A STUDY OF SINGLE AND MULTI-ROBOT LOCALIZATION: A MANIFOLDS APPROACH

By

JOSEPH KNUTH

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2013

© 2013 Joseph Knuth

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my adviser Dr. Prabir Barooah. In addition to guiding me throughout my graduate studies, he established the initial direction of my research and then allowed me the freedom to consider the problem in my own way. I would like to thank my committee members, Dr. Warren Dixon, Dr. Carl D. Crane III, and Dr. Paul Robinson for their guidance in directing the course of my work. I am especially grateful to Dr. Robinson for the time he was willing to dedicate introducing me to the field of differential geometry, a branch of mathematics that proved critical in my work. I would also like to thank my fellow lab mates who have, through their example, constantly driven me to work harder and strive to accomplish more in my own work. Finally, I would like to thank my wife for her support and encouragement throughout my time here at the University of Florida.

TABLE OF CONTENTS

page

ACKNOWLEDGMENTS . . . 3

LIST OF TABLES . . . 7

LIST OF FIGURES . . . 8

ABSTRACT . . . 10

CHAPTER

1 INTRODUCTION . . . 13

1.1 Motivation . . . 13
1.2 Related Work: Single Robot Error Growth . . . 18
1.3 Related Work: Collaborative Localization . . . 19
1.4 Related Work: Outlier rejection . . . 22

2 ERROR GROWTH IN POSE ESTIMATION FROM DEAD RECKONING . . . 23

2.1 Problem Statement . . . 23
2.2 Main Results . . . 29
2.2.1 General Trajectories . . . 29
2.2.2 Discussion on Theorem 2.1 and its Proof . . . 31
2.2.3 Special 2-D Trajectories . . . 34
2.3 Reducing the Bias . . . 37
2.4 Simulation Verification . . . 38
2.4.1 3-D Simulation . . . 39
2.4.2 Straight-Line 2-D Trajectory . . . 39
2.4.3 Periodic Trajectory . . . 44
2.5 Experimental Verification . . . 46
2.5.1 Test Set-Up . . . 46
2.5.2 Test Results . . . 47
2.6 Reducing the Bias . . . 49
2.7 Summary . . . 51

3 DISTRIBUTED COLLABORATIVE 3D POSE ESTIMATION . . . 53

3.1 Problem Statement . . . 53
3.1.1 The Collaborative Localization Problem . . . 53
3.1.2 The Distributed Collaborative Localization Problem . . . 58
3.2 Centralized Collaborative Localization Algorithm . . . 59
3.2.1 Case A: Homogeneous Measurements (Relative Pose) . . . 59
3.2.2 Case B: Heterogeneous Measurements . . . 63
3.3 Distributed Algorithm . . . 65
3.4 Simulation Results . . . 67

3.4.1 All Measurements are of Relative Pose . . . 68
3.4.2 Heterogeneous Measurements . . . 71
3.5 Experimental Results . . . 72
3.6 Summary . . . 77

4 EXISTING METHODS OF COLLABORATIVE LOCALIZATION . . . 79

4.1 Unit Quaternions . . . 79
4.2 The Euclidean Pose Graph Optimization Algorithm . . . 80
4.3 The Implicit Extended Kalman Filter Algorithm . . . 84
4.3.1 The Implicit (Error-State) Extended Kalman Filter . . . 84
4.3.2 Collaborative Localization Using the Indirect EKF . . . 86
4.4 Simulation Results . . . 90
4.5 Summary . . . 93

5 MAXIMUM LIKELIHOOD ESTIMATES . . . 95

5.1 The ML Estimates . . . 95
5.2 Simulation studies . . . 99
5.2.1 Performance in a single experiment . . . 100
5.2.2 Effect of measurement noise, and comparison with state-of-the-art camera network localization algorithm . . . 102
5.3 Summary . . . 104

6 OUTLIER REJECTION ON POSE GRAPHS . . . 105

6.1 Problem statement . . . 105
6.2 Outlier Rejection Algorithm . . . 105
6.3 Sliding Window Approximation . . . 109
6.4 Simulation . . . 111
6.4.1 Justification of the log-normal distribution . . . 111
6.4.2 Outlier Rejection . . . 112
6.5 Summary and Future Work . . . 114

7 CONCLUSION AND FUTURE WORK . . . 116

7.1 Conclusion . . . 116
7.2 Future Work . . . 119

APPENDIX

A SINGLE ROBT: PROOFS . . . 121

B PRODUCT RIEMANNIAN MANIFOLDS . . . 130

B.1 Riemannian Manifolds . . . 130
B.2 Product Manifold . . . 134
B.3 Proof of Theorem 3.1 . . . 136
B.4 Gradient of the Cost Function (5–9) for Heterogeneous Measurements . . . 138

REFERENCES 141
BIOGRAPHICAL SKETCH 147

LIST OF TABLES

Table page

5-1 The parameters of the measurement pdfs used in all simulations 100

6-1 The average number of measurements, outliers, measurements falsely rejected, and outlier measurements falsely accepted when the outlier rejection algorithm is simulated as a preprocessing step of collaborative localization. 114

LIST OF FIGURES

Figure page

2-1 A figure to explain the notation . . . 24

2-2 The 3-D path used for the simulation in this chapter . . . 40

2-3 Comparison of theoretical bounds vs. simulation results . . . 40

2-4 Comparison of theoretical results for straight line vs simulation data . . . 42

2-5 Comparison of theoretical results for straight line vs simulation camera . . . 43

2-6 Comparison of theoretical results for periodic motion vs simulation . . . 45

2-7 The robots used in the experiments . . . 45

2-8 Schematic of the test set-up . . . 47

2-9 Experimental results for the 2-D circular motion case . . . 48

2-10 A randomly generated path in 2-D used to test the bias-reduction method. . . . 49

2-11 Performance of the bias-reduction method . . . 50

3-1 The graph corresponding to a group of 3 robots at time $k = 3$ . . . 55

3-2 The local measurement graphs for the robots . . . 66

3-3 The trajectories and neighbor relations used in all simulations. . . . 69

3-4 Comparison between the centralized and distributed CL algorithm. . . . 70

3-5 Simulation results studying localization accuracy v.s. number of robots . . . 71

3-6 Simulation results studying the localization accuracy v.s. measurement type . . . 73

3-7 Two Pioneer P3-DX robots equipped with cameras and targets. . . . 74

3-8 Experimental: A plot of the location of robot 1 using RPGO . . . 75

3-9 Experimental results studying the localization accuracy v.s. measurement type. 77

4-1 Simulation results comparing the RPGO and EPGO algorithms . . . 91

4-2 Simulation results comparing the RPGO and IEKF algorithms . . . 93

5-1 The magnitude of the difference between p_R and $\bar{p}_R$ . . . 98

5-2 A single realization for a group of robots using various measurement types . . . 101

5-3 Comparison of ML-RPGO and the algorithm proposed by Tron and Vidal for varying levels of noise . . . 103

6-1 A comparison between the estimated pdf for the values of $\hat{D}(e)$ as compared to the corresponding log-likelihood distribution pdf 112

6-2 Simulation results studying the effect of outliers and the outlier rejection algorithm on localization accuracy 113

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

A STUDY OF SINGLE AND MULTI-ROBOT LOCALIZATION: A MANIFOLDS APPROACH

By

Joseph Knuth

May 2013

Chair: Prabir Barooah
Major: Mechanical Engineering

We consider the problem of localizing autonomous robots when GPS is not available. Our work consists of two parts. First we examine how the estimation error grows with time when a mobile robot estimates its location from inter-time relative pose measurements without global position or orientation sensors. We show that in both 2-D or 3-D space, both the bias and variance of the position estimation error grows at most linearly with time (or distance traversed) asymptotically. The bias is crucially dependent on the trajectory of the robot. Exact formulas for the bias and the variance of the position estimation error are provided for two specific 2-D trajectories- straight line and periodic. Experiments with a P3-DX wheeled robot and Monte-Carlo simulations are provided to verify the theoretical predictions. A method to reduce the bias is proposed based on the lessons learned.

We next consider a group of cooperating robots attempting to localize without the use of GPS. We propose a algorithm for estimating the 3-D pose (position and orientation) of each robot with respect to a common frame of reference. This algorithm does not rely on the use of any maps, or the ability to recognize landmarks in the environment. Instead we assume that noisy relative measurements between pairs of robots are intermittently available, which can be any one, or combination, of the following: relative pose, relative orientation, relative position, relative bearing, and relative distance. The additional information about each robots pose provided by these

measurements are used to improve over self-localization estimates. The proposed method is based on solving an optimization problem in an underlying Riemannian manifold $(SO(3) \times \mathbb{R}^3)^{n(k)}$ by a gradient descent law. The proposed algorithm is easily applicable to 3-D pose estimation, can fuse heterogeneous measurement types, and can handle arbitrary time variation in the neighbor relationships among robots. This algorithm is further refined by choosing a distribution for the various measurement types and developing a maximum likelihood estimator for collaborative localization. Simulations show that the errors in the pose estimates obtained using this algorithm are significantly lower than what is achieved when robots estimate their pose without cooperation. Results from experiments with a pair of ground robots with vision-based sensors reinforce these findings. Additionally, the question of trade-offs between cost (of obtaining a certain type of relative measurement) vs. benefit (improvement in localization accuracy) for the various types of relative measurements is considered. Finally, a set of simulations is present in which our proposed algorithm is compared against two state of the art collaborative localization algorithms. This comparison shows that the proposed method performs better when the error in orientation measurements is large, or when the time interval between inter-robot measurements is large.

Finally, we propose an outlier rejection algorithm that functions as a preprocessing step for a pose graph collaborative localization algorithm, such as the one proposed earlier in this work, when all measurements are of the relative pose. Outliers are identified using only the information contained in the remaining relative measurements. In particular, no a priori distribution on the relative measurements is assumed, nor is any information about the absolute pose of the robots utilized. This outlier rejection algorithm exploits properties of pose measurements concatenated over simple cycles in the measurement graph to define an edge consistency cost such that large values are indicative of the presence of an outlier. A hypothesis test approach is then utilized to identify the set of likely outlying measurements. Simulations utilizing the

proposed outlier rejection algorithm are presented. The outlier rejection algorithm is shown to successfully identify up to 95% of the outliers in the scenario considered, and successfully mitigate the effect of outliers on collaborative localization.

CHAPTER 1
INTRODUCTION

1.1 Motivation

Localization without the use of GPS is a crucial capability for any autonomous robot, as there are many situations in which GPS signals are either unavailable, or only intermittently available. These include operation in urban canyons and tunnels, inside buildings, under water, and extra-planetary exploration. In such a situation, localization with respect to an initial position is typically performed using a combination of sensors that are used to measure relative motion between two successive time instants, and then chaining them together in a process known as dead reckoning. Inertial sensors (gyroscopes and accelerometers), vision-based sensors (cameras, LIDARs, etc) and joint encoders (in case of ground vehicles) are examples of sensors that can be used to obtain such measurements. When robots are operating in a cooperative group, measurements between robots may also be available, and when utilized, improve the accuracy of the dead reckoning estimates.

In this work we first examine the growth rate in the position estimation error of a single robot that cannot directly measure either its global position or its global orientation. Specifically, we analyze the bias and the variance of the error. The robot is equipped with sensors that allows it to measure the relative pose (position and orientation) between its coordinate frames at two successive time instants, but not sensors that can measure its absolute pose with respect to a global coordinate frame. That is, the robot may have sensors such as wheel odometers, IMUs, and cameras, but does not have sensors such as GPS and compasses. The absolute position has to be estimated from the noisy relative pose measurements though dead reckoning.

When relative pose measurements obtained from sensors are concatenated to form an estimate of the robot's position in a global frame, errors in individual measurements accumulate. Over long time horizons, the resulting location estimates may become quite

poor. Though this is well recognized, a rigorous analysis of the asymptotic growth rate seems to be lacking. Olson et. al. [1] states that without a global orientation sensor, the error grows super-linearly with distance, and presents experimental evidence supporting that statement. They further state that position estimation error will grow as $O(s^{3/2})$, where s is the distance travelled. A number of papers have claimed, without proof, that the position estimation error grows super-linearly with distance or time in the absence of an absolute orientation sensor [2–7]. It is also not clear what is meant by "error" in these references, whether it is the mean, variance or some other measure. A parametric statistical model of the 2-norm of the position estimation error is proposed in [8], whose parameters have to be fitted from measured error.

We show in this work that the asymptotic growth rates of both the bias and variance of the position estimation error are upper bounded by linear functions of time. Thus, even without an absolute orientation sensor, the error growth (both bias and variance) is at most linear. We also show that the variance growth rate is lower bounded by a linear function of time as well, if the variance of the translation measurement is sufficiently large. Our analysis also provides insight into the mechanism of error growth, particularly its bias, that does not seem to have been recognized so far. In particular, we show that the expected value of the robot's position estimate converges to a point irrespective of whether the robot stays in a bounded region for all time or not. An outcome of this fact is that the growth in the bias depends crucially on the type of path the robot traverses even though the robot does not have - and does not use - information about its trajectory. The bias will be bounded or unbounded depending only on whether the robot stays within a bounded region or not. In addition, the asymptotic trends for the bias hold even if the measurements of relative translation and rotation are unbiased. In fact, they hold even if the relative translation measurements are error-free. The bias in the translation measurements that arise from vision-based sensors has been a topic of

research [9, 10]. However, the fact that large position estimation bias may occur even when all measurements are unbiased has not been emphasized in the literature.

We next consider the case when a group of robots, each without access to GPS, attempt to localize through the use of dead reckoning. We propose a method for fusing measurements of various types to perform collaborative localization that improves localization accuracy over dead reckoning alone. As in the single robot case, we assume all robots are equipped with proprioceptive sensors (cameras, IMUs, etc.) allowing each robot to measure its change in pose between time steps. We refer to these noisy measurements as inter-time relative pose measurements. In the absence of any other information, each robot can perform localization through dead reckoning using these noisy measurements. We further assume that each robot is equipped with exteroceptive sensors, so that noisy relative measurements between robots may become available intermittently. These measurements, which we call inter-robot relative measurements, can be one (or any combination) of the following: relative pose, relative orientation, relative position, relative bearing, and relative distance, between a pairs of robots. We provide a method to perform collaborative localization by fusing the inter-time and inter-robot relative measurements to obtain an estimate of the absolute pose of every robot.

We provide both a centralized and distributed algorithm for collaborative localization. In the centralized algorithm, all the measurements are assumed to be instantly available to a central processor. In the distributed algorithm, each robot performs localization using measurements from on-board sensors and information exchanged with neighboring robots. The complexity of the computations performed by a robot is only a function of the number of its neighbors at any given time, not of the total number of robots. This makes the distributed algorithm scalable to arbitrarily large groups of robots. In addition, the communication requirements are small. At every update, a pair of neighboring robots needs to exchange only (i) the relative measurement between them

and (ii) their current pose estimates. Though the distributed algorithm is not equivalent to the centralized, simulations are provided that indicate a comparable increase is localization accuracy is obtained with each.

The ability to fuse heterogeneous relative measurements provides an important functionality. In particular, many papers (see [11–15]) require measurements of relative pose, i.e., position and orientation, between robot pairs. While relative position can be obtained through stereo vision or laser range finders, obtaining relative orientation is quite challenging. It often requires robots to be equipped with recognizable targets with known geometry; and even then, orientation measurements can be quite noisy. In fact, obtaining relative position measurements with stereo vision is also challenging due to the ever decreasing size of robots limiting the baseline available for stereo vision. However, even if only one camera of a robot sees a neighboring robot, a bearing measurement can be obtained. Or, radio frequency-based techniques such as TOA (time of arrival) measurements can be used to measure distance during wireless communication. With the ability to fuse all types of relative measurements, all available measurements can be utilized for improvement in localization accuracy. Though [16, 17] consider the case of heterogeneous measurements, their algorithms are limited to the 2-D case. Our work extends to 3-D the comparison between various measurement types initiated in [16, 17].

Both simulations and experimental results (with a pair of P3-DX robots) are presented. In each case, the Monte-Carlo method was used to empirically obtain the bias and standard deviation of estimation error by conducting multiple trials. Results show that the proposed collaborative algorithm substantially improves upon dead-reckoning even when the team consists of a small number of robots. An examination of improvement vs. measurement type provides a basis for deciding whether the cost of the sensors required for obtaining certain types of measurements are commensurate with the localization accuracy they provide.

We next compare the proposed algorithm (referred to as the RPGO algorithm) against two state of the art collaborative localization algorithms. The first based on the standard pose graph optimization algorithm, and the second utilizing an indirect extended Kalman filter. An important distinction between these state of the art algorithms and the algorithm presented here is the necessity for parameterization of the underlying space of rotations.

The method proposed here also belongs to the pose graph based approach: the estimation problem is formulated as an optimization defined by a graph, where nodes represent robot poses at various times and edges represent inter-time and inter-robot measurements. There is a subtle - but major - distinction between existing pose graph optimization methods, which we call Euclidean pose graph optimization (EPGO), and our proposed method. While existing methods use a vector space parameterization of orientation (such as the complex part of the unit quaternion representation) and then apply vector-space minimization techniques to search for the minima, we perform the optimization directly on the product Riemannian manifold in which the problem is naturally posed without relying on a specific parameterization. A gradient-descent method on the Riemannian manifold is then used for searching for the minima. The gradient descent algorithm is independent of the parameterization as well; any parameterization of the orientations can be used for numerical implementation without affecting the solution obtained. The advantages of doing so are discussed in detail in Section 4.2 (see Remark 4.1). A similar use of parameterization is necessary in the application of an indirect extended Kalman filter (IEKF) to the collaborative localization problem.

For both the EPGO and IEKF algorithms, simulations are provided study the sample bias and variance in position estimation error though Monte-Carlo experiments. Scenarios in which the proposed algorithm performs best are identified. In particular, in the configurations considered, the RPGO algorithm outperforms the EPGO algorithm

when the measurement noise is large, and outperforms the IEKF algorithm when the time between inter-robot measurements is large.

The collaborative localization algorithm is then extended by identifying a distribution for each measurement type and utilizing the maximum likelihood estimate to localization the group of robots. The non-vector space nature of both the orientation and bearing measurements is considered and distributions chosen with respect to the appropriate measure on the given manifolds. Simulations are presented indicating a marked improvement in localization accuracy when the maximum-likelihood algorithm is used.

The collaborative localization algorithms presented in the work are analogous to a least squares optimization problem. Since lest squares optimization methods are often sensitive to outliers, identifying any outliers in the measurements can provide a considerable increase in performance. In general, the collaborative localization problem does not provide prior distributions for the measurements, and thus outliers must be identified based on consistency with other related measurements. In this work we specifically consider the case when all measurements are of the relative oration, or all are of the relative pose. In such a situation, simple cycles found in the measurement graph describing the localization problem can be used to provide a measure on the consistency of all measurements. We associate those edge measure's with a log-normal distribution, and utilize a standard Gaussian outlier rejection algorithm to identify the set of likely inconsistent measurements.

1.2 Related Work: Single Robot Error Growth

The papers by Smith and Cheesman [18], Su and Lee [19], and Wang and Chirikjian [20] derived recursive expressions for the covariance of the pose estimation error by assuming the errors are small, so that a first order approximation of the BCH (Baker-Campbell-Hausdorff) formula is valid. Recently, Wang and Chirikjian [21] developed a recursive formula for the covariance of the pose estimation error that retains the second order terms in the BCH formula. The paper [22] examines dead

recking error's probability density function for non-holonomic robots in 2-D. The work that comes closest to ours in spirit is [8], which proposed a parametric statistical model of the 2-norm of the position estimation error. Some of the parameters have to be fitted from measured error. However, the works mentioned above do not analyze asymptotic behavior of the error's mean and variance.

A related body of literature deals with problem of developing state estimation techniques for systems whose states, as well as the noisy measurements, are in $SO(3)$ or $SE(3)$ (see [23, 24] and references therein). The problem of position estimation of a mobile robot with noisy relative pose measurements between successive frames - one that is central to this work- falls into this category. However, our aim is not to develop an estimation technique, but to examine the growth of error in the position estimate when successive noisy relative pose measurements are chained together to obtain a global pose estimate.

1.3 Related Work: Collaborative Localization

Collaborative localization has been considered in the context of simultaneous localization and mapping (SLAM). In one class of approaches, robots exchange local maps which are aligned and merged to improve robots' location estimates as well as to improve the maps; see [25, 26] and references therein. This requires the ability to identify common features in distinct maps generated by the robots. A method based on pose graphs is developed in [27] that does not require exchange of maps. In [28], robots exchange images and an implicit extended Kalman filter is used to update the state of each robot when a common feature is found.

Recognizing common landmarks in distinct maps is often challenging. In addition, exchanging image data or maps between robots requires high bandwidth communication. A second body of work therefore considers the collaborative localization problem as one in which only relative measurements (of pose, position, orientation etc.) between pairs of robots are obtained and used to improve localization accuracy over self-localization.

Our work falls into this category, which we term collaborative localization from relative measurements.

Collaborative localization from relative measurements was first considered in [29], in which a set of robots were divided into two groups such that only one group was allowed to move at a time. The stationary group functioned as landmarks for the moving group. It was shown that for a group of three robots, this method of localization led to lower error than that from dead reckoning. Subsequent work allowed all robots to move simultaneously, and collaborative localization was often accomplished through the use of a Kalman filter (KF) or its extensions. In [30] the author models the problem as a linear estimation problem by assuming each robot's orientation is known exactly. Measurements of position between robots are then fused with the dead reckoning position estimates for each robot through the use of a Kalman filter. The author further proves that if the sensing and positioning errors are independent, application of their algorithm will always improve individual robots estimates. In [12], the authors consider the collaborative localization problem when noisy measurements of the absolute orientations are available. A Kalman filter is used to fuse the proprioceptive and exteroceptive measurements using linearized update equations; the computations are distributed among all robots. In [31] the author provides a distributed method of selecting measurements between robots that minimize the localization error for the entire team. Simulations are provided that show selecting optimal measurements provides smaller localization error over the case of random selection.

In [32] the authors consider a group of micro air vehicles (MAVs) in level flight that have on-board IMUs and are capable of measuring the relative position of neighboring MAVs. An EKF is used to fuse measurements, while a study of observability is performed providing insight into what measurements are necessary to localize the MAVs in 2-D. In [15] the authors propose a method for estimating the 2-D pose for a set of collaborating agents by decomposing the problem into multiple phases: first the absolute

orientations are estimated using relative orientation measurements, then these are used along with relative position measurements to obtain absolute position estimates, and finally an improved estimate of the absolute orientations are obtained from the position estimates. The estimation problem in each phase are solved using linear estimation techniques. The authors also propose a distributed algorithm using iterative updates and prove that the estimates found using the distributed algorithm converge asymptotically to those found using the centralized algorithm. In [33] the authors consider the problem of estimating the 2-D pose for a set of anonymous road vehicles. Each vehicle uses an extended Kalman filter (EKF) to estimate its own position and velocity along with the position and velocity for every vehicle it has seen recently. Identification of vehicles is performed using the Mahalanobis distance to compare measured and estimated pose, equating two vehicles only if the distance falls below some threshold. The final estimate of the pose for a given vehicle is obtained by fusing the state vectors for all neighboring vehicles.

A centralized algorithm for collaborative localization is presented in [14] that is applicable to both 2-D and 3-D. A nonlinear cost function is proposed that, when minimized, provides a maximum likelihood estimate of the pose for each robot. The nonlinear optimization problem is solved using standard optimization techniques. In [34] an algorithm for collaborative localization in 3-D by linearizing the relation between the pose measurements is presented. Results from a Monte-Carlo simulation are provided showing improved localization accuracy with increasing number of robots.

Leung et al. considers the problem of equivalency between centralized and decentralized collaborative localization algorithms [35]. No specific algorithm is proposed. Instead they allow for an arbitrary algorithm to fit within their framework. They show that given the Markov property holds for the estimated states, the time at which past measurements can be discarded can be determined in a decentralized manner such that the distributed algorithm will be equivalent to the centralized algorithm.

The aforementioned papers assume relative pose measurements between robots. In many applications, these may be difficult or expensive to obtain. To that end, [16] extends the particle filter approach for 2-D collaborative localization when robots take turns moving to include not only relative pose measurements, but also measurements of the relative orientation, position, bearing, and distance. Monte-Carlo simulations are used to compare the effect different types of inter-robot measurements, along with the number of robots, has on localization accuracy. In [17] the authors expand the use of the EKF to estimate the 2-D pose using measurements of the bearing, distance, or orientation between robots. Simulations are provided that indicate any combination of these measurements, with the exception of distance alone, provides sufficient additional information to improve over dead reckoning.

1.4 Related Work: Outlier rejection

Many collaborative localization algorithms attempt to detect data association errors or falsely identified loop closure events. In particular, [36–40] each attempt to perform simultaneous localization and mapping while remaining robust to outliers. However in each of the works mentioned above, outlier identification is not grantees, so a pre-processing algorithm to identify outliers is beneficial. In this work we present such a pre-processing algorithm.

The problem of identifying outliers prior to utilizing localizing was also considered by [41]. The authors utilize a least squares approach to pose graph optimization, with the inclusion of an additional state for each measurement indicating weather that measurement should be included. The optimal solution to the modified pose graph problem indicates what measurements are likely to be outliers.

CHAPTER 2
ERROR GROWTH IN POSE ESTIMATION FROM DEAD RECKONING

In this chapter we examine the growth rate in the position estimation error of a robot that cannot directly measure either its global position or its global orientation. Specifically, we analyze the bias and the variance of the error. The robot is equipped with sensors that allows it to measure the relative pose (position and orientation) between its coordinate frames at two successive time instants, but not sensors that can measure its absolute pose with respect to a global coordinate frame.

The rest of the chapter is organized as follows. Section 2.1 precisely formulates the problem under study, and Section 2.2 states the main results. A method for reducing the bias in position estimation error is presented in Section 2.3. Simulation verification is presented in Section 2.4 and experimental verification is presented in Section 2.5. Finally, simulations studying the bias-reduction algorithm presented in Section 2.3 are presented in Section 2.6.

2.1 Problem Statement

We measure time with a discrete index $k = 0, 1, \ldots$. Sensors used for relative localization of autonomous vehicles yield an estimate of the position and orientation of the vehicle at time k relative to that in the previous time instant, $k-1$. That is, they produce an estimate of the relative pose between frames attached to the robot at two successive time instants. Let $\mathbf{R}_{k+1}^{k}$ be the rotation between the local frames attached to the robot's body at time k and $k+1$. That is, if $\mathbf{u}^k$ is a vector expressed in the vehicle's frame at time k and $\mathbf{u}^{k-1}$ is the same vector expressed in the vehicle's frame at time $k-1$, then $\mathbf{u}^{k-1} = \mathbf{R}_{k}^{k-1}\ \mathbf{u}^k$. This notation is adopted from [42]. We will refer to the frame that is attached to the vehicle at time k as the "frame k". Similarly, let $\mathbf{t}_{i,j}^{k}$ be the relative translation from the frame i to the frame j, expressed in the frame k. The rotation $\mathbf{R}_{k}^{k-1} \in \mathrm{SO}(d)$ is usually expressed as a $d \times d$ matrix for $d \in \{2, 3\}$, while $\mathbf{t}_{i,j}^{k}$ is a vector in $\mathbb{R}^d$. Without loss of generality, the coordinate frame that is attached to the

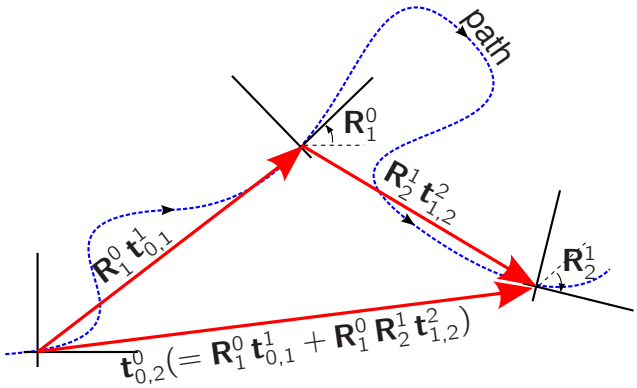


Figure 2-1. A figure to explain the notation: a robot's path (shown in dashed blue line) in 2-D and associated relative poses between time instants. $\mathbf{t}_{k-1,k}^0$ is the translation between the frames $k-1$ and k, expressed in the global frame 0, and $\mathbf{t}_{k-1,k}^k$ is the same vector expressed in the local frame k. The matrix $\mathbf{R}_k^0$ is the rotation between frame 0 and k, so that $\mathbf{R}_k^0\mathbf{t}_{k-1,k}^k$ is the translation from $k-1$ to k expressed in the global frame 0.

robot's body at the initial time $k=0$ is used as the global coordinate frame. We denote the rotation from frame k to the global coordinate frame (frame 0) by $\mathbf{R}_k^0$. Similarly, the translation from frame $k-1$ to k expressed in the global coordinate frame is denoted by $\mathbf{t}_{k-1,k}^0$. The position of the robot at any given time n is the vector $\mathbf{t}_{0,n}^0$.

With relative pose sensors such as inertial sensors, cameras, and wheel odometers, the measurements available at time k are estimates of the relative translation from $k-1$ to frame k expressed in frame k, i.e., of $\mathbf{t}_{k-1,k}^k$, and the rotation between the frames $k-1$ and k. i.e., of $\mathbf{R}_k^{k-1}$. The translation from $k-1$ to k, for $k\geq 1$, expressed in the global coordinate frame is

$$\mathbf{t}_{k-1,k}^0=\mathbf{R}_k^0\,\mathbf{t}_{k-1,k}^k, \qquad \text{where} \qquad \mathbf{R}_k^0=\mathbf{R}_1^0\,\mathbf{R}_2^1\ldots\mathbf{R}_k^{k-1}.$$

An example of a robots path along with its corresponding relative pose measurements can be seen in Figure 2-1. Estimates are denoted by hats on top of the corresponding symbols, and errors by tildes, so that $\hat{\mathbf{R}}_k^{k-1}$ and $\hat{\mathbf{t}}_{k-1,k}^k$ are the noisy estimates of $\mathbf{R}_k^{k-1}$

and $\mathbf{t}_{k-1,k}^{k}$, and the corresponding errors $\tilde{\mathbf{R}}_{k}^{k-1}$ and $\tilde{\mathbf{t}}_{k-1,k}^{k}$ are defined as

$$\begin{aligned} \tilde{\mathbf{R}}_{k}^{k-1} &:= (\mathbf{R}_{k}^{k-1})^{-1}\hat{\mathbf{R}}_{k}^{k-1}, \\ \tilde{\mathbf{t}}_{k-1,k}^{k} &:= \hat{\mathbf{t}}_{k-1,k}^{k} - \mathbf{t}_{k-1,k}^{k}. \end{aligned} \tag{2–1}$$

The absolute position of the robot at time k is determined by expressing the relative position measurements at each time step in the global coordinate frame and adding them together. The measurement of the translation from frame $k-1$ to k expressed in the global coordinate frame, which is denoted by $\hat{\mathbf{t}}_{k-1,k}^{0}$, is

$$\hat{\mathbf{t}}_{k-1,k}^{0} := \hat{\mathbf{R}}_{k}^{0}\,\hat{\mathbf{t}}_{k-1,k}^{k}, \tag{2–2}$$

where $\hat{\mathbf{R}}_{k}^{0}$ is an estimate of $\mathbf{R}_{k}^{0}$, which is computed from the relative rotation estimates as

$$\hat{\mathbf{R}}_{k}^{0} = \prod_{i=1}^{k} \hat{\mathbf{R}}_{i}^{i-1}. \tag{2–3}$$

Finally, the estimate of the position at time n in the global coordinate frame 0 is obtained by adding the relative translation estimates after transforming them to frame 0:

$$\hat{\mathbf{t}}_{0,n}^{0} := \sum_{k=1}^{n} \hat{\mathbf{t}}_{k-1,k}^{0}. \tag{2–4}$$

The error between the estimated position and the true position at time n is

$$\mathbf{e}(n) := \mathbf{t}_{0,n}^{0} - \hat{\mathbf{t}}_{0,n}^{0}. \tag{2–5}$$

The goal of this chapter is to study how the mean and covariance of the position estimation error $\mathbf{e}(n)$ scales with the time index n. If the robot's speed is upper and lower bounded by two constants, then the asymptotic trends with time are equivalent to those with distance travelled. Therefore we only study scaling with the time index n.

The straight-forward dead-reckoning formula (2–4) may not be used in practice. Typically a filtering-based algorithm is used to fuse relative pose measurements with the predictions of a model of the robot's motion. There are many variations possible in

terms of assumed model, states and input measurements; see [43] for a comparison among some of them. This renders examining the mechanism of error propagation and establishing asymptotic growth rates of such algorithms intractable. Therefore we adopt the simple dead-reckoning model that still captures the essential features of localization from relative pose measurements. We wish to emphasize that the estimation error resulting from the estimation method described above will have the same asymptotic trend as that of a filtering technique that uses a kinematic model of the robot motion. The reason is that a kinematic model essentially produces an independent noisy measurement of the relative pose. Thus, our investigations are useful in analyzing asymptotic performance of a wider class of estimation techniques. One situation where our model is not appropriate is when loop-closure is used to augment localization [44]. We focus on situations where loop closure is not applicable, e.g., an unmanned aerial vehicle flying in an expansive environment so that it may not come back to its earlier positions.

To state the assumptions on measurement error statistics, we establish a few conventions. A rotation matrix $R \in SO(3)$, where the special orthogonal group $SO(3)$ is the set of 3×3 real orthogonal matrices with unit determinant, can be represented by the exponential map: $R = e^{\omega^s}$, where ω^s is the 3×3 skew-symmetric matrix corresponding to the vector $\omega \in \mathbb{R}^3$ [45, Chapter 2]. A matrix in $SO(2)$ is uniquely specified by an angle $\theta \in [-\pi,\ \pi)$. A random rotation matrix $\mathbf{R} \in SO(3)$ (resp. $SO(2)$) can therefore be specified by a random vector $\boldsymbol{\omega} \in \mathbb{R}^3$ (resp. a scalar r.v. θ). We say that two random rotation matrices $\mathbf{R}_1, \mathbf{R}_2 \in SO(3)$ are independent if their corresponding $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$ are independent random vectors. For $SO(2)$, independence of rotations is defined as the independence of the scalar random variables θ_1, θ_2 that uniquely determine them. If $\mathbf{R}_1$ and $\mathbf{R}_2$ are independent, every entry of the matrix $\mathbf{R}_1$ is independent of every entry of $\mathbf{R}_2$. Similarly, we say that a rotation $\mathbf{R}_1 \in SO(3)$ (resp., $SO(2)$) and a random vector

$\mathbf{t} \in \mathbb{R}^3$ (resp. $\mathbb{R}^2$) are independent if $\boldsymbol{\omega}_1$ (resp., θ) and $\mathbf{t}$ are independent. In this case, too, every entry of $\mathbf{t}$ is independent of every entry of $\mathbf{R}$.

In this chapter, we use $\mathsf{E}[\mathbf{R}]$ (for a random rotation matrix $\mathbf{R}$) to denote the matrix whose i,j-th entry is $\mathsf{E}[(\mathbf{R})_{i,j}]$, i.e., the expected value of the i,j-th entry of $\mathbf{R}$. As a direct result of this convention, if $\mathbf{R}_1 \in SO(d)$ is independent of $\mathbf{R}_2 \in SO(d)$ and of $\mathbf{t} \in \mathbb{R}^3$, then $\mathsf{E}[\mathbf{R}_1\mathbf{R}_2] = \mathsf{E}[\mathbf{R}_1]\,\mathsf{E}[\mathbf{R}_2]$ and $\mathsf{E}[\mathbf{R}_1\mathbf{t}] = \mathsf{E}[\mathbf{R}_1]\,\mathsf{E}[\mathbf{t}]$.

In the sequel, $\mathrm{Tr}\,[\,\cdot\,]$ stands for trace of a matrix, and $\|\cdot\|_q$ denotes the (induced) q-norm of a (matrix) vector. When the subscript is omitted, it denotes the 2-norm.

We state the following assumptions for use in the rest of the chapter.

Assumption 2.1. *1. The robot's speed is uniformly bounded. More specifically, there exists a constant $\tau > 0$ such that $\|\mathbf{t}^k_{k-1,k}\| \leq \tau$.*

2. *The translation measurement errors $\tilde{\mathbf{t}}^k_{k-1,k}$ form a sequence of independent random vectors, with mean $\mathbf{b}_k := \mathsf{E}[\tilde{\mathbf{t}}^k_{k-1,k}]$ and covariance $\mathbf{P}_k := Cov(\tilde{\mathbf{t}}^k_{k-1,k}, \tilde{\mathbf{t}}^k_{k-1,k})$ that are uniformly bounded. That is, there exist scalar constants b, $\underline{p}$, $\overline{p}$ such that $0 \leq \|\mathbf{b}_k\| \leq b$ and $0 \leq \underline{p} \leq \mathrm{Tr}\,[\mathbf{P}_k] \leq \overline{p} < \infty$ for all k.*
3. *The rotation measurement errors $\tilde{\mathbf{R}}^{k-1}_k$ form a sequence of independent random matrices. The rotation and translation measurement errors $\tilde{\mathbf{R}}^{j-1}_j$ and $\tilde{\mathbf{t}}^k_{k-1,k}$ are mutually independent if $j \neq k$, and possibly dependent when $j = k$, with $\mathsf{E}[\tilde{\mathbf{R}}^{k-1}_k\,\tilde{\mathbf{t}}^k_{k-1,k}] =: \boldsymbol{\rho}_k \in \mathbb{R}^d$. There exists a scalar ρ such that $\|\boldsymbol{\rho}_k\| \leq \rho$ for all k.*
4. *The relative translation measurement errors $\{\tilde{\mathbf{t}}^k_{k-1,k}\}_{k=1}^{\infty}$ are uniformly absolutely integrable, i.e., there exists a scalar β so that $\beta_k \leq \beta < \infty$ for all k where $\beta_k := \mathsf{E}\,\|\tilde{\mathbf{t}}^k_{k-1,k}\|$.*
5. *The rotation measurement errors $\tilde{\mathbf{R}}^{k-1}_k$ are identically distributed, so that each $\tilde{\mathbf{R}}^{k-1}_k$ has the same distribution as that of some matrix $\tilde{\mathbf{R}} \in SO(d)$, $d \in \{2,3\}$. Moreover, $\tilde{\mathbf{R}}$ is not degenerate, i.e., its pdf (probability distribution function) is not concentrated on a set of measure zero.*

Apart from the assumptions on independence of measurement errors, the other assumptions, in particular those on the existence of the parameters τ, b, $\underline{p}$, $\overline{p}$, ρ, β, are trivially satisfied in any practical scenario. Finiteness of the displacement τ and bias norm b are easy to see; the parameters $\underline{p}$ and $\overline{p}$ are simply the lower and upper bounds on the eigenvalues of $\mathbf{P}_k$. The d-dimensional vector $\boldsymbol{\rho}_k$ is a measure of the correlation between the translation and rotation measurements, and the parameter ρ is an upper bound on the magnitude of the correlation. We allow the relative translation and rotation measurement errors at a particular time instant to be statistically dependent, since this may happen if there is overlap between the sensor suite used to obtain these two measurements. The parameter β is akin to an upper bound on the sum of bias and variance of the translation measurement error. To see this, consider not $\mathsf{E}[\|\tilde{\mathbf{t}}^k_{k-1,k}\|]$, but $\mathsf{E}[\|\tilde{\mathbf{t}}^k_{k-1,k}\|^2]$, which is the trace of the second moment of translation measurement error $\tilde{\mathbf{t}}^k_{k-1,k}$. Since the second moment is the sum of covariance and first moment, an upper bound on $\mathsf{E}[\|\tilde{\mathbf{t}}^k_{k-1,k}\|^2]$ is also an upper bound on sum of mean and variance (more precisely, on $\|\mathbf{b}_k\|^2 + \mathsf{Tr}\,[\mathbf{P}_k]$) of the translation measurement error.

The following technical result is crucial for the main results of this chapter and will be required for the subsequent discussions. We therefore state it here; the proof is provided in the Appendix.

Proposition 2.1. *Let* $\mathbf{R}$ *be a random rotation matrix with distribution defined over* $SO(d)$*,* $d \geq 2$ *and let* $\mathsf{E}[\mathbf{R}]$ *be the* $d \times d$ *matrix whose* i,j*-th entry is the expected value of the* $i,$*-th entry of* $\mathbf{R}$*. We have* $\|\mathsf{E}[\mathbf{R}]\| \leq 1$*, and the inequality is strict if the distribution of* $\mathbf{R}$ *is not degenerate*[1] *.* □

[1] Recall that we say the distribution of $\mathbf{R}$ is degenerate if its pdf is 0 everywhere except possibly in a set of measure 0.

