Cooperative Navigation for Low-bandwidth Mobile Acoustic Networks

by

Jeffrey M. Walls

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Mechanical Engineering)
in the University of Michigan
2015

Doctoral Committee:

Associate Professor Ryan M. Eustice, Chair
Assistant Professor Kira L. Barton
Professor Jessy W. Grizzle
Professor Louis L. Whitcomb, Johns Hopkins University

Acknowledgments

Instead of ‘Acknowledgments’, this section heading should really read ‘Coauthors’. I would not have written a single page if not for the enormous amount of help and support I have received from so many people (more than can be listed below). Knowing that the following cannot completely suffice, I’ll begin with a big ‘thank you’ to all.

First, Ryan, thank you for taking me on as a student six years ago. Thank you for trusting me to throw your AUVs over the side of a boat, and trusting that they would be returned. I’m glad that research has always been one part what we can show on paper, one part what we can demonstrate on real hardware. Hopefully, years from now, you’ll still think that time Paul sunburned his hands was pretty funny.

Next, to Professor Barton, Professor Grizzle, and Professor Whitcomb, thank you for serving on my committee. In particular, Louis, you provided an example of scholarship over the last five years that I have tried to emulate. Moreover, for what it counts coming from me, you do some really cool work in some really cool places.

To all PeRL members and affiliates, in order of appearance, Ayoung, Gaurav, Nick (glad you have a new desk), Paul (of sunburned hand fame), Sarah, Sweets (Chaves), Wolcott, Schuyler, Vittorio, Sweets (Ushani), Jie, GT, Alex, Enric, Katie, Sparkison, and Josh, thank you. 90% of the theory presented in this document was developed on the lab whiteboard with your help. 100% of the field results were collected with your help. I enjoyed every minute of it.

Ann Arbor friends, particularly the first few folks I met, Ryan, Ellen, Omar, and Sills, thank you for loads of laughs and heaps of great memories. All friends, past and present, get a job!

And, most importantly, to all my family. Ma, Pops, and Matt, let’s not get sentimental, but I would, of course, not be who I am without you, so thank you. Thank you for the constant encouragement. To my partner in life (and crime?), Alix, thank you for always keeping me laughing. Thank you for putting up with late nights and endless deadlines. Thank you for a wonderful life and adventures to come.

In the words of Maurice Sendak:

> “And now,” cried Max, “let the wild rumpus start!”

This work was funded in part by the National Science Foundation under awards IIS-0746455 and ANT-1039951, the Naval Sea Systems Command through the Naval Engineering Education Center under award N65540-10-C-0003, and the Office of Naval Research under award N00014-12-1-0092.

Contents

Chapter

List of Figures

List of Tables

List of Appendices

List of Acronyms

AHRS	attitude heading reference system
AUV	autonomous underwater vehicle
C-SAM	collaborative smoothing and mapping
CCL	compact control language
CDMA	code division multiple access
CEKF	centralized extended Kalman filter
CEIF	centralized extended information filter
CI	covariance intersection
CNA	cooperative navigation aid
DCCL	dynamic compact control language
DDF	decentralized data fusion
DDF-SAM	decentralized data fusion smoothing and mapping
DDP	differential dynamic programming
DEIF	decentralized extended information filter
DVL	Doppler velocity log
EIF	extended information filter
EKF	extended Kalman filter
FDMA	frequency-division multiple access
FSK	frequency-shift keying
GLC	generic linear constraint
GPS	global positioning system

iSAM	incremental smoothing and mapping
IU	interleaved update
KF	Kalman filter
KLD	Kullback-Leibler divergence
LBL	long-baseline
LQG	linear quadratic Gaussian
MAC	medium access control
MAP	maximum a posteriori
MDP	Markov decision process
MLE	maximum likelihood estimate
MMSE	minimum mean squared error
MRF	Markov random field
OSM	origin state method
OSP	origin state packet
OWTT	one-way-travel-time
PeRL	Perceptual Robotics Laboratory
POMDP	partially observable Markov decision process
PPS	pulse per second
PRM	probabilistic road map
PSK	phase-shift keying
RF	radio frequency
RRG	rapidly-exploring random graph
RRT	rapidly-exploring random tree
SAM	smoothing and mapping
SBL	short-baseline
SEIF	sparse extended information filter
SfM	structure-from-motion

SLAM	simultaneous localization and mapping
SOI	sign-of-innovations
TDMA	time division multiple access
TOA	time-of-arrival
TOF	time-of-flight
TOL	time-of-launch
TWTT	two-way travel-time
TXCO	temperature compensated crystal oscillator
UKF	unscented Kalman filter
UMBS	University of Michigan Biological Station
USBL	ultra-short-baseline
WHOI	Woods Hole Oceanographic Institution

Abstract

This thesis reports on the design and validation of estimation and planning algorithms for underwater vehicle cooperative localization. While attitude and depth are easily instrumented with bounded-error, autonomous underwater vehicles (AUVs) have no internal sensor that directly observes XY position. The global positioning system (GPS) and other radio-based navigation techniques are not available because of the strong attenuation of electromagnetic signals in seawater. The navigation algorithms presented herein fuse local body-frame rate and attitude measurements with range observations between vehicles within a decentralized architecture.

The acoustic communication channel is both unreliable and low bandwidth, precluding many state-of-the-art terrestrial cooperative navigation algorithms. We exploit the underlying structure of a post-process centralized estimator in order to derive two real-time decentralized estimation frameworks. First, the origin state method enables a client vehicle to exactly reproduce the corresponding centralized estimate within a server-to-client vehicle network. Second, a graph-based navigation framework produces an approximate reconstruction of the centralized estimate onboard each vehicle. Finally, we present a method to plan a locally optimal server path to localize a client vehicle along a desired nominal trajectory. The planning algorithm introduces a probabilistic channel model into prior Gaussian belief space planning frameworks.

