of IROS'06 Workshop on Benchmarks in Robotics Research*, 2006.

[3] S. Ceriani, G. Fontana, A. Giusti, D. Marzorati, M. Matteucci, D. Migliore, D. Rizzi, D. G. Sorrenti, and P. Taddei, "Rawseeds ground truth collection systems for indoor self-localization and mapping," *Autonomous Robots*, vol. 27, no. 4, pp. 353–371, 2009.

[4] J. Leonard, D. Barrett, J. How, S. Teller, M. Antone, S. Campbell, A. Epstein, G. Fiore, L. Fletcher, E. Frazzoli, *et al.*, "Team MIT urban challenge technical report," tech. rep., MIT Computer Science and Artificial Intelligence Laboratory (CSAIL), 2007.

[5] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pp. 127–136, IEEE, 2011.

[6] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald, "Kintinuous: Spatially extended KinectFusion," tech. rep., MIT Computer Science and Artificial Intelligence Laboratory (CSAIL), 2012.

[7] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Computer Vision–ECCV 2014*, pp. 834–849, Springer, 2014.

[8] R. Kalman, "A new approach to linear filtering and prediction," *Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.

[9] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE Transactions on Robotics*, vol. 24, pp. 1365–1378, 2008.

[10] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering," in *2011 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3281–3288, 2011.

[11] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g^2o: A general framework for graph optimization," in *2011 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3607–3613, 2011.

[12] A. Bachrach, S. Prentice, R. He, P. Henry, A. S. Huang, M. Krainin, D. Maturana, D. Fox, and N. Roy, "Estimation, planning and mapping for autonomous flight using an RGB-D camera in GPS-denied environments," *International Journal of Robotics Research*, vol. 31, no. 11, pp. 1320–1343, 2012.

[13] Microsoft, "Kinect for Windows." `https://www.microsoft.com/en-us/kinectforwindows/`, 2015.

[14] J. Tanz, "Kinect hackers are changing the future of robotics," *Wired*, 2011.

[15] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. I–652–I–659 Vol.1, June 2004.

[16] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2, pp. 2161–2168, IEEE, 2006.

[17] D. Galvez-Lopez and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, pp. 1188–1197, October 2012.

[18] P. Newman and K. Ho, "SLAM-loop closing with visually salient features," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pp. 635–642, IEEE, 2005.

[19] M. Milford, "Vision-based place recognition: how low can you go?," *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 766–789, 2013.

[20] N. Sünderhauf and P. Protzel, "Towards a robust back-end for pose graph slam," in *2012 IEEE International Conference on Robotics and Automation (ICRA),*, pp. 1254–1261, IEEE, 2012.

[21] E. Olson and P. Agarwal, "Inference on networks of mixtures for robust robot mapping," *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 826–840, 2013.

[22] Y. Latif, C. Cadena, and J. Neira, "Robust loop closing over time for pose graph SLAM," *The International Journal of Robotics Research*, pp. 1611–1627, 2013.

[23] T. Fawcett, "An introduction to ROC analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[24] C. Masreliez and R. Martin, "Robust Bayesian estimation for the linear model and robustifying the Kalman filter," *IEEE Transactions on Automatic Control*, vol. 22, pp. 361–371, Jun 1977.

[25] M. Tanaka and T. Katayama, "Robust fixed-lag smoother for linear systems including outliers in the system and observation noises," *International Journal of Systems Science*, vol. 19, pp. 2243–2259, 1988.

[26] B. Kovačević, Z. Durovic, and S. Glavaški, "On robust Kalman filtering," *International Journal of Control*, vol. 56, no. 3, pp. 547–562, 1992.

[27] I. C. Schick and S. K. Mitter, "Robust recursive estimation in the presence of heavy-tailed observation noise," *Annals of Statistics*, vol. 22, pp. 1045–1080, 1994.

[28] K. D. Rao, M. N. S. Swamy, and E. I. Plotkin, "GPS navigation with increased immunity to modeling errors," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, pp. 2–11, 2004.

[29] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust Statistics: The Approach Based on Influence Functions.* Wiley, 1986.

[30] P. J. Huber and E. M. Ronchetti, *Robust Statistics.* Wiley, 2009.

[31] M. A. Gandhi and L. Mili, "Robust Kalman filter based on a generalized maximum-likelihood-type estimator," *IEEE Transactions on Signal Processing*, vol. 58, pp. 2509–2520, 2010.

[32] J. H. Manton, V. Krishnamurthy, and H. V. Poor, "James-stein state filtering algorithms," *IEEE Transactions on Signal Processing*, vol. 46, no. 9, pp. 2431–2447, 1998.

[33] C. Stein, "Inadmissibility of the usual estimator for the mean of a multivariate normal distribution," in *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, (Berkeley, Calif.), pp. 197–206, University of California Press, 1956.