Due to its usefulness in later discussions, we define

$$\overline{R} := \mathrm{E}[\tilde{\mathbf{R}}]. \tag{2–6}$$

Recall that $\tilde{\mathbf{R}}$ is a rotation matrix that has the same distribution as all the rotation errors $\tilde{\mathbf{R}}_{k+1}^{k}$, $k = 1, \ldots$. It follows from Proposition 2.1 that under Assumption 2.1,

$$1 > \gamma := \|\overline{R}\|. \tag{2–7}$$

According to the convention used in this chapter, in general $\mathrm{E}[\mathbf{R}] \notin SO(d)$ even if $\mathbf{R} \in SO(d)$. It is important that the notation $\mathrm{E}[\mathbf{R}]$ is not to be understood as the expectation of the random variable $\mathbf{R}$ with a distribution defined over $SO(d)$, which we denote by μ_{R}, so that $\mu_{\mathrm{R}} \in SO(d)$. We call μ_{R} the "Lie-group mean" of $\mathbf{R}$. We call an estimate $\hat{\mathbf{R}}$ of a true rotation $\mathbf{R}$ unbiased if $\mu_{\hat{\mathrm{R}}} = \mathbf{R}$. A result of the adopted convention is that for an unbiased estimator $\hat{\mathbf{R}}$ of $\mathbf{R}$, in general $\mathrm{E}[\hat{\mathbf{R}}] \neq \mathbf{R}$. The reason the quantity $\mathrm{E}[\mathbf{R}]$ is more useful for this work than μ_{R} is that when $\mathbf{R}$ and $\mathbf{t}$ are independent, $\mathrm{E}[\mathbf{R}\mathbf{t}] = \mathrm{E}[\mathbf{R}]\,\mathrm{E}[\mathbf{t}]$ but in general $\mathrm{E}[\mathbf{R}\mathbf{t}] \neq \mu_{\mathrm{R}}\,\mathrm{E}[\mathbf{t}]$. The bias in translation measurements obtained from vision-based sensors has been the subject of research [9, 10]. The bias in rotation measurement, on the other hand, seem to have drawn limited attention. In [9], the error in 3-D rotation is described in terms of the corresponding Euler angles, and bias in rotation is also defined in terms of the bias in the Euler angles. An alternate definition of 3-D rotation error in terms of a 3-vector (involving angle and axis of rotation) is used in [46], but the question of its bias is not discussed.

2.2 Main Results

2.2.1 General Trajectories

Before stating the result, we review the asymptotic O, Ω, Θ notation. For two scalar-valued functions $f(n), g(n)$ taking non-negative integer arguments, the notation $f(n) = O(g(n))$ means there exists a positive integer n_1 and a positive constant c_1 so that $f(n) \leq c_1 g(n)$ for all $n \geq n_1$. The notation $f(n) = \Omega(g(n))$ means there exists a

positive integer n_2 and a positive constant c_2 so that $f(n) \geq c_2 g(n)$ for all $n \geq n_2$. The notation $f(n) = \Theta(g(n))$ means both $f(n) = \Omega(g(n))$ and $f(n) = O(g(n))$ hold.

Theorem 2.1. *Consider a robot moving in a 2-D or 3-D Euclidean space that performs position estimation from relative pose measurements as described in Section 2.1. Under Assumption 2.1, the following statements hold, where $\tau, \beta, b, \rho, \underline{p}, \overline{p}$ are parameters defined in Assumption 2.1 and γ is defined in (2–7).*

1. *The bias in the position estimation error satisfies $\| \mathrm{E}[\mathbf{e}(n)] \| = O(n)$. In particular,*

$$\max \left\{ 0, \|\mathbf{t}_{0,n}^0\| - \frac{1-\gamma^n}{1-\gamma} (\gamma\tau + \beta) \right\} \leq \| \mathrm{E}[\mathbf{e}(n)] \| \leq \|\mathbf{t}_{0,n}^0\| + \frac{1-\gamma^n}{1-\gamma} (\gamma\tau + \beta) . \tag{2–8}$$

2. *The position error covariance satisfies $\mathrm{Tr}\left[Cov(\mathbf{e}(n), \mathbf{e}(n))\right] = O(n)$, with upper bound given by*

$$\mathrm{Tr}\left[Cov(\mathbf{e}(n), \mathbf{e}(n))\right] \leq \overline{\alpha_0} \left(\frac{1+\gamma}{1-\gamma} \, n \right), \tag{2–9}$$

where

$$\overline{\alpha_0} = \max \left\{ (\tau^2 + 2\tau b + \overline{p} + b^2), (\tau + \frac{\beta}{\gamma})(\tau + b) \right\}. \tag{2–10}$$

If furthermore

$$\underline{p} \geq 2b\tau + \tau^2 + 2\frac{(\tau + \rho/\gamma)(\tau + b)}{1-\gamma}, \tag{2–11}$$

then

$$\mathrm{Tr}\left[Cov(\mathbf{e}(n), \mathbf{e}(n))\right] = \Theta(n). \qquad \square$$

Before discussing the implications of the theorem, we present a result in the form of a lemma that is useful in both the discussion and the proof of the theorem. The proof of the lemma is provided in the Appendix.

Lemma 1. *Under Assumptions 2.1, the first and second moment of the position estimate satisfies*

$$\| \mathrm{E}[\hat{\mathbf{t}}^0_{0,n}] \| \leq \frac{1-\gamma^n}{1-\gamma}(\gamma\tau+\beta), \qquad \mathrm{E}[\|\hat{\mathbf{t}}^0_{0,n}\|^2] \leq \overline{\alpha_0}\frac{1+\gamma}{1-\gamma}\, n,$$

where $\overline{\alpha_0}$ *is defined in* (2–10)*. Moreover, if condition* (2–11) *is satisfied, then we have* $\mathrm{E}[\|\hat{\mathbf{t}}^0_{0,n}\|^2] = \Theta(n)$*.* □

2.2.2 Discussion on Theorem 2.1 and its Proof

Theorem 2.1, and in particular the upper bound in (2–8), shows that if the robot's motion is confined to a bounded region, then the bias in the position estimation error stays uniformly bounded by a constant: $\| \mathrm{E}[\mathbf{e}(n)] \| = O(1)$. If the robot moves with a constant speed and with a constant (absolute) orientation, then its position grows linearly with time. In this case the theorem indicates that the bias grows linearly with time: $\| \mathrm{E}[\mathbf{e}(n)] \| = \Theta(n)$, since now both the upper and lower bounds are asymptotically linear in time. This implies that the asymptotic trend of the bias is crucially dependent on the robot's displacement.

This dependency of the bias on the robot's trajectory is a consequence of the fact that that the estimated position is always bounded in mean, even if the robot is moving out to infinity, which follows from Lemma 1. To obtain an intuitive understanding of Lemma 1, we first note that the estimated position is simply the sum of the translations after transforming them to the common global coordinate frame 0; see (2–4). Taking expectation on both sides of (2–4), we obtain

$$\mathrm{E}[\hat{\mathbf{t}}^0_{0,n}] = \mathrm{E}[\hat{\mathbf{t}}^0_{0,1}] + \mathrm{E}[\hat{\mathbf{t}}^0_{1,2}] + \cdots + \mathrm{E}[\hat{\mathbf{t}}^0_{n-1,n}]. \tag{2–12}$$

The k^{th} term in the sum above, assuming rotation and translation measurements are independent, is

$$\mathrm{E}[\hat{\mathbf{t}}^0_{k-1,k}] = \mathrm{E}\left[\left(\prod_0^{k-1} \hat{\mathbf{R}}^i_{i+1}\right)\hat{\mathbf{t}}^k_{k-1,k}\right] = \left(\prod_0^{k-1} \mathrm{E}[\hat{\mathbf{R}}^i_{i+1}]\right)\mathrm{E}[\hat{\mathbf{t}}^k_{k-1,k}]$$
$$= \left(\prod_0^{k-1} (\mathbf{R}^i_{i+1}\overline{R})\right)\mathrm{E}[\hat{\mathbf{t}}^k_{k-1,k}].$$

The magnitude of this term is of order γ^k, since it involves k products of $\overline{R}$, each of which has a norm equal to γ. Since $\gamma < 1$ (see (2–7)), the sum (2–12) is bounded for all n. The expected value of the position estimate therefore converges to a point. Notice that the bounds (2–8) on the bias does not depend on the error in the translation measurements. The conclusions drawn above remain the same even if the rotation and translation measurements are unbiased, i.e., $\mu_{\tilde{\mathrm{R}}} = I$, $\mathbf{b}_k = 0$, and in fact, even if the translation measurements are completely error free, $\tilde{\mathbf{t}}^k_{k-1,k} = 0$.

The discussion above can be summarised into the following conclusions about the bias:

(i) For large time index n, the main contributions to the bias in the position estimate are the displacement of the robot and the errors in the relative rotation measurements.

(ii) The asymptotic scaling of the bias does not change even when the translation and rotation measurements are unbiased, and in fact even if translation measurements are completely without error.

The first conclusion is well known, and is hardly surprising. However, conclusion (ii) does not seem to be recognized in the literature.

The variance growth rate does not seem to be sensitive to the trajectory of the robot. Furthermore, unlike the bias, the variance can grow without bound when the robot's trajectory is confined to a bounded region. We'll see evidence of this later in simulations and experiments reported in Sections 2.4 and 2.5. We believe that the sufficient condition (2–11) is conservative, and is an artifact of our proof technique.

The condition (2–11) is usually not satisfied in practice since it requires a very large translation measurement error. Yet the position estimation error variance seems to be $\Theta(n)$ in simulations and experiments reported in Section 2.5.2.

The results of the theorem are in contrast to the prevalent belief in the literature that the error growth is superlinear in time if absolute orientation measurements are not available [1–7]. The theorem shows that even without absolute orientation sensors, localization error - or more precisely its bias and variance - grows at most linearly with time. We believe that the belief about superlinear growth came about from the fact that experiments/simulations were not conducted long enough to draw reasonable conclusion about asymptotic trends. Through the root cause is the geometric decay due to γ, since γ is usually quite close to 1, there is an initial period where the error grows sharply until the geometric decay kicks in and the linear trend becomes obvious. More insight into this phenomenon will be obtained later in Section 2.2.3 that discusses 2D trajectories (see in particular Theorem 2.2).

The proof of Theorem 2.1, presented next, follows from Lemma 1 in a straightforward manner.

Proof 2.1.1. *2.1 It follows from* (2–5)*, by applying the triangle inequality that*

$$\| \mathrm{E}[\mathbf{e}(n)]\| \leq \|\mathbf{t}^0_{0,n}\| + \| \mathrm{E}[\hat{\mathbf{t}}^0_{0,n}]\| \tag{2–13}$$

$$\| \mathrm{E}[\mathbf{e}(n)]\| \geq \max\left\{0, \|\mathbf{t}^0_{0,n}\| - \| \mathrm{E}[\hat{\mathbf{t}}^0_{0,n}]\|\right\} \tag{2–14}$$

From Lemma 1, we have that $\| \mathrm{E}[\hat{\mathbf{t}}^0_{0,n}]\|$ *is upper bounded and so the first statement follows immediately from* (2–13) *and* (2–14)*.*

$$\psi = 2c\mathbf{r}^T (I - c\underline{\mathbf{R}})^{-1} \underline{\mathbf{R}}\mathbf{r} + \mathrm{Tr}\left[\mathbf{P} + \mathbf{b}\mathbf{b}^T\right] + (2\mathbf{b}^T + \mathbf{r}^T)(I - c\underline{\mathbf{R}})^{-1}\boldsymbol{\rho} \tag{2–15}$$

$$\omega(n) = \mathbf{r}^T (I - c\underline{\mathbf{R}})^{-2} \left(I - 4c\underline{\mathbf{R}} + 2(c\underline{\mathbf{R}})^2 + 2(c\underline{\mathbf{R}})^{n+1}\right)\mathbf{r} - 2\mathbf{b}^T (I - c\underline{\mathbf{R}})^{-2} \left(I - (c\underline{\mathbf{R}})^n\right)\boldsymbol{\rho}$$
$$+ \mathbf{b}^T (I - c\underline{\mathbf{R}})^{-1} \left[I - (c\underline{\mathbf{R}})^n\right]\mathbf{r} - \mathbf{r}^T (I - c\underline{\mathbf{R}})^{-2} \left[I - (c\underline{\mathbf{R}})^n\right]\boldsymbol{\rho} - \| \left[(I - c\underline{\mathbf{R}})^{-1}(I - (c\underline{\mathbf{R}})^n)(c\underline{\mathbf{R}}\mathbf{r} + \boldsymbol{\rho})\right] |$$

To prove the second statement, note that

$$\begin{aligned}
\mathrm{Tr}\left[Cov(\mathbf{e}(n), \mathbf{e}(n))\right] &= \mathrm{Tr}\left[Cov(\hat{\mathbf{t}}^0_{0,n},\ \hat{\mathbf{t}}^0_{0,n})\right] \\
&= \mathrm{Tr}\left[\mathrm{E}[\hat{\mathbf{t}}^0_{0,n}(\hat{\mathbf{t}}^0_{0,n})^T] - \mathrm{E}[\hat{\mathbf{t}}^0_{0,n}]\,\mathrm{E}[\hat{\mathbf{t}}^0_{0,n}]^T\right] \\
&= \mathrm{E}[(\hat{\mathbf{t}}^0_{0,n})^T \hat{\mathbf{t}}^0_{0,n}] - \|\,\mathrm{E}[\hat{\mathbf{t}}^0_{0,n}]\|^2 \\
&\leq \mathrm{E}[(\hat{\mathbf{t}}^0_{0,n})^T \hat{\mathbf{t}}^0_{0,n}].
\end{aligned}$$

Since $\|\,\mathrm{E}[\hat{\mathbf{t}}^0_{0,n}]\| = O(1)$*, the second statement follows from Lemma 1.* ■

2.2.3 Special 2-D Trajectories

In this section we provide non-asymptotic results on the error growth for the special case when the motion of the robot is confined to a 2D plane and its trajectory is limited to two particular types. In the 2-D scenario $\hat{\mathbf{t}}^i_{j,k}, \mathbf{t}^i_{j,k} \in \mathbb{R}^2$ and $\mathbf{R}^i_j, \hat{\mathbf{R}}^i_j \in SO(2)$ for every i, j, k. The x and y axes of a Cartesian coordinate frame that lies on this plane and is attached to the robot's body at the initial time $k = 0$ is used as the global coordinate frame. In the 2-D scenario, the robot's orientation at time n can be uniquely described by an angle $\theta_{0,n} \in [-\pi,\ \pi)$, which describes rotation of its local frame about the z-axis of the global frame. The relative rotation between the frames $k-1$ and k is uniquely determined by the angle by which the frame $k-1$ has to be rotated in the counter clockwise direction to reach frame k, which we denote by $\theta_{k-1,k}$. Figure 2-1 shows an example. A noisy measurement of the relative rotation, denoted by $\hat{\theta}_{k-1,k}$, is assumed available at time k. The error in the relative rotation measurement is

$$\tilde{\theta}_{k-1,k} := \hat{\theta}_{k-1,k} - \theta_{k-1,k}. \tag{2–16}$$

For future use, we define $f_R : [-\pi, \pi) \to SO(2)$ as

$$f_R(\alpha) := \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}.$$

The matrix $\mathbf{R}_k^{k-1}$ that describes the relative rotation between the frames $k-1$ and k is therefore given by $\mathbf{R}_k^{k-1} = f_R(\theta_{k-1,k})$. It can be shown from the definition (2–1) that

$$\tilde{\mathbf{R}}_k^{k-1} = f_R(\tilde{\theta}_{k-1,k}). \tag{2–17}$$

The estimate of the rotation $\mathbf{R}_k^{k-1}$ therefore is $\hat{\mathbf{R}}_k^{k-1} = f_R(\hat{\theta}_{k-1,k})$.

We first provide results for the case when the robot moves in a straight line with constant velocity and orientation. The proof of the theorem is in the Appendix.

Theorem 2.2. *Consider a robot that moves on a 2-D plane in a straight line with a constant orientation. Formally, for all k, $\theta_{k-1,k} = 0$ and $\mathbf{t}_{k-1,k}^k = \mathbf{r} \in \mathbb{R}^2$, for some vector $\mathbf{r}$. In addition to Assumption 2.1, assume that the relative orientation error $\tilde{\theta}$ has a pdf that is symmetric around its mean $\mathrm{E}[\tilde{\theta}]$, the translation measurement errors $\tilde{\mathbf{t}}_{k-1,k}^k$, $k = 1, \ldots$ are wide sense stationary with $\mathbf{b}_k = \mathbf{b}$, $\mathbf{P}_k = \mathbf{P}$, and $\boldsymbol{\rho}_k = \boldsymbol{\rho}$ for all k. In that case, we have*

$$\begin{aligned} \mathrm{E}[\mathbf{e}(n)] &= n\mathbf{r} - (I - c\underline{\mathbf{R}})^{-1}(I - (c\underline{\mathbf{R}})^n)(c\underline{\mathbf{R}}\mathbf{r} + \boldsymbol{\rho}), \\ \mathrm{Tr}\left[Cov(\mathbf{e}(n), \mathbf{e}(n))\right] &= \psi n + \omega(n), \end{aligned} \tag{2–18}$$

where

$$c := \mathrm{E}[\cos\left(\tilde{\theta} - \mathrm{E}[\tilde{\theta}]\right)], \qquad \underline{\mathbf{R}} := f_R(\mathrm{E}[\tilde{\theta}]), \tag{2–19}$$

and the scalars ψ, $\omega(n)$ are given in (2–15). □

Since the r.v. $\tilde{\theta}$ is not degenerate by Assumption 2.1, we have that $|c| < 1$. The spectral radius of $c\underline{\mathbf{R}}$ is strictly lower than unity since $|c| < 1$ and $\underline{\mathbf{R}} \in SO(2)$. Hence $I - c\underline{\mathbf{R}}$ is invertible and $\psi, \omega(n)$ in (2–15) are well defined.

An immediate corollary of Theorem 2.2 is that for straight line motion, both the bias and the variance of the position estimation error grow asymptotically linearly with time. This follows from the expressions for the bias and the variance upon using the fact that $c < 1$. However, due to the presence of the c^n terms, the growth looks superlinear for intermediate values of the time index n. Simulations presented in Section 2.4.2 verify this statement; see in particular Figure 2-4 and 2-5. The linear trend becomes visible only when large values of the time index n are considered. This may be one of the reasons that the error is believed to grow super-linearly with time in the literature.

The next case is a periodic trajectory in 2-D. We say the robot moves in a periodic trajectory with period p if the absolute orientation and position of the robot satisfies the following conditions: $\theta_{0,k} = \theta_{0,k+p}$ and $\mathbf{t}^0_{0,k} = \mathbf{t}^0_{0,k+p}$ for all k. The shape of the (closed) path along which the robot moves can be arbitrary. In the statement of the theorem, η denotes the number of periods up to time n, and q to denote the residual, i.e., $\eta(n) := \lfloor n/p \rfloor$ and $q := n - \eta p$.

Theorem 2.3. *Consider a robot moving in $\mathbb{R}^2$ whose trajectory is periodic with period p. In addition to Assumption 2.1.1-Assumption 2.1.4, assume that the first and second moments of the measurement errors are periodic with period p (so that $\mathbf{b}_k = \mathbf{b}_{k+p}$, $\boldsymbol{\rho}_k = \boldsymbol{\rho}_{k+p}$ and $\mathbf{P}_k = \mathbf{P}_{k+p}$). In that case,*

$$\begin{aligned} \mathrm{E}[\mathbf{e}(n)] = \mathbf{t}^0_{0,q} - (I - (c\underline{\mathbf{R}})^p)^{-1} \\ \times (I - (c\underline{\mathbf{R}})^{\eta p})\, w(p) - (c\underline{\mathbf{R}})^{\eta p} w(q), \end{aligned} \tag{2–20}$$

where $w(j)$ is given by

$$w(j) := \sum_{i=0}^{j-1} (c\underline{\mathbf{R}})^i \, \mathbf{R}^0_{i+1} \left(c\underline{\mathbf{R}}\, \mathbf{t}^{i+1}_{i,i+1} + \boldsymbol{\rho}_i \right),$$

where $c, \underline{\mathbf{R}}$ are as defined in Theorem 2.2.

The proof of the theorem is provided in the Appendix. The assumption of the moments $\boldsymbol{\rho}_k$ etc. being periodic with period p is motivated by the use of vision-based

sensors to measure relative poses. In that case the measurement error statistics may depend on the scene the camera sees, which will repeat itself every p instants due to the periodic nature of the robot's motion. Note that i.i.d. errors are a special case of errors with periodic statistics, and so the result is also valid in such a scenario.

It can be shown in a straightforward manner from (2–20) that the bias is $O(1)$, by using the fact that $|c| < 1$. This is consistent with Theorem 2.1 since the robot stays in a bounded region for all time when following a periodic trajectory.

2.3 Reducing the Bias

We now discuss a possible way to reduce the bias in the position estimate by using the lessons learned from the analysis that led to Theorem 2.1. First of all we note that computing $\mathrm{E}[\hat{\mathbf{t}}^0_{k-1,k}]$ requires knowledge of true relative rotations and translations, and therefore the bias cannot be eliminated by simply computing it and subtracting it from the estimated translation $\hat{\mathbf{t}}^0_{k-1,k}$ at every k. Instead the proposed method consist of modifying the raw measurements $\hat{\mathbf{R}}^{k-1}_k, \hat{\mathbf{t}}^k_{k-1,k}$ into the so-called modified measurements $(\hat{\mathbf{R}}^{k-1}_k)_{modif}, (\hat{\mathbf{t}}^k_{k-1,k})_{modif}$, that are defined below, and then using them in the position estimation.

$$(\hat{\mathbf{R}}^{k-1}_k)_{modif} := \hat{\mathbf{R}}^{k-1}_k(\overline{R})^{-1} \qquad\qquad (\hat{\mathbf{t}}^k_{k-1,k})_{modif} := \hat{\mathbf{t}}^k_{k-1,k} - \mathbf{b},$$

where

$$\mathbf{b} := \mathrm{E}[\tilde{\mathbf{t}}^k_{k-1,k}], \quad k \geq 1. \tag{2–21}$$

We assume that the translation measurements are stationary in mean so that $\mathbf{b}$ is a constant. The modified measurements can be computed from the raw measurements and knowledge of $\overline{R}, \mathbf{b}$, which can be determined from an analysis of sensor noise characteristic. For instance, the question of estimating $\mathbf{b}$ for vision-based sensors is examined in [9, 10]. The position at time k is now computed as before, but with the new corrected measurements in place of the raw sensor measurements $\hat{\mathbf{t}}^k_{k-1,k}$ and $\hat{\mathbf{R}}^{k-1}_k$.

Specifically,

$$(\hat{\mathbf{R}}_k^0)_{modif} := \prod_{i=1}^{k} (\hat{\mathbf{R}}_i^{i-1})_{modif}$$

$$(\hat{\mathbf{t}}_{k-1,k}^0)_{modif} := (\hat{\mathbf{R}}_k^0)_{modif}\, (\hat{\mathbf{t}}_{k-1,k}^k)_{modif},$$

$$\text{and finally, } (\hat{\mathbf{t}}_{0,n}^0)_{modif} = \sum_{k=1}^{n} (\hat{\mathbf{t}}_{k-1,k}^0)_{modif}.$$

The rationale for this proposal comes from the following relationships that can be shown from straightforward calculations:

$$\mathrm{E}[(\hat{\mathbf{R}}_k^{k-1})_{modif}] = \mathbf{R}_k^{k-1} \tag{2–22}$$

$$\mathrm{E}[(\hat{\mathbf{R}}_k^{k-1})_{modif} (\hat{\mathbf{t}}_{k-1,k}^k)_{modif}] = \mathbf{R}_k^{k-1} \mathbf{t}_{k-1,k}^k, \tag{2–23}$$

where the second relation (2–23) holds if the raw rotation and translation measurements $\hat{\mathbf{R}}_k^{k-1}$, $\hat{\mathbf{t}}_{k-1,k}^k$ are uncorrelated. The modification of the raw measurements eliminates the geometric decay of the length of the relative translation measurements after being transformed to frame 0. As discussed in Section 2.2.2, this decay was the main cause of the bias growth. If $\hat{\mathbf{R}}_k^{k-1}$, $\hat{\mathbf{t}}_{k-1,k}^k$ are correlated but the motion is limited to a 2-D space, a slightly different method can be used that ensures that (2–22),(2–23) hold.

2.4 Simulation Verification

In this section we empirically estimate the mean and covariance of the estimation error by conducting Monte-Carlo simulations and compare them with the theoretical predictions. In section 2.4.1 we simulate a robot moving along a randomly generated 3-D path and compare the results with the upper and lower bounds predicted in Theorem 2.1. In sections 2.4.2 and 2.4.3 we present simulations for the 2-D scenario with straight line and periodic trajectories so that empirical results can be compared with predictions of Theorem 2.2 and 2.3. In each case, the robot is simulated moving along either the straight line or periodic trajectory at a speed of $0.32\,\mathrm{m/s}$ for about 5.5 hours, traveling a distance of 6400 meters. In all three simulations, measurements of the robot's

relative pose were taken every 0.2 seconds. All simulations are conducted in MATLAB©. To the extent possible, the parameters used in the simulations are the same as those in the experiments.

2.4.1 3-D Simulation

For the 3-D case we simulate a robot moving along a path that is shown in Figure 2-2. The robot traverses this path from the starting point to the left and moving to the right. Measurement errors are generated as follows. The error in rotation $(\tilde{\mathbf{R}}_k^{k-1})$ is introduced by applying a random unit-quaternion at each time step drawn independently from a Von Mises-Fisher distribution with concentration parameter $k = 10000$. The reader is referred to [47] for details on Von Mises-Fisher distribution. The errors in relative translation at each time step $(\tilde{\mathbf{t}}_{k-1,k}^k)$ are drawn from a zero-mean normal random variable with covariance matrix $(2.5 \times 10^{-5})\, I_{3\times 3}$. The corresponding constants necessary to compute the upper bounds in Theorem 2.1 are obtained from randomly generated measurements to simulate a sensor characterization test and found to be $\gamma = 0.9997$, $\tau = 0.1295$ m, $\mathrm{b} = 0$ m, $\beta = 0.008$ m, $\underline{\mathrm{p}} = 7.45 \times 10^{-5}$ m^2, and $\overline{\mathrm{p}} = 7.55 \times 10^{-5}$ m^2

Figure 2-3 compares the empirically estimated bias and variance with the upper bounds given by Theorem 2.1. The empirical estimates are obtained from 4500 Monte Carlo simulations. As predicted by the theorem, the bias in the position estimate grows without bound since the robot's position is growing (in norm) without bound. We see that the bounds predicted by the theorem are of the same order of magnitude as values obtained empirically. However, the bounds for the variance are rather loose.

2.4.2 Straight-Line 2-D Trajectory

For the straight line case, we simulate a robot moving in a straight line on a plane with a constant velocity of $[0.2263, 0.2263]^{\mathsf{T}}$ m/s and constant orientation. Two types of simulations are conducted.

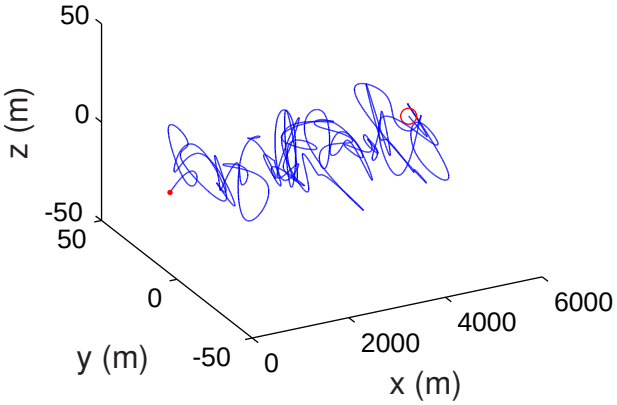


Figure 2-2. The 3-D path used for the simulation in section 2.4.1. The red dot indicates the robots initial location, while the red circle indicates the robots final location.

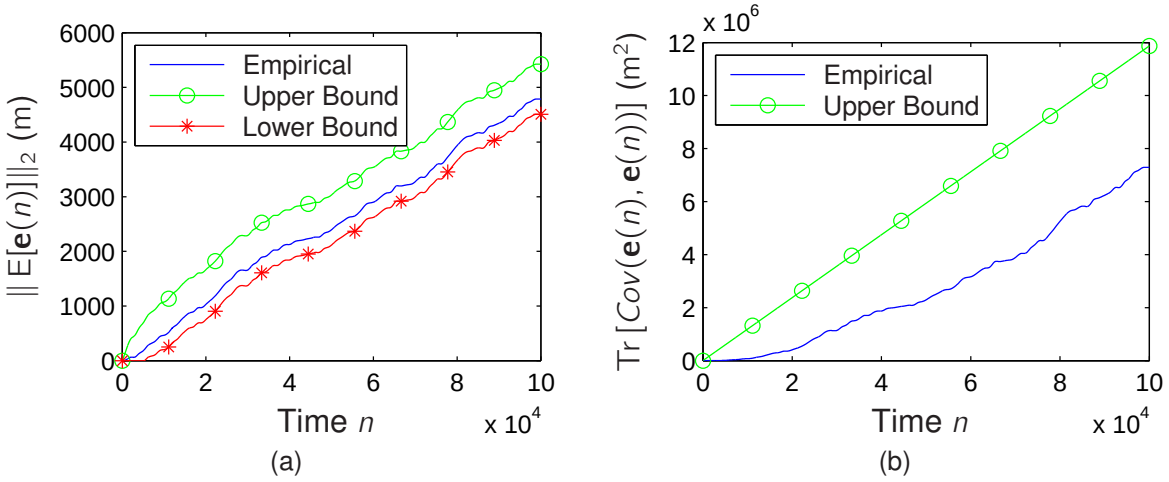


Figure 2-3. Comparison of theoretical bounds vs. simulation results. The labels “Upper Bound” or “Lower Bound” indicate the theoretical bounds found in Theorem 2.1, while the label “Empirical” indicates the sample (a) bias or (b) variance found using a 4500-iteration Monte-Carlo experiment.

In the first type, which we call simulated data, noisy measurements of the rotation, i.e., $\hat{\theta}_{k-1,k}$ are generated as a Laplace distributed random variable using a pseudo-random number generator. The reason for choosing a Laplace distribution, over the more commonly used Gaussian distribution, is the following. We obtained a large sample of 2D orientation estimates from images taken with a machine-vision camera and performed hypothesis testing for three distributions: Laplacian, Gaussian

and Fisher Von-Mises. Only the Laplace distribution passed the test. We refrain from giving details of the hypothesis testing here; they can be obtained from the authors upon request. Noisy measurements of the translations, i.e., $\hat{\mathbf{t}}^k_{k-1,k}$ were generated from noisy measurements of translation direction, which we call $\boldsymbol{\zeta}^k_{k-1,k}$, and translation magnitude, which we call $d_k \in \mathbb{R}^+$, as $\hat{\mathbf{t}}^k_{k-1,k} = \hat{d}_k \hat{\boldsymbol{\zeta}}^k_{k-1,k}$, where $\hat{d}_k$ and $\hat{\boldsymbol{\zeta}}^k_{k-1,k}$ are noisy estimates of d_k and $\boldsymbol{\zeta}^k_{k-1,k}$, respectively. Note that $\boldsymbol{\zeta}^k_{k-1,k}$ is a unit norm 2-vector. This is done to simulate relative pose measurement with IMU/wheel odometry and a monocular camera without scale information. The camera provides relative translation direction but not the magnitude of translation, which is measured by IMUs/wheel encoders.

In the second type of simulations, which we refer to as simulated camera, the vision-based relative pose estimation sensor is simulated in a more realistic fashion by generating synthetic image data, from which relative rotation and direction of translation are estimated. The magnitude of translation measurements are generated as in the "simulated data" case.

Simulated data: At each time step k, a measurement of the relative orientation is constructed numerically as $\hat{\theta}_{k-1,k} = 0 + \tilde{\theta}_{k-1,k}$, where the orientation error $\tilde{\theta}_{k-1,k}$ is chosen to be a 0-mean Laplace distributed r.v. Recall that a Laplace distribution with μ mean and variance $2\lambda^2$ has the pdf $f(\tilde{\theta}) = \dfrac{1}{2\lambda} e^{-|\tilde{\theta}-\mu|/\lambda}$. The value of λ chosen is 3.6×10^{-3}, which best fits the orientation measurement error statistics generated by the synthetic monocular camera-based relative pose sensor that is used in the experiments presented in the sequel. The noisy measurement of translation direction $\hat{\boldsymbol{\zeta}}^k_{k-1,k}$ generated as

$$\hat{\boldsymbol{\zeta}}^k_{k-1,k} = \begin{pmatrix} \cos\tilde{\phi}_{k-1,k} & -\sin\tilde{\phi}_{k-1,k} \\ \sin\tilde{\phi}_{k-1,k} & \cos\tilde{\phi}_{k-1,k} \end{pmatrix} \boldsymbol{\zeta}^k_{k-1,k}$$

where $\tilde{\phi}_{k-1,k}$ is a zero-mean Laplace random variable with variance $3.07 \times 10^{-2}\,\mathrm{rad}^2$, and $\boldsymbol{\zeta}^k_{k-1,k} = \frac{1}{\sqrt{2}}[1, 1]^T$ is the true translation direction. The magnitude of the translation

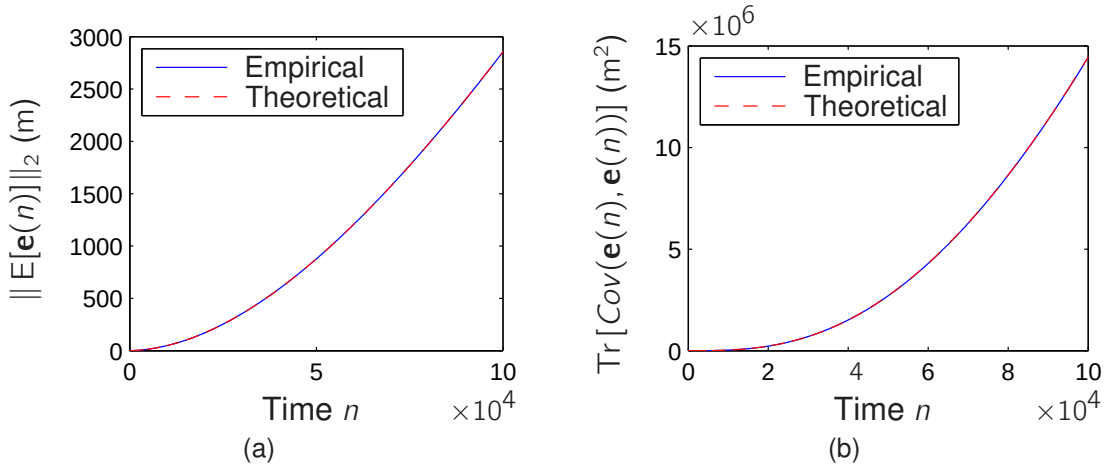


Figure 2-4. Comparison of theoretical results for straight line vs simulation data. The predicted value from Theorem 2.2 is given by the dashed line, while the sold line indicates the sample (a) bias or (b) variance found using a Monte-Carlo experiment for the "simulated data" case.

is $d_k = 6.4 \times 10^{-2}$ m and its noisy measurement is generated as $\hat{d}_k = d_k + \tilde{d}_k$, where $\tilde{d}_k$ is a zero-mean Gaussian random variable with mean 0 variance 8.5467×10^{-5} m^2. These numbers are chosen to be consistent with those seen in an experiment with a wheeled robot described later in Section 2.5. The parameters $\mathbf{b}$, c, $\mathbf{P}$, $\boldsymbol{\rho}$ that are needed to compute the predictions by Theorem 2.2, are estimated by a simulated sensor characterization test, i.e., by appropriate averaging of randomly generated data. They turn out to be $\mathbf{b} = [-0.6842, -0.6842] \times 10^{-3}$ m, $c = 1 - 1.2873 \times 10^{-5}$, $\mathrm{Tr}\,[\mathbf{P}] = 1.2479 \times 10^{-4}$ m^2, and $\boldsymbol{\rho} = c\mathbf{b}$.

The mean and covariance of the position estimation error at every time instant are empirically estimated by averaging over 76,600 Monte-Carlo simulations. Figure 2-4 presents the estimated mean and covariances, and the values predicted by Theorem 2.2. We see from the figure that the prediction from Theorem 2.2 matches estimates from Monte-Carlo simulations quite closely even for the large time intervals used in the simulations.

Simulated camera: We now simulate the scenario in which relative pose measurements are obtained by a calibrated monocular Prosilica EC 1020 camera

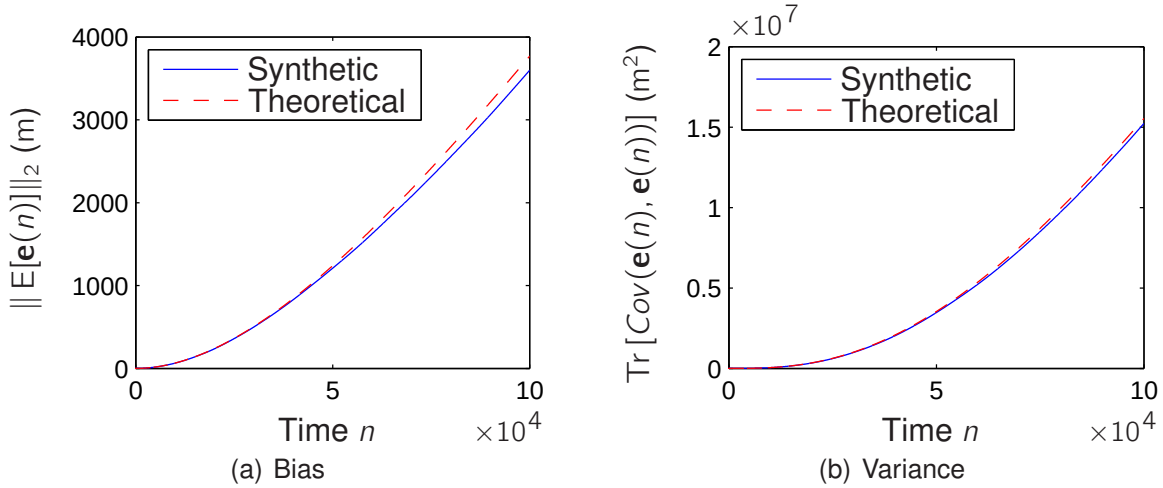


Figure 2-5. Comparison of theoretical results for straight line vs simulation camera. The predicted value from Theorem 2.2 is given by the dashed line, while the sold line indicates the sample (a) bias or (b) variance found using a Monte-Carlo experiment for the "simulated camera" case.

and wheel odometers found on a Pioneer P3-DX. To simulate an estimate of the camera ego-motion between consecutive time steps, suppose between k and $k+1$, a set of 50 3-D points are randomly generated in the volume visible to the camera at time step k, with their coordinates represented in the coordinate frame attached to the camera at time step k. The points are then acted on by the true transformation from k to $k+1$ to find the corresponding coordinates in the coordinate frame attached to the camera at time step $k+1$, Using a calibration matrix corresponding to the Prosilica EC 1020 camera, the points are projected into their corresponding image plane. This forms a set of correspondences analogous to the feature points extracted from actual image pairs. Each feature point is now corrupted by uniform noise with support lying in a 2×2 pixel square about the point. A RANSAC [48] assisted normalized 8-point algorithm [49] is used to estimate the rotation $\hat{R}$ and translation direction $\hat{\zeta}$ between the two time steps from these point correspondences. The axis of rotation was then aligned with the normal to the plane of motion and the component of the translation vector in that direction was dropped to insure the motion estimates remained in the plane. The magnitude of translation $\hat{d}$ is generated as in the Simulated Data case. The

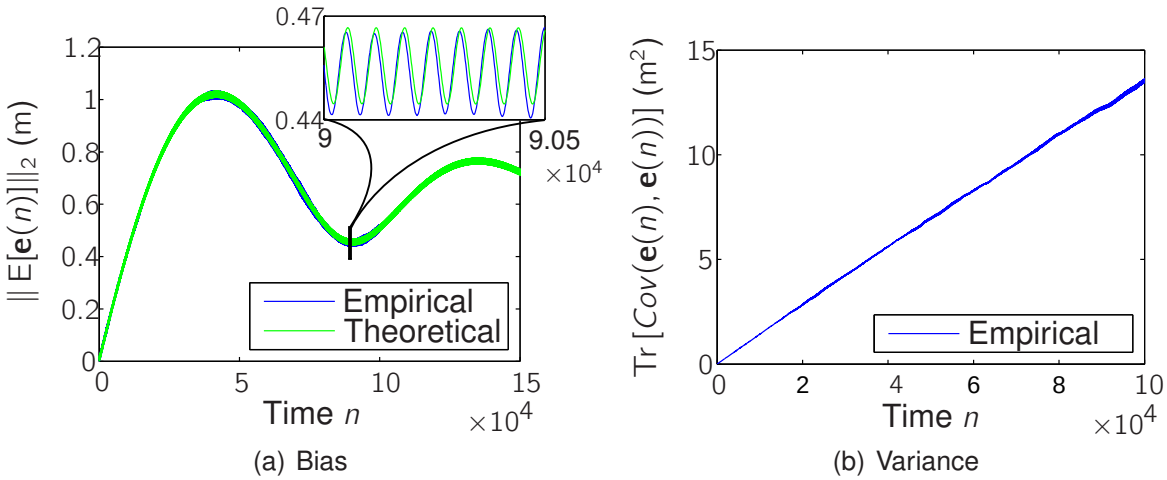
values of the parameters that are needed to compute the predictions by Theorem 2.2 are estimated from a simulated sensor characterization test like before. The values are found to be $\mathbf{b} = [-0.5767, -0.5904] \times 10^{-5}$ m, $\mathrm{Tr}\,[\mathrm{P}] = 1.6382 \times 10^{-4}$ m^2, and $c = 1 - 2.1462 \times 10^{-5}$.