In summary, cooperative localization reduces XY position error growth within underwater vehicle networks. Moreover, these methods remove the reliance on static beacon networks, which do not scale to large vehicle networks and limit the range of operations. Each proposed localization algorithm was validated in full-scale AUV field trials. The planning framework was evaluated through numerical simulation.

Chapter 1

Introduction

1.1 Motivation

Autonomous underwater vehicles (AUVs) have equipped oceanographers with the ability to safely and efficiently explore and research a variety of difficult environments (Bellingham and Rajan, 2007) including the under-ice Arctic Ocean (Jakuba et al., 2008; Kunz et al., 2009), the deepest regions of the Mariana trench (Bowen et al., 2009; Whitcomb et al., 2010), and the Great Barrier Reef (Williams and Mahon, 2004). AUVs require robust navigation algorithms to safely operate in such hostile environments and geo-reference the valuable scientific data that they collect.

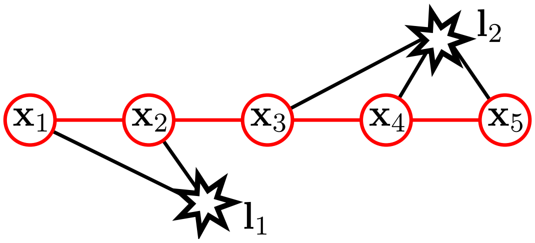
Water is opaque to electromagnetic signals, prohibiting the direct use of the global positioning system (GPS) and other radio frequency based navigation techniques while submerged. Terrain-aided navigation utilizes prior bathymetric terrain maps for localization, but is not applicable to operations in the mid-depth zone where rich altitude information is not available. Visually-aided navigation, commonly used in terrestrial and aerial robotics, is only possible when an AUV is close to the seafloor due to backscatter, strong attenuation, and a generally unstructured scene, which challenge state-of-the-art computer vision algorithms. Underwater vehicles typically employ acoustic beacon networks, such as narrowband long-baseline (LBL) and ultra-short-baseline (USBL), to obtain accurate bounded-error navigation in the mid-depth zone where there is no apparent visual information. Acoustic beacon methods, however, generally require additional infrastructure and limit vehicle operations to the acoustic footprint of the beacons.

Acoustic modems enable vehicles to both share data and observe their relative range. However, the underwater acoustic channel is unreliable, exhibits low-bandwidth, and suffers from high latency (sound is orders of magnitude slower than light). Despite these challenges, underwater navigation can benefit from cooperative localization—each vehicle is treated as a mobile acoustic navigation beacon, which requires no additional infrastructure and is not limited in the range of operations by static beacons.

This thesis reports on novel estimation and control algorithms for synchronous-clock cooperative underwater localization. The contributions of this work include: an exact one-to-many (server-to-client) localization algorithm, an approximate many-to-many (peer-to-peer) cooperative localization algorithm, and a planning framework for establishing more informative geometries for range-only server-client localization. These algorithms are designed to be tolerant to the extremely faulty and bandwidth-limited acoustic communication channel. Moreover, they are not limited to AUVs and can be applied to any network of platforms (terrestrial or aerial, manned or unmanned) in order to share navigation information with limited communication bandwidth. The work presented in this document builds upon the existing state-of-the-art in related work stemming from single-robot navigation, terrestrial cooperative localization, underwater navigation, and planning under uncertainty. The most salient features of the prior literature are discussed below.

1.2 A Review of Single Robot Navigation

Figure 1.1 Illustration of the full SLAM problem. A SLAM solution computes the unknown robot and landmark poses represented by circles and stars, respectively, given noisy observations illustrated by edges.



One of the fundamental challenges in robotics lies in answering the question ‘where am I?’. The problem of concurrently building a model of an *a priori* unknown environment and localizing within it has been addressed by robotics researchers over the last several decades by the development of simultaneous localization and mapping (SLAM) frameworks. Since we are interested in underwater localization, we focus on metric SLAM formulations. This class of algorithms computes a joint estimate over robot poses and a map—generally a collection of landmarks or features. The SLAM solution is often partitioned into two sub problems: inference (i.e., optimization) and data association (i.e., recognizing landmarks), known as the back-end and front-end problems, respectively. Here, we only review the back-end SLAM problem as it relates to cooperative localization and assume that the identities of robots and landmarks are known.

In general, the full metric SLAM solution is provided by the maximum *a posteriori* (MAP) estimate over a set of poses $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ and landmarks $\mathbf{L} = \{\mathbf{l}_1, \ldots, \mathbf{l}_m\}$ given the set of control inputs $\mathbf{U} = \{\mathbf{u}_1, \ldots, \mathbf{u}_{n-1}\}$ and observations $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_p\}$

$$\hat{\mathbf{X}}, \hat{\mathbf{L}} = \underset{\mathbf{X},\mathbf{L}}{\operatorname{argmax}}\, p(\mathbf{X}, \mathbf{L} | \mathbf{U}, \mathbf{Z}), \tag{1.1}$$

as illustrated in Fig. 1.1. In the following sections, we review various solutions to the general problem.

1.2.1 Filtering Approaches

The pioneering SLAM formulation by Smith et al. (1986) began to address the problem of fusing uncertain spatial information from sensors of varying quality for robot navigation. Their solution, among others including Leonard and Durrant-Whyte (1991), tracks the so-called stochastic map, a map composed of various landmarks such as corners, edges, or other fiducial markers scattered throughout the environment. The robot adds landmarks to the map relative to its current pose as it travels. Subsequent re-observation of the initialized landmarks provides a constraint on the robot's pose as a *loop closure* event. The power of SLAM algorithms rests in the ability to identify and incorporate these events, thereby strengthening map relationships.