[34] M. Grimble and A. El Sayed, "Solution of the H_∞ optimal linear filtering problem for discrete-time systems," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, pp. 1092–1104, Jul 1990.

[35] I. Yaesh and U. Shaked, "Game theory approach to optimal linear state estimation and its relation to the minimum H_∞-norm estimation," *IEEE Transactions on Automatic Control*, vol. 37, pp. 828–831, 1992.

[36] W. Shen and L. Deng, "Game theory approach to discrete H_∞ filter design," *IEEE Transactions on Signal Processing*, vol. 45, pp. 1092–1095, Apr 1997.

[37] D. Simon, *Optimal State Estimation: Kalman, H_∞, and Nonlinear Approaches.* Wiley, 2006.

[38] A. Giremus, J.-Y. Tourneret, and V. Calmettes, "A particle filtering approach for joint detection/estimation of multipath effects on GPS measurements," *IEEE Transactions on Signal Processing*, vol. 55, pp. 1275–1285, 2007.

[39] R. Kumar, D. Castañón, E. Ermis, and V. Saligrama, "A new algorithm for outlier rejection in particle filters," in *13th Conference on Information Fusion (FUSION)*, pp. 1 –7, july 2010.

[40] C. S. Maíz, E. M. Molanes-López, J. Míguez, and P. M. Djurić, "A particle filtering scheme for processing time series corrupted by outliers," *IEEE Transactions on Signal Processing*, vol. 60, pp. 4611 4627, September 2012.

[41] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics.* MIT Press, 2005.

[42] A. S. Willsky, "A survey of design methods for failure detection in dynamic systems," *Automatica*, vol. 12, pp. 601–611, 1976.

[43] J. Mattingley and S. Boyd, "Real-time convex optimization in signal processing," *IEEE Signal Processing Magazine*, vol. 27, pp. 50–61, May 2010.

[44] D. Kim, S.-G. Lee, and M. Jeon, "Outlier rejection methods for robust Kalman filtering," in *Future Information Technology* (J. J. Park, L. T. Yang, and C. Lee, eds.), vol. 184 of *Communications in Computer and Information Science*, pp. 316–322, Springer Berlin Heidelberg, 2011.

[45] S. Mohiuddin, D. E. Gustafson, and Y. Rachlin, "Mitigating the effects of GNSS multipath with a coded filter," in *Proceedings of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation*, pp. 2381–2394, 2011.

[46] M. Graham, T. Steiner, and J. How, "Robust vision-aided navigation in urban environments," in *AIAA Guidance Navigation and Control Conference 2013*, August 2013.

[47] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge University Press, 2004.

[48] D. Bertsekas, *Convex Optimization Theory.* Athena Scientific, 2009.

[49] S. Farahmand and G. Giannakis, "Robust RLS in the presence of correlated noise using outlier sparsity," *IEEE Transactions on Signal Processing*, vol. 60, pp. 3308 –3313, june 2012.

[50] S. Farahmand, G. B. Giannakis, and D. Angelosante, "Doubly robust smoothing of dynamical processes via outlier sparsity constraints," *IEEE Transactions on Signal Processing*, vol. 59, pp. 4529–4543, 2011.

[51] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[52] J.-A. Ting, A. D'Souza, and S. Schaal, "Automatic outlier detection: A Bayesian approach," in *IEEE International Conference on Robotics and Automation*, pp. 2489–2494, April 2007.

[53] J.-A. Ting, E. Theodorou, and S. Schaal, "Learning an outlier-robust Kalman filter," in *Machine Learning: ECML 2007*, vol. 4701 of *Lecture Notes in Computer Science*, pp. 748–756, Springer Berlin Heidelberg, 2007.

[54] J.-A. Ting, E. Theodorou, and S. Schaal, "A Kalman filter for robust outlier detection," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1514 –1519, 2007.

[55] S. Särkkä and A. Nummenmaa, "Recursive noise adaptive Kalman filtering by variational Bayesian approximations," *IEEE Transactions on Automatic Control*, vol. 54, pp. 596–600, March 2009.

[56] G. Agamennoni, J. I. Nieto, and E. M. Nebot, "An outlier-robust Kalman filter," in *IEEE International Conference on Robotics and Automation*, pp. 1551–1558, May 2011.

[57] G. Agamennoni, J. I. Nieto, and E. M. Nebot, "Approximate inference in state-space models with heavy-tailed noise," *IEEE Transactions on Signal Processing*, vol. 60, pp. 5024–5037, Oct 2012.

[58] N. Sunderhauf and P. Protzel, "Switchable constraints for robust pose graph slam," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1879–1884, IEEE, 2012.

[59] N. Sünderhauf, *Robust Optimization for Simultaneous Localization and Mapping*. PhD thesis, Chemnitz University of Technology, 2012.