Figure 2-5 compares the predictions of bias and variance by Theorem 2.2 to those estimated from 1000 Monte-Carlo simulations. The number of Monte-Carlo simulations is smaller in the synthetic data case due to the prohibitively high cost of conducting these simulations. We see from Figure 2-5 that Theorem 2.2 accurately predicts the position estimation error computed from synthetic image data. The prediction toward the end of the simulation time is not as accurate as in the simulated data case, which is due to the smaller number of Monte-Carlo trials.

2.4.3 Periodic Trajectory

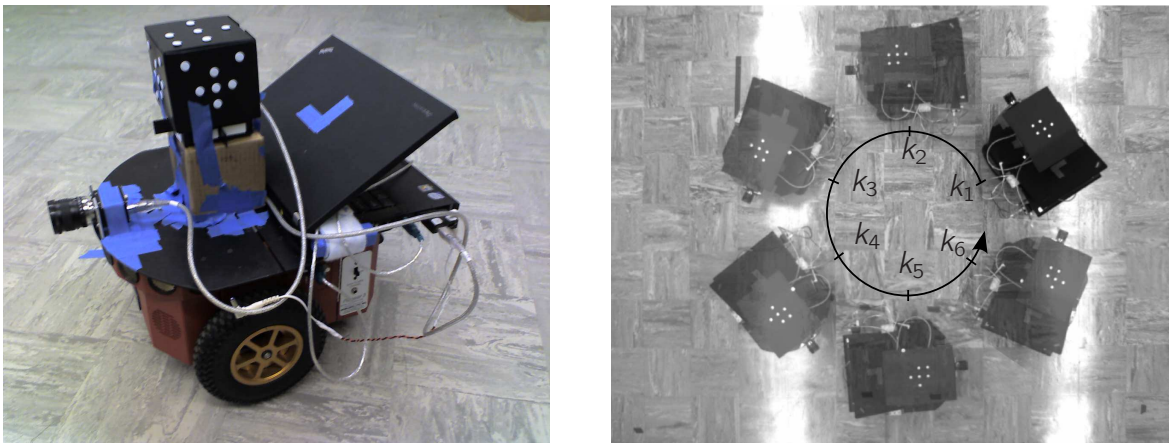
We next simulate a robot moving on a circle with circumference of 4.11 m so that its trajectory is periodic with period $p = 3020$. The speed of the robot is approximately $0.32\,\mathrm{m/s}$, so that it traverses the circle about 47 times before completing one period. The trajectory is chosen to be close to that encountered in an experiments with a Pioneer P3-DX robot, which will be presented in Section 2.5. Noisy relative pose measurements are generated as in the Simulated Data case in straight line motion. Orientation measurement errors are Laplace distributed (with mean $\mathrm{E}[\tilde{\theta}] = 6.8 \times 10^{-5}$ m and parameter $\lambda = 3.6 \times 10^{-3}$) while translation measurement errors are generated in the same manner, and with the same distributions as in the Simulated Data case in straight line motion; with the new true values given by $\boldsymbol{\zeta}^{k}_{k-1,k} = -[0.049, 0.999]^{T}$ and $\mathrm{d_k} = 0.064$ m.

Figure 2-6 shows the empirical estimates of bias and variance from $29,970$ Monte-Carlo simulations. It also presents the bias predicted from Theorem 2.3. We see from Figure 2-6(a) that the bias is quite accurately predicted by Theorem 2.3. The high frequency oscillation corresponds to the time it takes for the robot to traverse the



(a) Bias (b) Variance

Figure 2-6. Comparison of theoretical results for periodic motion vs simulation results. The results are for a 2-D scenario with periodic motion. The label "Theoretical" indicates predictions from Theorem 2.3. The label "Empirical" indicates estimates of the (a) bias or (b) variance found using a Monte-Carlo experiment.



(a) (b)

Figure 2-7. The robot used in the experiments. In (b) the trajectory of the robot is indicated by showing an image of the robot, as viewed by the overhead camera, at multiple time steps.

circle once. The lower frequency oscillation corresponds to the period of the trajectory. The variance seems to grow linearly with time, as one can see from Figure 2-6(b), but a formula is not available in the periodic case for comparison.

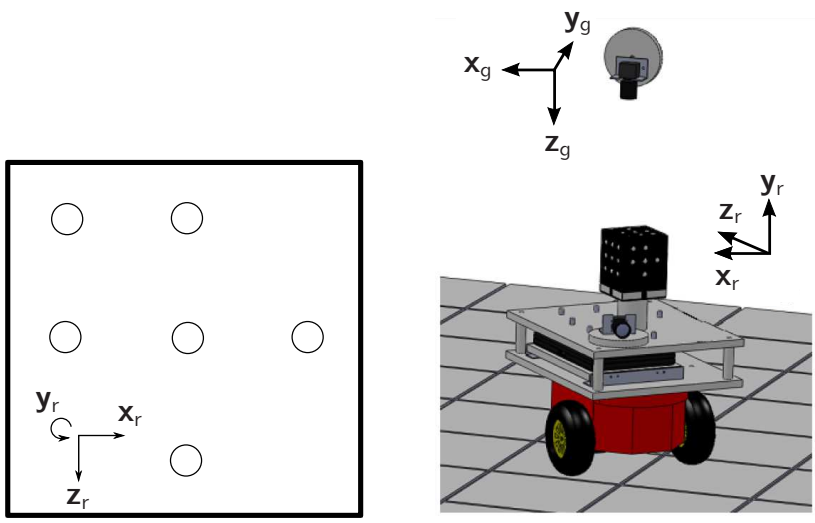
2.5 Experimental Verification

In this section we report results of experiments conducted with a wheeled Pioneer P3-DX robot that is equipped with a calibrated monocular Prosilica EC 1020 camera and wheel odometers. The images captured by the camera are used to estimate the relative rotation and direction of translation. The distance travelled is estimated by the wheel odometers and then fused with the direction of translation estimated from the camera to estimate the translation vector. The relative pose of the camera is measured every 0.2 seconds. An overhead camera is used to measure the true 2-D pose of the robot. Due to space constraints of the indoor test set-up, the trajectory of the robot was chosen to be an approximately circular one with radius 0.65 m and one rotation taking approximately 13 seconds (see Figure 2-7). Although the robot's trajectory is not truly periodic; it is approximately periodic with period $p = 3020$ (i.e., 604 seconds).

2.5.1 Test Set-Up

Figure 2-8(b) shows a schematic of the experimental set-up. The global coordinate frame is defined to coincide with the coordinate frame attached to an overhead camera viewing the plane of motion. That is, the origin of the global coordinate axes corresponds to the camera's focal point. The overhead camera is used to obtain the true pose of the robot. The robot's local coordinate frame was defined by a cube affixed to the top of the box. A grid consisting of six dots was placed atop the cube with a known geometry (see Figure 2-8(a)), which allows reconstruction of the full 3-D pose of the robot from the single monocular camera. Although some error between the true pose of the robot and that estimated from the overhead camera is unavoidable, this error did not have any cumulative effect over time. Therefore the pose estimated from the overhead camera is taken as the ground truth.

The KLT tracker [50] was used to track feature points across pairs of images, and a RANSAC-assisted normalized 8-point algorithm was used to estimate the relative rotation and direction of translation between every successive pairs of images. All



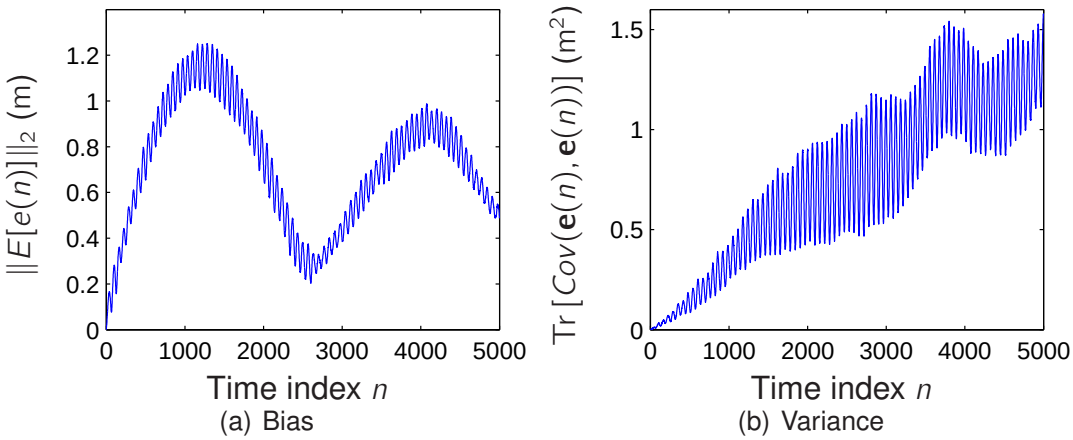
(a) Top view (b) Coordinate Axes

Figure 2-8. Schematic of the test set-up. The (a) overhead view and (b) side view, including overhead camera, are shown. The global coordinate frame and robot fixed coordinate frame are also indicated.

estimation was performed off-line. Even with RANSAC, outliers in point-correspondences can cause large errors in the relative pose estimates. An ad-hoc "filter" was implemented to reduce the effect of such errors as follows. If the estimated relative pose from the camera was deemed infeasible (which was determined by the known motion of the robot), the relative rotation and relative translation direction estimated in the previous time step was used as the estimate for the current time step. The relative translation between two time instants was estimated from the relative translation direction and the estimate of its magnitude, the latter being obtained from a wheel odometer. The relative poses so obtained were chained together to obtain an estimate of the global position and orientation of the robot at every time step, as described in Section 2.1.

2.5.2 Test Results

The position estimation error at each time step is computed by comparing the ground truth with the robot's position estimated from relative pose measurements. The bias and variance in the position estimation error at any given time step are determined by averaging over 17 experiments, where each experiment consists of the robot moving



(a) Bias (b) Variance

Figure 2-9. Experimental results for the 2-D circular motion case using a single P3-DX robot.

on its path for 1000 seconds (5000 time steps). The experimentally obtained bias and variance of position estimation error are shown in Figures 2-9(a) and 2-9(b).

From the figures we see that the experimentally obtained results - especially the bias – closely resemble those seen in simulations (cf. Figure 2-6(a),2-6(b)), which in turn are accurately predicted from the analysis. The experimentally obtained bias stays bounded, as Theorem 2.3 predicts. The variance also shows an on-average linear growth with time, which is consistent with Theorem 2.1. The experiment provides additional confidence in our theoretical results. In addition, we note that while the theoretical predictions are for a dead-reckoning type position estimation algorithm, the algorithm used in the experiments was more akin to a kinematic-model based filter. Still the theoretical predictions match the experimental results rather well. This is expected since - as argued earlier - the analysis is applicable to broader class of estimation algorithms; see the discussion in Section 2.1 after Eq. (2–5).

There are nevertheless some discrepancies between the experimentally obtained bias and variance values and those obtained from simulations, as can be seen comparing Figure 2-6(a) with Figure 2-9(a) and Figure 2-6(b) with Figure 2-9(b). These are due to the differences between the experiments and simulations. First, the experimental bias and variances values are computed by averaging over only

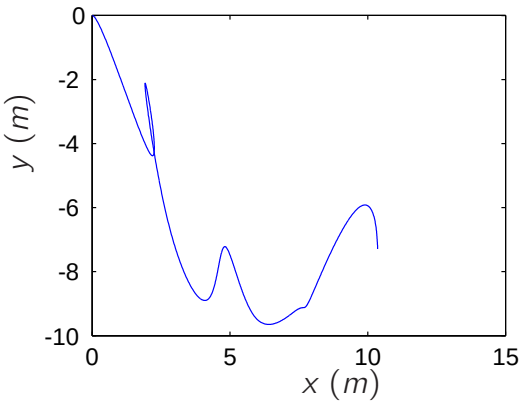


Figure 2-10. A randomly generated path in 2-D used to test the bias-reduction method.

17 experiments, whereas the simulation estimates are computed from at least 1000 Monte-Carlo simulations, in some cases many more. The reason for this smaller number of experimental trials is the difficulty and time needed in performing these experiments. The smaller number of trials available to average over produced less accurate estimates. Second, the characteristics of the camera error could not be modeled in any of our simulations. Third, it is not possible to ensure a truly periodic trajectory in a real experiment. The "high-frequency" oscillations in the experimental bias and variance plots are at 7.8×10^{-2} Hz, which correspond to the average time the robot takes to traverse the circle once. These are seen in the simulations as well; see in particular the inset in Figure 2-6(a). However, these oscillations are not particularly visible in the variance, one has to magnify the curve in Figure 2-6(b) considerably to see them. We believe the noticeable difference in case of the variance comes from the very small number of runs that we averaged over.

2.6 Reducing the Bias

The bias-reduction method, first presented in Section 2.3 was tested with the help of simulations to determine its effectiveness. The following types of trajectories in 2-D were used in the simulations: (i) straight line (ii) circular, (iii) random walk in a city-like grid, and (iv) a randomly generated smooth path. The performance was seen to be similar in all cases; so only the details for case (iv) a randomly generated smooth path

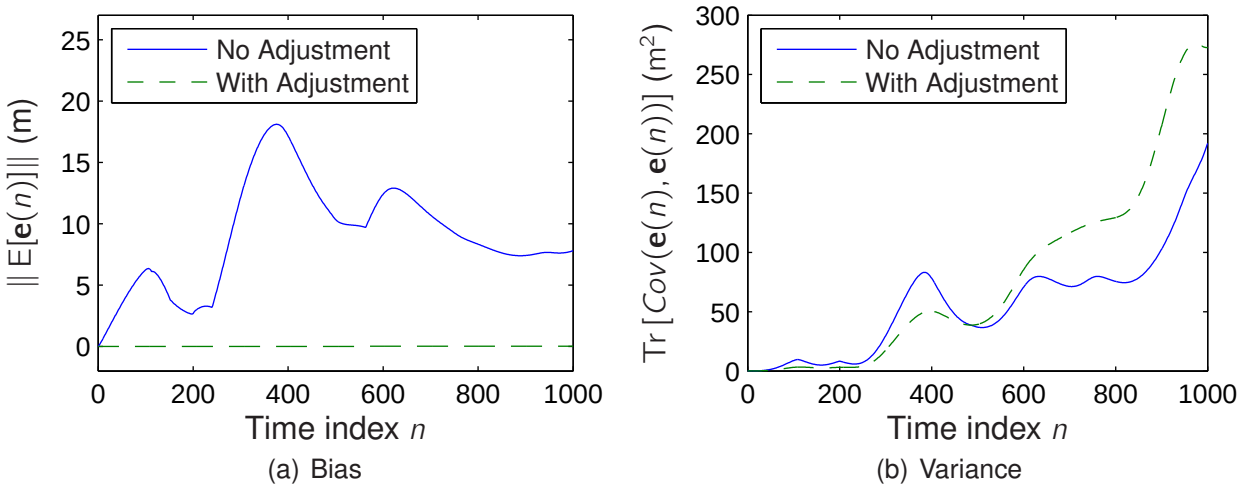


Figure 2-11. Performance of the bias-reduction method, for the path shown in Figure 2-10. The legend "With Adjustment" refers to the estimates obtained with the bias-reduction method of Section 2.3. The bias is reduced to almost zero with the proposed method. All quantities are estimated from more than a million Monte-Carlo simulations.

is presented. The path our robot traversed in the experiment is shown in Figure 2-10. Noise in the sensor measurements was simulated by adding i.i.d. Gaussian random vectors with mean $[0.05, 0.02]^T$ m and covariance matrix $0.05I$ to the relative translation measurements at each time step. The angle describing the relative rotation between each time step was corrupted by adding i.i.d. Gaussian random variables with mean 6.8×10^{-3} and variance 2.6×10^{-3}. The sensor characteristics $\overline{R}$ and $\mathbf{b}$ needed for the correction were determined a-priori; their values are $\overline{R} = 0.9987 f_R(6.8 \times 10^{-3})$, $\mathbf{b} = [0.05, 0.02]^T$. The estimates of the bias and variance in the position estimates were obtained from more than a million Monte-Carlo simulations. The comparison between the bias with the method described in Section 2.3 and that for the baseline case (no modification) is shown in Figure 2-11(a). The comparison of the variances is shown in Figure 2-11(b).

We see from the simulations that the proposed method of bias reduction significantly reduces the bias. The resulting variance is the same or smaller, for small values of time. For large values of time, the resulting variance is larger than that achieved if

measurements were not modified. This is expected since the modifications introduce additional uncertainty. In particular, the modified rotation measurements are no longer elements of $SO(d)$. A similar trend is seen for all other trajectories tested: the bias is significantly reduced for all time, while the variance is either smaller or almost the same for small values of time but is larger for large values time.

2.7 Summary

We examined the growth of error in position estimates obtained from noisy relative pose measurements. We showed that in both 2-D and 3-D, the bias and the variance of the position estimation error grows at most linearly with time or distance travelled. The precise growth rate of the bias depends on the trajectory of the robot. Specifically, if the robot stays in a bounded region, the bias is upper bounded by a constant for all time. It was proved that the variance growth rate is also lower bounded by linear function of time if the translation measurement errors are large enough. Exact formulas for the error bias and variance were obtained for two particular 2-D trajectories, straight line and periodic. Extensive Monte-Carlo simulations, and experiments with a wheeled robot, were used to verify the results.

The results of this chapter show that localization error growth rate is, in fact, not superlinear with time or distance even without absolute orientation sensors. In addition, it turns out that the asymptotic growth rate of the bias does not change even if all the measurements are unbiased or even if the translation measurements are completely error free. The bias growth is principally due to the fact that the expected value of the estimated position converges to a point, irrespective of how the robot is moving. This occurs since γ, the norm of the expected rotation error, is strictly less than unity. As a result, the magnitude of the measured translation, once the measurement is transformed to the global coordinate frame, decays geometrically with time.

One important assumptions made for the analysis was that the measurements collected at two distinct time instants are statistically independent. Though this may not

hold in practice, the results obtained from experiments and simulations with synthetic image data are consistent with the theoretical predictions. This shows that the analysis is not sensitive to the assumptions of independence. The sufficient condition (2–11) for the variance to be asymptotically linear in time is not satisfied in the simulations and the experiment. However, the empirically estimated variance from simulations and experiment seems to grow linearly with time. This indicates that the sufficient condition is conservative. Determining a necessary condition for variance growth to be linear is an open question.

A method to reduce the bias growth rate was suggested by the lessons learned in the analysis of error growth. Simulations showed that the proposed method reduces the bias significantly for all time, while having negligible effect on the variance for small values of time. The method can therefore be potentially used to improve localization accuracy for short periods of time. There are several issues that still need to be addressed. The method was observed to make the variance worse for large time. So an important research question is to determine the time period up to which the method can be used. The method requires knowledge of sensor characteristics. Its robustness to imprecise knowledge of sensor characteristics, and to time variations in those characteristics, also needs to be studied.

CHAPTER 3
DISTRIBUTED COLLABORATIVE 3D POSE ESTIMATION

In this chapter we propose a method for fusing measurements of various types to perform collaborative localization that improves localization accuracy over dead-reckoning. The problem is considered in such a fashion as to not require any parameterization of the space of rotations. We will refer to the proposed method as the Riemannian Pose Graph Optimization (RPGO) algorithm.

The rest of the chapter is organized as follows. The problem is stated in Section 3.1. A centralized estimation scheme is described in Section 3.2, and a distributed algorithm that is inspired by the centralized scheme is described in Section 3.3. Simulation and experimental results with the proposed algorithms are presented, in Sections 3.4 and 3.5, respectively. The chapter concludes with a discussion in Section 3.6.

3.1 Problem Statement

3.1.1 The Collaborative Localization Problem

Consider a group of r mobile robots indexed by $i = 1, \dots, r$. Time is measured by a discrete counter $k = 0, 1, 2, \dots$. Each robot is equipped with a local, rigidly attached frame of reference, that is, a coordinate system defined in the robot's local reference frame. We call the frame of reference attached to robot i at time k frame $i(k)$. Between any two frames of reference, say frame u and frame v, we denote the Euclidean transformation from v to u by $\mathbf{T}_{uv}$, where $\mathbf{T}_{uv}$ is an element of the special Euclidean group $SE(3)$. Specifically, if p_u is a point expressed in frame u and p_v is the same point expressed in frame v, then $p_u = \mathbf{T}_{uv} p_v$. We call $\mathbf{T}_{uv}$ the relative pose of frame v with respect to frame u.

Let frame 0 denote some fixed frame of reference that is common to all robots. The absolute pose of frame u is then given by the transformation $\mathbf{T}_{0u}$. We will often denote the absolute pose simply as $\mathbf{T}_u$. Robot i is said to be localized at time k when an

estimate is known for the absolute pose of frame $i(k)$. We denote such an estimate by $\hat{\mathbf{T}}_{i(k)}$.

Measurements of a robot's absolute pose, perhaps from measurements found using GPS and compass, are either not available or only rarely available. Instead, each robot is equipped with proprioceptive sensors such that, at every time k, the robot is able to obtain a relative pose measurement with respect to its previous pose, given by $\hat{\mathbf{T}}_{i(k-1)\,i(k)}$. We refer to these measurements as *inter-time relative pose measurements*. Such measurements can be obtained with inertial or vision based sensors. Additionally, they need not be obtained from a sensor alone. Instead, a measurement could also be the estimate computed by fusing sensor measurements with predictions of the robot's motion from a dynamic/kinematic model.

In addition to proprioceptive sensors, each robot is equipped with exteroceptive sensors so that occasionally, a robot i is able to obtain a relative measurement of one or more other robots. We call these *inter-robot relative measurements*. If a robot i collects a measurement of robot j at time k, it can be one of the following:

- **Relative pose**: The Euclidean transformation from frame $j(k)$ to frame $i(k)$; denoted by the symbol $\mathbf{T}$.
- **Relative orientation**: The element of $SO(3)$ that describes the change in orientation from frame $j(k)$ to frame $i(k)$; denoted by the symbol $\mathbf{R}$.
- **Relative position**: The vector in $\mathbb{R}^3$ that describes the change in position between the frames $i(k)$ and $j(k)$, expressed in frame $i(k)$; denoted by the symbol $\mathbf{t}$.
- **Relative bearing**: The vector of unit length that points from frame $i(k)$ to frame $j(k)$, expressed in frame $i(k)$; denoted by the symbol $\boldsymbol{\tau}$.
- **Relative distance**: The distance between frame $i(k)$ and frame $j(k)$; denoted by the symbol δ.

Which pairs of robots will be able to obtain an inter-robot relative measurement will depend on many factors, including the kind of sensors they have on-board, the range of sensors, etc and so the set of inter-robot measurements available varies with time. An

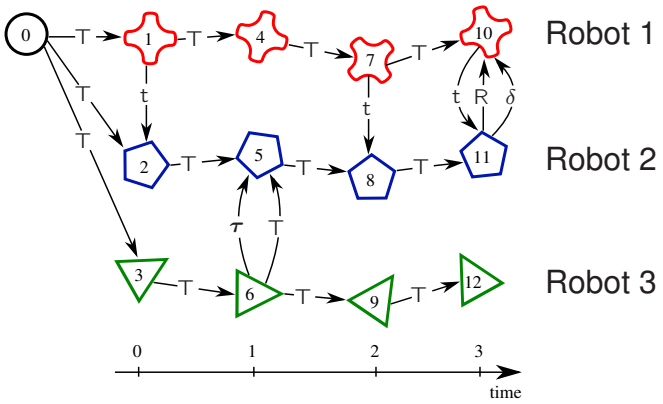


Figure 3-1. The graph corresponding to a group of 3 robots at time $k = 3$. Each (robot, time) pair is labeled with the corresponding node index from $\mathcal{V}_0(3)$. Arrows indicate edges in $\mathcal{E}(3)$, i.e., relative measurements. Each edge is labeled to indicate the type of measurement. Each robot had GPS and compass measurements at the initial time $k = 0$. Thereafter, no other GPS or compass measurements were available.

implicit assumption here is that a robot is able to uniquely identify another robot of which it obtains a relative measurement, so that there is no ambiguity on which pair of robots a relative measurement corresponds to.

The collaborative localization problem is the problem of estimating the pose of every robot at the current time k with respect to the common frame of reference by utilizing the inter-time and inter-robot measurements collected up to time k.

The situation above is best described in terms of a directed, time-varying, fully-labeled graph $\mathcal{G}(k) = (\mathcal{V}_0(k), \mathcal{E}(k), \ell(k))$, where nodes $\mathcal{V}_0(k)$ correspond to variables and edges $\mathcal{E}(k)$ to measurements, that shows how the noisy relative measurements relate to the absolute pose of each robot at every time step. The graph is defined as follows. For each robot $i \in \{1, \dots, r\}$ and each time $t \leq k$, a unique index (call it u) is assigned to the pair (i, t). How this indexing is done is immaterial. The indices $\{1, \dots, rk\}$ define a set $\mathcal{V}(k)$ that is a subset of the node set $\mathcal{V}_0(k)$ of the graph. We refer to the frame of reference attached to robot i at time t as frame u, where u is the node assigned to the pair (i, t). We introduce another node, denoted by 0, that corresponds to the common frame of reference in which every robot's pose is to be determined.

The node 0 is called the grounded node. The node set of the graph is then defined as $\mathcal{V}_0(k) := \mathcal{V}(k) \cup \{0\}$. The relative pose of frame u with respect to frame 0 is denoted simply by $\mathbf{T}_u$. We call the poses $\{\mathbf{T}_u\}_{u \in \mathcal{V}(k)}$ the node variables of $\mathcal{G}(k)$. Each relative pose (either measurement or node variable) $\mathbf{T}$ can be represented by a orientation and position pair, $\mathbf{R}, \mathbf{t}$. We will therefore freely refer to $\mathbf{T}_u$ and its corresponding orientation $\mathbf{R}_u$ or position $\mathbf{t}_u$ components as node variables.

The set of directed edges at time k, denoted $\mathcal{E}(k)$, corresponds to the noisy inter-time and inter-robot measurements collected up to time k. That is, suppose robot i is able to measure robot j's relative pose at time $\overline{k}$, and let u, v be the nodes corresponding to robots i, j at time $\overline{k}$, respectively. Then for all $k \geq \overline{k}$, there exists a directed edge $e \in \mathcal{E}(k)$ corresponding to this measurement. Since robot i measures robot j, the edge e leaves node u and arrives at v, we denote this by $e \triangleright (u, v)$. Similarly, each inter-time relative pose measurements of a robot also creates an edge in the graph. To delineate the type of measurement, a label from the set $\{$ $\mathbf{T}$ (pose), $\mathbf{R}$ (orientation), $\mathbf{t}$ (position), $\boldsymbol{\tau}$ (bearing), δ (distance) $\}$ is attached to each edge. The map from the set of edges to the set of labels is denoted by $\ell(k)$. For each edge $e \in \mathcal{E}(k)$ where $e \triangleright (u, v)$, if $\ell(k)(e) = s$ that indicates there is a measurement of type s for frame v with respect to frame u, where $s \in \{\mathbf{T}, \mathbf{R}, \mathbf{t}, \boldsymbol{\tau}, \delta\}$. The noisy relative measurement associated with edge $e \triangleright (u, v)$ is denoted by $\hat{\mathbf{T}}_{uv}, \hat{\mathbf{R}}_{uv}, \hat{\mathbf{t}}_{uv}, \hat{\boldsymbol{\tau}}_{uv}$, or $\hat{\delta}_{uv}$ for $\ell(k)(e) =$ $\mathbf{T}$ (pose), $\mathbf{R}$ (orientation), $\mathbf{t}$ (position), $\boldsymbol{\tau}$ (bearing), or δ (distance), respectively. A measurement of type $\mathbf{T}$ is really two measurements, one of type $\mathbf{R}$ and $\mathbf{t}$. We still use the nomenclature " measurement of type $\mathbf{T}$" for ease of comparison with prior work, since relative pose measurements are commonly considered in existing literature.

If the absolute pose in global GPS coordinates of at least one robot is known at time 0 through the use of a GPS and compass, then node 0 can be associated with a Terrestrial Reference Frame. When such measurements are not available, node 0

could correspond to the initial frame of reference of one of the robots. In either case, estimating the node variables is equivalent to determining the robots' poses with respect to frame of the grounded node 0. Without this grounded node, the problem of localization from relative measurements is indeterminate up to a rotation and a translation.

The graph $\mathcal{G}(k)$ is called the measurement graph at time k. Figure 3-1 shows an example of the graph corresponding to the measurements collected by 3 robots up to time index 3. Because each robot may be equipped with more than one sensor, multiple distinct edges may exist between a pair of nodes.

To ensure at least one estimate exists for every robot at each time k, we make the following assumption.

Assumption 3.1. *Each robot has access to an estimate of its absolute pose at time* 0*.*

Due to Assumption 3.1, an estimate of the pose of robot i at time time k (equivalently, the node variable $\mathbf{T}_u$, where node u corresponds to the pair (i, k)) can be computed by composing the inter-time relative pose measurements obtained by the robot i up to time k. This estimate is equivalent to robot i performing dead-reckoning. In practice, Assumption 3.1 holds if all robots have a GPS and compass measurement at time 0. If no GPS measurement is available, but each robot can obtain a relative pose measurement with respect to, say, robot 1, then again the assumption holds.

Often many more edges are present in the graph $\mathcal{G}(k)$ than those necessary to form a single estimate of the node variables; robots can benefit from collaborative localization in such a scenario. As an illustrative example, consider the example shown in Figure 3-1. The path $(0, 1, 4, 7, 10)$ provides the dead-reckoning estimate of robot 1 at time 3, or equivalently an estimate of $\mathbf{T}_{10}$. Similarly, the path $(0, 2, 5, 8, 11)$ provides as estimate for the node variable $\mathbf{T}_{11}$. Additionally, the three edges between nodes 10 and 11 corresponding to relative orientation, position, and distance measurements all provide additional information about how $\mathbf{T}_{10}$ and $\mathbf{T}_{11}$ relate. Because there is noise

in each relative measurement, by incorporating the information found in these three edges, we expect to get a better estimate of both $\mathbf{T}_{10}$ and $\mathbf{T}_{11}$. The goal of collaborative localization is to utilize all edges in the graph $\mathcal{G}(k)$ to improve over any estimates of the node variables that are obtained using only a single path. When the measurements are linearly related to the node variables, this can be accomplished by using the Best Linear Unbiased Estimator, as done in [15, 51]. In our case, the relationship between the measurements and node variables is nonlinear.

Remark 3.1. *The discussion above also indicates that Assumption 3.1 is not necessary, but merely sufficient for localization. For robot i to be localized, all that is needed is that there exists a time k so that there is an undirected path from node u (where u corresponds to (i, k)) to node 0 such that the edges along this path are of type $\mathbf{T}$, that is, correspond to measurements of the relative pose. In that case, an estimate of $\mathbf{t}_u$ can be obtained by concatenating the relative pose measurements along the path from 0 to u, and hence robot i is localized at time k. After that, even if no inter-robot relative measurements are available, robot i can perform dead reckoning. Assumption 3.1 is taken in order to simplify the presentation.*

3.1.2 The Distributed Collaborative Localization Problem

The problem stated above does not put any restriction on the access to the measurements collected up to time k. In particular, a method that assumes that all measurements collected by all the robots up to time k are instantly available to a central computer with no computation or memory constraints is allowed. However, when a large number of robots are involved, with communication accomplished through the use of low-bandwidth wireless links with limited range, such a centralized scheme is not feasible. In addition, retaining past measurements indefinitely will quickly exhaust both available memory and available processing capabilities of a centralized computing unit.

We now modify the problem by including constraints on communication and computation. In particular, a robot is now required to localize itself by using information

that is available from on-board sensors and data it can collect from its neighbors. Two robots i and j are said to be neighbors at time k if they can communicate at time k.

We now pose the distributed collaborative localization problem as follows: each robot is to localize itself by using measurements collected by on-board sensors and information it can obtain by communicating with its neighbors. To simplify the development of the distributed algorithm, the following assumption is made:

Assumption 3.2. *If robot i can obtain a relative measurement of a robot j at time k, then i and j can communicate at that time.*

Though this assumption may seem strict, it can, in fact, always be satisfied: robot i simply drops any measurements involving j if it cannot communicate with j.

3.2 Centralized Collaborative Localization Algorithm

In this section we present a solution to the collaborative localization problem where all the relative measurements are instantly available to a central processor at each time k. The centralized solution naturally leads to a distributed scheme, which will be described in the next section.

For ease of exposition, we first consider the case when every inter-robot relative measurement is a measurement of the relative pose. The algorithm is described in Section 3.2.1. Then in Section 3.2.2 we expand our algorithm to consider measurements of the relative pose, orientation, position, bearing, or distance between nodes.

3.2.1 Case A: Homogeneous Measurements (Relative Pose)

Instead of addressing the problem of estimating the robots' current poses at time k, we examine the more general problem of estimating all the node variables $\{\mathbf{T}_u\}_{u\in\mathcal{V}(k)}$ of the measurement graph $\mathcal{G}(k)$, using the noisy relative measurements, $\{\hat{\mathbf{T}}_{uv}\}_{(u,v)\in\mathcal{E}(k)}$.

We pose the collaborative localization problem as an optimization of a cost function over the set of node variables, where the cost function measures how well a given set of absolute poses explains the noisy relative measurements collected up to time k. The

initial condition for each node variable $\mathbf{T}_u$, $u \in \mathcal{V}(k)$ is given by the dead reckoning estimate, the existence of which is assured by Assumption 3.1.

To derive a suitable cost function, we break each pose (both noisy relative pose measurements and node variables) into its corresponding rotation $\mathbf{R} \in SO(3)$ and translation $\mathbf{t} \in \mathbb{R}^3$. In this work, a rotation $\mathbf{R} \in SO(3)$ is considered to be an abstract operator and not necessarily equated to its matrix representation. When the relative pose measurements are completely error free, $\hat{\mathbf{R}}_{uv}$ is the true relative rotation of frame v with respect to frame u. This rotation can also be expressed as $\mathbf{R}_u^T\mathbf{R}_v$, where $\mathbf{R}_u^T$ is the adjoint of the operator $\mathbf{R}_u$. Thus, if no noise were present, $\hat{\mathbf{R}}_{uv}$ would be equal to $\mathbf{R}_u^T\mathbf{R}_v$. Similarly, in the absence of noise, $\hat{\mathbf{t}}_{uv}$ and $\mathbf{R}_u^T(\mathbf{t}_v - \mathbf{t}_u)$ would be equal, since both are the relative translation of frame v with respect to frame u. Since noise is present in the measurements, how much $\hat{\mathbf{R}}_{uv}$ differs from $\mathbf{R}_u^T\mathbf{R}_v$ (and $\hat{\mathbf{t}}_{uv}$ from $\mathbf{R}_u^T(\mathbf{t}_v - \mathbf{t}_u)$) - measured by a suitable distance function - provides a measure of how a given set of node variables explains the noisy measurements. Distance between two translation, as elements of $\mathbb{R}^3$, is given by the 2-norm of the difference. To measure the distance between $p, q \in SO(3)$, we use a Riemannian distance $d(p, q)$:

$$d(p, q) = \sqrt{-\frac{1}{2}\,\mathrm{Tr}\left[\log^2(p^{-1}q)\right]}. \tag{3–1}$$

More details on this distance function can be found in Appendix B.1. The cost function at each time k is chosen as a sum of edge-costs over all edges (measurements):

$$f(\{\mathbf{T}_u\}_{u\in\mathcal{V}(k)}) := \sum_{(u,v)\triangleleft e\in\mathcal{E}(k)} c_e(\mathbf{R}_u, \mathbf{t}_u, \mathbf{R}_v, \mathbf{t}_v), \tag{3–2}$$

$$\text{where } c_e(\mathbf{R}_u, \mathbf{t}_u, \mathbf{R}_v, \mathbf{t}_v) := \frac{1}{2}\Big(d^2(\hat{\mathbf{R}}_{uv}, \mathbf{R}_u^T\mathbf{R}_v) + \|\hat{\mathbf{t}}_{uv} - \mathbf{R}_u^T(\mathbf{t}_v - \mathbf{t}_u)\|^2\Big) \tag{3–3}$$

where $\|\cdot\|$ denotes the Euclidean 2-norm.

If the relative pose measurements were completely error free, the minimum value of the cost function would be 0. By minimizing the cost function, we expect to find an improved estimate for the absolute pose of each robot over what can be found through

dead reckoning alone. The cost function in (3–3) is similar to one proposed in [52] for a static camera network; here the cost function changes with time.

Finding the minimum of a function defined over a vector space has been studied extensively. However the function $f(\cdot)$ in (3–3) is defined on a curved surface, specifically, the product Riemannian Manifold $(SO(3) \times \mathbb{R}^3)^{n(k)}$ where $n(k) = |\mathcal{V}(k)|$, the cardinality of the set $\mathcal{V}(k)$. One option for this optimization is to use a parameterization of the rotations using, say, 3×3 rotation matrices or unit quaternions, and then embedding the manifold in an vector space of higher dimension. Optimization techniques applicable to vector spaces can then be used, with the constraints on the parameterization of rotations appearing as Lagrange multipliers. This however, leads to an increase in the dimensionality of the optimization problem. Even when a parameterization is chosen that doesn't lead to an increase in dimensionality, such as Euler angles, Hopf Coordinates, or the axis-angle representation, the optimization step still requires more computation than what is possible when the geometry of $SO(3)$ is utilized. Additionally, such parameterizations have problems such as failing to be bijective in certain regions; see [53] for a discussion of the relevant parameterizations of $SO(3)$ and their associated problems. Instead, our goal is to find a provably correct algorithm that utilizes the geometry of the space without relying on any particular parameterization. We accomplish this through use of a gradient descent algorithm on the product manifold.

Gradient descent in a Riemannian manifold is analogous to gradient descent in a vector space in the following sense. Given a smooth real valued function f defined on a manifold M, the gradient of f at $p \in M$, denoted $grad\, f(p)$, is a vector in the tangent space of M at p, which is denoted by T_pM. Just as in Euclidean space, $grad\, f(p)$ points in the direction of greatest rate of increase of f. An explicit expression for the gradient of the cost function (3–3) is provided in the next theorem; the proof is in Appendix B.3.

Theorem 3.1. *The gradient of the cost function shown in (3–3) at* $p = (\mathbf{R}_1, \mathbf{t}_1, \dots, \mathbf{R}_{n(k)}, \mathbf{t}_{n(k)}) \in \left(SO(3) \times \mathbb{R}^3\right)^{n(k)}$ *is*

$$grad\, f(p) = \left(grad\, f(\mathbf{R}_1), grad\, f(\mathbf{t}_1), \dots, grad\, f(\mathbf{R}_{n(k)}), grad\, f(\mathbf{t}_{n(k)})\right),$$

where, for $u = 1, \dots, n(k)$,

$$grad\, f(\mathbf{R}_u) = -\mathbf{R}_u \Bigg(\sum_{(u,v)\in\mathcal{E}(k)} \left[\mathbf{R}_u^T(\mathbf{t}_v - \mathbf{t}_u)\hat{\mathbf{t}}_{u\,v}^T - \hat{\mathbf{t}}_{u\,v}(\mathbf{t}_v - \mathbf{t}_u)^T\mathbf{R}_u + \log(\mathbf{R}_u^T\mathbf{R}_v\hat{\mathbf{R}}_{u\,v}^T)\right] + \sum_{(v,u)\in\mathcal{E}(k)} \log(\mathbf{R}_u^T\mathbf{R}_v\hat{\mathbf{R}}_{v\,u}) \Bigg)$$

$$grad\, f(\mathbf{t}_u) = \sum_{(u,v)\in\mathcal{E}(k)} \left(\mathbf{t}_u + \mathbf{R}_u\hat{\mathbf{t}}_{u\,v} - \mathbf{t}_v\right) + \sum_{(v,u)\in\mathcal{E}(k)} \left(\mathbf{t}_u - \mathbf{R}_v\hat{\mathbf{t}}_{v\,u} - \mathbf{t}_v\right).$$

Minimizing a function f using gradient descent requires the current estimate to be updated during each iteration by moving in the direction of the negative gradient. In a vector space this is accomplished by simply subtracting $\eta\, grad\, f$ from the current estimate for some appropriate scalar η. On a Riemannian manifold, moving in the direction of $-grad\, f$ is accomplished through parallel transport. The parallel transport map at a point

$$p := (\mathbf{R}_1, \mathbf{t}_1, \dots, \mathbf{R}_{n(k)}, \mathbf{t}_{n(k)}) \in (SO(3) \times \mathbb{R}^3)^{n(k)}, \tag{3–4}$$

denoted by $\exp_p$, is given by

$$\exp_p(\xi) = \left(\mathbf{R}_1 \exp(\mathbf{R}_1^T\xi_{\mathrm{R}_1}), \mathbf{t}_1 + \xi_{\mathrm{t}_1}, \dots, \mathbf{R}_{n(k)} \exp(\mathbf{R}_{n(k)}^T\xi_{\mathrm{R}_{n(k)}}), \mathbf{t}_{n(k)} + \xi_{\mathrm{t}_{n(k)}}\right) \tag{3–5}$$

where $\xi = (\xi_{\mathrm{R}_1}, \xi_{\mathrm{t}_1}, \dots, \xi_{\mathrm{R}_{n(k)}}, \xi_{\mathrm{t}_{n(k)}})$ is an element of the tangent space $T_p\left[(SO(3) \times \mathbb{R}^3)^{n(k)}\right] = T_{\mathrm{R}_1}SO(3) \times \cdots \times T_{\mathrm{t}_{n(k)}}\mathbb{R}^3$, and the $\exp(\cdot)$ function appearing in the right hand side of (3–5) is the map $\exp : \mathbb{L}(\mathbb{R}^3) \to \mathbb{L}(\mathbb{R}^3)$ defined by $exp(X) = \sum_{k=0}^{\infty} \frac{X^k}{k!}$ for $x \in \mathbb{L}(\mathbb{R}^3)$. The derivation of (3–5) is provided in Appendix B.2. The gradient descent

law is

$$p_{t+1} = \exp_{p_t}(-\eta_t grad\, f(p_t)), \quad t = 0, 1, \dots,$$

where $\eta_t \geq 0$ is the step size for iteration t. The parameter η_t is chosen as the *Armijo step size* $\eta_t^{(A)} = \beta^{N_t}\alpha$, where N_t is the smallest nonnegative integer such that

$$f(p_t) - f(exp_{p_t}(\beta^{N_t}\alpha\, grad\, f(p_t))) \geq \sigma\beta^{N_t}\alpha \|grad\, f(p_t)\|, \tag{3–6}$$

for scalar tuning parameters $\alpha > 0$, $\beta, \sigma \in (0, 1)$. The norm $\|\cdot\|$ defined on the vector space $T_p\left[(SO(3) \times \mathbb{R}^3)^n\right]$ is given by

$$\|(\xi_{\mathrm{R}_1}, \xi_{\mathrm{t}_1}, \dots, \xi_{\mathrm{R}_{n(k)}}, \xi_{\mathrm{t}_{n(k)}})\|^2 = \sum_{u=1}^{n(k)} \left(\frac{1}{2}\,\mathrm{Tr}\left[\xi_{\mathrm{R}_u}^T \xi_{\mathrm{R}_u}\right] + \xi_{\mathrm{t}_u}^T \xi_{\mathrm{t}_u}\right).$$

which comes from the *Riemannian Metric*. More details on the Riemannian metric can be found in Appendix B.1. Theorem 4.3.1 in [54] grantees that the iterates p_t converges to a critical point of the cost function f defined in (3–7) as $t \to \infty$.