Early SLAM solutions recursively track a state vector composed of the current robot pose and landmark positions, $\boldsymbol{x} = [\mathbf{x}_n^\top, \mathbf{l}_1^\top, \ldots, \mathbf{l}_m^\top]^\top$, within a Bayesian estimation framework, e.g., the extended Kalman filter (EKF). The filtering solution corresponds to the Bayes filter formulation of (1.1). Given linear Gaussian noise models, the Kalman filter provides the optimal minimum mean squared error (MMSE) estimate. System nonlinearities are dealt with in ad hoc estimators such as the EKF by repeated linearization or the unscented Kalman filter (UKF) by applying the unscented transform. The posterior distribution over state, however, is assumed to be approximately Gaussian, $\boldsymbol{x} \sim \mathcal{N}\big(\boldsymbol{\mu}, \Sigma\big)$ with mean $\boldsymbol{\mu}$ and covariance Σ. Filtering complexity generally grows with state size because map updates involve expensive matrix operations (in general, matrix multiplication is $\mathcal{O}(n^3)$ in the state dimension). Additionally, the fully dense covariance matrix has a quadratic memory requirement.

The information filter (Bar-Shalom et al., 2001) is the information (inverse covariance) form of the Kalman filter, $\boldsymbol{x} \sim \mathcal{N}^{-1}\big(\boldsymbol{\eta}, \Lambda\big)$, where $\boldsymbol{\eta}$ and Λ are the information vector and matrix, respectively. Thrun et al. (2004) empirically observed that the information matrix over landmark states is dominated by a few terms that encode relationships between nearby landmarks, suggesting that the dense information matrix can be approximated as a sparse matrix. By performing a sparsification procedure, the authors improved upon previous EKF based implementations, allowing constant time updates and memory requirements linear in the number of map landmarks. The sparsification procedure, however, introduced the potential for overconfidence as it throws away information, i.e., ignores conditional relationships between map elements.

Particle filters (Doucet et al., 2001) are also a popular tool for filter-based SLAM (Montemerlo et al., 2002). Unlike the previously mentioned methods, the particle filter is able to represent arbitrary posterior distributions—not only unimodal Gaussian. However, the complexity of particles filters is difficult to evaluate due to the ambiguity in defining adequate particle density.

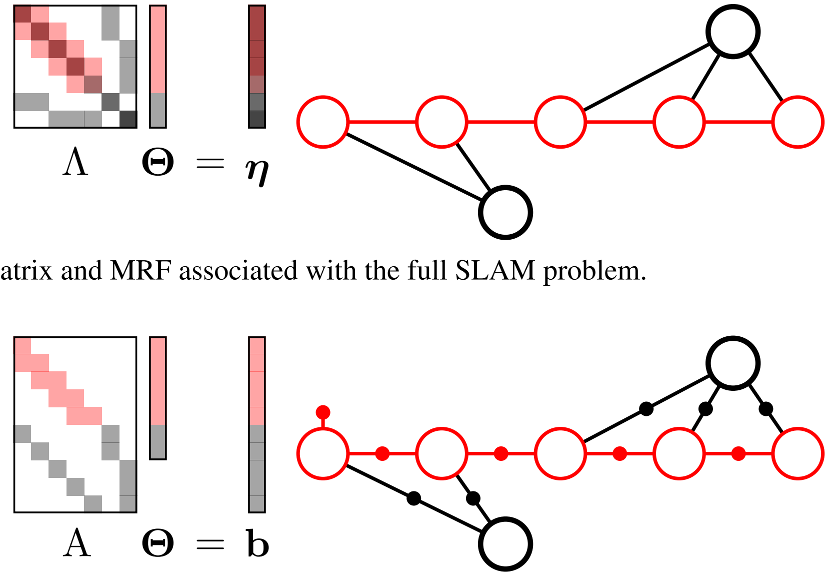
1.2.2 Graph-based Approaches

The challenge of the data association problem is defining a set of repeatably observable landmarks or features. Lu and Milios (1997) proposed a SLAM solution that copes with this problem through an alternate map representation: modeling the environment as a collection of robot poses or views as opposed to a set of landmarks or features. The collection of historic robot poses is commonly referred to as the pose-graph. Lu and Milios (1997) showed that the pose-graph formulation leads to an equivalent optimization problem that minimizes the error of reprojected data between overlapping views. Although their original solution solved a batch maximum likelihood estimate (MLE) problem, Gutmann and Konolige (1999) later presented a real-time implementation that essentially performs a batch update over a sliding window of robot poses. Several derivative works employ an equivalent 'delayed-state', i.e., view-based, filtering framework (Eustice et al., 2006a; Walter et al., 2007) to achieve real-time performance.

Based on the work of Lu and Milios (1997) and Thrun et al. (2004), Eustice et al. (2006a) observed that the information matrix of a delayed-state representation is *exactly* sparse, i.e., requires no approximate sparsification procedure. The authors noted that the information matrix becomes fully dense as a result of marginalizing out robot poses, whereas maintaining the robot trajectory preserves sparsity. The distribution over the delayed-state vector can be factored as a Markov random field (MRF) and represented graphically (Fig. 1.2a). The information matrix exactly corresponds to the adjacency matrix of the the MRF. Marginalization introduces new edges in the graph called fill-in, which generally increases the graph density and therefore the complexity of the solution. This insight allows the benefits of constant time updates without the drawbacks of an approximate sparsification that can lead to overconfidence. Furthermore, Eustice et al. (2006a) noted that the information form encodes a sparse linear system in which most updates involve only a small portion of the system, suggesting a fast approximate method for state-recovery.