[60] P. Agarwal, G. D. Tipaldi, L. Spinello, C. Stachniss, and W. Burgard, "Robust map optimization using dynamic covariance scaling," in *2013 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 62–69, 2013.

[61] E. Olson and P. Agarwal, "Inference on networks of mixtures for robust robot mapping," in *Proceedings of Robotics: Science and Systems*, (Sydney, Australia), July 2012.

[62] G. H. Lee, F. Fraundorfer, and M. Pollefeys, "Robust pose-graph loop-closures with expectation-maximization," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[63] M. Graham and J. How, "Robust simultaneous localization and mapping via information matrix estimation," in *ION/IEEE Position, Location and Navigation Symposium*, pp. 937–944, May 2014.

[64] Y. Latif, C. C. Lerma, and J. Neira, "Robust loop closing over time," in *Proceedings of Robotics: Science and Systems*, (Sydney, Australia), July 2012.

[65] Y. Latif, C. Cadena, and J. Neira, "Realizing, reversing, recovering: Incremental robust loop closing over time using the iRRR algorithm," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4211–4217, 2012.

[66] L. Carlone, A. Censi, and F. Dellaert, "Selecting good measurements via l_1 relaxation: a convex approach for robust estimation over graphs," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014.

[67] B. Anderson and J. Moore, *Optimal Filtering.* Prentice-Hall, 1978.

[68] R. Brown and P. Hwang, *Introduction to Random Signals and Applied Kalman Filtering.* Wiley, 1991.

[69] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation.* Wiley, 2001.

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

[71] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *Journal of Fluids Engineering*, vol. 83, no. 1, pp. 95–108, 1961.

[72] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Autonomous Robot Vehicles*, pp. 167–193, Springer, 1990.

[73] J. J. Leonard and H. F. Durrant-Whyte, *Directed sonar sensing for mobile robot navigation*, vol. 448. Kluwer Academic Publishers Dordrecht, 1992.

[74] J. Nocedal and S. J. Wright, *Numerical Optimization.* Springer Series in Operations Research and Financial Engineering, Springer, second ed., 2006.

[75] K. Levenberg, "A method for the solution of certain nonlinear problems in least squares," *Quarterly of Applied Mathematics*, vol. 2, pp. 164–168, 1944.

[76] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for Industrial & Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[77] M. J. Powell, "A new algorithm for unconstrained optimization," *Nonlinear programming*, pp. 31–65, 1970.

[78] D. M. Rosen, M. Kaess, and J. J. Leonard, "An incremental trust-region method for robust online sparse least-squares estimation," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 1262–1269, IEEE, 2012.

[79] D. M. Rosen, M. Kaess, and J. J. Leonard, "Rise: An incremental trust-region method for robust online sparse least-squares estimation," *IEEE Transactions on Robotics*, 2014.

[80] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[81] P. M. Newman, *On the Structure and Solution of the Simultaneous Localisation and Map Building Problem*. PhD thesis, University of Sydney, 1999.

[82] U. Frese, "A proof for the approximate sparsity of SLAM information matrices," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 329–335, IEEE, 2005.

[83] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[84] A. Blake and A. Zisserman, *Visual reconstruction*, vol. 2. MIT press Cambridge, 1987.

[85] E. D. Kaplan and C. J. Hegarty, eds., *Understanding GPS: Principles and Applications*. Artech House, 2006.

[86] B. D. Tapley, B. E. Schutz, and G. H. Born, *Statistical Orbit Determination*. Elsevier Academic Press, 2004.

[87] E. J. Candes and P. A. Randall, "Highly robust error correction by convex programming," *IEEE Transactions on Information Theory*, vol. 54, pp. 2829–2840, July 2008.

[88] Y. Eldar and G. Kutyniok, eds., *Compressed Sensing: Theory and Applications*. Cambridge University Press, 2012.

[89] E. J. Candes, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted l_1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 877–905, 2008.

[90] D. Donoho, "For most large underdetermined systems of linear equations, the minimal l_1 norm solution is also the sparsest solution," *Communications on Pure and Applied Mathematics*, vol. 59, no. 6, pp. 797–829, 2006.

[91] M. Figueiredo, R. Nowak, and S. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, pp. 586–597, Dec 2007.

[92] P. S. Maybeck, *Stochastic Models Estimation and Control*, vol. 1. Academic Press, 1979.

[93] D. S. Bernstein, *Matrix Mathematics: Theory, Facts, and Formulas.* Princeton University Press, 2005.

[94] E. van den Berg and M. P. Friedlander, "Probing the Pareto frontier for basis pursuit solutions," *SIAM Journal on Scientific Computing*, vol. 31, pp. 890–912, 2008.

[95] E. Candes and J. Romberg, "l_1-Magic : Recovery of sparse signals via convex programming," tech. rep., Caltech, 2005.

[96] S. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale l_1-regularized least squares," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, pp. 606–617, 2007.