3.2.2 Case B: Heterogeneous Measurements

We now extend our consideration to the case when the inter-robot noisy relative measurements can be any combination of relative pose, orientation, position, bearing, and distance. The inter-time relative measurements are still of relative pose. To accommodate heterogeneous measurement types, we modify the cost function (3–3) so that it measures how well a given set of node variables fit each of the measurements. The new cost function is:

$$f(\{\mathbf{T}_u\}_{u\in\mathcal{V}(k)}) := \sum_{(u,v)\lhd e\in\mathcal{E}(k)} c_e(\mathbf{R}_u, \mathbf{t}_u, \mathbf{R}_v, \mathbf{t}_v), \tag{3–7}$$

where $c_e(\mathbf{R}_u, \mathbf{t}_u, \mathbf{R}_v, \mathbf{t}_v)$ is the cost for edge $e \triangleright (u, v)$ that is given by

$$c_e(\mathbf{R}_u, \mathbf{t}_u, \mathbf{R}_v, \mathbf{t}_v) := \frac{1}{2}\times \begin{cases} d^2(\hat{\mathbf{R}}_{uv}, \mathbf{R}_u^T\mathbf{R}_v) + \|\hat{\mathbf{t}}_{uv} - \mathbf{R}_u^T(\mathbf{t}_v - \mathbf{t}_u)\|^2 & \text{if } \ell(k)(e) = \mathbf{T} \\ d^2(\hat{\mathbf{R}}_{uv}, \mathbf{R}_u^T\mathbf{R}_v) & \text{if } \ell(k)(e) = \mathbf{R} \\ \|\hat{\mathbf{t}}_{uv} - \mathbf{R}_u^T(\mathbf{t}_v - \mathbf{t}_u)\|^2 & \text{if } \ell(k)(e) = \mathbf{t} \\ \|\left(\hat{\boldsymbol{\tau}}_{uv}\|\mathbf{t}_v - \mathbf{t}_u\|\right) - \mathbf{R}_u^T(\mathbf{t}_v - \mathbf{t}_u)\|^2 & \text{if } \ell(k)(e) = \boldsymbol{\tau} \\ \|\left(\hat{\delta}_{uv} - \|\mathbf{t}_v - \mathbf{t}_u\|\right)\|^2 & \text{if } \ell(k)(e) = \delta \end{cases} \tag{3–8}$$

The cost associated with full pose, orientation, and position measurements have been discussed in Section 3.2.1. The cost for edges associated with noisy relative bearing measurements is motivated by the fact that when no noise is present in the measurements, $\hat{\boldsymbol{\tau}}_{uv}\|\mathbf{t}_v - \mathbf{t}_u\|$ is the unit vector pointing from frame u to frame v expressed in frame u, as is $\mathbf{R}_u^T(\mathbf{t}_v - \mathbf{t}_u)$. Similarly, when no noise is present, $\hat{\delta}_{uv}$ is the distance between frame u and frame v, as is $\|\mathbf{t}_v - \mathbf{t}_u\|$. In the presence of noise, how much $\hat{\boldsymbol{\tau}}_{uv}\|\mathbf{t}_v - \mathbf{t}_u\|$ differs from $\mathbf{R}_u^T(\mathbf{t}_v - \mathbf{t}_u)$ (and $\hat{\delta}_{uv}$ from $\|\mathbf{t}_v - \mathbf{t}_u\|$) provides a measure of how a given set of node variables explains the noisy measurements.

We again minimize the cost function (3–7) through the use of a gradient descent algorithm. The gradient of (3–7) can be computed using techniques very similar to those used to compute the gradient of (3–3). The gradient of (3–7) is provided in Appendix B.4. As in the previous case, we take the dead-reckoning estimate as the initial guess for each node variable $\mathbf{T}_u$, $u \in \mathcal{V}(k)$. The pseudo-code of the gradient descent algorithm on the manifold $(SO(3) \times \mathbb{R}^3)^{n(k)}$ is given in Algorithm 1, where $\epsilon > 0$ is a user-specified accuracy threshold. Correctness of the Algorithm 1 (convergence to a critical point of the cost function f in (3–7)) follows from [54, Theorem 4.3.1].

It should be noted that the algorithm presented above is independent of the parameterization used to represent rotations. One could use unit quaternions, 3×3 rotation matrices, etc.

Algorithm 1: Riemannian Pose Graph Optimization

Input: $\mathcal{G}(k)$, all noisy measurements on $\mathcal{E}(k)$, an initial guess for each node variable $(\mathbf{R}_u, \mathbf{t}_u)$, $u \in \mathcal{V}(k)$.

Output: $\left\{(\hat{\mathbf{R}}_u, \hat{\mathbf{t}}_u)\right\}_{u \in \mathcal{V}(k)}$

$\hat{p} \rightarrow$ initial guess (p defined in (3–4));

repeat

  foreach $u \in \mathcal{V}(k)$ **do**

    Compute $grad\, f(\hat{\mathbf{R}}_u)$ and $grad\, f(\hat{\mathbf{t}}_u)$ for the cost f in (3–7) (as shown in Appendix B.4);

  end

  Determine $\eta^{(A)}$, the Armijo step size from (3–6) (with $\hat{p}$ for p_t);

  foreach $u \in \mathcal{V}(k)$ **do**

    $\hat{\mathbf{R}}_u \rightarrow \hat{\mathbf{R}}_u \exp\left(-\eta^{(A)} \hat{\mathbf{R}}_u^T grad\, f(\hat{\mathbf{R}}_u)\right)$;

    $\hat{\mathbf{t}}_u \rightarrow \hat{\mathbf{t}}_u - \eta^{(A)}\, grad\, f(\hat{\mathbf{t}}_u)$;

  end

until $\| grad\, f\left(\left\{(\hat{\mathbf{R}}_u, \hat{\mathbf{t}}_u)\right\}_{u \in \mathcal{V}(k)}\right) \| > \varepsilon$;

3.3 Distributed Algorithm

In this section we propose an algorithm for solving the distributed localization problem. The algorithm requires limited memory, processor power, and communication bandwidth for its execution.

For each robot i, let $N_i^{(+)}(k)$ denote the set of all robots $j \in \{1, \dots, r\}$ such that, at time k, robot i can obtain a relative measurement with respect to j. Similarly, let $N_i^{(-)}(k)$ denote the set of all robots $j \in \{1, \dots, r\}$ such that, at time k, robot j can obtain a relative measurement with respect to i. The neighbors of robot i at time k are then given by the set $N_i(k) = N_i^{(+)}(k) \cup N_i^{(-)}(k)$. Due to Assumption 3.2, robot i can communicate with its neighbors $N_i(k)$ during time k.

Consider the local measurement graph $\mathcal{G}_i(k) = (\mathcal{V}_i(k), \mathcal{E}_i(k), \ell_i(k))$ of robot i, whose node set is simply the neighbors of i at time k along with the grounded node 0 and i itself: $\mathcal{V}_i(k) = N_i(k) \cup \{0, i\}$. The edges of $\mathcal{G}_i(k)$ correspond to the inter-robot measurements at time k between i and its neighbors, along with an edge $e \triangleright (0, j)$ for each $j \in \mathcal{V}_i(k)$ (see Figure 3-2 for an example). Each node in the local measurement

graph $\mathcal{G}_i(k)$ is associated with an absolute pose of a robot at time k. No past poses belong to this graph. An edge $(p, q) \triangleleft e \in \mathcal{G}_i(k)$ (where $i = p$ or q) corresponds to an inter-robot relative measurement between robots p and q at time k. The additional edges $e \triangleright (0, j), j \in \mathcal{V}_i(k)$ correspond to the "initial" estimate of robot j's absolute pose at time k, denoted $\hat{\mathbf{T}}_{0j}(k)$. Each robot j obtains the estimate $\hat{\mathbf{T}}_{0j}(k)$ at time k by concatenating its pose estimate obtained at time $k-1$ with the noisy inter-time relative pose measurement describing its motion from $k-1$ to k. The estimate $\hat{\mathbf{T}}_{0j}(k)$ is then used as the measurement associated with edge $e \triangleright (0, j)$ and broadcast to each of that robots neighbors for inclusion in their local measurement graphs. The graph $\mathcal{G}_i(k)$ is a now a measurement graph since each edge has an associated noisy relative measurement. The edges $(0, j) \triangleleft e \in \mathcal{E}_i(k)$ for $j \in \mathcal{V}_i(k)$ ensures that Assumption 3.1 is satisfied for the local measurement graph $\mathcal{G}_i(k)$ for each i.

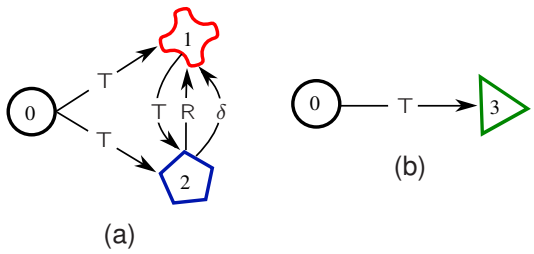


Figure 3-2. The local measurement graphs for the robots of Figure 3-1: (a) $\mathcal{G}_1(3)$ and $\mathcal{G}_2(3)$ (in this example they are the same) and (b) $\mathcal{G}_3(3)$.

The distributed algorithm works as follows. At each time k, every robot $i \in \{1, \ldots, r\}$ forms an initial estimate $\hat{\mathbf{T}}_{0i}(k)$ of its absolute pose as described above and obtains inter-robot relative measurements of each of its neighbors $j \in N_i^{(+)}(k)$. Robot i then transmits to each $j \in \mathcal{V}_i(k)$ its initial estimate of its absolute pose $\hat{\mathbf{T}}_{0i}(k)$, along with all inter-robot relative measurements between itself and j that it obtained at time k. Robot i receives in turn robot j's estimate of its current absolute pose $\hat{\mathbf{T}}_{0j}(k)$, along with relative measurements involving itself that j collected at that time. Robot i then executes Algorithm 1 on the local measurement graph $\mathcal{G}_i(k)$. The unknown node variables in this

graph consist of its own pose and the poses of its neighbors (all at time k). After the computation, only the estimate of i's own current pose is retained in its local memory; all other computed values are discarded. Since the distributed algorithm is simply Algorithm 1 applied to a local measurement graph, it inherits the correctness property of Algorithm 1. Note that if robot i has no neighbors at time k, the distributed collaborative localization algorithm is equivalent to performing self-localization from inter-time relative measurements.

The local measurement graphs $\mathcal{G}_1(3)$, $\mathcal{G}_2(3)$ and $\mathcal{G}_3(3)$ corresponding to the example of Figure 3-1 are shown in Figure 3-2. At time $k = 3$, robot 3 can see no other robots, so it will update its absolute pose using the inter-time relative pose measurement alone, without the aid of any inter-robot relative measurements. Robots 2 and 1, in contrast, will use the relative measurements between them obtained at time $k = 3$ to update their pose estimates.

Note that this treatment of the distributed algorithm is a special case of the sliding window approximation when the length of the window is set to 0. For the general sliding window approximation, see Section 6.3.

3.4 Simulation Results

In this section we present simulations comparing the centralized and distributed algorithms and study the effect of collaboration on localization accuracy. First, in Section 3.4.1, we consider the case that all inter-robot relative measurements are of the relative pose. We examine the difference in localization accuracy between the centralized and distributed algorithms, as well as the effect of increasing number of robots on localization accuracy. Then in Section 3.4.2 we consider the heterogeneous measurement case in which the inter-time relative measurements are of relative pose, but the inter-robot relative measurements may be of the relative pose, orientation, position, bearing, or distance. We examine the effects these various measurement types have on the accuracy of the location estimates.

The following definitions will be of use when evaluating the effectiveness of the various algorithms. Given a set of robots, the position estimation error of robot i is defined as $\mathbf{e}_i(k) := \hat{\mathbf{t}}_i(k) - \mathbf{t}_i(k)$, where $\mathbf{t}_i(k)$ is its absolute position at k and $\hat{\mathbf{t}}_i(k)$ is the estimate. The bias in the position estimation error of robot i is defined as $\| \mathrm{E}[\mathbf{e}_i(k)] \|$, where $\| \cdot \|$ is the 2-norm and $\mathrm{E}[\cdot]$ denotes expectation. The standard deviation is defined as $\sqrt{\mathrm{Tr}\left[Cov(\mathbf{e}_i(k), \mathbf{e}_i(k))\right]}$, where $Cov(\cdot)$ stands for covariance. Similarly, the orientation (estimation) error of robot i is defined as the scalar $\mathbf{e}_i^{\mathrm{R}}(k) := d\big(\hat{\mathbf{R}}_i, \mathbf{R}(k)\big)$, where $\hat{\mathbf{R}}_i(k), \mathbf{R}_i(k)$ are the estimated and true orientations of robot i at k and $d(\cdot, \cdot)$ is the distance function on $SO(3)$, defined in 3–1. The bias and standard deviation in the orientation estimation error are defined in the usual way. In each scenario described below, the bias and variance in position estimation error is estimated through the use of a Monte Carlo simulation.

3.4.1 All Measurements are of Relative Pose

To compare the centralized and distributed algorithms, we examine the localization of a group of 5 robots. All inter-robot relative measurements are of the relative pose. Each of the 5 robots travels along a distinct zig-zag path in 3-D, shown in Figure 3-3(a). Two robots can obtain relative pose measurements at time k if the Euclidean distance between them at that time is less than $7\mathrm{m}$. Due to assumption 3.2, communication between robots is possible between those pairs with a distance less than $7\mathrm{m}$. Furthermore, 25% of these potential measurements were dropped to simulate random failure. A plot of the number of neighbors of robot 1 over time is shown in Figure 3-3(b). The orientation measurements for each relative pose (both inter-robot and inter-time) were corrupted by independent identically distributed (i.i.d.) unit quaternions drawn from a Von Mises-Fisher distribution [55] centered around the zero-rotation quaternion and with a concentration parameter of $10,000$. Noise in the relative translation measurements was simulated by adding i.i.d zero-mean normal random variables with covariance matrix $I_{3\times 3} \times 10^{-6}$.

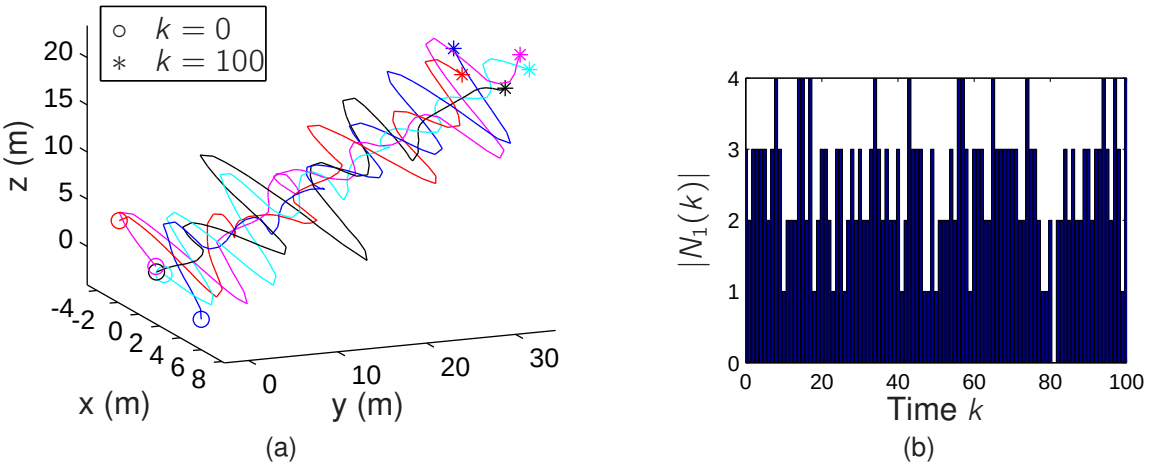


Figure 3-3. The trajectories and neighbor relations used in all simulations. In (a) the 3D trajectories for each the 5 robots are shown and (b) shows the number of neighbors for robot 1 as a function of time when all 5 robots are cooperating.

Figure 3-4 shows the bias and standard deviation in position error of a single robot acting in a 5 robot team, estimated using a 100-iteration Monte Carlo simulation. The group of robots performed localizing using either the centralized or distributed collaborative localization algorithms with all inter-robot measurements being of the relative pose. The plots indicate that the improvement in localization accuracy with the distributed algorithm is quite close to that with the centralized algorithm. This is promising since the distributed algorithm is applicable to large teams of robots in highly dynamic scenarios that can lead to arbitrary time variation in neighbor relationships. The centralized algorithm is not applicable in a realistic setting, but it provides a measure of the best performance possible using this algorithm. From this point forward, we will only study the distributed algorithm.

We next consider a group of robots utilizing the distributed collaborative localization algorithm, this time varying the number of robots in the group. In each of the sample runs of the Monte Carlo simulation, the robots traveled the same paths - those shown in Figure 3-3(a). Measurement noise was introduced in the same manner as in the previous simulations. Neighbor relations were again determined as described earlier,

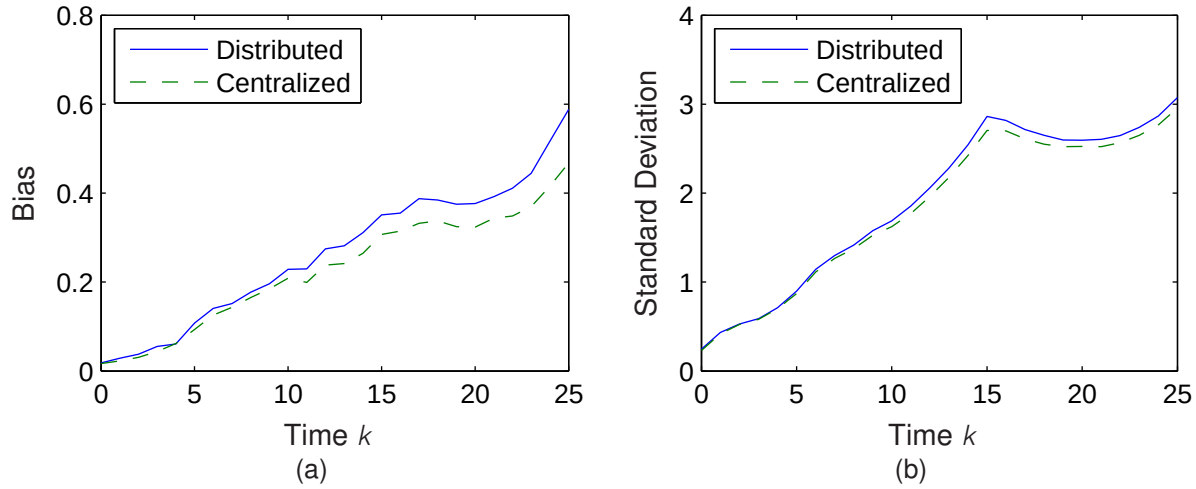


Figure 3-4. Comparison between the centralized and distributed collaborative localization algorithm. A Monte-Carlo simulations with 50 iterations was used to estimate the (a) bias and (b) standard deviation is position estimation error.

and kept the same from run to run to preclude that from being an additional source of randomness. Simulations for robot teams of size $1, 2, 3, 4$ and 5 were carried out. In each case $1,000$ iterations were performed. When only one robot is present in the team, collaborative localization is equivalent to self-localization without the aid of any inter-robot relative pose measurement. As the number of robots in the team increases, the number of neighbors for a robot at any given time will tend to increase and so greater improvement in localization accuracy is expected.

The bias and standard deviation in the position estimation error $\mathbf{e}_i(k)$ for robot 1 ($i = 1$) are shown in Figure 3-5. Both bias and standard deviation show significant improvement with distributed collaborative localization over self-localization. This is evident even for a team of only two robots. As the number of robots in the team increases, the localization error of robot 1 decreases. The improvement in accuracy however, shows a diminishing return with increasing team size. Similar results were seen when the bias and standard deviation in orientation estimation error was considered.

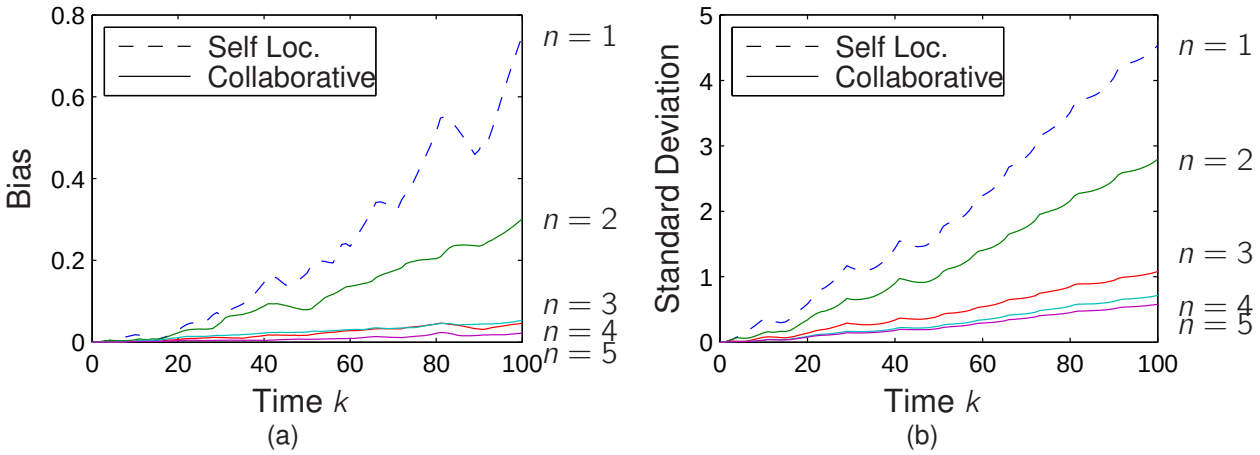


Figure 3-5. Simulation results studying the localization accuracy v.s. the number of robots. The sample (a) bias and (b) standard deviation in the position estimation error for robot 1 are found using a Monte-Carlo experiment when the distributed algorithm is used for communicating robot groups of size 1 (self localization), $2, 3, 4$ and 5.

3.4.2 Heterogeneous Measurements

We now perform simulations for the case when inter-robot relative measurements are again allowed to be of the relative pose, orientation, position, bearing, or distance. To examine the effect that measurement type has on localization accuracy, in each experiment, we let all inter-robot relative measurements be of the same type. Monte Carlo simulations were conducted in which we again consider the 5 robots traveling in the zig-zag paths shown in Figure 3-3(a). Errors in the pose, orientation and position measurements are introduced as described in Section 3.4.1. Noise in the bearing measurements is induced by rotating the true bearing through the application of a unit quaternion generated from an i.i.d. Von Mises-Fisher distribution. The noise in the distance measurements is normally distributed. The bias and standard deviation of position error for robot 1 are shown in Figure 3-6(a,b), while those of the orientation error are shown in Figure 3-6(c, d).

We see from the plots that improvement over self-localization occurs for all measurement types, with the exception of distance measurements. While distance

measurements improve the standard deviation of the position estimates, they have little or no effect on the bias. That distance has little effect on the accuracy is consistent with the conclusions in the study [17] for 2-D localization. The fact that bearing measurements lead to higher improvement over distance or position measurements was also observed in [17] for the 2-D case. While the conclusions in [17] were based on single simulations, ours are based on Monte Carlo simulations.

Trade-offs between cost of sensors and the resulting benefit in localization can be analyzed from these empirically observed trends. Though full pose provides the most benefit to localization accuracy, it is clear that any of the considered inter-robot measurement types can be used to improve localization accuracy over dead-reckoning. In particular, after relative pose, relative position seems to be the most valuable types of inter-robot measurements, leading to significant reduction in both bias and variance over dead reckoning. This means that the cost of having sensors capable of measuring relative position (stereo vision, laser range finder, or monocular camera based bearing sensor along with a RF-based distance measurement) may very well be justified by the localization accuracy they lead to. On the other hand, it is also apparent that the improvement due to inter-robot bearing measurements is quite comparable to that due to inter-robot relative position measurements. Yet, only a single camera is necessary to measure the bearing, where as (in general) stereo vision is necessary to measure the full relative position, which is still quite prone to large errors unless large baseline stereo is used. Thus, given cost, payload, and reliability constraints, monocular cameras might be a better choice than stereo vision. These conclusions come with the caveat that they have been drawn from one set of Monte Carlo simulations; more extensive studies are needed to establish how general these trends are.

3.5 Experimental Results

In this section we present results for experiments conducted with two Pioneer P3-DX robots; they are shown in Figure 3-7. Each robot was equipped with a calibrated

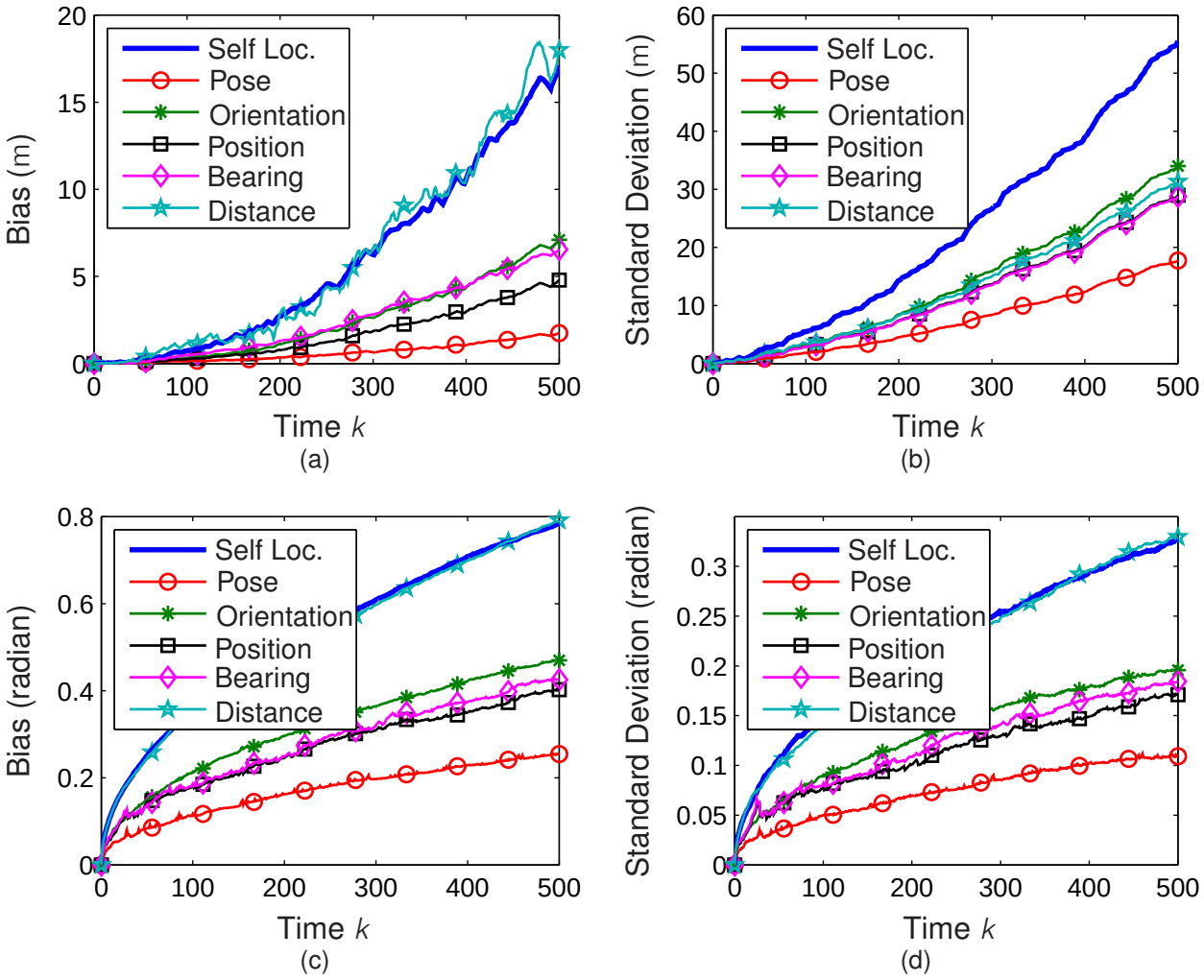


Figure 3-6. Simulation results studying the localization accuracy v.s. measurement type. The sample (a) bias and (b) standard deviation in position error and (c) bias and (d) standard deviation in orientation error for robot 1, found using a Monte-Carlo experiment, are shown for each measurement type when the distributed algorithm is used to perform localization. when the distributed The label "Self Loc." refers to a robot using dead reckoning to localize, without the use of any inter-robot measurements. The labels "Pose", "Orientation", "Position", "Bearing" and "Distance" indicate the group of robots used inter-robot measurements of the respective types to perform collaborative location using the distributed algorithm.

monocular Prosillica EC 1020 camera and wheel odometers. Measurements from these sensors were fused to obtain the noisy inter-time relative pose measurements. Each robot is additionally equipped with a target allowing the on-bard cameras to measure

Figure 3-7. Two Pioneer P3-DX robots equipped with cameras and targets. Robot 1 is shown on the left, while robot 2 is on the right.

the inter-robot relative pose by exploiting the known geometry of each target. The true pose of each robot was determined using an overhead camera capable of tracking each robot's target. The sensor suite was polled every 0.2 seconds with the noisy inter-robot relative pose measurements available at most, but not all, times.

All robots moved in straight lines with their paths approximately parallel. Six different pose estimates of the robots were obtained at each time. The first was a dead-reckoning estimate, obtained from the inter-time relative pose measurements alone. The remaining 5 estimates were obtained by using the distributed collaborative localization algorithm with the inter-robot noisy relative measurements being of full pose, orientation only, position only, bearing only, or distance only, respectively. These measurements were obtained by projecting the relative pose measurements onto the appropriate measurement types, then discarding the remaining information. The resulting global position estimates, along with the true positions, for robot 1 are reported in Figure 3-8. Simulations presented in Section 3.4 indicate that we should see a significant improvement in localization accuracy even in this small team, and the experimental results are consistent with that conclusion. As in the simulations, distinct improvement in localization accuracy is seen when collaborative localization is performed, irrespective of the type of inter-robot measurement, though the improvement varied depending on the measurement type.

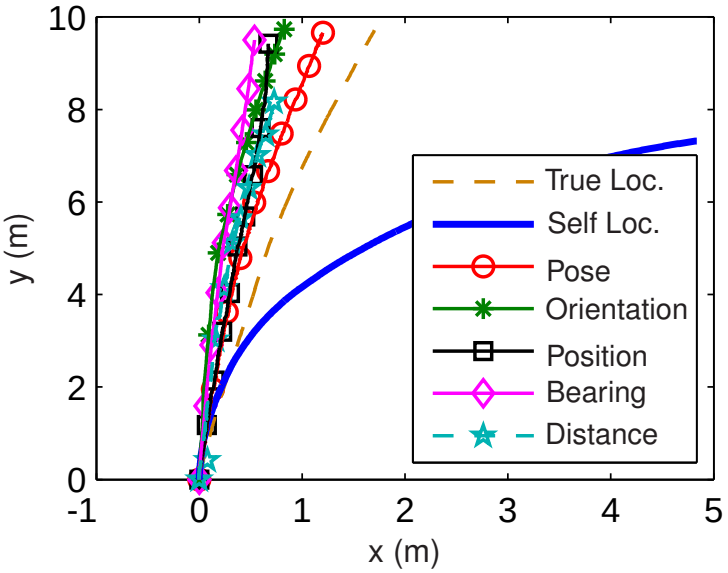


Figure 3-8. Experimental: A plot of the location of robot 1 in the overhead camera frame of reference when both robots move in a straight line. The true path (found using the overhead camera), estimated path using self localization, and estimated path using the distributed collaborative localization algorithm are all reported. The various curves correspond to the type of inter-robot relative measurement used.

To show that the improvements seen in the previous experiments are not just a random occurrence, the experiment was repeated 24 times to produce an empirical estimate of the bias and standard deviation in position error (by taking appropriate averages) for each type of inter-robot measurement. The results are reported in Figure 3-9. The experimental results again show that all measurement types lead to an increase in localization accuracy, with relative pose measurements leading to the maximum improvement, as expected. This is also the same trend that was observed in the simulations. In contrast to the simulations, here we see that distance measurements do lead to a non-negligible improvement in the bias of localization accuracy. As in the simulations, we see that both bearing and position measurements lead to similar improvement in the bias. However, in contrast to the simulations, in the experiments orientation measurements seem to improve the bias more than bearing or position measurements. The trend of the standard deviation improvement with

orientation, position and orientation measurements are similar in both simulation and experiments. The most significant difference between the trends observed in simulations and experiments are in the standard deviation improvement between pose, orientation, and position measurements: they are much closer in the experiments than in the simulations. In fact, the standard deviation with position or orientation measurements seem to be a little smaller than that with pose measurements. We believe this is an artifact of the small number of experimental samples averaged to estimate the bias and standard deviation empirically, which limits the accuracy of the estimates.

In short, the experiments verify that the proposed algorithm for distributed collaborative localization leads to statistically significant improvement in localization accuracy even with a small number of robots. Several trends seen in the experiments about the relative merits of the different types of measurements are consistent with those in the simulations. However, there are a few noticeable differences as well. Though the exact cause of these differences is not clear, one should note that the simulations and experiments differ in a number of ways. The biggest difference is that the bias and standard deviations estimated from the experiments are likely to have higher error than in the simulations due to the small number of experimental samples, which in turn is due to the difficulty associated with conducting repeated experiments. The other significant difference comes from the measurement noise distributions. In the simulations, measurement noise was drawn from distributions that were somewhat arbitrarily chosen, while the noise distributions in the experiments are unknown. The third potential source of difference is the paths of the robots. In [56], which examined the growth of dead-reckoning error, we showed that the path a robot traverses plays a crucial role in the bias and standard deviation of the localization error. In particular, the bias grows without bound if the robot moves in a straight line, but stays uniformly bounded by a constant if the robot stays inside a bounded region. Similar effects are likely in collaborative localization. Therefore, the difference between the paths used in

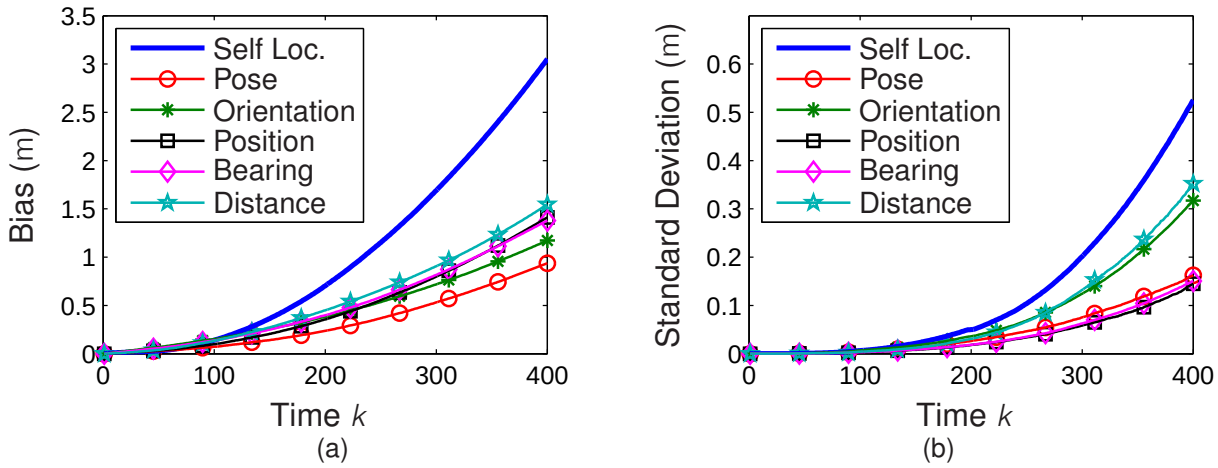


Figure 3-9. Experimental results studying the localization accuracy v.s. measurement type. The sample (a) bias and (b) standard deviation in position error for robot 1 are shown for each measurement type when the distributed algorithm is used to perform localization on a pair of robots moving in an approximately straight line. The bias and standard deviation were empirically estimated by averaging over 24 repeated trials.

the simulation and experiments might be another source of the difference in localization accuracy observed.

3.6 Summary

In this chapter we introduced a novel distributed algorithm for estimating the 3-D pose of multiple robots. The algorithm utilizing noisy inter-robot measurements of various types (relative pose, orientation, position, bearing, or distance) between pairs of robots, when available, along with the noisy inter-time relative pose measurements usually utilized for dead reckoning. The distributed algorithm is inspired by a centralized algorithm for solving a least-squares type problem in which the natural manifold structure of the space of rotations is utilized. The least-squares like cost function is chosen to measure how well the estimates explain the relative measurements. A gradient-descent in a product Riemannian manifold is used to solve the optimization problem ensuring the estimates remain on the manifold without the need for any projection. The proposed algorithm does not rely on any particular parameterization of the underlying manifold; allowing any parameterization to be used in numerical

computations without affecting the resulting estimates. The algorithm is provably correct in the sense that the solution converges to a critical point of the cost function as the number of iterations (gradient descent steps) increases.

CHAPTER 4
EXISTING METHODS OF COLLABORATIVE LOCALIZATION

In this chapter we compare the RPGO algorithm presented in Chapter 3 with existing methods used for collaborative localization. Two state-of-the-art algorithms in particular are considered. The first is referred to here as the Euclidean Pose Graph Optimization (EPGO) algorithm. This is based on standard pose graph optimization methods, see [57] and references therein.

The second algorithm considered utilizes an indirect extended Kalman filter (IEKF) to perform collaborative localization. Though the specific form of this algorithm is original work, it was developed in a similar manner as the IEKF observer for collaborative localization in [58]. We will refer to this as the IEKF algorithm.

In each case, as with the RPGO algorithm, the input to each algorithm is a fully labeled graph, an initial guess for the node variables, and a set of noisy relative measurements.

The remainder of this Chapter is organized as follows. In section 4.1 we review the JPL stand for unit quaternions as a parameterization of $SO(3)$. In Section 4.2 the EPGO algorithm is presented. Then in Section 4.3 the IEKF algorithm is presented. Simulations comparing these two algorithms with $RPGO$ are presented in Section 4.4.

4.1 Unit Quaternions

In this section, we will review the unit quaternion as it relate to the space $SO(3)$. Rather then using the standard definition of quaternions proposed by Hamilton, a more convenient definition is often used in the robotics community. Specifically, the standard convention proposed by JPL [59].

Under this convention, the unit quaternion is defined as

$$\mathbf{q} = q_1 i + q_2 j + q_3 k + q_4$$

where i, j, and k are hyperimaginary numbers such that

$$i^2 = j^2 = k^2 = -1, \quad ijk = 1.$$

We will often represent $\mathbf{q}$ by a column vector such that $\mathbf{q} = [q_1, q_2, q_3, q_4]^T$.