Filtering solutions all suffer from eventual divergence that occurs as a result of linearization error (Julier and Uhlmann, 2001). Lu and Milios (1997) computed the full robot trajectory as the solution to a sparse nonlinear system over robot poses—minimizing the squared error over robot poses. Similarly, the full SLAM problem defines a sparse system over both robot poses and landmarks as in Fig. 1.2. Graphical SLAM by Folkesson and Christensen (2004), Square root smoothing and mapping (SAM) by Dellaert and Kaess (2006), and GraphSLAM by Thrun

Figure 1.2 The graphical representation of the SLAM problem (Fig. 1.1) exposes its sparse underlying structure.


$\Lambda \quad \boldsymbol{\Theta} = \boldsymbol{\eta}$
$\mathrm{A} \quad \boldsymbol{\Theta} = \mathbf{b}$

(a) Information matrix and MRF associated with the full SLAM problem.

(b) Measurement Jacobian and factor graph associated with the full SLAM problem. Small circle nodes represent factors, i.e., measurements. The factor graph corresponds to a least-squares problem that computes the parameter vector, $\boldsymbol{\Theta}$, in order to best explain the observations.

and Montemerlo (2006) considered the efficient solution of this system using the Gauss-Newton algorithm, involving iteratively solving a large sparse linear system. Current state-of-the-art SLAM algorithms leverage well established sparse linear algebra routines to quickly and efficiently solve the underlying large linear system (Dellaert and Kaess, 2006; Olson et al., 2006b; Kaess et al., 2008, 2012; Kummerle et al., 2011; Polok et al., 2013).

For additive Gaussian noise models, the MAP SLAM formulation (1.1) can be written

$$\begin{aligned}\hat{\boldsymbol{\Theta}} &= \underset{\boldsymbol{\Theta}}{\operatorname{argmax}}\, p(\boldsymbol{\Theta}|\mathbf{U},\mathbf{Z}) \\ &= \underset{\boldsymbol{\Theta}}{\operatorname{argmin}} - \log p(\boldsymbol{\Theta}|\mathbf{U},\mathbf{Z}) \\ &= \underset{\boldsymbol{\Theta}}{\operatorname{argmin}} \sum_i \left\| f_i(\boldsymbol{\Theta}) - \mathbf{u}_i \right\|^2_{\mathrm{Q}_i} + \sum_j \left\| h_j(\boldsymbol{\Theta}) - \mathbf{z}_j \right\|^2_{\mathrm{R}_j}, \end{aligned} \tag{1.2}$$

where the parameter vector $\boldsymbol{\Theta}$ includes poses $\mathbf{X}$ and landmarks $\mathbf{L}$, and $f_i(\,\cdot\,)$ and $h_i(\,\cdot\,)$ are the process and observation models with noise covariance matrices Q_i and R_j, respectively. We recognize (1.2) as a nonlinear least-squares problem. Within a nonlinear least-squares solver (e.g., Gauss-Newton), (1.2) is linearized to produce a large linear least-squares problem in the estimate update $\delta\boldsymbol{\Theta}$

$$\begin{aligned} \delta\hat{\boldsymbol{\Theta}} &= \underset{\delta\boldsymbol{\Theta}}{\operatorname{argmin}} \left\| \mathrm{A}\delta\boldsymbol{\Theta} - \mathbf{b} \right\|^2 \\ &= (\mathrm{A}^\top \mathrm{A})^{-1}\mathrm{A}^\top \mathbf{b}, \end{aligned} \tag{1.3}$$

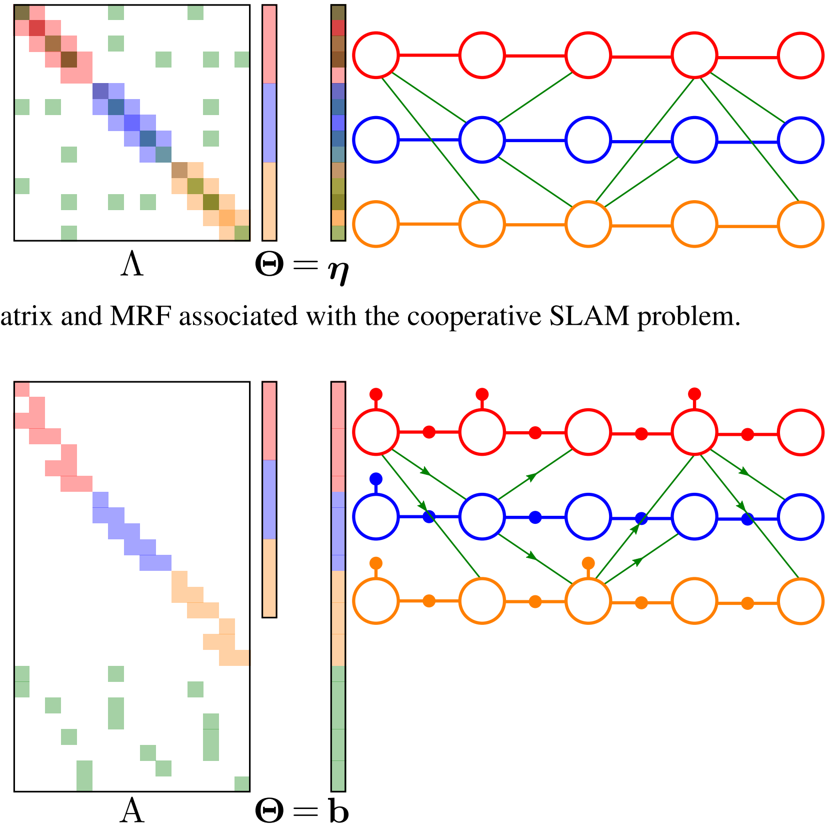
where the matrix A represents the stacked whitened measurement Jacobian and $\Lambda = \mathrm{A}^\top\mathrm{A}$ is the information matrix (inverse error covariance matrix, i.e., Σ^{-1}). Probabilistic graphical models corresponding to this problem are discussed in Appendix A.