[97] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *Proceedings of Signal Processing, Sensor Fusion, and Target Recognition VI*, 1997.

[98] J. Rehder, K. Gupta, S. Nuske, and S. Singh, "Global pose estimation with limited GPS and long range visual odometry," in *IEEE International Conference on Robotics and Automation*, pp. 627–633, 2012.

[99] G. Sibley, G. Sukhatme, and L. Matthies, "The iterated sigma point Kalman filter with applications to long range stereo," in *Proceedings of Robotics: Science and Systems*, (Philadelphia, USA), August 2006.

[100] A. Davison, I. Reid, N. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 1052–1067, June 2007.

[101] K. Konolige and M. Agrawal, "FrameSLAM: From bundle adjustment to real-time visual mapping," *IEEE Transactions on Robotics*, vol. 24, pp. 1066–1077, Oct 2008.

[102] H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Real-time monocular SLAM: Why filter?," in *IEEE International Conference on Robotics and Automation*, pp. 2657–2664, May 2010.

[103] T. J. Steiner, "A unified vision and inertial navigation system for planetary hoppers," Master's thesis, Massachusetts Institute of Technology, 2012.

[104] S. Roumeliotis, A. Johnson, and J. Montgomery, "Augmenting inertial navigation with image-based motion estimation," in *IEEE International Conference on Robotics and Automation*, vol. 4, pp. 4326–4333, 2002.

[105] E. Olson, *Robust and Efficient Robotic Mapping.* PhD thesis, Massachusetts Institute of Technology, 2008.

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

[107] G. Celeux and G. Govaert, "A classification EM algorithm for clustering and two stochastic versions," *Computation Statistics and Data Analysis*, vol. 14, no. 3, pp. 315–332, 1992.

[108] A. Gelman, J. B. Carlin, H. S. Stern, and D. R. Rubin, *Bayesian Data Analysis.* Chapman and Hall, 1995.

[109] R. Neal and G. E. Hinton, "A view of the EM algorithm that justifies incremental, sparse and other variants," in *Learning in Graphical Models*, pp. 355–368, 1998.

[110] N. Sunderhauf and P. Protzel, "Switchable constraints vs. max-mixture models vs. RRR - a comparison of three approaches to robust pose graph SLAM," in *2013 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2013.

[111] A. Howard and N. Roy, "The robotics data set repository (radish)," 2003.

[112] P. Agarwal, E. Olson, and W. Burgard, "Max-mixture - open source implementation with g2o." `https://github.com/agpratik/max-mixture`, 2012.

[113] Y. Latif, C. C. Lerma, and J. Neira, "RRR." `https://github.com/ylatif/rrr`, 2014.

[114] L. Carlone and A. Censi, "From angular manifolds to the integer lattice: Guaranteed orientation estimation with application to pose graph optimization," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 475–492, 2014.

[115] T. D. Barfoot and P. T. Furgale, "Associating uncertainty with three-dimensional poses for use in estimation problems," *IEEE Transactions on Robotics*, vol. 30, no. 3, pp. 679–693, 2014.

[116] R. Kannan and C. Monma, "On the computational complexity of integer programming problems," in *Optimization and Operations Research* (R. Henn, B. Korte, and W. Oettli, eds.), vol. 157 of *Lecture Notes in Economics and Mathematical Systems*, pp. 161–172, Springer Berlin Heidelberg, 1978.

[117] R. Kuemmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization." `https://github.com/RainerKuemmerle/g2o`, 2014.

[118] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, "Visual odometry and mapping for autonomous flight using an RGB-D camera," in *International Symposium on Robotics Research (ISRR)*, pp. 1–16, 2011.

[119] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3400–3407, IEEE, May 2011.

[120] M. Kaess, "Apriltags c++ library," March 2013.

[121] J. How, B. Bethke, A. Frank, D. Dale, and J. Vian, "Real-time indoor autonomous vehicle test environment," *IEEE Control Systems Magazine*, vol. 28, no. 2, pp. 51–64, 2008.

[122] R. Finman, T. Whelan, M. Kaess, and J. J. Leonard, "Toward lifelong object segmentation from change detection in dense RGB-D maps," in *2013 European Conference on Mobile Robots (ECMR)*, pp. 178–185, IEEE, 2013.

[123] D. M. Rosen, M. Kaess, and J. J. Leonard, "Robust incremental online inference over sparse factor graphs: Beyond the gaussian case," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 1025–1032, IEEE, 2013.

[124] D. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.

[125] L. Carlone and F. Dellaert, "Duality-based verification techniques for 2D SLAM," in *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015. To appear.

[126] H.-C. Wang, C. Finn, L. Paull, M. Kaess, R. Rosenholtz, S. Teller, and J. Leonard, "Bridging text spotting and SLAM with junction features," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015. submitted.