Multiplication on the set of unit quaternions, denoted by the symbol $\otimes$, is defined by

$$p \otimes q = \begin{bmatrix} p_4\bar{\mathbf{q}} + q_4\bar{\mathbf{p}} + \lfloor \bar{\mathbf{q}} \times \rfloor \bar{\mathbf{p}} \\ p_4 q_4 - \bar{\mathbf{p}}^T \bar{\mathbf{q}} \end{bmatrix}$$

where $\bar{\mathbf{q}} = [q_1, q_2, q_3]^T$ and $\lfloor \bar{\mathbf{q}} \times \rfloor$ is the cross product matrix given by

$$\lfloor \bar{\mathbf{q}} \rfloor = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}.$$

Unit quaternions are of interest to us because each unit quaternion maps to a unique element of $SO(3)$ in a smooth fashion. Given such a quaternion $\mathbf{q}$, the corresponding 3×3 rotation matrix is given by

$$C(\mathbf{q}) := (2q_4^2 - 1)\mathbf{I}_{3\times 3} - 2q_4\lfloor \bar{\mathbf{q}} \times \rfloor + 2\bar{\mathbf{q}}\bar{\mathbf{q}}^T$$

In the following sections, both the IEKF and EPGO algorithms will utilize unit quaternions as their parameterization of $SO(3)$.

4.2 The Euclidean Pose Graph Optimization Algorithm

The RPGO algorithm, first presented in Section 3.2, is a special case of a larger class of algorithms called pose graph optimization algorithms. In contrast to the RPGO algorithm, which utilizes a cost function that is not dependent on any particular parameterization of $SO(3)$, standard pose graph optimization techniques require such a suitable parameterization to be used. In this section we consider one such popular

parameterization given by the set of unit quaternions [59]. We refer to this algorithm as the Euclidean Pose Graph Optimization (EPGO) algorithm.

We again utilize the fully labeled, time varying graph presented in Section 3.1. For each node $u \in \mathcal{V}(k)$, let $\mathbf{q}_u$ denote the unit quaternion corresponding to the node variable $\mathbf{R}_u$. Similarly, given a orientation measurement $\hat{\mathbf{R}}_{uv}$, let $\hat{\mathbf{q}}_{uv}$ denote the corresponding unit quaternion.

Estimates for the node variables at time k are determined by minimizing a cost function

$$f(\{\mathbf{q}_u, \mathbf{t}_u\}_{u\in\mathcal{V}(k)}) := \sum_{(u,v)\triangleleft e\in\mathcal{E}(k)} g_e(\mathbf{q}_u, \mathbf{t}_u, \mathbf{q}_v, \mathbf{t}_v)^T P_e g_e(\mathbf{q}_u, \mathbf{t}_u, \mathbf{q}_v, \mathbf{t}_v) \tag{4–1}$$

where the positive definite matrix P_e is a scaling matrix and $g_e(\mathbf{q}_u, \mathbf{t}_u, \mathbf{q}_v, \mathbf{t}_v)$ is a suitable vector edge error defined for each measurement type. Though standard pose graph optimization is able to handle all measurement types considered thus far, for ease of exposition, only measurements of the relative position and orientation will be considered in the following discussion.

Given the unit quaternion parameterization, no canonical vector edge error exists. Instead, many choices are possible. One suitable choice, which is the vector edge error used in the subsequent comparisons, is given as follows:

$$g_e(\mathbf{q}_u, \mathbf{t}_u, \mathbf{q}_v, \mathbf{t}_v) = \begin{cases} \mathbf{q}_u^{-1} \otimes \mathbf{q}_v \otimes \hat{\mathbf{q}}_{uv}^{-1} - 1 & \text{if } \ell(k)(e) = \mathbf{R} \\ C(\mathbf{q}_u)^T(\mathbf{t}_v - \mathbf{t}_u) - \hat{\mathbf{t}}_{uv} & \text{if } \ell(k)(e) = \mathbf{t} \end{cases} \tag{4–2}$$

where $e \triangleright (u, v)$, $\otimes$ denotes quaternion multiplication as defined in Section 4.1, and C denote the map that takes a unit quaternion to its corresponding 3×3 rotation matrix representation.

Many vector-space optimization algorithms can be used to search for the minimum of such a cost function. One common choice is to use Levenberg-Marquadt, as was done in [57]. To implement Levenberg-Marquadt, a minimal parameterization is utilized.

Given a unit quaternion $\mathbf{q}_u$, we denote the corresponding minimal parameterization as $\overline{\mathbf{q}_u}$ such that $\left[\overline{\mathbf{q}_u}^T \mathbf{q}_u^4\right]^T = \mathbf{q}_u^T$. In addition, a new operator $\boxplus : \text{Dom}(\mathbf{q}_u) \times \text{Dom}(\overline{\mathbf{q}_u}) \rightarrow \text{Dom}(\mathbf{q}_u)$ is defined by

$$\mathbf{q}_u \boxplus \overline{\mathbf{q}_v} = \mathbf{q}_v \otimes \mathbf{q}_u. \tag{4–3}$$

To simplify the discussion, the parameterized node variables, minimally parameterized node variables, and vector edge error, and scaling matrices are all stacked as follows: $\mathbf{X} = \left[\mathbf{t}_1^T, \mathbf{q}_1^T, \dots, \mathbf{t}_n^T, \mathbf{q}_n^T\right]^T$, $\mathbf{\Delta X} = [\mathbf{t}_{1'}^T, \overline{\mathbf{q}_{1'}}^T, \dots, \mathbf{t}_{n'}^T, \overline{\mathbf{q}_{1'}}^T]^T$, $g = [g_1^T, \dots, g_m^T]^T$, $P = \text{diag}(P_1, \dots P_m)$, where for all $i \in \mathcal{V}(k)$, $\mathbf{t}_{i'} \in \mathbb{R}^3$, $\overline{\mathbf{q}_{i'}} \in Dom(\overline{\mathbf{q}_i})$. Finally, we extend the $\boxplus$ operator to act on the new stacked state vectors as

$$\mathbf{X} \boxplus \mathbf{\Delta X} = [(\mathbf{X}_1 \boxplus \mathbf{\Delta X}_1)^T, \dots, (\mathbf{X}_{2n} \boxplus \mathbf{\Delta X}_{2n})^T]^T$$

where the $\boxplus$ is defined for unit quaternions in (4–3) and reduces to the addition operator for position vectors. The Jacobian of the stacked vector error function g is given by

$$J = \left[\frac{\partial g_i(\mathrm{X} \boxplus \mathbf{\Delta X})}{\partial \mathbf{\Delta X}_j} \Big|_{\mathbf{\Delta X}=0} \right]_{ij}$$

For the measurement types considered, the Jacobian is explicitly given as follows. Consider an edge $e \triangleright (u, v)$ corresponding to an orientation measurement. To simplify the notation, let $\mathbf{q} = \mathbf{q}_u^{-1}$ and $\mathbf{p} = \mathbf{q}_v \otimes \hat{\mathbf{q}}_{uv}^{-1}$. Then

$$\frac{\partial g_e(\mathbf{X} \boxplus \mathbf{\Delta X})}{\partial \Delta \mathbf{q}_h}\Big|_{\Delta \mathrm{X}=0} = \begin{bmatrix} p_4 \lfloor \overline{\mathbf{q}} \times \rfloor - p_4 q_4 id - q_4 \lfloor \overline{\mathbf{p}} \times \rfloor + \lfloor \overline{\mathbf{p}} \times \rfloor \lfloor \overline{\mathbf{q}} \times \rfloor \\ p_4 \overline{\mathbf{q}}^T + q_4 \overline{\mathbf{p}}^T - \overline{\mathbf{p}}^T \lfloor \overline{\mathbf{q}} \times \rfloor \end{bmatrix} I_{u,v}(h)$$

$$\frac{\partial g_e(\mathbf{X} \boxplus \mathbf{\Delta X})}{\partial \Delta \mathbf{t}_h}\Big|_{\mathbf{\Delta X}=0} = 0$$

Where $I_{u,v}(h) = 1$ if $h = u$, -1 if $h = v$, 0 otherwise, and $\lfloor \mathbf{v}\times \rfloor$ is the cross product matrix given by

$$\lfloor \bar{\mathbf{v}}\times \rfloor = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}$$

for $\mathbf{v} = [v_1\ v_2\ v_3]^T$. Similarly, when e corresponds to a position measurement we find

$$\left.\frac{\partial g_e(\mathbf{X} \boxplus \mathbf{\Delta X})}{\partial \Delta \mathbf{q}_h}\right|_{\mathbf{\Delta} \mathrm{X}=0} = -2\lfloor (\mathbf{t}_v - \mathbf{t}_u)\times \rfloor I_u(h)$$
$$\left.\frac{\partial g_e(\mathbf{X} \boxplus \mathbf{\Delta X})}{\partial \Delta \mathbf{t}_h}\right|_{\mathbf{\Delta} \mathrm{X}=0} = -I_{u,v}(h) C(\mathbf{q}_u)^T$$

Where I_u is the indicator function.

Levenberg-Marquadt is then carried out in the usual fashion, setting $H = J^T P J$, $b = -J^T P g(\mathbf{X})$. Fixing a time k, at each iteration of the Levenberg-Marquadt algorithm, the update state vector $\mathbf{\Delta X}$ is given by solving the linear equation $(H + \lambda id)\mathbf{\Delta X} = b$ for a suitably large λ. The current estimate of $\mathbf{X}$ is then updated by setting $\mathbf{X} \leftarrow \mathbf{X} \boxplus \mathbf{\Delta X}$. This process is repeated until $\|\mathbf{\Delta X}\|$ is appropriately small.

Remark 4.1. *The orientations ($\mathbf{R}_{(\cdot)}$) that appear in our problem formations and through out the development of the RPGO algorithm in Chapter 3 are abstract rotation operators, or elements of $SO(3)$. No particular parameterization (quaternions, rotation matrices, etc.) is assumed during the development. In contrast, the cost function for the EPGO algorithm, (4–2) utilizes a particular parameterization in the form of unit quaternions. A different parameterization would necessarily lead to a different cost function, and perhaps a different estimate delivered by the corresponding algorithm. Thus, the estimates found using the EPGO algorithm are a function of both the measurements and the chosen parameterization, rather then a function of the measurements alone.*

4.3 The Implicit Extended Kalman Filter Algorithm

The IEKF algorithm presented blow is similar to that found in [58]. The two primary differences are as follows: (i) Inter-time relative measurements are of the relative pose, rather then measurements of the linear and angular velocities. This is motivated by the idea that measurements are captured by a camera, rather then an inertial measurement unit. (ii) The IEKF algorithm utilizes measurements of the relative position between robots, where as the algorithm presented in [58] uses measurements of the bearing with respect to a time varying set of feature points. Under these changes, the IEKF algorithm is able to function under the same circumstances as the RPGO algorithm.

The IEKF algorithm utilizes an indirect EKF. Also sometimes referred to as an error-state EKF. A brief review of the indirect EKF is provided below.

4.3.1 The Implicit (Error-State) Extended Kalman Filter

Consider a state vector $\mathbf{X}_k$ with the (possibly) non-linear state transition and observation models

$$\mathbf{X}_{k+1} = f(\mathbf{X}_k) + g(\eta_k) \tag{4–4}$$

$$\mathbf{z}_{k+1} = h(\mathbf{X}_{k+1}) + \xi_{k+1} \tag{4–5}$$

where η_k and ξ_k are assumed to be white, zero mean, and mutually independent, with covariance Q_k and R_k respectively.

Let $\hat{\mathbf{X}}_{k|k\prime}$ denote an estimate of the state at time k given all measurements up to time $k\prime$ obtained directly from those measurements (such as integration of IMU data). The error-state is defined as

$$\tilde{\mathbf{X}}_k = \mathbf{X}_k - \hat{\mathbf{X}}_k. \tag{4–6}$$

In the following, $\hat{\mathbf{X}}_{k+1|k}, \hat{\tilde{\mathbf{X}}}_{k+1|k}$ denotes a prediction of the state at time $k+1$ based on measurements upto time k. Similarly, $\hat{\mathbf{X}}_{k+1|k+1}, \hat{\tilde{\mathbf{X}}}_{k+1|k+1}$ denotes the estimated state after all measurements at time $k+1$ have been considered.

The propagation step of the indirect EKF is as follow.

$$\hat{\mathbf{X}}_{k+1|k} = f(\hat{\mathbf{X}}_{k|k}) \tag{4–7}$$

$$\hat{\tilde{\mathbf{X}}}_{k+1|k} = f(\hat{\tilde{\mathbf{X}}}_{k|k} + \hat{\mathbf{X}}_{k|k}) - f(\hat{\mathbf{X}}_{k|k}) \tag{4–8}$$

$$P_{k+1|k} = FP_{k|k}F^T + GQ_kG^T \tag{4–9}$$

where $P_{k|k}$ is the covariance estimate of $\mathbf{X}_k$ and thus of $\tilde{\mathbf{X}}_k$ given all measurements up to time k, and F, G are the Jacobian matrices

$$F = \left.\frac{\partial f(\mathbf{X})}{\partial \mathbf{X}}\right|_{\mathrm{X}=\hat{\mathrm{X}}_{k|k}} \qquad G = \left.\frac{\partial g(\eta)}{\partial \eta}\right|_{\eta=0}. \tag{4–10}$$

By construction, we will show that for all k, $\hat{\tilde{\mathbf{X}}}_{k|k} = 0$. Combining this fact with (4–9) gives $\hat{\tilde{\mathbf{X}}}_{k+1|k} = 0$.

Given a measurement at time k, z_k, the error-state measurement is defined as $\tilde{z}_k = z_k - \hat{z}_k$ where $\hat{z}_k = h(\hat{\mathbf{X}}_k)$. The linearized error-state measurement is then given by

$$\tilde{z}_{k+1} = H_{k+1}\tilde{\mathbf{X}}_{k+1} + \xi_{k+1}, \tag{4–11}$$

where

$$H_{k+1} = \left.\frac{\partial h(\mathbf{X})}{\partial \mathbf{X}}\right|_{\mathrm{X}=\hat{\mathrm{X}}_{k|k}}.$$

Let $S_{k+1} = H_{k+1}P_{k+1|k}H_{k+1}^T + R_{k+1}$. The Kalman gain is $K_{k+1} = P_{k+1|k}H_{k+1}^T S_{k+1}^{-1}$ and the update state in $\Delta\mathbf{X} = K\tilde{z}$. The indirect EKF state is then updated to include the measurement z by

$$\begin{aligned}
\hat{\tilde{\mathbf{X}}}'_{k+1|k+1} &= \hat{\tilde{\mathbf{X}}}_{k+1|k} + \Delta\mathbf{X} \\
&= \Delta\mathbf{X} \\
P_{k+1|k+1} &= (I - K_{k+1}H_{k+1})P_{k+1|k}(I - K_{k+1}H_{k+1})^T + K_{k+1}R_{k+1}K_{k+1}^T.
\end{aligned}$$

The prime denotes that this is only temporarily the updated state, as $\hat{\tilde{\mathbf{X}}}'_{k+1|k+1}$ need not equal zero. To avoid the need to propagate both an estimate of $\mathbf{X}$ and the error-state, a reset of of the error-state is necessary. We reset the error-state by including the information it holds in the estimate $\hat{\mathbf{X}}_{k+1|k+1}$. Note that $\hat{\tilde{\mathbf{X}}}'_{k+1|k} + \hat{\mathbf{X}}_{k+1|k}$ is its self an estimate of the state $\mathbf{X}$. Any update to the error-state estimate should not change the total state estimate. This leads to the following definition

$$\hat{\mathbf{X}}_{k+1|k+1} = \hat{\tilde{\mathbf{X}}}'_{k+1|k+1} + \hat{\mathbf{X}}_{k+1|k}.$$

This definition ensures that $\hat{\tilde{\mathbf{X}}}_{k+1|k+1} = 0$.

Two important facts arise from the proposed reset. The first is that the covariance estimate of the error-state is unchanged by the reset. The second is that the error-state estimate need never be explicitly computed. The update and reset are combined to give

$$\hat{\mathbf{X}}_{k+1|k+1} = \hat{\mathbf{X}}_{k+1|k} + \Delta\mathbf{X}. \tag{4–12}$$

4.3.2 Collaborative Localization Using the Indirect EKF

As with the EPGO algorithm, and in contrast to the RPGO algorithm, the IEKF algorithm requires a suitable parameterization of $SO(3)$ to be used. Unit quaternions are again chosen for this parameterization. To remain consistent with the notation used in the literature, a rotation will be parameterized by its inverse quaternion. That is, given a node variable $\mathbf{R}_u$ where u maps to time k and robot i, the corresponding quaternion $\mathbf{q}_i^k$ is given by

$$\mathbf{q}_i^k = C(\mathbf{R}_u^T) \tag{4–13}$$

where C denotes the map that takes a unit quaternion to its corresponding 3×3 rotation matrix representation.

The previous definition introduces a new notation, disjoint from the rest of the chapter, but necessary to properly describe the IEKF algorithm. For the remainder of

this section, unless specifically noted otherwise, superscript notation will denote time and subscript will denote robot index.

The state vector for robot i at time k is given by

$$\mathbf{X}_i^k = [(\mathbf{q}_i^k)^T \, (\mathbf{t}_i^k)^T]^T \tag{4–14}$$

where $\mathbf{t}_i^k$ is the position of robot i at time k and $\mathbf{q}_i^k$ is the orientation of robot i and time k as defined in (4–13). The full state vector $\mathbf{X}^k$ is given by stacking the state vector for each robot and $\hat{\mathbf{q}}_i^k$, $\hat{\mathbf{t}}_i^k$, $\hat{\mathbf{X}_i^k}$, $\hat{\mathbf{X}^k}$ denote estimates of the corresponding true states.

The error state vector for robot i at time k corresponding to (4–14) is given by

$$\tilde{\mathbf{X}}_i^k = [(\boldsymbol{\delta\theta}_i^k)^T \, (\tilde{\mathbf{t}}_i^k)^T]^T \tag{4–15}$$

where $\boldsymbol{\delta\theta}_i^k$ is the angle-error vector defined by the error quaternion $\tilde{\mathbf{q}}_i^k = \mathbf{q}_i^k \otimes (\hat{\mathbf{q}}_i^k)^{-1} \approx [\frac{1}{2}(\boldsymbol{\delta\theta}_i^k)^T \; 1]^T$. This approximation is made by assuming that $d(id, C(\tilde{\mathbf{q}}_i^k))$ is sufficiently small. The state propagation model is chosen to correspond with the kinematic model when inter-time relative position and orientation measurements are available,

$$\mathbf{q}_i^{k+1} = (\mathbf{q}_i^{k,k+1})^{-1} \otimes \mathbf{q}_i^k \tag{4–16}$$

$$\mathbf{t}_i^{k+1} = \mathbf{t}_i^k + C(\mathbf{q}_i^k)^T \, \mathbf{t}_i^{k,k+1} \tag{4–17}$$

where $\mathbf{q}_i^{k,k+1}$, $\mathbf{t}_i^{k,k+1}$ denote the change in orientation and position respectively for robot i between time k and time $k+1$. The measured change in orientation and position for robot i is modeled as $\hat{\mathbf{q}}_i^{k,k+1} = \tilde{\mathbf{q}}_i^{k,k+1} \otimes (\mathbf{q}_i^{k,k+1})^{-1}$ and $\hat{\mathbf{t}}_i^{k,k+1} = \mathbf{t}_i^{k,k+1} + \tilde{\mathbf{t}}_i^{k,k+1}$ where $\tilde{\mathbf{q}}_i^{k,k+1} \approx [\frac{1}{2}(\boldsymbol{\delta\theta}_i^{k,k+1})^T \; 1]^T$. Here $\boldsymbol{\delta\theta}_i^{k,k+1}$ and $\tilde{\mathbf{t}}_i^{k,k+1}$ are assumed to be zero-mean white Gaussian noise processes. The linearized error state equation is then given by $\tilde{\mathbf{X}}^{k+1} = F^k \tilde{\mathbf{X}}^k + G^k[(\boldsymbol{\delta\theta}_1^{k,k+1})^T \; (\tilde{\mathbf{t}}_1^{k,k+1})^T \ldots (\boldsymbol{\delta\theta}_n^{k,k+1})^T \; (\tilde{\mathbf{t}}_n^{k,k+1})^T]^T$ for $F^k = \text{diag}(F_i^k)$,

$G^k = \text{diag}(G_i^k)$ and

$$F_i^k = \begin{bmatrix} \mathbf{I}_3 & -C(\hat{\mathbf{q}}_i^k)^T \lfloor \hat{\mathbf{t}}_i^{k,k+1} \times \rfloor \\ 0 & C(\hat{\mathbf{q}}_i^{k,k+1})^T \end{bmatrix}$$

$$G_i^k = \begin{bmatrix} -C(\hat{\mathbf{q}}_i^k)^T & 0 \\ 0 & C(\hat{\mathbf{q}}_i^{k,k+1})^T \end{bmatrix}.$$

Where $\lfloor \mathbf{v} \times \rfloor$ is defined as in Section 4.2.

At each time step, the noisy measurements of the relative change in position and orientation are used to update the state estimate and covariance as

$$\hat{\mathbf{X}}_i^{k+1|k} = \begin{bmatrix} (\hat{\mathbf{q}}_i^{k,k+1})^{-1} \otimes \hat{\mathbf{q}}_i^k \\ \hat{\mathbf{t}}_i^k + C(\hat{\mathbf{q}}_i^k)^T \hat{\mathbf{t}}_i^{k,k+1} \end{bmatrix}$$

and

$$P^{k+1|k} = F^k P^k (F^k)^T + G^k Q^k (G^k)^T$$

where $Q^k = \text{E}[[(\boldsymbol{\delta\theta}_1^{k,k+1})^T \ (\tilde{\mathbf{t}}_1^{k,k+1})^T \ldots (\boldsymbol{\delta\theta}_n^{k,k+1})^T \ (\tilde{\mathbf{t}}_n^{k,k+1})^T]^T \ [(\boldsymbol{\delta\theta}_1^{k,k+1})^T \ (\tilde{\mathbf{t}}_1^{k,k+1})^T \ldots (\boldsymbol{\delta\theta}_n^{k,k+1})^T \ (\tilde{\mathbf{t}}_n^{k,k+1})^T$
Here the superscript $k+1|k$ is used to indicate this is a temporary estimate of the state at time $k+1$ utilizing only the inter-robot relative measurements available up to time k. If no inter-robot measurements are available at time $k+1$, then $\hat{\mathbf{X}}_i^{k+1} = \hat{\mathbf{X}}_i^{k+1|k}$, $P^k = P^{k+1|k}$. When inter-robot measurements are available, they are utilized as follows. Given an edge $e_{ij} \in \mathcal{E}(k)$ corresponding to an inter-robot measurement at time k from robot i to j, the measurement is modeled by

$$\mathbf{z}_{e_{ij}}^k = C(\mathbf{q}_i^k)(\mathbf{t}_j^k - \mathbf{t}_i^k) + \zeta_{e_{ij}}$$

The noise $\zeta_{e_{ij}}$ is assumed to be a zero-mean, normally distributed with $\text{E}[\zeta_{e_{ij}}(\zeta_{e_{ij}})^T] =: R^k$. The linearized error-measurement model is then given

$$\tilde{\mathbf{z}}_{e_{ij}}^k = \mathbf{z}_{e_{ij}}^k - \hat{\mathbf{z}}_{e_{ij}}^k \approx H_{e_{ij}}^k \tilde{\mathbf{X}}^k + \zeta_{e_{ij}}$$

where

$$\hat{\mathbf{z}}^k_{e_{ij}} := C(\hat{\mathbf{q}}^k_i)\left(\hat{\mathbf{t}}^k_j - \hat{\mathbf{t}}^k_i\right)$$
$$H^k_{e_{ij}} = \mathbf{e}_i^T \odot H^k_i + \mathbf{e}_j^T \odot H^k_j.$$

In the previous equation $\odot$ refers to the Kronecker product, $\mathbf{e}_i$ indicates the ith standard basis vector, and H^k_i, H^k_j are

$$H^k_i = \begin{bmatrix} -C(\hat{\mathbf{q}}^k_i) & \lfloor C(\hat{\mathbf{q}}^k_i)\left(\hat{\mathbf{t}}^k_j - \hat{\mathbf{t}}^k_i\right)\times \rfloor \end{bmatrix}$$
$$H^k_j = \begin{bmatrix} C(\hat{\mathbf{q}}^k_i) & 0 \end{bmatrix}.$$

Finally, given the set of measurements $\left\{\mathbf{z}^k_{e_1}, \cdots, \mathbf{z}^k_{e_m}\right\}$ acquired at time k, the total error-measurement $\tilde{\mathbf{z}}^k$ and error-measurement matrix H^k are given by stacking all the error-measurements $\tilde{\mathbf{z}}^k_{e_{ij}}$ and matrices $H^k_{e_{ij}}$ respectively.

The IEKF is then implemented as follows. Let $S^{k+1} = H^{k+1}P^{k+1|k}(H^{k+1})^T + R^{k+1}$. The Kalman gain is $K^{k+1} = P^{k+1|k}(H^{k+1})^T(S^{k+1})^{-1}$ and the update state is $\Delta\mathbf{X}^k = K^{k+1}(z^{k+1} - H^{k+1}\hat{\mathbf{X}}^{k+1|k})$. The IEKF state is then updated to include the measurement z by

$$\hat{\mathbf{X}}^{k+1} = \hat{\mathbf{X}}^{k+1|k} \oplus \Delta\mathbf{X}^{k+1}$$
$$P^{k+1|k+1} = (I - K^{k+1}H^{k+1})P^{k+1|k}(I - K^{k+1}H^{k+1})^T + K^{k+1}R^{k+1}(K^{k+1})^T.$$

where $\oplus$ is defined component wise as follows.

$$\hat{\mathbf{X}}^{k+1|k}_i \oplus \Delta\mathbf{X}^{k+1}_i =: \begin{bmatrix} \hat{\mathbf{q}}^{k+1|k}_i \\ \hat{\mathbf{t}}^{k+1|k}_k \end{bmatrix} \oplus \begin{bmatrix} \Delta\theta^{k+1}_i \\ \Delta\mathbf{t}^{k+1}_i \end{bmatrix} = \begin{bmatrix} \Delta\mathbf{q}^{k+1}_i \otimes \hat{\mathbf{q}}^{k+1|k}_i \\ \hat{\mathbf{t}}^{k+1|k}_i + \Delta\mathbf{t}^{k+1}_i \end{bmatrix}$$

where

$$\Delta\mathbf{q}^{k+1}_i = \begin{bmatrix} \frac{1}{2}(\Delta\theta^{k+1}_i)^T & \left(1 - \frac{1}{4}\Delta\theta^{k+1}_i\right)^{1/2} \end{bmatrix}^T.$$

4.4 Simulation Results

In this section we present simulations comparing the collaborative localization algorithm first presented in Chapter 3 against the two state of the art competing collaborative localization algorithms presented in Section 4.2 and Section 4.3.1.

The following definitions will be of use when evaluating the effectiveness of the various algorithms. Given a set of robots, the position estimation error of robot i is defined as $\mathbf{e}_i(k) := \hat{\mathbf{t}}_i(k) - \mathbf{t}_i(k)$, where $\mathbf{t}_i(k)$ is its absolute position at k and $\hat{\mathbf{t}}_i(k)$ is the estimate. The bias in the position estimation error of robot i is defined as $\| \mathrm{E}[\mathbf{e}_i(k)] \|$, where $\| \cdot \|$ is the 2-norm and E denotes expectation. The standard deviation is defined as $\sqrt{\mathrm{Tr}\left[Cov(\mathbf{e}_i(k), \mathbf{e}_i(k))\right]}$, where $Cov(\,\cdot\,)$ stands for covariance. In each scenario described below, the bias and variance in position estimation error is estimated through the use of a Monte Carlo simulation.

When comparing with the EPGO algorithm, all inter-robot relative measurements are of the relative pose. When comparing with the IEKF algorithm, all inter-robot relative measurements are of the relative position.

We first consider the standard pose graph algorithm EPGO, in which optimal robot poses are computed by minimizing the cost function (4–1). Recall that rotations are parameterized by the complex part of the corresponding unit-quaternion, and the optimization problem is set up as in [57]. Searching for the optima is performed by Levenberg-Marquardt algorithm. To maintain comparability, we provide the same local measurement graph to both the RPGO algorithm as well as to the EPGO algorithm.

A group of 5 are robots are simulated to move along the 3-D path described above. Error in the pose measurements were induced as in simulations in Section 3.4.2. Simulations were performed varying the concentration parameter K in the Von Mises-Fisher distribution from which the noisy rotations (quaternions) used to corrupt the inter-robot orientation measurements are drawn. These simulations show that, when K is very large, that is, the variance is very low, EPGO does very well, even

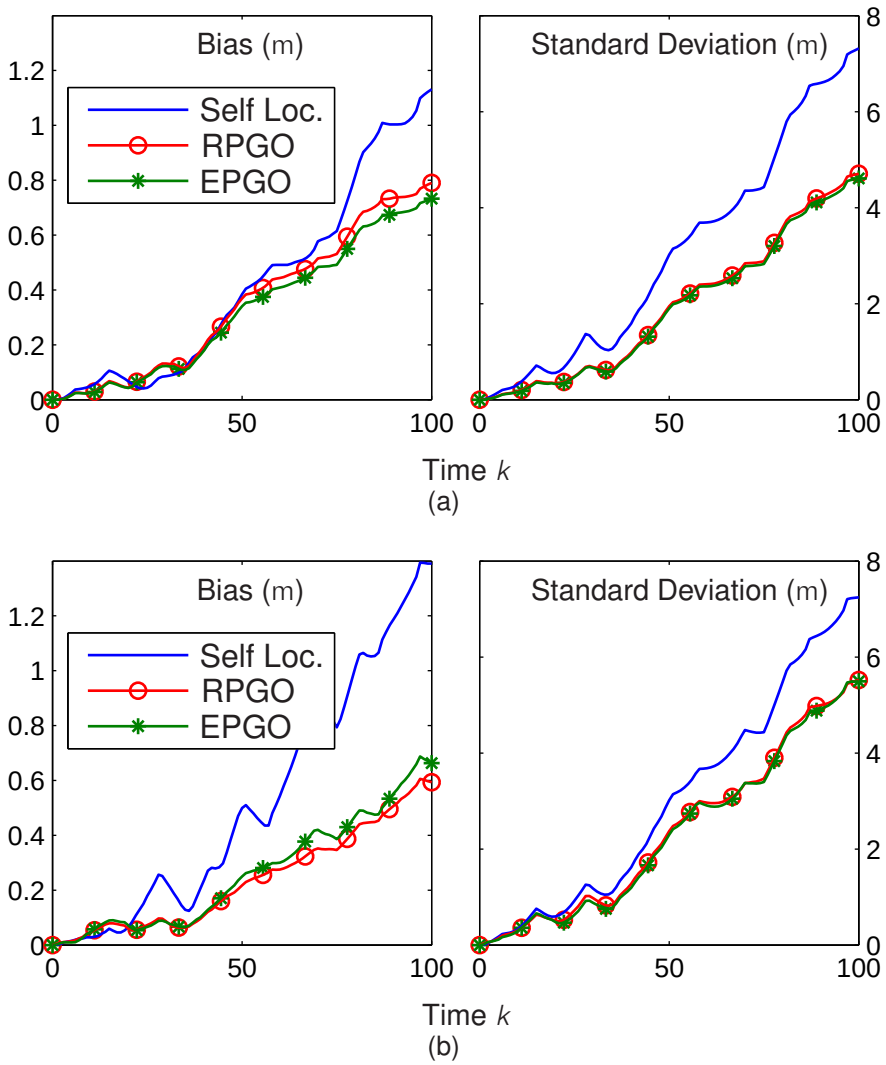


Figure 4-1. Simulation results comparing the RPGO and EPGO algorithms. The position estimation error of robot 1 when (a) $K = 10000$ and (b) $K = 100$ (in a group of 5 robots utilizing noisy inter-robot relative pose measurements), computed with both algorithms, with the same input data are shown. The label "Self Loc." refers to a robot localizing by dead reckoning alone.

outperforming the RPGO algorithm. However, when K is small, that is, the noise variance is large, RPGO outperforms EPGO. Figure 4-1 shows the bias and standard deviation for both $k = 1,0000$ and $K = 100$. For the case of $K = 100$, it is clear that the proposed RPGO algorithm outperforms the EPGO algorithm. A detailed comparison is a topic of future work.

We next consider IEKF algorithm, present in section 4.3. The IEKF algorithm utilizes an Extended Kalman Filter (EFK), developed in a similar manner as the EKF observer for collaborative localization in [58]. An indirect form filter is used, with error between the true pose and the estimated pose (before fusing the current inter-robot measurements) being the filter state. A pair of robots is simulated traveling along distinct sinusoidal paths in 3-D space. Measurements are generated as described earlier.

The trends observed from extensive simulations can be summarized as follows. When the time interval between successive inter-robot measurements, call it ΔT, is small, the IEKF algorithm performs as well, or better than, the RPGO algorithm. However, when the time between measurements is large, the RPGO algorithm provides significantly better estimates of the robots' poses compared to the IEKF. Figure 4-2 provides numerical results for both the case of a small ΔT (0.1 seconds) and a large ΔT (30 seconds in this example). As expected the IEKF does very well when ΔT is small, but performs poorly for the larger ΔT. How small ΔT has to be for the IEKF algorithm to perform well depends on many factors, including the motion of the robots, noise in the measurements, etc. For the parameters used in the simulations mentioned above, ΔT has to be smaller than 1 sec for the IEKF algorithm to perform as well as the RPGO algorithm.

We believe the reason for this behavior of the IEKF algorithm is the error introduced by the linearization involved in covariance propagation. The linearized state equations rely on the assumption that the angle between the true and estimated orientation is very small. When the time interval between inter-robot measurements is sufficiently small, this approximation holds. In that case the error in the covariance matrix due to linearization is small enough that it does not out weigh the added benefit of using covariance information. However, the small angle approximation is violated for large time intervals, leading to quite poor covariance estimates, which in turn lead to poor pose estimates.

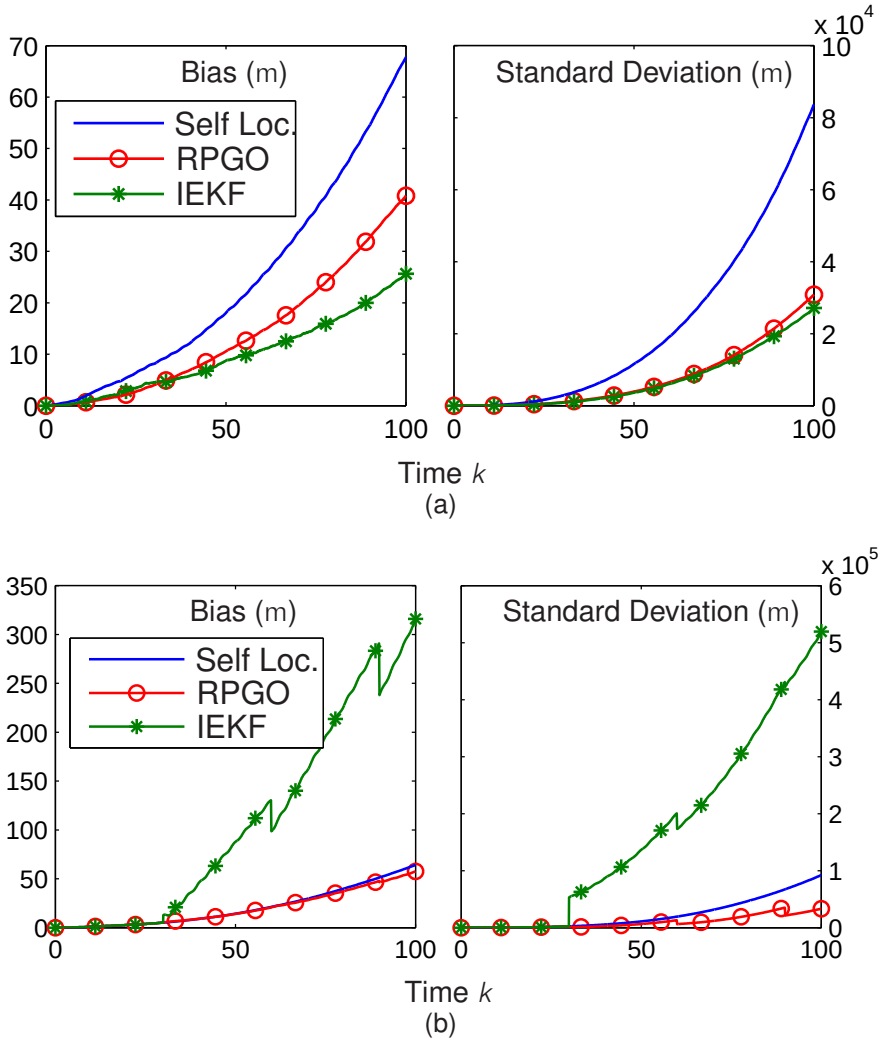


Figure 4-2. Simulation results comparing the RPGO and IEKF algorithms with (a)$\Delta T = 0.1$ and (b)$\Delta T = 30$ sec. The position estimation error of robot 1 (in a group of 2 robots utilizing noisy inter-robot relative pose measurements), computed with both algorithms. The label "Self Loc." refers to a robot localizing by dead reckoning alone.

4.5 Summary

In this chapter, two state of the art collaborative localization algorithms were compared with the RPGO algorithm presented in Chapter 3. The first algorithm, referred to as the Euclidean Pose Graph Optimization algorithm, was a classical least squares based pose graph optimization algorithm. The second algorithm was based on a

indirect extended Kalman filter. In each case, simulations were presented that identified situations under which the RPGO provided more accurate pose estimates.

CHAPTER 5
MAXIMUM LIKELIHOOD ESTIMATES

In Chapter 3 we presented an algorithm for collaborative localization when inter-robot relative measurements can be of the relative position, orientation, bearing, or distance. In this chapter we extend this algorithm by adding a weight associated with each edge. To develop a systematic way to define such weights, the maximum likelihood (ML) estimator of the node variables is considered. We will refer to the algorithm developed in this chapter as the Maximum Likelihood Riemannian Pose Graph Optimization (ML-RPGO) algorithm.

5.1 The ML Estimates

Consider the collaborative localization problem as described in Section 3.1. The fully labeled time varying measurement graph is given by $\mathcal{G}(k) = (\mathcal{V}(k), \mathcal{E}(k), \ell(k))$. Let $\left\{\hat{\mathbf{M}}_e\right\}_{e\in\mathcal{E}(k)}$ denote the corresponding set of inter-robot and inter-time relative measurements. We again consider inter-robot relative measurements of the relative pose, orientation, position, bearing, and distance. For ease of exposition, throughout this chapter we will freely identify pose measurements with the corresponding orientation and position pair. We will therefore often not explicitly consider measurements of type pose.

We assume that measurements on distinct edges are statistically independent. Under this assumption, the joint pdf of all the measurements at time k, given by the set $\left\{\hat{\mathbf{M}}\right\}_{\mathcal{E}(k)}$, satisfies:

$$p\left(\left\{\hat{\mathbf{M}}\right\}\right) = \prod_{e\in\mathcal{E}(k)} p_e(\hat{\mathbf{M}}_e), \tag{5–1}$$

where $p_e(\hat{\mathbf{M}}_e)$ is the pdf describing the probability of observing the measurement $\hat{\mathbf{M}}_e$. For each of the four types of measurements, a corresponding class of pdfs must be specified. Choosing appropriate pdfs for the orientation and bearing measurements is challenging as these densities are not defined over any vector space, but rather

over the curved surfaces $SO(3)$ and S^2 respectively. We assume that each relative orientation measurement $\hat{\mathbf{R}}_{ij}$ comes from a wrapped Gaussian distribution on $SO(3)$ with mean direction $\mathbf{R}_i^T\mathbf{R}_j$ and covariance matrix $\sigma^2 I$. The corresponding density function $p_{\mathrm{R}} : SO(3) \to \mathbb{R}^+$ is given by

$$f_{\mathrm{R}}(\hat{\mathbf{R}}_{ij}) = K_R \sum_{k=-\infty}^{\infty} \exp\Big(-\frac{1}{2\sigma_e^2}\big(d(\hat{\mathbf{R}}_{ij}, \mathbf{R}_i^T\mathbf{R}_j) - 2\pi k\big)^2\Big) \tag{5–2}$$

for appropriate normalizing constant $K_R(\sigma_e)$ [60]. Here the distance function $d(\,\cdot\,,\,\cdot\,)$ in $SO(3)$ is given by the Riemannian distance

$$d(A, B) = \sqrt{-\frac{1}{2}\,\mathrm{Tr}\left[\log^2(A^T B)\right]}, \quad A, B \in SO(3). \tag{5–3}$$

Thought a characterization of the WG distribution on SO(3) is still an open problem, It is known that the corresponding WG distribution on $SO(2)$ is the solution to the heat equation [60]. The Normal distribution solves the heat equation on a vector space, and so we can say the WG distribution is "Gaussian-like".