Algorithms to solve linear systems are classified as direct methods (e.g., matrix factorization as in Cholesky, LU, or QR), or iterative methods (e.g., relaxation or conjugate gradients). Direct methods are more common for SLAM solutions. Dellaert and Kaess (2006) considered direct factorization methods (LU and QR) in their Square root SAM algorithm. Kaess et al. (2008) proposed recursively updating the QR factorization within the incremental smoothing and mapping (iSAM) framework. Direct solution methods rely on variable elimination (equivalent to marginalization). As such, an optimal variable ordering exists which reduces fill-in as variables are eliminated, therefore preserving sparsity and boosting performance. Both Square root SAM and iSAM exploit efficient variable ordering heuristics to speed up inference.

Iterative linear solvers offer several advantages over direct methods in terms of memory management, parallelization, and distribution. However, they may require many iterations for convergence. Duckett et al. (2000, 2002) proposed a Gauss-Seidel relaxation based SLAM algorithm that was capable of producing consistent maps. Both Howard et al. (2001) and Frese et al. (2005) considered multi-grid relaxation methods popularized in computational mechanics for the solution of large partial differential equations. Relaxation based methods unfortunately have no upper bound on the number of iterations required to reach a desired tolerance of a fixed-point solution. Conjugate gradients only requires a finite number of iterations to converge, but cannot guarantee that intermediate steps monotonically approach the minimum. Moreover, the required number of iterations may still be prohibitively large. Thrun and Montemerlo (2006) suggested conjugate gradients for a batch solution, while Konolige (2004) implemented a real-time iterative method based around conjugate gradients similar to Gutmann and Konolige (1999). Dellaert et al. (2010) proposed a special preconditioned conjugate gradients method for large scale SLAM problems that outperforms direct methods for dense maps. Preconditioning is extended to extremely large structure-from-motion (SfM) problems (analogous to the batch full SLAM problem) by Agarwal et al. (2010) and Jian et al. (2012, 2013).

1.3 A Review of Cooperative Navigation

Figure 1.3 The graphical representation of cooperative SLAM illustrates the sparse structure as well as the origin of information, i.e., edges and factors are colored by their origin platform. $\boldsymbol{\Theta}$ includes poses of all platforms, while no landmarks are present in this example—observations are only directly between platforms.



Λ
$\boldsymbol{\Theta} = \boldsymbol{\eta}$
A
$\boldsymbol{\Theta} = \mathbf{b}$

(a) Information matrix and MRF associated with the cooperative SLAM problem.

(b) Measurement Jacobian and factor graph associated with the cooperative SLAM problem.

Fielding teams of robots promises to boost reliability and performance over single robot deployments. Additionally, information can be shared across the team to improve navigation. Similar to single-robot SLAM, cooperative multiple-platform SLAM is interested in answering the question ‘where are *we*?’. These algorithms fuse information originating from several platforms including relative vehicle observations in addition to odometry and landmark measurements. Cooperative SLAM imposes a larger computational cost (as the state dimension grows linearly with the number of platforms) and increased communication cost when compared to single robot SLAM. Additionally, cooperative SLAM algorithms may have to determine the initial correspondence between local reference frames defined for each separate platform. For the cooperative AUV algorithms presented in this thesis, all platforms are initialized in the same local reference frame given GPS observations at the surface.

The multi-platform SLAM problem can be solved as a batch process within a centralized estimator as suggested by Howard et al. (2002) and Dellaert et al. (2003) using an MLE framework. A centralized system has direct access to all measurements as they become available and hence provides the benchmark solution. This solution corresponds to (1.1) where all vehicle poses are now included as illustrated in Fig. 1.3. Centralized solutions, however, are not generally attainable online due to their large communication and processing requirements. Decentralized solutions are necessary for real-time implementation within the underwater domain given the constraints of the acoustic communication channel (Section 1.5). Below, we classify distributed systems as a superset of decentralized in that distributed systems may require some centralized coordination (Mu et al., 2011).

1.3.1 Distributed SLAM

Early work on decentralized cooperative networks by Kurazume et al. (1994) and Rekleitis et al. (2000) assigned alternating dynamic and static roles to each robot team member. Each moving team member performs single-robot SLAM treating the stationary robots as landmarks. Although these methods do not recreate the centralized benchmark, they do not require any (or have little) communication overhead and greatly reduce the rate of error accumulation. However, error growth is still unbounded in time since the 'landmarks' must be re-initialized each time the role designation switches. Moreover, this scheme slows the overall rate of team movement as several team members are static at any given time vitiating several benefits of multiple vehicle operations.

Egocentric algorithms, e.g., Fox et al. (2000), consider the problem of strictly localizing a vehicle given the estimated pose of another vehicle and a relative pose measurement. These algorithms apply the naive Bayes assumption, i.e., that all vehicle estimates are independent, and are so-called egocentric since they track only their own state estimate. After incorporating information from another platform, however, the two vehicle states are dependent. At the next measurement update, the vehicles (erroneously) assume that their estimates are independent. These methods incur very low communication cost as only a single pose vector and covariance ever need to be transmitted. Egocentric methods ignore dependency, and are inconsistent (overconfident) as a result, but are trivially tolerant to communication failure. Problems associated with 'double-counting' information are a hallmark of the cooperative localization literature. A consistent estimate is one that does not produce an overconfident covariance estimate (Bar-Shalom et al., 2001), i.e., $\hat{\Sigma} \succeq \Sigma$, where $\hat{\Sigma}$ and Σ are the estimated and true covariance, respectively.