For bearing measurements, we choose the Von Mises-Fisher (VMF) distribution [47]. This is a well-known density in the literature on directional statistics and considered a reasonable analog of the Gaussian density in the d-Sphere. Specifically, each relative bearing measurement $\hat{\boldsymbol{\tau}}_{ij}$ is assumed to be distributed according to the Von Mises-Fisher distribution with mean direction $\mu_{\boldsymbol{\tau}} := \frac{\mathrm{R}_i^T(\mathrm{t}_j - \mathrm{t}_i)}{\|\mathrm{t}_j - \mathrm{t}_i\|}$ and concentration parameter k_e. The density function $p_{\boldsymbol{\tau}} : S^2 \to \mathbb{R}^+$ is given by

$$p_{\boldsymbol{\tau}}(\hat{\boldsymbol{\tau}}_{ij}) = K_{\boldsymbol{\tau}} \exp\Big(\frac{k_e}{\|\mathbf{t}_j - \mathbf{t}_i\|}(\mathbf{t}_j - \mathbf{t}_i)^T\mathbf{R}_i\hat{\boldsymbol{\tau}}_{ij}\Big) \tag{5–4}$$

for appropriate normalization constant $K_{\boldsymbol{\tau}}(k_e)$. The VMF distribution is a "Gaussian-like" distribution, in that it is the equilibrium distribution for the Ornstein-Uhlenbeck process on the unit sphere, just as the Normal distribution is the equilibrium distribution for vector spaces [61].

Each relative position measurement $\hat{\mathbf{t}}_{ij}$ is assumed to be distributed according to the multivariate normal distribution with mean $\mu_{\mathrm{t}} := \mathbf{R}_i^T(\mathbf{t}_j - \mathbf{t}_i)$ and covariance matrix Σ_e. The density function $p_{\mathrm{t}} : \mathbb{R}^3 \to \mathbb{R}^+$ is given by

$$p_{\mathrm{t}}(\hat{\mathbf{t}}_{ij}) = K_{\mathrm{t}} \exp\Big(-\frac{1}{2}(\hat{\mathbf{t}}_{ij} - \mu_{\mathrm{t}})^T \Sigma_e^{-1} (\hat{\mathbf{t}}_{ij} - \mu_{\mathrm{t}}) \Big) \tag{5–5}$$

for appropriate normalization constant $K_{\mathrm{t}}(\Sigma_e)$.

Finally, each relative distance measurement $\hat{\delta}_{ij}$ is assumed Normally distributed with mean $\|\mathbf{t}_j - \mathbf{t}_i\|$ and variance σ_e^2. The density function $p_\delta : \mathbb{R} \to \mathbb{R}^+$ is given by

$$p_\delta(\hat{\delta}_{ij}) = K_\delta \exp\Big(\frac{-(\hat{\delta}_{ij} - \|\mathbf{t}_j - \mathbf{t}_i\|)^2}{2\sigma_e^2} \Big) \tag{5–6}$$

for appropriate normalizing constant $K_\delta(\sigma_e)$. We assume that the standard deviation σ_e is small enough with respect to the mean $\|\mathbf{t}_j - \mathbf{t}_i\|$ so that the probability of $\hat{\delta}_{ij}$ being negative is negligible.

Among these distributions, the wrapped Gaussian is the most cumbersome due to the infinite series in its definition. We therefore approximate p_{R} by the function

$$\bar{p}_{\mathrm{R}}(\hat{\mathbf{R}}_{ij}) = K_R \exp\big(-\frac{1}{2\sigma^2} d^2(\hat{\mathbf{R}}_{ij}, \mathbf{R}_i^T \mathbf{R}_j) \big). \tag{5–7}$$

Note that $\bar{p}_{\mathrm{R}}$ is not a probability density function. To justify the approximation of p_{R} by $\bar{p}_{\mathrm{R}}$, the 1-norm of $p_{\mathrm{R}} - \bar{p}_{\mathrm{R}}$ was computed using Monte-Carlo integration with $100,000$ samples. The value of σ_e was in the range $[0^+, 2]$. The results are reported in Figure 5-1. For $\sigma < 0.7$ radians, the norm of the difference is near enough to zero to be indistinguishable. We therefore conclude that the approximation $\bar{p}_{\mathrm{R}}$ of the wrapped Gaussian distribution p_{R} is reasonably accurate for values of $\sigma < 0.7$.

We are now ready to characterize the ML estimate of the node variables (absolute poses).

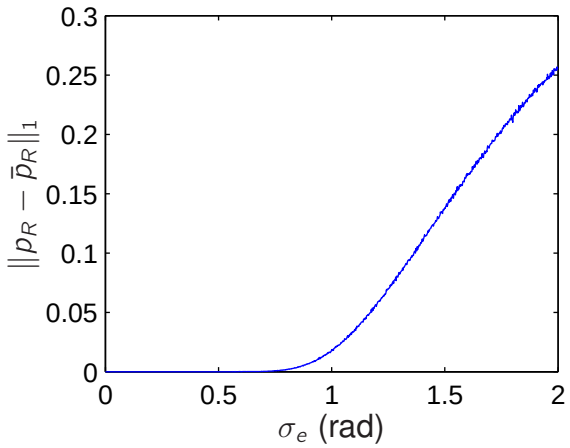


Figure 5-1. The magnitude of the difference between p_{R} and $\bar{p}_{\mathrm{R}}$, the pdf for orientation measurements and its approximation respectively

Proposition 5.1. *An approximation of the maximum likelihood ML estimate* $\left\{(\hat{\mathbf{R}}, \hat{\mathbf{t}})\right\}_{\mathcal{V}(k)}$ *of the node variables* $\left\{\mathbf{T}_u\right\}_{u\in\mathcal{V}(k)}$ *based on the measurements* $\left\{\hat{\mathbf{M}}\right\}_{\mathcal{E}(k)}$ *is given by*

$$\left\{(\hat{\mathbf{R}}, \hat{\mathbf{t}})\right\}_{\mathcal{V}(k)} = \underset{\{(\mathrm{R},\mathrm{t})\}_{\mathcal{V}(k)}\in(SO(3)\times\mathbb{R})^{|\mathcal{V}(k)|}}{\arg\min} f\left(\left\{\mathbf{T}_u\right\}_{u\in\mathcal{V}(k)}\right) \tag{5–8}$$

where $f : (SO(3)\times\mathbb{R})^{|\mathcal{V}(k)|} \to \mathbb{R}$ *is a cost function given by*

$$f\left(\left\{\mathbf{T}_u\right\}_{u\in\mathcal{V}(k)}\right) := \sum_{(i,j)=e\in\mathcal{E}(k)} g_e(\mathbf{R}_i, \mathbf{t}_i, \mathbf{R}_j, \mathbf{t}_j) \tag{5–9}$$

in which $g_e(\mathbf{R}_i, \mathbf{t}_i, \mathbf{R}_j, \mathbf{t}_j)$ *is the cost for edge* e *defined as*

$$g_e(\mathbf{R}_i, \mathbf{t}_i, \mathbf{R}_j, \mathbf{t}_j) = \begin{cases} \frac{1}{2\sigma_e^2} d^2(\hat{\mathbf{R}}_{ij}, \mathbf{R}_i^T\mathbf{R}_j) & \text{if } \ell(e) = \mathbf{R} \\ \frac{1}{2}\Big(\big(\hat{\mathbf{t}}_{ij} - \mathbf{R}_i^T(\mathbf{t}_j - \mathbf{t}_i)\big) \times \Sigma_e^{-1}\big(\hat{\mathbf{t}}_{ij} - \mathbf{R}_i^T(\mathbf{t}_j - \mathbf{t}_i)\big)\Big) & \text{if } \ell(e) = \mathbf{t} \\ \frac{-k_e}{\|\mathbf{t}_j - \mathbf{t}_i\|}(\mathbf{t}_j - \mathbf{t}_i)^T\mathbf{R}_i\hat{\boldsymbol{\tau}}_{ij} & \text{if } \ell(e) = \boldsymbol{\tau} \\ \frac{1}{2\sigma_e^2}\big(\hat{\delta}_{ij} - \|\mathbf{t}_j - \mathbf{t}_i\|\big)^2 & \text{if } \ell(e) = \delta \end{cases} \tag{5–10}$$

Proof. As described above, the pdf describing the probability of observing a measurement depends on the value of the node variables. To emphasize this, we rewrite (5–1) as

$$p(\{\hat{\mathbf{M}}\}_{\mathcal{E}(k)}|\{\mathbf{T}_u\}_{u\in\mathcal{V}(k)}) = \prod_{e\in\mathcal{E}(k)} p_e(\hat{\mathbf{M}}_e \,|\, \{\mathbf{T}_u\}_{u\in\mathcal{E}(k)})$$

where the dependency on the unknown parameters $\{\mathbf{T}_u\}_{u\in\mathcal{V}(k)}$ is shown clearly. The likelihood function $L(\,\cdot\,)$ is the density viewed as a function of the unknown parameters. The log-likelihood function $\log L$ satisfies

$$\begin{aligned}\log L(\{\mathbf{T}_u\}_{u\in\mathcal{V}(k)} \,|\, \{\hat{\mathbf{M}}\}_{\mathcal{E}(k)}) &= \log\left(p(\{\hat{\mathbf{M}}\}_{\mathcal{E}(k)}|\{\mathbf{T}_u\}_{u\in\mathcal{V}(k)})\right)\\ &\propto \log(\prod_{e\in\mathcal{E}(k)} \operatorname{Ker}\left(p_e(\hat{\mathbf{M}}_e \,|\, \{\mathbf{T}_u\}_{u\in\mathcal{V}(k)})\right)\end{aligned}$$

where $\operatorname{Ker} p_{\hat{\mathbf{M}}}$ is the kernel of the corresponding pdf. The max-likelihood estimate is obtained by maximizing the right hand side of the relation above. When we use the approximation $\bar{p}_{\mathrm{R}}$ instead of p_{R}, it follows from straightforward calculations that the right hand side is equal to $-f(\{\mathbf{T}_u\}_{u\in\mathcal{V}(k)})$, were f is as defined in (5–9). The corresponding (approximate) maximum likelihood estimate for $\{\mathbf{T}_u\}_{u\in\mathcal{V}(k)}$ given $\{\hat{\mathbf{M}}\}_{\mathcal{E}(k)}$ is computed by minimizing f. This estimate is not strictly equal to the maximum likelihood estimate because of the approximation of p_{R} by $\bar{p}_{\mathrm{R}}$. Since the approximation is quite accurate for $\sigma < 0.7$, we expect the estimate obtained to be a close approximation of the ML estimate for $\sigma < 0.7$. □

The cost function 5–9 can be minimized as described in Chapter 3.

5.2 Simulation studies

In this section we present simulations studying the performance of the ML-RPGO algorithm in terms of localization accuracy. For ease of exposition, we consider the robot group at a single time instant, say $k = k_0$, where the only inter robot relative measurements available are those captured at time k_0. This will allow us to compare the

	Orient.(σ_e)	Pos.(Σ_e)	Bearing (k_e)	Dist. (σ_e)
Initial Est.	0.26 rad	4*I* m	–	–
Inter-Cam.	0.087 rad	0.25 *I* m	20	0.5m

Table 5-1. The parameters of the measurement pdfs used in all simulations (see (5–2)- (5–6)).

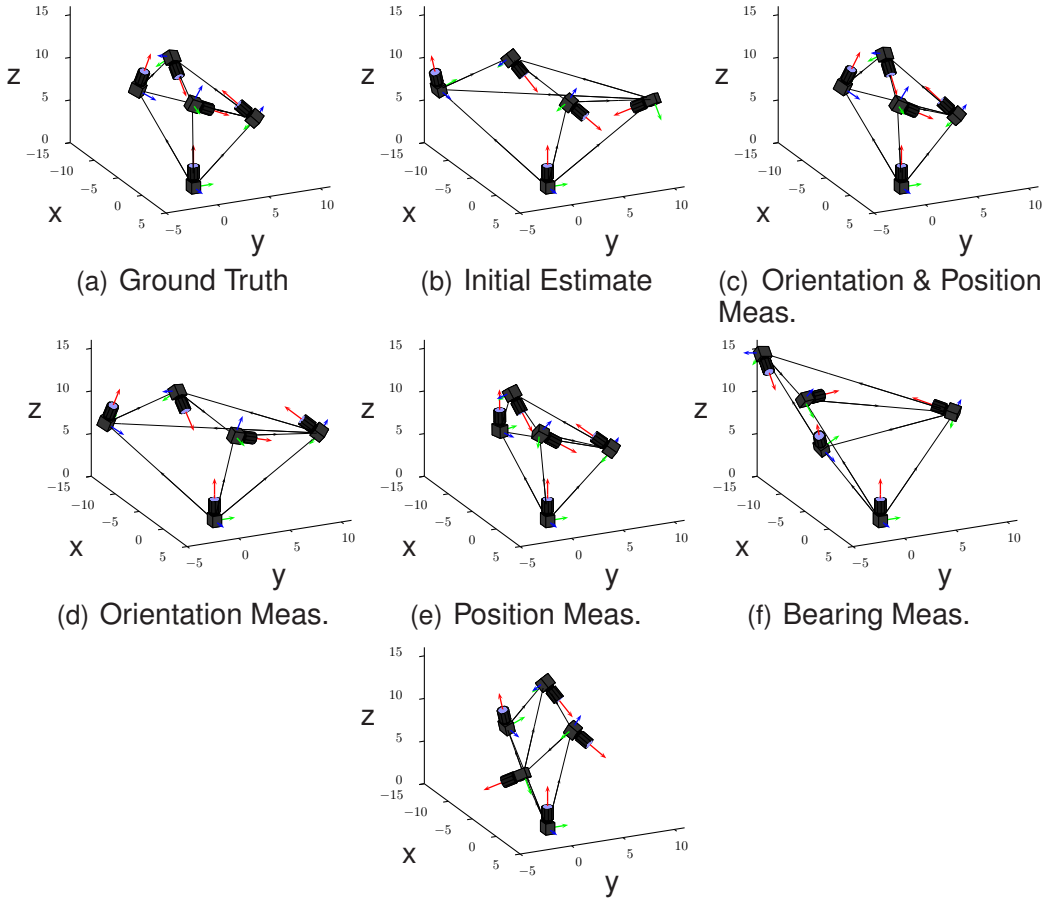
ML-RPGO algorithm with one presented in [52] in which camera networks, instead of robots, are considered.

We examine the following questions. One, how does estimation accuracy of the ML-RPGO algorithm change as the connectivity of the measurement graph increases due to the increase in the number of relative measurements for the same number of robots/cameras, and how does accuracy depend on the type of those measurements? Two, how does ML-RPGO perform compared to the alternative method proposed in [52]?

The following definitions are required. The error in an estimate $\hat{\mathbf{R}}_i$ of the orientation for a camera i is $e_R(i) := d(\mathbf{R}_i, \hat{\mathbf{R}}_i)$, where $d(\cdot, \cdot)$ is defined in (5–3). The error in an estimate $\hat{\mathbf{t}}_i$ of the position of camera i is $e_p(i) := \|\mathbf{t}_i - \hat{\mathbf{t}}_i\|_2$. The total r.m.s. error in the orientation and position estimate is defined as $\sqrt{\mathrm{E}[\sum_{i=1}^{n} e_R^2(i)]}$ and $\sqrt{\mathrm{E}[\sum_{i=1}^{n} e_p^2(i)]}$, respectively, where $\mathrm{E}[\cdot]$ denotes expectation. The expected value $\mathrm{E}[e_{[\cdot]}(i)]$ is also referred to as the bias in that error. All expectations are computed from appropriate averaging from random samples obtained through simulations.

5.2.1 Performance in a single experiment

We consider a network of 5 cameras. One possible graph on such a network is presented in Figure 5-2(a). Both the initial guess for the pose of each camera, and the noisy inter-camera measurements are drawn from the distributions described in Section 5.1. The parameters for each distribution are reported in Table 5-1. The variances of the initial guess of poses are chosen to be much higher then those for the inter-camera measurements to simulate a realistic situation when the initial guess is poor.



(a) Ground Truth (b) Initial Estimate (c) Orientation & Position Meas.

(d) Orientation Meas. (e) Position Meas. (f) Bearing Meas.

(g) Distance Meas.

Figure 5-2. A single realization for a group of robots using various measurement types and the ML-RPGO algorithm. Each of the (a) true values (b) the initial guess for the poses of the robots, and (c-g) the estimated poses after the ML-RPGO algorithm has been used with inter-robot relative measurements. Each of the plots in (c)-(g) correspond to a distinct type of relative measurement.

The ML-RPGO algorithm is applied to one realization of the noisy measurements for each of the measurement types, as well as for the case when both relative orientation and position measurements are available. The resulting estimated positions and orientations are shown in Figures 5-2(c-g). We see that relative orientation measurements allow for an accurate estimation of the orientation of each camera without having any effect on the estimated position. In contrast, relative position measurements improve both the orientation and position estimates for each camera. This is expected since

absolute orientations of the cameras affect the relative positions while absolute positions do not affect relative orientations. When noisy measurements of both orientation and position are available, the pose of every camera can be estimated with high accuracy. In contrast to the position and orientation measurements, both the bearing and distance measurements lead to poor estimates, at least for this set of measurements.

5.2.2 Effect of measurement noise, and comparison with state-of-the-art camera network localization algorithm

We now examine the effect on localization accuracy when some of the measurements are noisier than others. We also compare ML-RPGO's performance with the algorithm in [52] for varying measurement noise levels.

A network of 3 cameras is considered, in which every camera has a measurement for every other camera in the network. That is, the corresponding graph is fully connected. Each inter-camera measurement is of the relative orientation and bearing. These measurement types are chosen to enable comparison with the algorithm in [52], since the same measurement types are considered in [52]. The parameters σ_e and k_e for distributions of relative orientation and bearing measurements of camera 2 and 3 are given in Table 5-1. However, the noise parameters for measurements obtained by camera 1 are allowed to vary as follows: for orientation measurements, $\sigma_e = 0.087 \times K$, and for bearing measurements, $k_e = 20 \times K$ where $K \in [2^{-8}, 2^8]$. The wrapped Gaussian distribution (for relative orientation measurements) and Von Mises - Fisher distributions (for relative bearing measurements) produce more or less noisy measurements depending on the value of K. For larger values of K, orientation measurements become more noisy, while bearing measurements become less noisy. For each value of K considered, noisy measurements are generated from the corresponding distributions. These measurements are then used by the ML-RPGO algorithm and the algorithm in [52] to estimate the pose of each camera. To coincide with the assumptions made in [52], the initial pose estimates are not used as measurements in the ML-RPGO

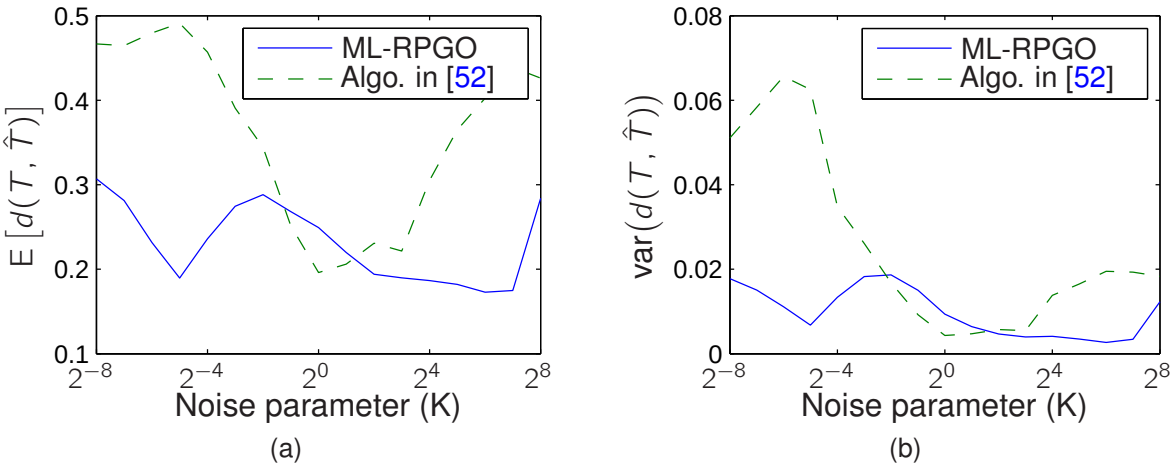


Figure 5-3. Comparison of ML-RPGO and the algorithm proposed by Tron and Vidal in [52] for varying levels of noise in the relative measurements for a 3-camera network. Larger K corresponds to noisier relative orientation measurements and less noisy bearing measurements. Both the sample (a) Bias and (b) variance are shown.

algorithm. This will be reflected in our choice of distance metric. For each method of estimation and each value of K, the bias and variance of the distance between pose estimates is computed from a Monte-Carlo simulation with 200 samples. The distance between a pose $T = \{(\mathbf{R}_i, \mathbf{t}_i)\}_{i=1}^{n}$ of a network of n cameras and its estimate $\hat{T}$ is defined as

$$d(T, \hat{T}) := \left(\sum_{i=1}^{n} (d^2(\mathbf{R}_i, \hat{R}_i) + \left\| \frac{\mathbf{t}_i}{\|\mathbf{t}_i\|} - \frac{\hat{t}_i}{\|\hat{t}_i\|} \right\|^2) \right)^{1/2} \tag{5–11}$$

Using the orientation and bearing measurements alone, the pose of all cameras can only be estimated up to a scale ambiguity. For that reason we have normalized the position estimates to adjust for the ambiguity in scale.

The results are reported in Figure 5-3. Though both algorithms provide accurate estimates, we see that when all measurements have equal amount of noise, the algorithm in [52] proves to be more accurate. This is likely due to an additional optimization step found in [52] in which the initial estimates are improved by minimizing an additional cost function. The ML-RPGO algorithm does not perform this additional

step, though it could be implemented if desirable. However the ML-RPGO algorithm provides more accurate estimates when the difference between the noise levels in the various measurements is large. This occurs since the ML-RPGO algorithm takes into account the noise in each measurement in a principled way to compute the most likely estimates given this information. This reduces the effect of measurements that are highly noisy, while heightening the effects of measurements with lower noise. No such weights are present in the algorithm in [52]. While it is possible to modify the cost function in [52] to include weights, due to the non-Euclidean nature of the relative orientation measurements, it is not clear how one would determine these weights.

5.3 Summary

In this chapter we first identified reasonable distributions for each of the considered measurement types. The distributions were chosen in such a way as to depend on the true value of the given measurement. Additionally, as in previous chapters and in contrast to much of the previous work in this area, the orientations measurements were given their natural manifold structure, and so the corresponding distribution was defined over the manifold $SO(3)$ with respect to the appropriate probability measure. A novel collaborative localization algorithm, referred to as the ML-RPGO algorithm, was then proposed based on the maximum likelihood estimator for the absolute pose of each of the robots. The resulting algorithm was given by minimizing the negative log-likelihood function. As in Chapter 3 this results in an optimization problem of the product Riemannian manifold.

Simulations were presented that studied the effect of increasing numbers of neighbors on estimation accuracy. Finally, simulations were presented comparing the ML-RPGO algorithm with a state of the art camera network localization algorithm.

CHAPTER 6
OUTLIER REJECTION ON POSE GRAPHS

In Chapter 3 we presented an algorithm for collaborative localization when inter-robot relative measurements can be of the relative position, orientation, bearing, or distance. In this section, we restrict our focus to only measurements of the relative orientation or pose (orientation and position), and consider how outliers in measurements of these types might be identified and rejected.

6.1 Problem statement

Consider the collaborative localization problem as described in Section 3.1 with the restriction that all inter robot measurements are of the relative pose. The time varying measurement graph is given by $\mathcal{G}(k) = (\mathcal{V}(k), \mathcal{E}(k))$. Note that the labeling function $\ell(k)$ (and thus the fully labeled nature of the graph) is unnecessary as all measurements are of the same type. Let $\left\{\hat{\mathbf{M}}_e\right\}_{e\in\mathcal{E}(k)}$ denote the corresponding set of inter-robot and inter-time relative measurements. An outlier is a measurement $\hat{\mathbf{M}}_{e'} \in \left\{\hat{\mathbf{M}}_e\right\}_{e\in\mathcal{E}(k)}$ that is inconsistent with the remaining measurements. An outlier is most often a grossly inaccurate measurement in a set of measurements that are relatively accurate (for which the noise is small). In such a case, if $\hat{\mathbf{M}}_{e'}$ is an outlier, we expect the estimate of the node variables corresponding to the measurement graph $(\mathcal{V}(k), \mathcal{E}(k) \setminus \{e'\})$ to be more accurate then the estimates corresponding to $\mathcal{G}(k)$, in which the outlier is still included. The outlier rejection problem is to identify and remove such outliers using only the information provided by the measurements, and no information about the node variables. In particular, any knowledge of a prior distribution on the value of the node variables is not available.

6.2 Outlier Rejection Algorithm

Consider a set of robots attempting to perform collaborative localization using a pose graph optimization CL algorithm in the presence of outliers when all measurements

are of the relative pose. Let $G(k) = (\mathcal{V}(k), \mathcal{E}(k))$ denote the corresponding measurement graph and $\left\{\hat{\mathbf{M}}_e\right\}_{e\in\mathcal{E}(k)}$ the set of all inter-robot and inter-time relative measurements.

For each edge $e \in \mathcal{E}(k)$, write $e \triangleright (a, b)$ to denote edge e is from node a to node b. A simple cycle is an ordered collection of edges $c = (e_0, \cdots, e_\ell)$ such that

- $a_i = b_{i-1},\ i = 1, \cdots, \ell,$
- $a_0 = b_k,$
- $a_i \neq a_j,\ \forall i \neq j.$

We will refer to the set of all such cycles in the graph $\mathcal{G}(k)$ as $\mathcal{C}(k)$. For a measurement graph in which all measurements are of the relative pose, composing noise-free measurements along any cycle yields the identity. To utilize this fact, for the simple cycle $c = (e_0, \cdots, e_\ell) \in \mathcal{C}(k)$ define the cycle measurement $\hat{\mathbf{M}}_c = \hat{\mathbf{M}}_{e_0}\hat{\mathbf{M}}_{e_1}\cdots\hat{\mathbf{M}}_{e_\ell}$. Since $\hat{\mathbf{M}}_c = id \in SE(3)$ whenever the measurements are noise-free, a suitable distance metric on the product manifold $(SO(3) \times \mathbb{R}^3)$ provides a measure of the noise encountered in cycle c when measurements do contain noise. Towards this end, we define the cycle consistency cost $D_{\mathcal{C}} : \mathcal{C}(k) \to \mathbb{R}^+$ as

$$D_{\mathcal{C}}(c \in \mathcal{C}(k)) = \frac{\sqrt{d^2(id_{SO(3)}, \hat{\mathbf{R}}_c) + \|\hat{\mathbf{t}}_c\|^2}}{|c|} \tag{6–1}$$

where $\hat{\mathbf{M}}_c = (\hat{\mathbf{R}}_c, \hat{\mathbf{t}}_c)$ and $d(\cdot, \cdot)$ is the Riemannian distance [62] given by

$$d(A, B) = \sqrt{-\frac{1}{2}\,\mathrm{Tr}\left[\log^2(A^T B)\right]}, \quad A, B \in SO(3).$$

While $D_{\mathcal{C}}(c)$ provides a measure of the average noise encountered along the simple cycle $c \in \mathcal{C}(k)$, using $D_{\mathcal{C}}(c)$ alone little can be said about the accuracy of any particular measurement. Instead, we will consider the simple cycle consistency costs for all cycles containing a given edge of interest. Let $\mathcal{C}_e(k) \subset \mathcal{C}(k)$ denote the set of all cycles that

include edge e. The edge consistency cost $D : \mathcal{E}(k) \to \mathbb{R}$ is then defined as

$$D(e \in \mathcal{E}(k)) = \min_{c \in \mathcal{C}_e(k)} \{D_{\mathcal{C}}(c)\}. \tag{6–2}$$

$D(e)$ provides a measure of the noise in measurement $\hat{\mathbf{M}}_e$ based only on the given measurements consistency with other measurements in the graph.

As the number of measurements in a graph $\mathcal{G}(k)$ grow, finding all cycles, and subsequently the value of $D(e)$ for $e \in \mathcal{E}(k)$, can become infeasible. Instead, we will will consider some subset found through a depth first search (DFS) on the graph [63]. In particular, let m indicate the number of edges in $\mathcal{E}(k)$ and define a tuning parameter $M > 0$. We then find $m\,M$ random cycles by using the DFS. The set of cycles found in this manner is denoted by $\hat{\mathcal{C}}(k) \subset \mathcal{C}(k)$. We then define the approximate edge consistency cost $\hat{D}(e \in \mathcal{E}(k))$ as

$$\hat{D}(e) = \min_{c \in \hat{\mathcal{C}}_e(k)} \{D_{\mathcal{C}}(c)\} \tag{6–3}$$

where $\hat{\mathcal{C}}_e(k) \subset \hat{\mathcal{C}}(k)$ is the set of all cycles found that contain the edge e.

For the algorithm to perform well, the number of outliers, the topology of the graph, and the value of the tuning parameter M should be such that the following condition is satisfied: If $e \in \mathcal{E}(k)$ is not an outlier, then there exists a simple cycle $c \in \hat{\mathcal{C}}_e(k)$ such that every edge in c is not an outlier. When this condition is satisfied, if a large value of $D_{\mathcal{C}}(c)$ indicates the presence of an outlier in the cycle c, then we expect $\hat{D}(e)$ to be large if and only if e itself is an outlier.

To identify outliers based on the edge consistency costs, a hypothesis test of the set $\left\{\hat{D}(e) \,|\, e \in \mathcal{E}(k)\right\}$ will be utilized. Though the values of $\hat{D}(e)$ are not $i.i.d.$ (independent, identically distributed) we will make the simplifying assumption that they in fact are i.i.d. Further, we choose the log-normal distribution to describe the identical distribution for the edge consistency cost of each edge. Evidence supporting this choice of distribution will be presented in Section 6.4.1. Finally the set $\mathcal{S}(k) = \left\{\log(\hat{D}(e)) \,|\, e \in \mathcal{E}(k)\right\}$ is

considered. Under the simplifying assumption that the values $\hat{D}(e)$ are distributed $i.i.d.$ log-normal, the set $\mathcal{S}(k)$ will be distributed $i.i.d.$ normal. The one sided version of Grubbs' test for outliers [64] can then be used to identify likely outliers in the measurements as follows. Given the data set $S(k)$, we say a value $s \in S(k)$ is an outlier (in distribution) if it is not distributed according to the same $i.i.d.$ normal distribution describing the probability of seeing the other values in $S(k)$. The null hypothesis, H_0, is that there are no positive outliers in the set $S(k)$. Here "positive" indicates we are only considering outliers to the right of the mean. For instance, outliers that are very negative would not be rejected. Rejecting the null hypothesis is equivalent to accepting the alternate hypothesis, which states that the largest value in $S(k)$ is an outlier (in distribution). The one sided Grubbs' test statistic is given by

$$G = \frac{s_{\max} - \bar{s}}{\sigma_s}$$

where $s_{\max}$ denotes the maximum value in $S(k)$, and $\bar{s}$, σ_s are the sample mean and sample standard deviation of $S(k)$ respectively. The null hypothesis is rejected at a significance level of α if

$$G > \frac{N-1}{\sqrt{N}} \sqrt{\frac{t^2_{\alpha/N,N-2}}{N-2+t^2_{\alpha/N,N-2}}}$$

where $t_{\alpha/N,N-2}$ denotes the upper critical value of the t-distribution with $N-2$ degrees of freedom and a significance level of α/N.

If the null hypothesis is rejected, $s_{\max}$ is removed from $S(k)$ and the hypothesis test is repeated until the null hypothesis can not be rejected. Each time an outlier (in distribution) is removed from the set $S(k)$, the edge that generated that particular value is also removed from the graph, thus discarding the measurement as a suspected outlier.

Remark 6.1. *Although the outlier rejection algorithm is developed for the case when all measurement are of the relative pose, extending the algorithm to the case when all*

measurements are of the relative orientation instead is straightforward. Utilizing the fact that noise free orientation measurements composed over cycles also yield the identity, the algorithm as presented need only be modified by redefining the cycle consistency cost function D_c as

$$D_c(c \in \mathcal{C}(k)) = \frac{d(id_{SO(3)}, \hat{\mathbf{R}}_c)}{|c|}.$$

6.3 Sliding Window Approximation

Straightforward application of the method described in the previous section is only possible up to a certain time, beyond which the size of the graph makes computations infeasible. Under such a condition, a sliding window approximation can be used. Sliding window approximation is commonly used in both pose graph CL and graph SLAM (see [65–67]). In this section we briefly review the sliding window approximation and provide a modification under which the sliding window approximation can be used in the outlier identification problem.

The sliding window measurement graph at time k is given by removing all measurements that occurred before time $k - s$. Often this cut removes all edges leading to node 0, resulting in a disconnected graph. For example, such a disconnected graph would eventually result if all robots have access to GPS and compass measurements at time 0, but at no later time. To reconnect the graph, the most recent node variable estimates for the nodes introduced at time $k - s$ (the earliest nodes still in the graph) are used as measurements between those nodes and node 0. Specifically, the sliding window measurement graph $\mathcal{G}_s(k) = (\mathcal{V}_s(k), \mathcal{E}_s(k))$ where

$$\begin{aligned}
\mathcal{V}_s(k) &= \Big(V(k) \setminus \mathcal{V}\big(k - (s+1)\big)\Big) \cup \{0\} \\
\mathcal{E}_s(k) &= \Big(E(k) \setminus \mathcal{E}\big(k - (s+1)\big)\Big) \\
&\qquad \cup \{e = (0, j) \,|\, j \in \mathcal{V}(k-s) \setminus \mathcal{V}(k - (s+1))\}.
\end{aligned}$$

and the additional edges $(0, j)$ in $\mathcal{E}_s(k)$ correspond to measurements given by the node variable estimates $(\hat{\mathbf{R}}_j, \hat{\mathbf{t}}_j) \in \left\{(\hat{\mathbf{R}}_i, \hat{\mathbf{t}}_i)\right\}_{\mathcal{V}(k-1)\setminus\mathcal{V}(k-(s+1))}$ found using a pose graph optimization collaborative localization algorithm.

The description above corresponds to the standard sliding window implementation most often seen in literature. When using the sliding window approximation to identify outliers, a simple modification is necessary. Let $\mathcal{G}_s(k) = (\mathcal{V}_s(k), \mathcal{E}_s(k))$ denote a sliding window pose graph. Rather then attempting to identify outliers in $\mathcal{G}_s(k)$, a reduced graph $\mathcal{G}_r(k) = (\mathcal{V}_r(k), \mathcal{E}_r(k))$ is used, where $\mathcal{V}_r(k) = \mathcal{V}_s(k) \setminus \{0\}$ and $\mathcal{E}_r(k) = \mathcal{E}_s(k) \setminus \{(0, j) \mid$ the edge $(0, j)$ corresponds to an estimate of the node variable, not a true measurement $\}$. In simple terms, we remove the additional edges connected to 0 added in the construction of the sliding window pose graph. This reduction of the sliding window graph is necessary as, for sufficiently large k, the measurements (really node variable estimates) corresponding to these edges are expected to be more noisy then the inter-robot and inter-time measurements. In fact, in many cases the uncertainty in these node variable estimates will grow without bound while the uncertainty in inter-robot and inter-time measurements remains constant. Finally, the outlier rejection algorithm described in Section 6.2 is now applied to the reduced sliding window graph so constructed at every time index k. The original (non-reduced) sliding window graph, minus the identified outliers, can then be passed to any appropriate pose graph CL algorithm.

Remark 6.2. *Though only the centralized solution is discussed in this work, the outlier rejection algorithm utilizing the sliding window approximation can be distributed in such a way as to have each robot perform its own outlier rejection using only measurements for neighboring robots. In particular, each robot will maintain contact with any robots it has measured (or that has measured it) since time $k - s$ (we refer to such robots a neighbors) and construct a local sliding window measurement graph consisting of inter-robot measurements between its self and its neighbors, as well as inter-time*

measurements for itself and for each of its neighbors. Outlier rejection can then be performed on the local sliding window measurement graph.

6.4 Simulation

In this section we first present results justifying the use of the log-normal distribution to describe the edge consistency cost distribution, which is necessary in the application of Grubbs' test for outliers. Then in Section 6.4.2 we present simulations in which outliers are present during collaborative localization and the results of applying the outlier rejection algorithm.

6.4.1 Justification of the log-normal distribution

A pose measurement graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with 50 nodes and 202 edges was used in two simulations in which all measurements were of the relative pose. Outliers were generated by corrupting 3% of the relative pose measurements by 10m in position and 90° in orientation (about a random axis). The robots moved at an average speed of $.1\text{m}$ per time step, so a 10m error is enough to make a measurement an outlier. The 90° error in the orientation measurement was chosen because it is commonly seen when utilizing vision-based sensors (see [68]). The average angular speed of the robots was 8° per time step, and so the error in orientation is also sufficiently large to be classified as an outlier. The tunning parameters M and α were set at 10 and 0.025 respectively.

The values of $\left\{\hat{D}(e) \mid e \in \mathcal{E}\right\}$ were computed for the graph $\mathcal{G}$. Gaussian kernel density approximation was then used to compute an estimate of the pdf describing the values found. In addition, the sample mean and variance were computed, and the log-normal density with the equivalent mean and variance was identified.

In Figure 6-1 both the estimated pdf, along with the corresponding log-normal density function can be seen. A comparison of the estimated and log-normal pdf shows that, while not an exact match, the approximate shape is adequately captured. Most importantly to the success of the hypothesis test based algorithm presented in

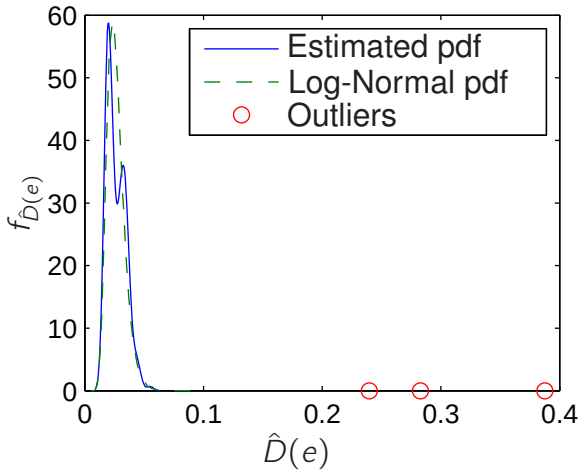


Figure 6-1. A comparison between the estimated pdf for the values of $\hat{D}(e)$ as compared to the corresponding log-likelihood distribution pdf. The locations of the outliers are also indicated.

Section 6.2, the value of $\hat{D}(e)$ when e is out outlier is also an outlier for the identified log-normal distribution.

6.4.2 Outlier Rejection

We now present a set of simulations that provide some insight into the effectiveness of the proposed algorithm. First we define a set of convenient performance metrics. The position estimation error of robot i is defined as $\mathbf{e}_i(k) := \hat{\mathbf{t}}_i(k) - \mathbf{t}_i(k)$, where $\mathbf{t}_i(k)$ is its absolute position at k and $\hat{\mathbf{t}}_i(k)$ is the estimate. The bias in the position estimation error of robot i is defined as $\| \mathrm{E}[\mathbf{e}_i(k)] \|$, where $\| \cdot \|$ is the 2-norm and E denotes expectation. The standard deviation is defined as $\sqrt{\mathrm{Tr}\left[Cov(\mathbf{e}_i(k), \mathbf{e}_i(k))\right]}$, where $Cov(\cdot)$ stands for covariance. In each scenario described below, the bias and standard deviation in position estimation error is estimated through the use of a Monte Carlo simulation with 300 sample runs.

A group of 4 robots were simulated to move along distinct 3-D paths. Two robots were able to obtain noisy relative pose measurements at time k if the Euclidean distance between them at that time was less then $7\mathrm{m}$. The Riemannian pose graph optimization algorithm presented in [69] was used to perform the collaborative localization in

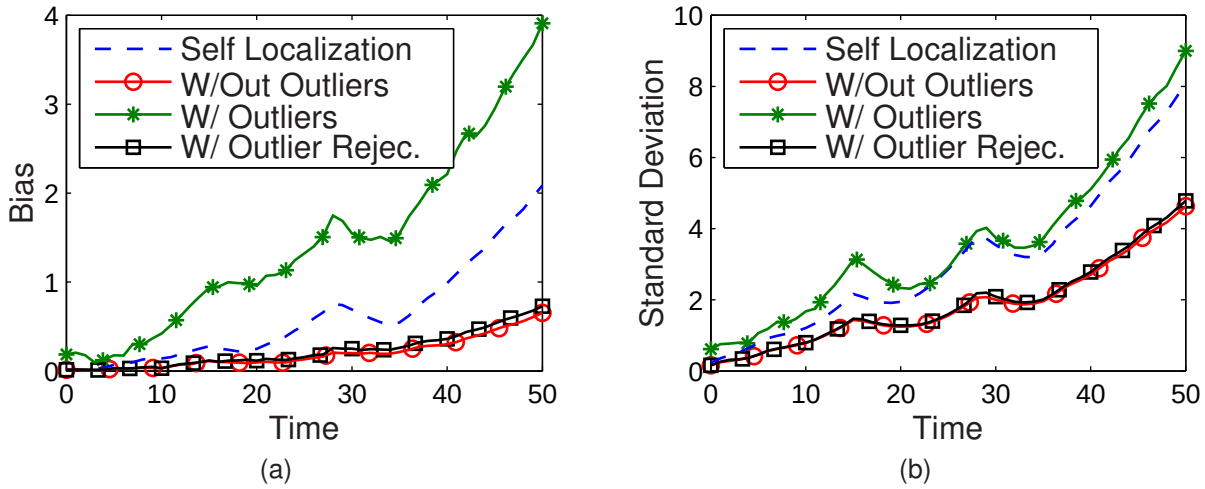


Figure 6-2. Simulation results studying the effect of outliers and the outlier rejection algorithm on localization accuracy. Results are shown for collaborative localization without outliers, with outliers and no rejection, and with outliers when the rejection algorithm is utilized. The (a) Bias and (b) Standard Deviation in position estimation error are estimated using a 300-iteration Monte-Carlo Simulation.

three case. In the first simulation, all measurements were equally accurate, that is, no outliers were present. In the second simulation, outliers were generated by corrupting approximately 4% of the measurements by 90° in orientation (about a random axis) and 10m in position. Finally, the same set of outlier induced measurements was considered, but the outlier rejection algorithm presented in Section 6.2 was utilized to identify and reject likely outliers. For this simulation the sliding window approximation described in Section 6.3 was utilized with a window length (s) of 3. The tunning parameters M and α were set at 10 and 0.025 respectively. The bias and standard deviation of the position estimation error, defined above, estimated using a 300-iteration Monte-Carlo simulation are presented in Figure 6-2. A summary of the average number of measurements, outliers, and false negatives/positives is reported in Table 6-1.

A number of observations can be made from the plots in Figure 6-2. The first observation is that the presence of outliers can cause the estimates found using collaborative localization to become even less accurate than those found using dead

	Average	Percentage
Measurements	48.1	-
Outliers	1.85	3.7%
False Rejects	0.22	0.5%
False Accepts	0.087	5%

Table 6-1. The average number of measurements, outliers, measurements falsely rejected, and outlier measurements falsely accepted when the outlier rejection algorithm is simulated as a preprocessing step of collaborative localization. The averages are with respect to all 300 Monte-Carlo simulations over all 50 time steps. The percentages are with respect to total measurements for Outliers, non-outlier measurements for falsely rejected measurements, and outlier measurements for falsely accepted measurements.

reckoning alone. To see why this is so, it is important to note that the collaborative localization algorithm used in this simulation is essentially a least squares optimization problem. As such, the algorithm is sensitive to outliers, to the extent that the estimated node variables may better reflect the information contained in the outliers, rather then the remaining, more accurate measurements.

The second observation to be made from Figure 6-2 is the considerable improvement to localization accuracy when the proposed outlier rejection algorithm is utilized before performing collaborative localization. In fact, accuracy is almost as good as that if no outliers were present. This is unsurprising as, from Table 6-1 it is evident that the majority of outliers were correctly identified, and very few additional measurements were falsely rejected. A sufficient number of relative measurements, approximately 96%, remained after the outlier rejection preprocessing; enough to allow the collaborative location algorithm to perform well.

6.5 Summary and Future Work

In this chapter we presented an algorithm to identify and rejection outliers as a preprocessing step to pose graph optimization based collaborative localization. The algorithm utilized properties of relative pose measurements composed over cycles

to develop a metric on the set of edges indicative of the presence of an outlier. A hypothesis test was then utilized to identify the likely set of outliers.

Simulations were presented that studied the effectiveness of the outlier rejection algorithm when utilized before performing collaborative location on a set of robots. It was shown that, while the presence of outliers can cause collaborative localization to perform even worse then dead reckoning, the outlier rejection algorithm succeeds in the task of removing outliers to such a extent that performance is increased nearly to that of the no-outlier case.

In addition, simulations were presented that explored the validity of the assumption that edge consistency costs were i.i.d. log-normal. The critical conclusion of this simulation was that, if an edge e' is an outlier, the corresponding approximate edge consistency cost $\hat{D}(e')$ is an outlier of the log-normal distribution with mean and variance given by the sample mean and variance of the set $\left\{\hat{D}(e)\right\}$. Because of this, we expect to correctly identify outlying measurements using Grubbs' test for outliers.

CHAPTER 7
CONCLUSION AND FUTURE WORK

7.1 Conclusion

In this dissertation we presented work considering both single robot localization and multi-robot collaborative localization. In Chapter 2 we studied the growth of error in position estimates when dead reckoning is used to localize a robot without access to GPS. We examined the growth of error in position estimates obtained from noisy relative pose measurements. We showed that in both 2-D and 3-D, the bias and the variance of the position estimation error grows at most linearly with time or distance travelled. The precise growth rate of the bias depends on the trajectory of the robot. Specifically, if the robot stays in a bounded region, the bias is upper bounded by a constant for all time. It was proved that the variance growth rate is also lower bounded by linear function of time if the translation measurement errors are large enough. Exact formulas for the error bias and variance were obtained for two special 2-D trajectories, straight line and periodic. Extensive Monte-Carlo simulations, and experiments with a wheeled robot, were used to verify the results.

The results of this chapter show that localization error growth rate is not superlinear with time or distance even without absolute orientation sensors. In addition, it turns out that the asymptotic growth rate of the bias does not change even if all the measurements are unbiased or even if the translation measurements are completely error free. The bias growth is principally due to the fact that the expected value of the estimated position converges to a point, irrespective of how the robot is moving. This occurs since γ, the norm of the expected rotation error, is strictly less than unity. As a result, the magnitude of the measured translation, once the measurement is transformed to the global coordinate frame, decays geometrically with time.

In Chapter 3 we introduced a novel distributed algorithm for estimating the 3-D pose of multiple robots, referred to as the RPGO algorithm. The algorithm utilizing noisy inter-robot measurements of various types (relative pose, orientation, position, bearing, or distance) between pairs of robots, when available, along with the noisy inter-time relative pose measurements usually utilized for dead reckoning. The distributed algorithm is inspired by a centralized algorithm for solving a least-squares type problem in which the natural manifold structure of the space of rotations is utilized. The least-squares like cost function is chosen to measure how well the estimates explain the relative measurements. A gradient-descent in a product Riemannian manifold is used to solve the optimization problem ensuring the estimates remain on the manifold without the need for any projection. The proposed algorithm does not rely on any particular parameterization of the underlying manifold; allowing any parameterization to be used in numerical computations without affecting the resulting estimates. The algorithm is provably correct in the sense that the solution converges to a critical point of the cost function as the number of iterations (gradient descent steps) increases. Then in Chapter 4, two state of the art collaborative localization algorithms were compared with the RPGO algorithm. The first algorithm, referred to as the Euclidean Pose Graph Optimization algorithm, was a classical least squares based pose graph optimization algorithm. The second algorithm was based on a indirect extended Kalman filter. In each case, simulations were presented that identified situations under which the RPGO provided more accurate pose estimates.

In Chapter 5 we identified reasonable distributions for each of the considered measurement types. The distributions were chosen in such a way as to depend on the true value of the given measurement. Additionally, as in previous chapters and in contrast to much of the previous work in this area, the orientations measurements were given their natural manifold structure, and so the corresponding distribution was defined over the manifold $SO(3)$ with respect to the appropriate probability measure.

A novel collaborative localization algorithm, referred to as the ML-RPGO algorithm, was then proposed based on the maximum likelihood estimator for the absolute pose of each of the robots. The resulting algorithm was given by minimizing the negative log-likelihood function. As in Chapter 3 this results in an optimization problem of the product Riemannian manifold. Simulations were presented that studied the effect of increasing numbers of neighbors on estimation accuracy. Finally, simulations were presented comparing the ML-RPGO algorithm with a state of the art camera network localization algorithm.

Finally, in Chapter 6 we presented an algorithm to identify and rejection outliers as a preprocessing step to pose graph optimization based collaborative localization (such as the algorithm presented in Chapter 3). The algorithm utilized properties of relative pose measurements composed over cycles to develop a metric on the set of edges indicative of the presence of an outlier. A hypothesis test was then utilized to identify the likely set of outliers.

Simulations were presented that studied the effectiveness of the outlier rejection algorithm when utilized before performing collaborative location on a set of robots. It was shown that, while the presence of outliers can cause collaborative localization to perform even worse then dead reckoning, the outlier rejection algorithm succeeds in the task of removing outliers to such a extent that performance is increased nearly to that of the no-outlier case.

In addition, simulations were presented that explored the validity of the assumption that edge consistency costs were i.i.d. log-normal. The critical conclusion of this simulation was that, if an edge e' is an outlier, the corresponding approximate edge consistency cost $\hat{D}(e')$ is an outlier of the log-normal distribution with mean and variance given by the sample mean and variance of the set $\left\{\hat{D}(e)\right\}$. Because of this, we expect to correctly identify outlying measurements using Grubbs' test for outliers.

7.2 Future Work

Throughout the later chapters of this work, there existed the implicit underlying assumption that robots were capable of being uniquely identified. Though this is a common assumption in the literature on collaborative localization, it is not easy to achieve in practice. It may be possible to use the inter-robot measurements to identify a robot, as done in [33]. A study of how such methods might be utilized, and how those methods will affect the localization estimates is an area of future study.

In Chapter 5, statistical information about the relative measurements was utilized to develop a maximum likelihood estimator. One interesting area of future work is to further investigate the validity of the chosen distributions through the use of hypothesis testing on experimental data. Another critical area of future study is that of propagating the statistical information. That is, identifying the distributions of the pose estimates after the ML-RPGO algorithm has been utilized. Once that propagation step has been considered, the ML-RPGO algorithm can be distributed in much the same was as the RPGO algorithm presented in Section 3.3.

The success of the outlier rejection algorithm presented in Chapter 6 is highly dependent topology of the underlying measurement graph. In particular, while it is clear that the number of edges affect the robustness of the outlier rejection algorithm, an in-depth study of the connection between the maximum number of outliers for which the rejection algorithm succeeds versus the topology of the measurement graph remains an active area for future work. Future studies will also attempt to extend the proposed outlier rejection scheme to other measurement types: position, bearing, and distance.

Finally, in addition to the centralized algorithm presented in Section 6.2, a method to distributed the outlier rejection algorithm was also briefly outlined in Remark 6.2. In such a distributed scheme, each robot poses a different local measurement graph that is only a small part of the total graph. How the outlier rejection algorithm performs on

these local graphs as compared to the full centralized solution presented here requires further study.

APPENDIX A
SINGLE ROBT: PROOFS

Proof of Prop. 2.1. Let $\mathbf{y}$ be a d-dimensional random vector. Since $Cov(\mathbf{y}, \mathbf{y}) = \mathrm{E}[\mathbf{y}\mathbf{y}^T] - \mathrm{E}[\mathbf{y}]\,\mathrm{E}[\mathbf{y}]^T$, we have upon taking the trace of both sides

$$\|\,\mathrm{E}[\mathbf{y}]\|^2 = \mathrm{E}[\|\mathbf{y}\|^2] - \mathrm{Tr}\,[Cov(\mathbf{y}, \mathbf{y})] \leq \mathrm{E}[\|\mathbf{y}\|^2],$$

since $\mathrm{Tr}\,[Cov(\mathbf{y}, \mathbf{y})] \geq 0$. Moreover equality in the above inequality holds if and only if the variance of each of the components of $\mathbf{y}$ is 0, that is, $\mathbf{y}$ is degenerate. We now apply this result to the random vector $\mathbf{y} := \mathbf{R}x$, where x is a deterministic d-dimensional vector while $\mathbf{R}$ is a random rotation matrix:

$$\|\,\mathrm{E}[\mathbf{R}]x\|^2 \leq \mathrm{E}[\|\mathbf{R}x\|^2] = \mathrm{E}[\|x\|^2] = \|x\|^2, \tag{A–1}$$

where the first equality is due to the fact that rotation doesn't change the 2-norm of a vector, and the second equality is due to x being deterministic. This proves that $\|\,\mathrm{E}[\mathbf{R}]\| \leq 1$. Since $\mathbf{y}$ is degenerate if only if $\mathbf{R}$ is, the inequality in (A–1) is strict if $\mathbf{R}$ is non-degenerate. This proves the result. ■

The following additional technical result is needed for the proof of Lemma 1.

Proposition A.1. *If X_i is a sequence of random vectors such that $\mathrm{E}[X_i^T X_j] \leq \overline{\alpha_0}\eta^{|i-j|}$, where $|\eta| < 1$ and $\overline{\alpha_0}$ is an arbitrary constant, then*

$$\mathrm{E}[(\sum_{i=1}^{n} X_i)^T(\sum_{i=1}^{n} X_i)] \leq \overline{\alpha_0}\frac{1+\eta}{1-\eta}n.$$

If in addition $\underline{\alpha_0}\eta^{|i-j|} \leq \mathrm{E}[X_i^T X_j]$ for $i \neq j$ and $0 < \beta_0 \leq \mathrm{E}[X_i^T X_i]$, where $\underline{\alpha_0}, \underline{\beta_0}$ are constants such that $\underline{\beta_0} > 2\frac{|\underline{\alpha_0}|}{1-|\eta|}$, then $\mathrm{E}[(\sum_{i=1}^{n} X_i)^T(\sum_{i=1}^{n} X_i)] = \Theta(n)$.

Proof of Prop. A.1. Expanding the sum, we obtain

$$\mathrm{E}[(\sum_{i=1}^{n} X_i)^T(\sum_{i=1}^{n} X_i)] = \sum_{i=1}^{n} T_i, \tag{A–2}$$

where

$$T_i := \sum_{j=1}^{n} \mathrm{E}[X_i^T X_j]. \tag{A–3}$$

It follows from (A–3) and the hypothesis that

$$\begin{aligned} T_i &\le \overline{\alpha_0}(\eta^{i-1} + \eta^{i-2} + \cdots + \eta + 1 + \eta + \cdots + \eta^{n-i}) \\ &= \overline{\alpha_0}(-1 + \sum_{k=0}^{i-1} \eta^k + \sum_{k=0}^{n-i} \eta^k) \le \overline{\alpha_0}(-1 + \sum_{k=0}^{\infty} |\eta|^k + \sum_{k=0}^{\infty} |\eta|^k) \\ &= \overline{\alpha_0}(-1 + \frac{1}{1-|\eta|} + \frac{1}{1-|\eta|}) = \overline{\alpha_0}\frac{1+|\eta|}{1-|\eta|}, \end{aligned}$$

where the second inequality follows from $|\eta| < 1$. The upper bound now follows from (A–2). This proves the first statement.

When the additional hypothesis holds, we have

$$\begin{aligned} T_i &\ge \underline{\alpha_0}(\eta^{i-1} + \eta^{i-2} + \cdots + \eta) + \underline{\beta_0} + \underline{\alpha_0}(\eta + \cdots + \eta^{n-i}) \\ &\ge -2|\underline{\alpha_0}| \sum_{k=0}^{\infty} |\eta|^k + \underline{\beta_0} = \underline{\beta_0} - 2\frac{|\underline{\alpha_0}|}{1-|\eta|} =: \ell_0 > 0 \end{aligned}$$

It follows from (A–2) that $\mathrm{E}[(\sum_{i=1}^n X_i)^T(\sum_{i=1}^n X_i)] \ge n\ell_0 = \Omega(n)$. Combining the asymptotic lower and upper bounds, we get $\mathrm{E}[(\sum_{i=1}^n X_i)^T(\sum_{i=1}^n X_i)] = \Theta(n)$. ■

Proof of Lemma 1. It follows from (2–4) that

$$\mathrm{E}[\hat{\mathbf{t}}_{0,n}^0] = \sum_{k=1}^{n} \mathrm{E}[\hat{\mathbf{t}}_{k,k+1}^0] \tag{A–4}$$

From (2–2)-(2–3) we get

$$\begin{aligned} \hat{\mathbf{t}}_{k,k+1}^0 &= \mathbf{R}_1^0 \tilde{\mathbf{R}}_1^0 \ldots \mathbf{R}_{k+1}^k \tilde{\mathbf{R}}_{k+1}^k \left(\mathbf{t}_{k,k+1}^{k+1} + \tilde{\mathbf{t}}_{k,k+1}^{k+1}\right) \\ \Rightarrow \mathrm{E}[\hat{\mathbf{t}}_{k,k+1}^0] &= \mathbf{R}_k^0 \overline{R} \ldots \mathbf{R}_{k+1}^k \left(\overline{R}\ \mathbf{t}_{k,k+1}^{k+1} + \boldsymbol{\rho}_{k+1}\right) \end{aligned}$$

where the second equality follows from the assumption that the orientation measurement errors are i.i.d. Since a rotation does not change the 2-norm of a vector,

$$\| \mathrm{E}[\hat{\mathbf{t}}^0_{k,k+1}]\| \leq \|\overline{R}^k\| \left(\|\overline{R}\| \ \ \|\mathbf{t}^{k+1}_{k,k+1}\| + \|\boldsymbol{\rho}_{k+1}\| \right)$$

where the inequality follows from applying triangle inequality and using sub-multiplicative property of induced norms. Since $\|\overline{R}^k\| \leq \|\overline{R}\|^k$, we obtain upon using Proposition 2.1 and the definition $\gamma = \|\overline{\mathbf{R}}\|$ that

$$\| \mathrm{E}[\hat{\mathbf{t}}^0_{k,k+1}]\| \leq \gamma^k a,$$

where $a := \sup_k (\|\overline{R}\|\|\mathbf{t}^{k+1}_{k,k+1}\| + \|\boldsymbol{\rho}_{k+1}\|) \leq \gamma\tau + \beta$. Applying triangle inequality to (A–4), we get

$$\| \mathrm{E}[\hat{\mathbf{t}}^0_{0,n}]\| \leq \sum_{k=0}^{n-1} \| \mathrm{E}[\hat{\mathbf{t}}^0_{k,k+1}]\| \qquad \leq a \sum_{k=0}^{n-1} \gamma^k \leq a \frac{1-\gamma^n}{1-\gamma},$$

since $0 < \gamma < 1$. This proves the result about the mean.

The proof for the second moment result proceeds by first showing that $\mathrm{E}[(\hat{\mathbf{t}}^0_{j,j+1})^T \hat{\mathbf{t}}^0_{i,i+1}]$ satisfies the hypothesis of Proposition A.1 and then applying the proposition. We note that for $i \leq j$,

$$\begin{aligned}
(\hat{\mathbf{t}}^0_{i,i+1})^T \hat{\mathbf{t}}^0_{j,j+1} &= (\hat{\mathbf{t}}^{i+1}_{i,i+1} \hat{\mathbf{R}}^0_{i+1})^T \hat{\mathbf{R}}^0_{j+1} \hat{\mathbf{t}}^{j+1}_{j,j+1} \\
&= (\hat{\mathbf{t}}^{i+1}_{i,i+1})^T \hat{\mathbf{R}}^{i+1}_{j+1} \hat{\mathbf{t}}^{j+1}_{j,j+1} \\
&= V_1 + V_2 + V_3 + V_4,
\end{aligned}$$

where

$$\begin{aligned}
V_1 &:= (\mathbf{t}^{i+1}_{i,i+1})^T \ \ \hat{\mathbf{R}}^{i+1}_{j+1} \ \ \mathbf{t}^{j+1}_{j,j+1} \\
V_2 &:= (\tilde{\mathbf{t}}^{i+1}_{i,i+1})^T \ \ \hat{\mathbf{R}}^{i+1}_{j+1} \ \ \mathbf{t}^{j+1}_{j,j+1} \\
V_3 &:= (\mathbf{t}^{i+1}_{i,i+1})^T \ \ \hat{\mathbf{R}}^{i+1}_{j+1} \ \ \tilde{\mathbf{t}}^{j+1}_{j,j+1} \\
V_4 &:= (\tilde{\mathbf{t}}^{i+1}_{i,i+1})^T \ \ \hat{\mathbf{R}}^{i+1}_{j+1} \ \ \tilde{\mathbf{t}}^{j+1}_{j,j+1}.
\end{aligned}$$

We now evaluate the expected values of these four terms. By using the Independence of the orientation measurement errors and, we get

$$\begin{aligned} \mathrm{E}[V_1] &= (\mathbf{t}_{i,i+1}^{i+1})^T \ \mathbf{R}_{i+2}^{i+1}\overline{R}\ldots\mathbf{R}_{j+1}^{j}\overline{R}\ \mathbf{t}_{j,j+1}^{j+1} \\ \Rightarrow |\,\mathrm{E}[V_1]| &\leq \|\mathbf{t}_{i,i+1}^{i+1}\|\ \|\overline{R}^{j-i}\ \mathbf{t}_{j,j+1}^{j+1}\| \\ &\leq \|\overline{R}^{j-i}\|\ \|\mathbf{t}_{i,i+1}^{i+1}\|\ \|\mathbf{t}_{j,j+1}^{j+1}\| \leq \gamma^{j-i}\tau^2, \end{aligned}$$

where the first inequality uses the fact that rotations do not change the 2-norm. For V_2, since $\tilde{\mathbf{t}}_{i,i+1}^{i+1}$ is statistically dependent only on $\tilde{\mathbf{R}}_{i+1}^{i}$ and not on $\tilde{\mathbf{R}}_{i+2}^{i+1}$, ..., $\tilde{\mathbf{R}}_{j+1}^{j}$, it is also independent of $\hat{\mathbf{R}}_{j+1}^{i+1}$. Hence,

$$|\,\mathrm{E}[V_2]| = |\mathbf{b}_i\ \mathbf{R}_{i+2}^{i+1}\ \overline{R}\mathbf{R}_{j+1}^{j}\ \overline{R}\mathbf{t}_{j,j+1}^{j+1} \qquad \Rightarrow \qquad |\,\mathrm{E}[V_2]| \leq \gamma^{j-i}\ b\tau.$$

Similarly, we have, for $i < j$,

$$\begin{aligned} \mathrm{E}[V_3] &= (\mathbf{t}_{i+1,i+1}^{i})^T\ \mathbf{R}_{j+1}^{i+1}\ \overline{R}\ \mathbf{R}_{j}^{j-1}\overline{R}\ \boldsymbol{\rho}_{j+1} \\ \Rightarrow |\,\mathrm{E}[V_3]| &\leq \gamma^{j-i}\frac{1}{\gamma}\tau\rho. \end{aligned}$$

and for $i = j$, $|\,\mathrm{E}[V_3]| \leq \tau b$. For V_4, when $i < j$, we have

$$\begin{aligned} V_4 &= (\tilde{\mathbf{t}}_{i,i+1}^{i+1})^T\ \mathbf{R}_{i+2}^{i+1}\ \tilde{\mathbf{R}}_{i+2}^{i+1}\ldots\mathbf{R}_{j+1}^{j}\ \tilde{\mathbf{R}}_{j+1}^{j}\ \tilde{\mathbf{t}}_{j,j+1}^{j+1}, \\ \Rightarrow |\,\mathrm{E}[V_4]| &\leq \|\mathbf{b}_i\|\ \|\overline{\mathbf{R}}\|^{j-i-1}\|\boldsymbol{\rho}_{j+1}\| \leq \gamma^{j-i}\frac{1}{\gamma}b\rho. \end{aligned}$$

When $i = j$, we have $V_4 = (\tilde{\mathbf{t}}_{j,j+1}^{j+1})^T\ \tilde{\mathbf{t}}_{j,j+1}^{j+1}$, which implies $\mathrm{E}[V_4] = \mathrm{Tr}\,[\mathbf{P}_{j+1}] + \mathbf{b}_{j+1}^T\mathbf{b}_{j+1}$, by definition. Therefore,

$$0 < \underline{p} \leq \mathrm{E}[V_4] \leq \overline{p} + b^2. \quad (i = j).$$

Combining all four terms, we get,

$$\begin{aligned} \underline{\alpha_0}\gamma^{j-i} \leq \mathrm{E}[(\hat{\mathbf{t}}_{i,i+1}^{0})^T\hat{\mathbf{t}}_{j,j+1}^{0}] &\leq \alpha_0\gamma^{j-i}, \quad (i < j) \\ \underline{\beta_0} \leq \mathrm{E}[(\hat{\mathbf{t}}_{i,i+1}^{0})^T\hat{\mathbf{t}}_{i,i+1}^{0}] &\leq \beta_0, \end{aligned}$$

where $\underline{\alpha_0} := -(\tau^2+\tau b+\frac{1}{\gamma}\tau\rho+\frac{1}{\gamma}b\rho)$, $\alpha_0 := \tau^2+\tau b+\frac{1}{\gamma}\tau\rho+\frac{1}{\gamma}b\rho$, and $\underline{\beta_0} := \underline{p}-(\tau^2+2\tau b)$, $\beta_0 := \tau^2+2\tau b+\overline{p}+b^2$. Note that in case $\underline{\beta_0}$ is negative, it is a poor lower lower bound since $(\hat{\mathbf{t}}^0_{i,i+1})^T\hat{\mathbf{t}}^0_{i,i+1} > 0$. Repeating these arguments for $i \geq j$ and combining, we find that

$$\underline{\alpha_0}\gamma^{|i-j|} \leq \mathsf{E}[(\hat{\mathbf{t}}^0_{i,i+1})^T\hat{\mathbf{t}}^0_{j,j+1}] \leq \overline{\alpha_0}\gamma^{|i-j|}, \quad (i \neq j)$$
$$\max\{0, \underline{\beta_0}\} \leq \mathsf{E}[(\hat{\mathbf{t}}^0_{i,i+1})^T\hat{\mathbf{t}}^0_{i,i+1}] \leq \overline{\alpha_0},$$

where $\overline{\alpha_0} := \max\{\alpha_0, \beta_0\}$. Now call $X_i := \hat{\mathbf{t}}^0_{i,i+1}$, so that $\hat{\mathbf{t}}^0_{0,n} = \sum_{i=0}^{n-1} X_i$. Hence, $\mathsf{E}[(\hat{\mathbf{t}}^0_{0,n})^T\hat{\mathbf{t}}^0_{0,n}] = \mathsf{E}[(\sum_{i=0}^{n-1} X_i)^T(\sum_{j=0}^{n-1} X_j)]$. It now follows from Proposition A.1 that $\mathsf{E}[(\hat{\mathbf{t}}^0_{0,n})^T\hat{\mathbf{t}}^0_{0,n}]$ is $O(n)$, and is $\Theta(n)$ if $\underline{\beta_0} > 2\frac{|\alpha_0|}{1-\gamma}$. Since $|\underline{\alpha_0}| = \tau^2+\tau b+\tau\rho$, the condition $\underline{\beta_0} > 2\frac{|\alpha_0|}{1-\gamma}$ is equivalent to $\underline{p} > 2b\tau+\tau^2+2\dfrac{(\tau+\rho/\gamma)(\tau+b)}{1-\gamma}$, which proves the result. ■

Proof of 2.2. Define a new random variable, $\delta\tilde{\theta}_{k-1,k} := \tilde{\theta}_{k-1,k} - \mathsf{E}[\tilde{\theta}_{k-1,k}]$. Then $\{\delta\tilde{\theta}_{k-1,k}\}_{k=0}^{\infty}$ is an i.i.d. sequence and the marginal density of $\delta\tilde{\theta}_{k-1,k}$ is symmetric about 0. We define the corresponding rotation matrices $\delta\tilde{R}^i_j := f_R(\delta\tilde{\theta}_{i,j})$. Utilizing the commutative property of 2-D rotation matrices, we have $\tilde{\mathbf{R}}^i_j = \left(\underline{\mathbf{R}}^{j-i}\right)\delta\tilde{R}^i_j$. It then follows from (2–5) that

$$\mathbf{e}(n) = n\mathbf{r} - \hat{\mathbf{t}}^0_{0,n}$$

and from (2–4), (2–3), and (2–2) that

$$\hat{\mathbf{t}}^0_{0,n} = \sum_{k=1}^{n}\left(\prod_{i=1}^{k}\underline{\mathbf{R}}\delta\tilde{R}^{i-1}_i\right)\left(\mathbf{r}+\tilde{\mathbf{t}}^k_{k-1,k}\right),$$

where we have used the fact that $\hat{\mathbf{R}}^{i-1}_i = \mathbf{R}^{i-1}_i\tilde{\mathbf{R}}^{i-1}_i = \underline{\mathbf{R}}\delta\tilde{R}^{i-1}_i$ since $\mathbf{R}^{i-1}_i = I$ due to the nature of the trajectory. We define two new random variables

$$\mathbf{f}_n := \sum_{k=1}^{n}\left(\prod_{i=1}^{k}\underline{\mathbf{R}}\delta\tilde{R}^{i-1}_i\right)\mathbf{r}$$
$$\mathbf{g}_n := \sum_{k=1}^{n}\left(\prod_{i=1}^{k}\underline{\mathbf{R}}\delta\tilde{R}^{i-1}_i\right)\tilde{\mathbf{t}}^k_{k-1,k},$$

so that

$$\hat{\mathbf{t}}^0_{0,n} = \mathbf{f}_n + \mathbf{g}_n. \tag{A–5}$$

By the i.i.d. assumption on the sequence $\{\tilde{\theta}_{k-1,k}\}_k$, the sequence $\{\delta\tilde{R}_k^{k-1}\}_k$ is also i.i.d., so that

$$\mathrm{E}[\delta\tilde{R}_j^i] = \mathrm{E}[\prod_{k=i+1}^{k} \delta\tilde{R}_k^{k-1}] = \prod_{k=i-1}^{j} \mathrm{E}[\delta\tilde{R}_k^{k-1}] = c^{j-i} I, \tag{A–6}$$

where we have used the fact that $\mathrm{E}[\sin\delta\tilde{\theta}_{i-1,i}] = 0$, which follows from Assumption 2.1. It is then straightforward to show that

$$\mathrm{E}[\mathbf{f}_n] = \sum_{k=1}^{n} (c\underline{\mathbf{R}})^k \, \mathbf{r} = (I - c\underline{\mathbf{R}})^{-1} \left(I - (c\underline{\mathbf{R}})^n\right) c\underline{\mathbf{R}}\mathbf{r}$$
$$\mathrm{E}[\mathbf{g}_n] = \sum_{k=0}^{n-1} (c\underline{\mathbf{R}})^k \, \boldsymbol{\rho} = (I - c\underline{\mathbf{R}})^{-1} \left(I - (c\underline{\mathbf{R}})^n\right) \boldsymbol{\rho}$$

The expected value $\mathbf{e}(n)$ is now

$$\mathrm{E}[\mathbf{e}(n)] = n\,\mathbf{r} - (I - c\underline{\mathbf{R}})^{-1} \left(I - (c\underline{\mathbf{R}})^n\right) (c\underline{\mathbf{R}}\mathbf{r} + \boldsymbol{\rho}) \tag{A–7}$$

which proves the first equality in (2–18).

For the variance, it follows from (A–5) that

$$\begin{aligned} \mathrm{Tr}\left[Cov(\mathbf{e}(n), \mathbf{e}(n))\right] &= \mathrm{Tr}\left[Cov(\hat{\mathbf{t}}^0_{0,n}, \hat{\mathbf{t}}^0_{0,n})\right] \\ &= \mathrm{E}[\mathbf{f}_n^T \mathbf{f}_n] + \mathrm{E}[\mathbf{g}_n^T \mathbf{g}_n] + 2\,\mathrm{E}[\mathbf{f}_n^T \mathbf{g}_n] \\ &\quad - \mathrm{E}[\hat{\mathbf{t}}^0_{0,n}]^T \,\mathrm{E}[\hat{\mathbf{t}}^0_{0,n}]. \end{aligned} \tag{A–8}$$

$$\begin{aligned}
\mathrm{E}[\mathbf{f}_n^T \mathbf{f}_n] &= \mathbf{r}^T \,\mathrm{E}\left[\sum_{i=1}^{n}\left(\prod_{j=1}^{i}\underline{\mathbf{R}}\;\delta\tilde{R}_j^{j-1}\right)^T \sum_{k=1}^{n}\left(\prod_{\ell=1}^{k}\underline{\mathbf{R}}\;\delta\tilde{R}_\ell^{\ell-1}\right)\right]\mathbf{r} \\
&= \mathbf{r}^T\Big[\left(I + c\underline{\mathbf{R}}^T + \cdots + (c\underline{\mathbf{R}}^T)^{n-1}\right) \\
&\quad + \left(c\underline{\mathbf{R}} + I + c\underline{\mathbf{R}}^T + \cdots + (c\underline{\mathbf{R}}^T)^{n-2}\right) \\
&\quad \cdots + \left((c\underline{\mathbf{R}})^{n-1} + \cdots + I\right)\Big]\mathbf{r}
\end{aligned}$$

where we have used the independence of the sequence $\{\delta\tilde{R}_k^{k-1}\}_k$ and the fact that $\delta\tilde{R}_k^{k-1}\delta\tilde{R}_k^{k-1} = I = \underline{\mathbf{R}}\,\underline{\mathbf{R}}^T$. The expression above simplifies to

$$\begin{aligned}
\mathrm{E}[\mathbf{f}_n^T \mathbf{f}_n] = \mathbf{r}^T\left[nI + 2\sum_{k=1}^{n-1}(n-k)\,(c\underline{\mathbf{R}})^k\right]\mathbf{r} &= \mathbf{r}^T\,(I - c\underline{\mathbf{R}})^{-2} \\
\times\left(I + 2(n-2)c\underline{\mathbf{R}} - 2(n-1)\,(c\underline{\mathbf{R}})^2 + 2\,(c\underline{\mathbf{R}})^{n+1}\right)&\mathbf{r}.
\end{aligned}$$

To examine $\mathrm{E}[\mathbf{g}_n^T\mathbf{g}_n]$, we express the product as $\mathbf{g}_n^T\mathbf{g}_n = \sum_{k=1}^{n} T_k$ where

$$\begin{aligned}
T_k = (\tilde{\mathbf{t}}_{k-1,k}^{k})^T\left((\delta\tilde{R}_k^{k-1})^T\;\;(\delta\tilde{R}_{k-1}^{k-2})^T \ldots (\delta\tilde{R}_1^{0})^T\right)(\underline{\mathbf{R}}^k)^T \\
\times\left(\underline{\mathbf{R}}\;\;\delta\tilde{R}_1^0\tilde{\mathbf{t}}_{0,1}^{1} + \cdots + \underline{\mathbf{R}}^n\;\delta\tilde{R}_1^0 \ldots \delta\tilde{R}_k^{k-1}\tilde{\mathbf{t}}_{k-1,k}^{k}\right).
\end{aligned}$$

Taking expectation and using the assumptions on the noise correlations, we get for $k > 1$,

$$\begin{aligned}
\mathrm{E}[T_k] = \mathrm{Tr}\left[P + \mathbf{b}\mathbf{b}^T\right] + \mathbf{b}^T((c\underline{\mathbf{R}})^{k-2} + (c\underline{\mathbf{R}})^{k-3} + \cdots + I \\
+ I + (c\underline{\mathbf{R}}) + (c\underline{\mathbf{R}})^2 + \cdots + (c\underline{\mathbf{R}})^{n-1-k})\boldsymbol{\rho},
\end{aligned}$$

and for $k = 1$, $\mathrm{E}[T_k] = \mathrm{Tr}\left[\mathbf{P} + \mathbf{b}\mathbf{b}^{\mathrm{T}}\right] + \mathbf{b}^T(I + c\underline{\mathbf{R}} + (c\underline{\mathbf{R}})^2 + \cdots + (c\underline{\mathbf{R}})^{n-1-k})\boldsymbol{\rho}$. Repeating this for all the T_k's we get:

$$\begin{aligned}
\mathrm{E}[\mathbf{g}_n^T\mathbf{g}_n] &= n\,\mathrm{Tr}\left[\mathbf{P} + \mathbf{b}\mathbf{b}^T\right] + \mathbf{b}^T\left[2\sum_{k=0}^{n-2}(n-k-1)\,(c\underline{\mathbf{R}})^k\right]\boldsymbol{\rho} \\
&= n\,\mathrm{Tr}\left[\mathbf{P} + \mathbf{b}\mathbf{b}^T\right] + \mathbf{b}^T\,(I - c\underline{\mathbf{R}})^{-2} \times \\
&\quad [2(n-1)I - 2nc\underline{\mathbf{R}} + 2\,(c\underline{\mathbf{R}})^n]\,\boldsymbol{\rho}.
\end{aligned}$$

Similar tedious calculations lead to the following

$$\begin{aligned}\mathrm{E}[\mathbf{f}_n^T \mathbf{g}_n] &= \left[\sum_{k=0}^{n-1} \mathbf{b}^T (c\underline{\mathbf{R}})^k + \sum_{k=0}^{n-2} (n-k-1)\boldsymbol{\rho}^T (c\underline{\mathbf{R}}^T)^k\right] \mathbf{r} \\ &= \mathbf{b}^T (I - c\underline{\mathbf{R}})^{-2} \left[I - c\underline{\mathbf{R}} - (c\underline{\mathbf{R}})^n + (c\underline{\mathbf{R}})^{n+1}\right] \mathbf{r} \\ &\quad + \mathbf{r}^T (I - c\underline{\mathbf{R}})^{-2} \left[(n-1)I - nc\underline{\mathbf{R}} + (c\underline{\mathbf{R}})^n\right] \boldsymbol{\rho}.\end{aligned}$$

Plugging all of this back in (A–8), we get $\mathrm{Tr}\left[Cov\left(\mathbf{e}(n), \mathbf{e}(n)\right)\right] = \psi n + \omega(n)$, where $\psi, \omega(n)$ are given in (2–15). This proves the second equality in (2–18). ■

Proof of Thrm. 2.3. Define a new random variable, $\delta\tilde{\theta}_{k-1,k} := \tilde{\theta}_{k-1,k} - \mathrm{E}[\tilde{\theta}_{k-1,k}]$. The sequence $\{\delta\tilde{\theta}_{k-1,k}\}_{k=0}^{\infty}$ are then i.i.d. and the marginal density of $\delta\tilde{\theta}_{k-1,k}$ is symmetric about the origin for each k. We define the corresponding rotation matrices $\delta\tilde{R}_j^i := f_R(\delta\tilde{\theta}_{i,j})$. Utilizing the commutative property of rotations in 2-D, we have the following relation

$$\tilde{\mathbf{R}}_j^i = \left(\underline{\mathbf{R}}^{j-i}\right) \delta\tilde{R}_j^i \tag{A–9}$$

To examine the bias, we first re-write the position estimate $\hat{\mathbf{t}}_{0,n}^0$ as

$$\begin{aligned}\hat{\mathbf{t}}_{0,n}^0 = \sum_{i=0}^{n} \hat{\mathbf{t}}_{i,i+1}^0 &= \sum_{k=0}^{\eta-1} \left(\sum_{m=1}^{p} \hat{\mathbf{t}}_{kp+m-1,kp+m}^0\right) \\ &\quad + \sum_{j=1}^{q} \hat{\mathbf{t}}_{\eta p+j-1,\eta p+j}^0,\end{aligned} \tag{A–10}$$

where the first term is sum is over all time steps up to the end of the last (η-th) period and the second term for the time steps after that. For any $0 \leq m < p$, we have

$$\begin{aligned}\hat{\mathbf{t}}_{kp+m-1,kp+m}^0 &= \hat{\mathbf{R}}_{kp+m}^0 \ \hat{\mathbf{t}}_{kp+m-1,kp+m}^{kp+m} \\ &= \mathbf{R}_m^0 \ \tilde{\mathbf{R}}_{kp+m}^0 (\mathbf{t}_{m-1,m}^m + \tilde{\mathbf{t}}_{kp+m-1,kp+m}^{kp+m}),\end{aligned}$$

where apart from $\hat{R} = R\tilde{R}$, we have used the periodic nature of the trajectory that leads to $\mathbf{R}^0_{kp+m} = \mathbf{R}^0_m$ and $\mathbf{t}^{kp+m}_{kp+m-1,kp+m} = \mathbf{t}^m_{m-1,m}$. Taking expectation and using (A–9), we obtain

$$\mathrm{E}[\hat{\mathbf{t}}^0_{kp+m-1,kp+m}] = \mathbf{R}^0_m (c\underline{\mathbf{R}})^{kp+m-1}(c\underline{\mathbf{R}}\mathbf{t}^m_{m-1,m} + \boldsymbol{\rho}_m)$$

This expression is used to evaluate $\mathrm{E}[\mathbf{t}^0_{0,n}]$ by taking expectation of the right hand side of (A–10). After grouping terms, we obtain

$$\mathrm{E}[\mathbf{t}^0_{0,n}] = \left(\sum_{k=0}^{\eta-1}(c\underline{\mathbf{R}})^{kp}\omega(p)\right) + (c\underline{\mathbf{R}})^{\eta p}\omega(q) \tag{A–11}$$

Using techniques similar to those used in the proof of Theorem 2.2, it can be shown that

$$\begin{aligned}\mathrm{E}[\hat{\mathbf{t}}^0_{0,n}] &= \sum_{i=0}^{\eta-1}(c\underline{\mathbf{R}})^{ip} w + (c\underline{\mathbf{R}})^{\eta p} w(q) \\ \Rightarrow \mathrm{E}[\mathbf{e}(n)] &= \sum_{k=0}^{q-1}\mathbf{R}^0_{k+1}\mathbf{t}^{k+1}_{k,k+1} - \sum_{k=0}^{\eta-1}(c\underline{\mathbf{R}})^{kp}\, w - (c\underline{\mathbf{R}})^{\eta p} w(q).\end{aligned}$$

By replacing the summation we arrive at (2–20). ■

APPENDIX B
PRODUCT RIEMANNIAN MANIFOLDS

B.1 Riemannian Manifolds

In this section we provide greater detail on the building blocks of Algorithm 1, including Riemannian manifolds. A full introduction to the study of Riemannian geometry is outside the scope of this work; the interested reader is referred to [54, 62].