Roumeliotis and Bekey (2002) first noted that ignoring dependencies that develop between estimates can lead to gross inconsistency and overconfidence. They proposed a distributed Kalman filter approach, which accounts for the interdependency of information at the cost of significantly increased communication. Thrun and Liu (2003) presented the distributed version of the sparse extended information filter (SEIF), which is similar in spirit to Roumeliotis and Bekey (2002), although it performs a sparsification procedure to reduce complexity. These methods distribute the full computation of the centralized filter, which is guaranteed to be consistent as it accounts for all information, amongst the vehicle network. Nerurkar and Roumeliotis (2008) and Nerurkar et al. (2009) developed a distributed nonlinear MAP estimator. The authors leverage a distributed conjugate gradients algorithm (Bertsekas and Tsitsiklis, 1997) to distribute the computation of the centralized linear subproblem.

Instead of attempting to reconstruct a centralized filter, Bahr et al. (2009b) proposed a bookkeeping strategy to prevent fusing correlated data, termed the interleaved update (IU) algorithm. While IU avoids inconsistency, the communication cost is significantly higher than its egocentric counterpart as it maintains and communicates a bank of filters, which grows exponentially in the size of the vehicle network. Moreover, IU only uses a subset of relative platform information as it periodically discards correlated information.

Decentralized data fusion (DDF) for general multiple sensor systems has recently informed cooperative navigation research. The key philosophy of DDF is that locally obtained information is first fused locally and then communicated where it is fused with all accumulated messages from the network. In this way, each platform has a consistent estimate that is tolerant to communication failure using all received information. Covariance intersection (CI) presented by Julier and Uhlmann (1997), later expanded by Mahler (2000) and generalized by Bailey et al. (2012), suggests a conservative approach for combining estimates with unknown dependence. Grime et al. (1992) reported the channel filter, which is capable of distributing the linear information filter for known communication topologies. In the case of DDF frameworks, each locally computed estimate will not exactly match the centralized solution because each platform may have only received a subset of the information.

Graph-based cooperative localization research shares the DDF mentality by combining a local and global fusion layer. As such, many algorithms require that each platform perform the following three tasks:

i. Build a local graph fusing local information (as in the single robot SLAM scenario),

ii. Communicate and collect local graphs from other platforms,

iii. Optimize the global graph that fuses the received local graphs.

The decentralized data fusion smoothing and mapping (DDF-SAM) algorithm, proposed by Cunningham et al. (2010, 2012, 2013) explicitly performs exactly these tasks within the iSAM framework. Each robot builds a local graph just as in single robot SLAM and communicates it across the network. Concurrently, each robot collects local graphs from each of its teammates. The set of received local graphs is then optimized in a single framework. These methods attempt to reconstruct the centralized graph (Fig. 1.3). Unfortunately, the communication cost grows with the size of each local graph.

The DDF model for distributed robot localization, such as in Cunningham et al. (2010), remains the standard. Kim et al. (2010) report on the alignment of multiple pose-graphs using iSAM. They employ the ‘anchor-node’ concept, equivalent to the base-nodes of Ni et al. (2007), to simultaneously compute the initial correspondence between robot reference frames. Howard et al. (2003) distribute their earlier centralized MLE approach (Howard et al., 2002) using a system similar to relaxation. The advantage to these solutions is that the estimate is guaranteed to be consistent because the full pose history is maintained.

The above algorithms may consider a communication limit, but do not constrain it to the point of affecting performance. Ribeiro et al. (2006) suggested a noteworthy approach to multi-sensor data fusion, transmitting a single bit per measurement, the sign-of-innovations. Within this framework, each platform attempts to approximate the centralized estimate using each quantized measurement. The authors derive an algorithm that closely mirrors the standard Kalman filter by approximating the posterior probability given each quantized measurement as a Gaussian. Nerurkar et al. (2011b) elaborated on this work by allowing local analog (not quantized) measurements in addition to the remote quantized observations. Trawny et al. (2009) presented a MAP estimator integrating sign-of-innovations quantized observations. Although there is a small bandwidth requirement, these algorithms require a non-lossy all-to-all communication topology.

Leung et al. (2010) attempted to reduce the required bandwidth of communicated information by leveraging the Markov structure of local graphs. This approach greatly reduces the required bandwidth, although relies on acknowledgments to bound communication. Without acknowledgment, the required communication may grow too large.

1.3.2 Submap Approaches

Submap-based SLAM, multiple session SLAM, and the multiple platform cooperative localization problem are in many ways strongly related. Submap and multiple session approaches seek to align several maps originating from a single robot at different times, whereas multiple platforms may align several maps from different robots collected simultaneously. In order to efficiently cope with large environments, many single-robot SLAM algorithms employ submap-based distribution schemes such as ATLAS SLAM (Bosse et al., 2004; Newman et al., 2002). A submap-based approach essentially performs SLAM within small regions of the environment to remain tractable, then performs a separate optimization step to align each of the submaps. Multiple session algorithms (Ozog et al., 2015) may be employed to support long-term single-robot applications.

Ni et al. (2007) proposed a submap-based SAM approach termed Tectonic SAM. The authors partition the measurement Jacobian by metric submaps, and perform the optimization first over submaps and then over the map separators. The global alignment can be performed efficiently by adding 'base-nodes'—variables that dictate the relative coordinate frame of each submap. Base-nodes reduce the complexity of the measurement Jacobian. Tectonic SAM was then trivially extended to a distributed multi-robot application by Andersson and Nygards (2008) in an algorithm dubbed collaborative smoothing and mapping (C-SAM) wherein each submap was defined as each robot's local map. Each robot then communicates information regarding the separator variables across the network.

1.3.3 Network Localization

Relative robot localization shares a high dot product with network localization from the distributed sensor network community. Researchers here often want to localize static nodes given only relative range or bearing information to a number of neighboring nodes (Hendrickson, 1995; Jacobs and Hendrickson, 1997; Patwari et al., 2005; Aspnes et al., 2006). These algorithms are often based on geometrical methods aiming to avoid ambiguities endemic to the partial-state information contained in range- or bearing-only observations (Anderson et al., 2008). Another body of work considers the optimization problem given a set of range-only measurements. In particular, Barooah et al. (2006) proposes a decentralized relaxation-based algorithm for localization and time-synchronization. While these applications involve static nodes, a relaxation-based method is attractive for its low-bandwidth requirement and ability to reconstruct the centralized solution after a number of iterations.

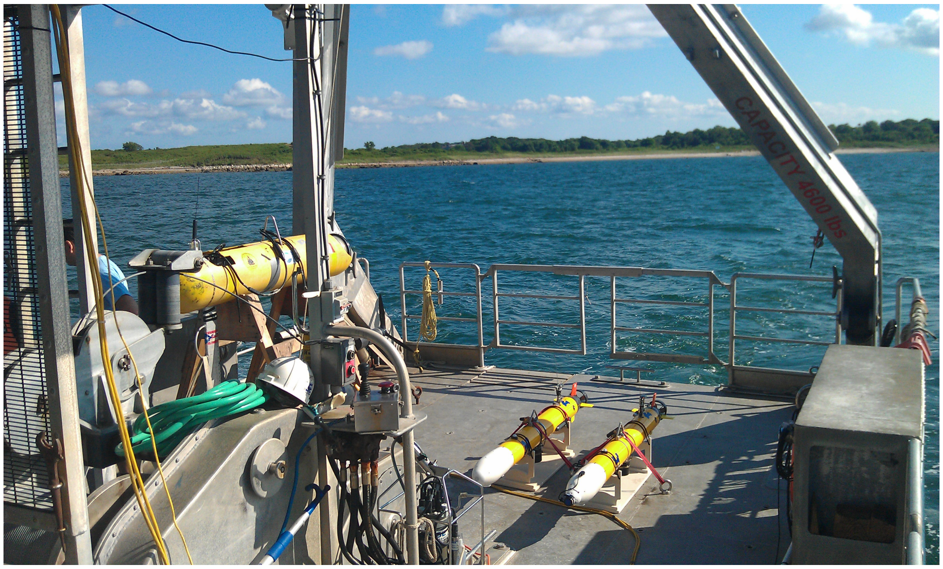
1.3.4 Consensus

Consensus algorithms have gained popularity in multiple platform systems for their distributive properties (Ren et al., 2005). Consensus describes methods for summarizing and sharing information in order to reach an agreement. Other than fairly weak assumptions on network connectivity, communication in a consensus network only depends on the estimated quantity—independent of the number of platforms, or even the origin of each message. In the cooperative control community, agreement may refer to the relative geometry within a vehicle formation. Within sensor networks, agreement could pertain to estimating a particular quantity, e.g., ocean temperature or salinity (Lynch et al., 2008). Peterson and Paley (2013) proposed a consensus-information filter for estimating flow-field parameters in mobile networks based on the distributed Kalman filter by Olfati-Saber (2007).

Aragues et al. (2012) proposed a multiple platform consensus-based SLAM formulation to estimate the global landmark map. Within a cooperative localization context, instantaneous vehicle poses involved in relative measurements could be considered landmarks. The size of the map, then, would monotonically increase for each new relative measurement, quickly exceeding allowable bandwidth in the underwater domain. While ill-suited to the cooperative localization problem, consensus deserves mention for its low-bandwidth decentralized estimation capabilities and use within cooperative control architectures.

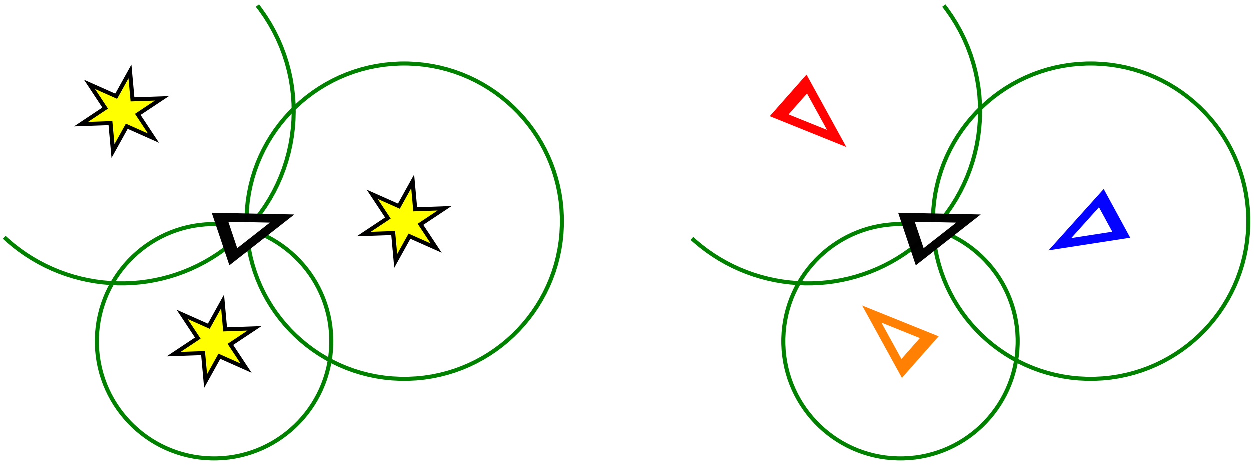
1.4 A Review of Underwater Navigation

Figure 1.4 AUVs onboard the *R/V Tioga* in Buzzard's Bay, MA. The two adjacent AUVs on deck were used for experimental validation throughout this thesis.