A manifold M of dimension d is a topological space that can locally be represented by $\mathbb{R}^d$. The set of all linear operators on the vector space $\mathbb{R}^3$ is denoted by $\mathbb{L}(\mathbb{R}^3)$. Elements of $\mathbb{L}(\mathbb{R}^3)$ can be represented by 3×3 matrices. We can then describe the manifold of 3-D rotations, denoted $SO(3)$, by the set $\left\{R \in \mathbb{L}(\mathbb{R}^3) : R^T R = id, det(R) = 1\right\}$. The symbol id denotes the identity operator. Given a point $p \in SO(3)$, the tangent space of $SO(3)$ at p, denoted $T_pSO(3)$ is given by

$$T_pSO(3) = \left\{p\hat{v} : \hat{v} \in \mathbb{L}(\mathbb{R}^3), \hat{v}^T = -\hat{v}\right\}.$$

A *Riemannian metric* on a manifold M is given by defining $\forall p \in M\ g_p : T_pM \times T_pM \to \mathbb{R}$ such that

- $\forall p \in M$, g_p is an inner product
- given two vector fields X and Y on M, the map $g : M \to \mathbb{R}$, $p \mapsto g_p(X_p, Y_p)$ is smooth.

The Riemannian metric g gives rise to an inner-product norm on the tangent space T_pM for each $p \in M$. That is, for $\xi \in T_pM$, $\|\xi\|^2 = g_p(\xi, \xi)$. A *Riemannian manifold* is a manifold equipped with a Riemannian metric. The first Riemannian manifold we consider is $(SO(3), g)$ were g is the Riemannian metric given by

$$g_p(A, B) = \frac{1}{2}\,\mathrm{Tr}\left[A^T B\right] \tag{B–1}$$

for all $p \in SO(3)$, $A, B \in T_pSO(3)$. From this point on, when we refer to $SO(3)$ we mean this Riemannian manifold. To further simplify the notation, when the argument $p \in SO(3)$ is clear, we will denote $g_p(X_p, Y_p)$ by $g(X_p, Y_p)$. For a Riemannian manifold,

the analog to a straight line in Euclidean space is given by a *geodesic*. For two points p and q on a manifold M, a geodesic from p to q, denoted γ_{pq}, is a "shortest" path from p to q. More precisely, if we consider the set of all parameterized paths from p to q, given by $\Gamma = \{\gamma : [0,1] \to M : \gamma(0) = p, \gamma(1) = q\}$ then a geodesic γ_{pq} is given by a path that minimizes

$$\int_0^1 \|\gamma'(t)\| dt$$

over all paths in Γ, and $\gamma'(t)$ is the derivative of $\gamma(t)$ with respect to t, which is defined by the following relationship:

$$\gamma'(t_0)f = \frac{d(f(\gamma(t))}{dt}|_{t_0}$$

for all $f : M \to \mathbb{R}$. Note that the derivative $\gamma'(0)$ is an element of the tangent space $T_{\gamma(0)}M$. For the manifold $SO(3)$, the geodesic from p to q ($\in SO(3)$) is given by

$$\gamma_{pq}(t) = p \exp(tv), \ t \in [0,1]$$

where $v = \log(p^{-1}q)$, the map $\exp : \mathbb{L}(\mathbb{R}^3) \to \mathbb{L}(\mathbb{R}^3)$ is given by $\exp(X) = \sum_{k=0}^{\infty} \frac{X^k}{k!}$ and log is the inverse map of exp[1] .

For an arbitrary Riemannian manifold, the Riemannian metric defines a corresponding distance function on the manifold as follow. For a Riemannian manifold M, the distance between two points $p, q \in M$ is given by

$$d_M(p,q) = \int_0^1 \|\gamma_{pq}(t)\| dt$$

[1] This use of $\log(p)$ for $p \in SO(3)$ is only valid on the region where $\exp$ is a diffeomorphism, which it is at least on the open ball about $I \in SO(3)$ of radius π [70].

In particular, for the manifold $SO(3)$,

$$d_{SO(3)}(p, q) = \sqrt{-\frac{1}{2}\,\mathrm{Tr}\left[\log^2(p^T q)\right]}.$$

The subscript $SO(3)$ on the distance function will be omitted when the arguments make it clear what manifold the distance refers to. The second manifold we consider is $(\mathbb{R}^3, < \cdot, \cdot >)$, standard Euclidean 3-space with the Riemannian metric given by the inner product on $\mathbb{R}^3$. Given points $p, q \in \mathbb{R}^3$ the (unique) geodesic connecting p and q is the straight line from p to q and $d(p, q) = \|p - q\| := < p - q, p - q >^{1/2}$, the standard Euclidean inner product norm.

In addition to the distance function, the Riemmanian metric also specifies a parallel transport function. Let M be an arbitrary Riemmanian manifold. For $p \in M$ and $\xi \in T_pM$, the *parallel transport* function, denoted $\exp_p$, finds a new point $q \in M$ given by moving along a geodesic beginning at p and with initial velocity ξ. More precisely,

$$q = \exp_p(\xi) = \gamma(1)$$

where γ is a parameterized geodesic with $\gamma(0) = p$ and ξ being the tangent to $\gamma(t)$ at $t = 0$. In the case of $SO(3)$, $\exp_p(\xi) = p\exp(p^T\xi)$.

Finally, we consider the gradient for real value functions defined on the manifold. Given a Riemannian manifold M and a scalar, real-valued function $f : M \to \mathbb{R}$, the gradient of f at $p \in M$, denoted by $grad\ f(p)$, can be defined by the following relation:

$$g_p(grad\, f(p), \xi_p) = (f \circ \gamma)'(0) \quad \forall p \in M. \tag{B–2}$$

where $\gamma : [0, 1] \to M$ is any curve with $\gamma(0) = p$, and $(f \circ \gamma)'(0) := \frac{d(f\circ\gamma)(t)}{dt}|_0$. For an example of a gradient calculation that will be useful later, consider the following function

$$f_1 : SO(3) \to \mathbb{R},\ p \mapsto \frac{1}{2}d^2(p, q) \tag{B–3}$$

for some fixed $q \in SO(3)$. For an arbitrary $p \in SO(3)$, we consider the geodesic $\gamma_{pq}(t) = p\exp(tv)$ where $v = log(p^{-1}q)$. It can be shown that

$$(f_1 \circ \gamma)(t) = \frac{1}{2}(1-t)^2 d^2(p,q)$$

and thus

$$(f_1 \circ \gamma)'(0) = -d^2(p,q) = \frac{1}{2}\,\mathrm{Tr}\left[v^2\right].$$

The corresponding tangent vector to $\gamma(t)$ at p is given by $\xi_p := \gamma'(0) = pv$. It follows from (B–1) that

$$g_p(grad\, f_1(p), \xi_p) = \frac{1}{2}\,\mathrm{Tr}\left[(grad\, f_1(p))^T pv\right].$$

Applying (B–2) we have

$$grad\, f_1(p) = -pv = -p\log(p^{-1}q) \quad \forall p \in SO(3). \tag{B–4}$$

For a function $f : SO(3) \to \mathbb{R}$ the gradient at $\mathbf{R} \in SO(3)$ is given by [68]:

$$grad\, f(\mathbf{R}) = f_{\mathrm{R}} - \mathbf{R} f_{\mathrm{R}}^T \mathbf{R} \tag{B–5}$$

where $f_{\mathrm{R}} \in L(\mathbb{R}^3)$ is the linear operator whose matrix representation (using the canonical basis vectors for $\mathbb{R}^3$) is given by: $(f_{\mathrm{R}})_{ij} = \frac{\partial f}{\partial \mathrm{R}_{ij}}$, where $\mathbf{R}_{ij}$ represents the (i,j)-th entry of the 3×3 matrix representation of $\mathbf{R}$. Gradient calculation for a function $f : \mathbb{R}^3 \to \mathbb{R}$ is straightforward.

Apart from $SO(3)$ and $\mathbb{R}^3$, the other Riemannian manifold that is useful in this study is $SE(3)$. Instead of $SE(3)$, however, we consider the equivalent manifold $SO(3) \times \mathbb{R}^3$. There exists a natural way to define the Riemannian metric on this manifold, based on the Riemannian metrics in $SO(3)$ and $\mathbb{R}^3$ so that we can deal with geodesics and gradients. This has to do with the fact that $SO(3) \times \mathbb{R}^3$ is a product manifold, which is the topic of the next section.

B.2 Product Manifold

Let $\left\{(M_i, g^{(i)})_{i=1}^n\right\}$ be a set of n Riemannian manifolds. We define the product Riemannian manifold (M, g) as follows:

$$\begin{aligned} M &:= M_1 \times \cdots \times M_n \\ g_p(\xi, \zeta) &:= \sum_{i=1}^{n} g_{p_i}^{(i)}(\xi_i, \zeta_i) \end{aligned} \tag{B–6}$$

for all $p = (p_1, \dots, p_n) \in M$ and all $\xi = (\xi_1, \dots, \xi_n),\ \zeta = (\zeta_1, \dots, \zeta_n) \in T_pM$. We first restrict our considerations to product Riemannian manifolds of the form $M = M_1 \times M_2$ (i.e. $n = 2$). The extension to any finite combination is straight forward. The following lemma will be useful in the sequel. The proof of the lemma can be found in [68].

Lemma 2 ([68]). *Let $M = M_1 \times M_2$ be a product Riemannian manifold as defined in (B–6) and consider a smooth function $f : M \to \mathbb{R}$. Then for any parameterized path $\gamma = (\gamma_1, \gamma_2) : [0, 1] \to M$,*

$$\frac{d}{dt} f(\gamma(t))|_{t_0} = \frac{d}{dt} f(\gamma_1(t), \gamma_2(t_0))\Big|_{t_0} + \frac{d}{dt} f(\gamma_1(t_0), \gamma_2(t))\Big|_{t_0}$$

where $t_0 \in [0.1]$.

Theorem B.1. *Let $M = M_1 \times M_2$ be a product Riemannian manifold, as defined in (B–6). Consider a smooth function $f : M \to \mathbb{R}$ and for all $(p, q) \in M$ ($p \in M_1$, $q \in M_2$), define*

$$f_1^q : M_1 \to \mathbb{R}, \quad a \mapsto f(a, q)$$

$$f_2^p : M_2 \to \mathbb{R}, \quad b \mapsto f(p, b).$$

Then for all $(p, q) \in M$

$$grad\, f(p, q) = \left(grad\, f_1^q(p), grad\, f_2^p(q)\right).$$

Proof. Fix $(p, q) \in M$ and consider an arbitrary tangent vector $\xi = (\xi_1, \xi_2) \in T_{(p,q)}M$. Choose a parameterized path $(\gamma_1, \gamma_2) = \gamma : [0, 1] \to M$ such that ξ_1 is tangent to $\gamma_1(t)$ at $p = \gamma_1(t_0)$ and ξ_2 is tangent to $\gamma_2(t)$ at $q = \gamma_2(t_0)$ (and thus ξ is tangent to $\gamma(t)$ at

$(p, q) = \gamma(t_0)$). Using lemma 2, we have

$$(f \circ \gamma)'(t_0) = \frac{d}{dt}(f_1^q \circ \gamma_1)|_{t_0} + \frac{d}{dt}(f_2^p \circ \gamma_2)|_{t_0}.$$

From this relation and (B–2), we find that

$$g(grad\, f(p, q), \xi) = g^{(1)}(grad\, f_1^q(p), \xi_1) + g^{(2)}(grad\, f_2^p(q), \xi_2). \tag{B–7}$$

Let $grad\, f(p, q) = (A, B) \in T_{(p,q)M}$ for some unknown $A \in T_pM,\ B \in T_qM$. Using the definition of g given in (B–6), we can rewrite (B–7) as

$$g^{(1)}(A, \xi_1) + g^{(2)}(B, \xi_2) = g^{(1)}(grad\, f_1^q(p), \xi_1) + g^{(2)}(grad\, f_2^p(q), \xi_2).$$

Since this equality holds for all $\xi_1 \in T_pM$ and all $\xi_2 \in T_qM$, we have $A = grad\, f_1^q(p)$, $B = grad\, f_2^p(q)$, which completes the proof. □

As the number of manifolds increases, the notation $grad\, f_1^q(p)$ becomes more cumbersome. We will therefore simplify the notation by writing $grad\, f(p)$ (to mean $grad\, f_1^q(p)$) whenever the manifold the gradient found with respect to is made obvious by the argument p and q is clear from context.

The following corollary is a direct consequence of Theorem B.1.

Corollary 1. *Given a set of n Riemannian manifolds* $\{(M_i, g_i)\}_{i=1}^n$*, if the product Riemannian metric on the product manifold* (M, g)*, where* $M = M_1 \times \cdots \times M_n$*, is defined as*

$$g_p(X, Y) = g_1(X_1, Y_1) + \cdots + g_n(X_n, Y_n)$$

for $p = (p_1, \ldots, p_n) \in M$ *and* $X = (X_1, \ldots, X_n)$, $Y \in T_pM = T_{p_1}M_1 \times \ldots T_{p_n}M_n$*, then*

$$grad\, f(p) = (grad\, f(p_1), \ldots, grad\, f(p_n)).$$

We next consider what geodesics look like on the product manifold.

Theorem B.2. *If M is a product Riemannian manifold as defined in (B–6), then a geodesic between $p = (p_1, p_2)$ and $q = (q_1, q_2) \in M$ can be obtained by the product of geodesics between p_1 and q_1, and p_2 and q_2, on M_1 and M_2, respectively.*

Proof. Let $\gamma_{pq}, \gamma_{p_1q_1}, \gamma_{p_2q_2}$ denote the geodesics on M, M_1, and M_2 respectively. By definition, $\gamma_{pq} =: (\gamma^{(1)}, \gamma^{(2)})$ is the path that minimizes

$$\gamma_{pq} = \arg\min_{\gamma \in \Gamma_{pq}} \int_0^1 \|\gamma'(t)\|^2 dt$$

where Γ_{pq} denotes the set of all paths on M from p to q and, for $\gamma \in \Gamma_{pg}$, $\gamma(t)$ denotes a parameterization of path γ such that $\gamma(0) = p$ and $\gamma(1) = q$. Using the definition of the inner-product norm on the product manifold, we see that

$$\begin{aligned}
\gamma_{pq} &= \arg\min_{(\gamma^{(1)}, \gamma^{(2)}) \in \Gamma_{pq}} \int_0^1 (\|\gamma^{(1)\prime}(t)\|^2 + \|\gamma^{(2)\prime}(t)\|^2) dt \\
&= \left(\arg\min_{\gamma \in \Gamma_{p_1q_1}} \int_0^1 \|\gamma'(t)\|^2 dt,\ \arg\min_{\gamma \in \Gamma_{p_2q_2}} \int_0^1 \|\gamma'(t)\|^2 dt \right) \\
&= (\gamma_{p_1q_1}, \gamma_{p_2q_2}).
\end{aligned}$$

□

That parallel transport on M is given by parallel transport on each individual M_i immediately follows. Thus proving (3–5).

B.3 Proof of Theorem 3.1

proof. The cost function (3–3) is a function from $\mathcal{M}$ to $\mathbb{R}$ where $\mathcal{M}$ is the product manifold

$$\mathcal{M} := \left(SO(3) \times \mathbb{R}^3\right)^n, \text{ where } n := |\mathcal{V}(k)|.$$

We define the Riemannian metric on this product manifold as follows. For $p = (\mathbf{R}_1, \mathbf{t}_1, \ldots, \mathbf{R}_n, \mathbf{t}_n) \in \mathcal{M}$ and $X, Y \in T_p\mathcal{M}$, where $X \in T_p\mathcal{M}$ is expanded as

$(X_{\mathrm{R}_1}, X_{\mathrm{t}_1}, \ldots, X_{\mathrm{R}_n}, X_{\mathrm{t}_n})$,

$$g_p(X, Y) := \sum_{i \in n} \left(g_{\mathrm{R}_i}(X_{\mathrm{R}_i}, Y_{\mathrm{R}_i}) + < X_{\mathrm{t}_i}, Y_{\mathrm{t}_i} > \right)$$

This meets the conditions of Corollary 1 and so for $p = (\mathbf{R}_1, \mathbf{t}_1, \ldots, \mathbf{R}_n, \mathbf{t}_n) \in \mathcal{M}$, we have

$$grad\, f(p) = (grad\, f(\mathbf{R}_1), grad\, f(\mathbf{t}_1), \ldots, grad\, f(\mathbf{R}_h), grad\, f(\mathbf{t}_h)) .$$

All that remains is to determine $grad\, f(\mathbf{R}_u)$ and $grad\, f(\mathbf{t}_u)$ for $u = 1, \ldots, n$. Using linearity of the gradient operator, we have

$$grad\ f(\mathbf{R}_u) = \sum_{e \in \mathcal{E}(k)} grad\ c_e(\mathbf{R}_u), \quad \text{and } grad\ f(\mathbf{t}_u) = \sum_{e \in \mathcal{E}(k)} grad\ c_e(\mathbf{t}_u).$$

First define $f_2 := \frac{1}{2} d^2(\mathbf{R}_u^T \mathbf{R}_v, \hat{\mathbf{R}}_{uv})$, $f_3 := \frac{1}{2} \|\hat{\mathbf{t}}_{uv} - \mathbf{R}_u^T(\mathbf{t}_v - \mathbf{t}_u)\|^2$, so that for $e \triangleright e(u, v)$, $c_e = f_2 + f_3$. For $h = u$ or $h = v$, we therefore have

$$grad\ \ c_e(\mathbf{R}_h) = grad\ \ f_2(\mathbf{R}_h) + grad\ \ f_3(\mathbf{R}_h) \tag{B–8}$$

$$grad\ \ c_e(\mathbf{t}_h) = 0 + grad\ \ f_3(\mathbf{t}_h). \tag{B–9}$$

The Riemannian distance $d(p, q)$ is bi-invariant, meaning that for arbitrary $R, \bar{R} \in SO(3)$, $d(p, q) = d(Rp\bar{R}, Rq\bar{R})$. Using bi-invariance, we obtain $d^2(\mathbf{R}_u^T \mathbf{R}_v, \hat{\mathbf{R}}_{uv}) = d^2(\mathbf{R}_v, \mathbf{R}_u \hat{\mathbf{R}}_{uv})$, which implies $grad\ \ f_2(\mathbf{R}_v) = grad\ \ f_1(\mathbf{R}_v | q)$ where $f_1 : SO(3) \to \mathbb{R}$ is given by (B–3) and $q = \mathbf{R}_u \hat{\mathbf{R}}_{uv}$. Applying formula (B–4), we obtain

$$grad\ \ f_2(\mathbf{R}_v) = -\mathbf{R}_v \log(\mathbf{R}_v^T \mathbf{R}_u \hat{\mathbf{R}}_{uv}).$$

The expression for $grad\ f_2(\mathbf{R}_u)$ is similarly obtained; by thinking of f_2 as a function of $\mathbf{R}_u$ and keeping $\mathbf{R}_v, \hat{\mathbf{R}}_{uv}$ fixed. These gradients are compactly represented as:

$$grad\, f_2(\mathbf{R}_h) = \begin{cases} -\mathbf{R}_h \log\left(\mathbf{R}_h^T \mathbf{R}_v \hat{\mathbf{R}}_{uv}^T\right) & \text{if } h = u \\ -\mathbf{R}_h \log\left(\mathbf{R}_h^T \mathbf{R}_u \hat{\mathbf{R}}_{uv}\right) & \text{if } h = v \\ 0 & \text{otherwise} \end{cases}$$

The gradient of the function $f_3 : SO(3) \to \mathbb{R}$ (thinking of f_3 only as a function of $\mathbf{R}_u$ while $\mathbf{t}_u, \mathbf{t}_v, \hat{\mathbf{t}}_{uv}$ are fixed) can be obtained using (B–5) with straightfrward but tedious calculations; which turn out to be $grad\, f_3(\mathbf{R}_u) = -(\mathbf{t}_v - \mathbf{t}_u)\hat{\mathbf{t}}_{uv}^T + \mathbf{R}_u \hat{\mathbf{t}}_{uv}(\mathbf{t}_v - \mathbf{t}_u)^T \mathbf{R}_u$. The gradient of $f_3 : \mathbb{R}^3 \to \mathbb{R}$ is simply $(\frac{\partial f_3(x)}{\partial x})^T$, which is

$$grad\, f_3(\mathbf{t}_h) = \begin{cases} \left(\mathbf{t}_h + \mathbf{R}_h \hat{\mathbf{t}}_{uv} - \mathbf{t}_v\right) & \text{if } h = u \\ \left(\mathbf{t}_h - \mathbf{R}_u \hat{\mathbf{t}}_{uv} - \mathbf{t}_u\right) & \text{if } h = v \\ 0 & \text{o.w.} \end{cases}$$

These formulae completely specify the gradient of the edge cost, $grad\ c_e(\cdot)$ in (B–8). The expressions for $grad\ f(\mathbf{R}_u)$ and $grad\ f(\mathbf{t}_u)$ for $u = 1, \ldots, n$ that are provided in the theorem are obtained simply by adding the components of the gradients that are derived above. □

B.4 Gradient of the Cost Function (5–9) for Heterogeneous Measurements

As in the proof of Theorem 3.1, the gradient of the function f in (5–9) at a point $p = (\mathbf{R}_1, \mathbf{t}_1, \ldots, \mathbf{R}_n, \mathbf{t}_n) \in \left(SO(3) \times \mathbb{R}^3\right)^n$ is given by

$$grad\, f(p) = \sum_{e \in \mathcal{E}(k)} grad\, c_e(p) =: (grad\, f(\mathbf{R}_1), \ldots, grad\, f(\mathbf{t}_n)) \tag{B–10}$$

where $grad\, g_e(p)$ is the gradient of the edge cost function for edge $e = (u, n)$. Finding the gradient of the cost function (5–9) then reduces to finding the gradients of the edge costs c_e (specified in (5–10)) for each edge $e \in \mathcal{E}(k)$.

It follows from Corollary 1 that the gradient of the cost function c_e in (5–10) for the edge $e = (u, v) \in \mathcal{E}(k)$ is

$$grad\ c_e(p) = \Big(grad\ c_e(\mathbf{R}_1), grad\ c_e(\mathbf{t}_1), \dots, grad\ c_e(\mathbf{R}_n), grad\ c_e(\mathbf{t}_n)\Big)$$

For $\ell(k)(e) = \mathbf{T}$ (pose), the gradients have already been computed in the previous section, which can be compactly represented as $(e \triangleright (u, v))$

$$grad\ c_e(\mathbf{R}_h) = \begin{cases} -2\mathbf{R}_h\Big(\log(\mathbf{R}_h^T\mathbf{R}_v\hat{\mathbf{R}}_{uv}^T) \\ \quad +\mathbf{R}_h^T(\mathbf{t}_v - \mathbf{t}_u)\hat{\mathbf{t}}_{uv}^T - \hat{\mathbf{t}}_{uv}(\mathbf{t}_v - \mathbf{t}_u)^T\mathbf{R}_h\Big) & \text{if } h = u \\ -2\mathbf{R}_h\log(\mathbf{R}_h^T\mathbf{R}_u\hat{\mathbf{R}}_{uv}) & \text{if } h = v \\ 0 & \text{o.w.} \end{cases}$$

$$grad\ c_e(\mathbf{t}_h) = 2l_{uv}(h)(\mathbf{t}_u + \mathbf{R}_u\hat{\mathbf{t}}_{uv} - \mathbf{t}_j)$$

where $l_{uv}(h) = 1$ if $h = u$, -1 if $h = v$ and 0 otherwise. If $\ell(k)(e) = \mathbf{R}$ (orientation) or $\ell(k)(e) = \mathbf{t}$ (position), we have the following expressions for the gradient from the previous section. If $\ell(k)(e) = \mathbf{R}$,

$$grad\ c_e(\mathbf{R}_h) = \begin{cases} -2\mathbf{R}_h\Big(\log(\mathbf{R}_h^T\mathbf{R}_v\hat{\mathbf{R}}_{uv}^T) & \text{if } h = u \\ -2\mathbf{R}_h\Big(\log(\mathbf{R}_h^T\mathbf{R}_u\hat{\mathbf{R}}_{uv}) & \text{if } h = v \\ 0 & \text{otherwise} \end{cases}$$

$$grad\ c_e(\mathbf{t}_h) = 0.$$

and if $\ell(k)(e) = \mathbf{t}$, then

$$grad\ c_e(\mathbf{R}_h) = \begin{cases} -2\mathbf{R}_h\Big(\mathbf{R}_h^T(\mathbf{t}_v - \mathbf{t}_u)\hat{\mathbf{t}}_{uv}^T - \hat{\mathbf{t}}_{uv}(\mathbf{t}_v - \mathbf{t}_u)^T\mathbf{R}_h\Big) & \text{if } h = u \\ 0 & \text{otherwise} \end{cases}$$

$$grad\ c_e(\mathbf{t}_h) = 2l_{uv}(h)(\mathbf{t}_u + \mathbf{R}_u\hat{\mathbf{t}}_{uv} - \mathbf{t}_v).$$

If $\ell(k)(e) = \boldsymbol{\tau}$ (bearing) or $\ell(k)(e) = \delta$ (distance), the gradient $grad\ \ c_e(\mathbf{R}_h)$ can be computed by using formula (B–5). The gradient $grad\ \ c_e(\mathbf{t}_h)$ is obtained by differentiation. We obtain, for $\ell(k)(e) = \boldsymbol{\tau}$ (bearing)

$$grad\ \ c_e(\mathbf{R}_h) = \begin{cases} -2\mathbf{R}_h\Big(\mathbf{R}_h^T(\mathbf{t}_v - \mathbf{t}_h)\hat{\boldsymbol{\tau}}_{uv}^T\|\mathbf{t}_u - \mathbf{t}_v\| - \hat{\boldsymbol{\tau}}_{uv}\|\mathbf{t}_v - \mathbf{t}_u\|(\mathbf{t}_v - \mathbf{t}_u)^T\mathbf{R}_h\Big) & \text{if } h = u \\ 0 & \text{otherwise} \end{cases}$$

$$grad\ \ c_e(\mathbf{t}_h) = -4l_{uv}(h)[(\mathbf{t}_v - \mathbf{t}_u) - \|\mathbf{t}_v - \mathbf{t}_u\|\mathbf{R}_u\hat{\boldsymbol{\tau}}_{uv}]$$

and if $\ell(k)(e) = \delta$ (dist)

$$grad\ c_e(\mathbf{R}_h) = 0$$

$$grad\ c_e(\mathbf{t}_h) = -2l_{uv}(h)\frac{(\hat{\delta}_{uv} - \|\mathbf{t}_v - \mathbf{t}_u\|)}{\|\mathbf{t}_v - \mathbf{t}_u\|}(\mathbf{t}_v - \mathbf{t}_u).$$

The gradient $grad\ \ f(p)$ can now be computed by using these formulas.

REFERENCES

[1] C. F. Olson, L. H. Matthies, M. Schoppers, and M. W. Maimone, “Rover navigation using stereo ego-motion,” Robotics and Autonomous Systems, vol. 43, no. 4, pp. 215–229, June 2003.

[2] H. Badino, “A robust approach for ego-motion estimation using a mobile stereo platform,” in International Workshop on Complex Motion, 2004, pp. 198–208.

[3] M. Klopschitz and D. Schmalstieg, “Automatic reconstruction of wide-area fiducial marker models,” in International Symposium on Mixed and Augmented Reality, ser. ISMAR ’07, Washington, DC, USA, 2007.

[4] B. Stenning and T. Barfoot, “Path planning on a network of paths,” in Aerospace Conference, 2011 IEEE, march 2011, pp. 1 –12.

[5] S. Se, H.-K. Ng, P. Jasiobedzki, and T.-J. Moyung, “Vision based modeling and localization for planetary exploration rovers,” in 55th International Astronautical Congress, 2004.

[6] A. Lambert, P. Furgale, T. Barfoot, and J. Enright, “Visual odometry aided by a Sun sensor and inclinometer,” in Aerospace Conference, 2011 IEEE, march 2011, pp. 1 –14.

[7] D. Nistér, O. Naroditsky, and J. Bergen, “Visual odometry for ground vehicle applications,” Journal of Field Robotics, vol. 23, p. 2006, 2006.

[8] R. Jiang, R. Klette, and S. Wang, “Modeling of unbounded long-range drift in visual odometry,” in Fourth Pacific-Rim Symposium on Image and Video Technology, November 2010, pp. 121–126.

[9] G. Dubbelman and F. Groen, “Bias reduction for stereo based motion estimation with applications to large scale visual odometry,” in International conference on Computer Vision and Pattern Recognition, June 2009, pp. 1–8.

[10] G. Sibley, L. Matthies, and G. Sukhatme, “Bias reduction and filter convergence for long range stereo,” in Robotics Research, ser. Springer Tracts in Advanced Robotics, S. Thrun, R. Brooks, and H. Durrant-Whyte, Eds. Springer Berlin / Heidelberg, 2007, vol. 28, pp. 285–294.

[11] G. Grisetti, S. Grzonka, C. Stachniss, P. Pfaff, and W. Burgard, “Efficient Estimation of Accurate Maximum Likelihood Maps in 3D,” Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3472–3478, 2007.

[12] S. Roumeliotis and G. Bekey, “Distributed multirobot localization,” IEEE Transactions on Robotics and Automation, vol. 18, no. 5, pp. 781 – 795, oct 2002.

[13] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, “A probabilistic approach to collaborative multi-robot localization,” Autonomous robots, vol. 8, no. 3, pp. 325–344, 2000.

[14] A. Howard, M. Matark, and G. Sukhatme, “Localization for mobile robot teams using maximum likelihood estimation,” in IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, Switzerland, September 2002.

[15] R. Aragues, L. Carlone, G. Calafiore, and C. Sagues, “Multi-agent localization from noisy relative pose measurements,” in IEEE International Conference on Robotics and Automation, Shanghai, China, Mar 2011.

[16] I. Rekleitis, G. Dudek, and E. Milios, “Multi-robot cooperative localization: a study of trade-offs between efficiency and accuracy,” in IEEE/RSJ International Conference on Intelligent Robots and Systems, EPFL, Switzerland, September 2002.

[17] A. Martinelli, F. Pont, and R. Siegwart, “Multi-robot localization using relative observations,” in IEEE International Conference on Robotics and Automation, Barcelona, Spain, April 2005.

[18] R. C. Smith and P. Cheeseman, “On the representation and estimation of spatial uncertainty,” International Journal of Robotics Research, vol. 5, no. 4, pp. 56–68, 1986.

[19] S. Su and C. S. G. Lee, “Manipulation and propagation of uncertainty and verification of applicability of actions in assembly tasks,” IEEE Transactions on Systems, Man, and Cybernetics, vol. 22, no. 6, pp. 1376–1389, 1982.

[20] Y. Wang and G. S. Chirikjian, “Error propagation on the euclidean group with applications to manipulator kinematics,” IEEE Transactions on Robotics, vol. VOL. 22, no. 4, pp. 1–12, Jul 2006.

[21] Y. Wang and G. Chirikjian, “Nonparametric-second order theory of error propagation on motion groups,” International Journal of Robotics Research, vol. 27, no. 11-12, pp. 1258–1273, 2008.

[22] Y. Zhou and G. S. Chirikjian, “Probabilistic models of dead-reckoning error in nonholonomnic mobile robots,” in ICRA, 2003, pp. 1594–1599.

[23] P. Smith, T. Drummond, and K. Roussopoulos, “Computing MAP trajectories by representing, propagating and combining pdfs over groups,” in IEEE International Conference on Computer Vision, vol. 2, 2003, pp. 1275–1282.

[24] J. Kwon, M. Choi, F. Park, and C. Chun, “Particle filtering on the euclidean group: framework and applications,” Robotica, vol. 25, pp. 725–737, 2007.

[25] K. Ni, D. Steedly, and F. Dellaert, “Tectonic SAM: Exact, out-of-core, submap-based SLAM,” in IEEE International Conference on Robotics and Automation, Roma, Italy, May 2007.

[26] L. A. A. Andersson and J. Nygards, “C-SAM : Multi-robot SLAM using square root information smoothing,” in IEEE International Conference on Robotics and Automation, Pasadena, California, USA, May 2008.

[27] B. Kim, M. Kaess, L. Fletcher, J. Leonard, A. Bachrach, N. Roy, and S. Teller, “Multiple relative pose graphs for robust cooperative mapping,” in IEEE International Conference on Robotics and Automation, Anchorage, Alaska, May 2010.

[28] V. Indelman, P. Gurfil, E. Rivlin, and H. Rotstein, “Distributed vision-aided cooperative localization and navigation based on three-view geometry,” in IEEE Aerospace Conference, Big Sky, Montana, USA, March 2011.

[29] R. Kurazume, S. Nagata, and S. Hirose, “Cooperative positioning with multiple robots,” in IEEE International Conference on Robotics and Automation, San Diego, CA, May 1994.

[30] C. Sanderson, “A distributed algorithm for cooperative navigation among multiple mobile robots,” Advanced Robotics, vol. 12, no. 4, pp. 335 – 349, 1998.

[31] V. Caglioti, A. Citterio, and A. Fossati, “Cooperative, distributed localization in multi-robot systems: a minimum-entropy approach,” in IEEE Workshop on Distributed Intelligent Systems, Czech Republic, June 2006.

[32] R. Sharma and C. Taylor, “Cooperative navigation of mavs in gps denied areas,” in IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, Seoul, Korea, Aug. 2008.

[33] N. Karam, F. Chausse, R. Aufrere, and R. Chapuis, “Localization of a group of communicating vehicles by state exchange,” in IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, October 2006.

[34] J. Knuth and P. Barooah, “Distributed collaborative localization of multiple vehicles from relative pose measurements,” in 47th Annual Allerton Conference on Communication, Control, and Computing, Allerton, IL, October 2009, pp. 314 –321.

[35] K. Y. Leung, T. Barfoot, and H. Liu, “Decentralized localization of sparsely-communicating robot networks: A centralized-equivalent approach,” IEEE Transactions on Robotics, vol. 26, no. 1, pp. 62 –77, 2010.

[36] E. Olson, M. Walter, J. Leonard, and S. Teller, “Single cluster graph partitioning for robotics applications,” in Proceedings of Robotics Science and Systems, 2005, pp. 265–272.

[37] J. McDonald, M. Kaess, C. Cadena, J. Neira, and J. Leonard, “6-DOF multi-session visual SLAM using anchor nodes,” in European Conference on Mobile Robots, ECMR, Orebro, Sweden, Sep 2011, pp. 69–76.

[38] K. Konolige, J. Bowman, J. D. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua, “View-based maps,” in Proceedings of Robotics: Science and Systems, Seattle, USA, June 2009.

[39] C. D. Cadena Lerma, J. McDonald, J. J. Leonard, and J. Neira, “Place recognition using near and far visual information,” in World Congress, vol. 18, no. 1, 2011, pp. 6822–6828.

[40] M. Cummins and P. Newman, “Probabilistic localization and mapping in the space of appearance,” The International Journal of Robotics Research, vol. 27, no. 6, pp. 647–665, 2008.

[41] N. Sunderhauf and P. Protzel, “Towards a robust back-end for pose graph slam,” in Robotics and Automation (ICRA), 2012 IEEE International Conference on, May 2012, pp. 1254–1261.

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

[43] G. Bleser and D. Strickera, “Advanced tracking through efficient image processing and visual-inertial sensor fusion,” Computers & Graphics, vol. 33, no. 1, pp. 59–72, 2009.

[44] B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, and J. Tardós, “A comparison of loop closing techniques in monocular SLAM,” Robotics and Autonomous Systems, vol. 57, pp. 1188–1197, December 2009.

[45] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, An invitation to 3-D vision, ser. Interdisciplinary applied mathematics. Springer, 2006.

[46] K. Kanatani, “Statistical optimization and geometric inference in computer vision [and discussion],” Philosophical Transactions: Mathematical, Physical and Engineering Sciences, vol. 356, no. 1740, pp. 1303–1320, May 1998, (with R. Cipolla, R. I. Hartley, W. Triggs, A. Fitzgibbon, O. Faugeras in discussions).

[47] N. I. Fisher, T. Lewis, and B. J. J. Embleton, Statistical Analysis of Spherical Data. Cambridge University Press, 1997.

[48] M. Fischler and L. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” Communications of the ACM, vol. 24, pp. 381–385, 1981.

[49] R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision. Cambridge University Press, 2004.

[50] C. Tomasi and T. Kanade, “Detection and tracking of point features,” Carnegie Mellon University, Tech. Rep., April 1991.

[51] P. Barooah, W. J. Russell, and J. P. Hespanha, “Approximate distributed kalman filtering for cooperative multi-agent localization,” in International conference in Distributed Computing in Sensor Systems (DCOSS), Santa Barbara, CA, June 2010.

[52] R. Tron and R. Vidal, “Distributed image-based 3-d localization of camera sensor networks,” in IEEE Conference on Decision and Control, Shanghai, China, December 2009.

[53] A. Yershova, S. Jain, S. LaValle, and J. Mitchell, “Generating uniform incremental grids on SO(3) using the hopf fibration,” The International Journal of Robotics Research, vol. 29, no. 7, 2010.

[54] P. Absil, R. Mahony, and R. Sepulchre, Optimization Algorithms on Matrix Manifolds. Princeton, NJ: Princeton University Press, 2008.

[55] K. V. Mardia and P. E. Jupp, Directional Statistics, ser. Wiley series in probability and statistics. Wiley, 2000.

[56] J. Knuth and P. Barooah, “Error growth in position estimation from noisy relative pose measurements,” Robotics and Autonomous Systems, vol. 61, no. 3, pp. 229–244, March 2013.

[57] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, “$g^2$0: a general framework for graph optimization,” in IEEE International Conference on Robotics and Automation, 2011, pp. 3607–3613.

[58] I. V. Melnyk, J. A. Hesch, and S. I. Roumeliotis, “Cooperative Vision-aided Inertial Navigation Using Overlapping Views,” IEEE International conference on Robotics and Automation, pp. 1–8, 2012.

[59] W. G. Breckenridge, “Quaternions-proposed standard conventions,” JPL, INTEROFFICE MEMORANDUM IOM 343-79-1199, Tech. Rep., 1999.

[60] P. T. Fletcher, S. Joshi, C. Lu, and S. Pizer, “Gaussian Distributions on Lie Groups and Their Application to Statistical Shape Analysis,” in Information Processing in Medical Imaging, May 2003, pp. 450–462.

[61] J. Kent, “Some probabilistic properties of bessel functions,” The Annals of Probability, vol. 6, no. 5, pp. pp. 760–770, 1978.

[62] S. Gallot, D. Hulin, and J. LaFontaine, Riemannian Geometry, 3rd ed. Springer, 2004.

[63] D. B. Johnson, “Finding all the elementary circuits of a directed graph,” SIAM Journal on Computing, vol. 4, no. 1, pp. 77–84, 1975.

[64] F. E. Grubbs, “Procedures for detecting outlying observations in samples,” Technometrics, vol. 11, no. 1, pp. 1–21, 1969.

[65] G. Sibley, G. S. Sukhatme, and L. Matthies, “Constant time sliding window filter slam as a basis for metric visual perception,” in Proc. IEEE International Conference on Robotics and Automation Workshop, 2007, pp. 10–14.

[66] G. Sibley, L. Matthies, and G. Sukhatme, “A sliding window filter for incremental slam,” Unifying perspectives in computational and robot vision, pp. 103–112, 2008.

[67] S. Lim, T. kyeong Lee, S. Lee, S. An, and S. young Oh, “Adaptive sliding window for hierarchical pose-graph-based slam,” in Control, Automation and Systems (ICCAS), 2012 12th International Conference on, Oct. 2012, pp. 2153–2158.

[68] Y. Ma, J. Košecká, and S. Sastry, “Optimization criteria and geometric algorithms for motion and structure estimation,” International Journal of Computer Vision, vol. 44, no. 3, pp. 219–249, 2001.

[69] J. Knuth and P. Barooah, “Collaborative localization with heterogeneous inter-robot measurements by Riemannian optimization,” in IEEE International Conference on Robotics and Automation, May 2013.

[70] K. Grove, H. Karcher, and E. A. Ruh, “Jacobi fields and finsler metrics on compact lie groups with an application to differentiable pinching problems.” Mathematische Annalen, vol. 211, pp. 7–21, 1974.

BIOGRAPHICAL SKETCH

Joseph Knuth received his Bachelor of Science degree in Computer Engineering from the University of Illinois at Urbana-Champaign in 2004. In 2008 Joseph joined the Distributed Control and Estimation Lab in the Mechanical and Aerospace Engineering Department at the University of Florida, Gainesville. He received a Master of Science in 2010 and is currently pursuing his doctoral degree under the advisement of Dr. Prabir Barooah.