Underwater vehicles (Fig. 1.4) typically integrate body-frame velocity and attitude measurements to compute a dead-reckoned navigation solution. While depth and attitude are instrumented with bounded-error, dead-reckoned XY-horizontal position errors grow unbounded in time without absolute position measurements, (e.g., GPS). Higher quality sensors are only capable of reducing the rate of error growth. Therefore, alternative methods for constraining navigation errors are required. A detailed survey of the state-of-the-art in underwater navigation is presented by Kinsey et al. (2006). We review typical Doppler navigation techniques and acoustic beacon-based navigation for underwater vehicles.

Figure 1.5 Acoustic navigation example. (a) illustrates the trilateration of a vehicle position (black triangle) given ranges to known beacons (yellow stars). (b) shows trilateration using mobile beacons whose positions are not well known.



(a) Fixed-beacon LBL example

(b) Mobile-beacon example

1.4.1 Doppler Navigation

The Doppler velocity log (DVL) is capable of accurately measuring seafloor referenced (bottom-lock) or water-column referenced (water-lock) velocity (RD Instruments, 1996). DVLs observe velocity by measuring the time-dilation (either frequency shift or phase) of a reflected acoustic beam. Vehicles compute a dead-reckoned navigation estimate by transforming observed beam velocities into the world frame using observed attitude and then integrating in time. Whitcomb et al. (1999a) report that noisy heading measurements are the most significant source of error for bottom-track DVL navigation. For vehicle altitudes in excess of the bottom-lock maximum range, water-lock navigation is possible, although vehicles must also estimate the water currents. Without an absolute position reference, dead-reckoned position uncertainty will grow unbounded in time.

1.4.2 Fixed-beacon Acoustic Methods

Underwater acoustic beacon navigation systems attain bounded-error navigation by measuring their relative range to beacons with known position (Vickery, 1998). These systems include: long-baseline (LBL), short-baseline (SBL), and ultra-short-baseline (USBL) systems.

Vehicles observe their range to a beacon by measuring the acoustic time-of-flight (TOF) and assuming a known sound velocity profile. Two-way travel-times (TWTTs)—round-trip TOF—are obtained when a vehicle interrogates and then receives a response from the beacon network. The rate at which vehicles can receive range measurements decreases with the number of vehicles in the network. In contrast, the one-way-travel-time (OWTT) is computed with knowledge of the exact transmit and receive times; the OWTT is the time difference between the time-of-launch (TOL) and the time-of-arrival (TOA). This is accomplished by synchronizing all clocks in the network. Synchronous-clock acoustic networks can scale to arbitrarily many vehicles because all vehicles within acoustic range of the transmitting platform passively receive a range measurement leading to constant time update rates. The synchronous-clock hardware employed in this thesis is detailed in Appendix C.

The LBL navigation framework, for example in Fig. 1.5a, employs a network of fixed reference beacons to which vehicles can measure range via TWTT (Hunt et al., 1974; Milne, 1983; Whitcomb et al., 1999b; Yoerger et al., 2007) and provides the standard in underwater acoustic positioning. LBL, however, limits the area of operations to the coverage footprint of the reference beacons. LBL range and accuracy is dependent on the acoustic frequency employed since attenuation of acoustic signals is frequency dependent. Low frequency systems ($8 - 16 \text{ kHz}$) can achieve a maximum range greater than 10 km, while more accurate high frequency systems ($200 - 300 \text{ kHz}$) only provide coverage up to a few hundred meters.

Most LBL systems require a lengthy calibration procedure to survey the beacon network, spending valuable ship time (Jakuba et al., 2008). Range-only landmark-based SLAM (discussed in Section 1.2) with *a priori* unknown beacon positions was initially proposed for terrestrial navigation with radio frequency (RF) beacons by Kantor and Singh (2002) and later expanded by Djugash et al. (2006) using the traditional EKF. This concept was later adapted to underwater acoustic networks by Newman and Leonard (2003) and Olson et al. (2006a). The important insight in these works was to treat beacon TOL positions as landmarks in the SLAM map.

Some recent work has addressed localization given only a single static beacon with known position. The vehicle is not able to trilaterate its position given a single range observation. However, the vehicle can estimate its position given multiple range observations to the beacon and its dead reckoned navigation. The observability requirements of such a system have been studied by several authors including Song (1999), Gadre and Stilwell (2005), Zhou and Roumeliotis (2008), and Fallon et al. (2010b) using both linear (Chen, 1984) and nonlinear observability analysis (Hermann and Krener, 1977). The literature shows that only a few trivial scenarios exist that are unobservable. In general, if the relative velocity between the platforms change, their relative position is observable. In Chapter 4, we consider the information utility of relative vehicle trajectories.

1.4.3 Mobile-beacon Acoustic Methods

The idea of single fixed beacon navigation has led to the development of algorithms considering a single *mobile* beacon. Scherbatyuk (1995) proposed an early least-squares algorithm to estimate the initial position of an AUV and water current given ranges to a single beacon with known location. Similar solutions using an EKF are presented by Baccou and Jouvencel (2003) and Gadre and Stilwell (2005). Both Scherbatyuk (1995) and Baccou and Jouvencel (2003) note that the beacon could be another mobile vehicle given that it is able transmit its own position estimate. Likewise, Vaganay et al. (2004) also reports an EKF-based solution where a well instrumented AUV acts as a communication/navigation aid, providing an accurate position estimate to a 'client' vehicle.

Recent algorithms for mobile-beacon acoustic navigation are instances of the more general cooperative localization algorithms reviewed in Section 1.3. The challenge for these approaches is in performing distributed estimation over the acoustic channel; each platform must decide what information to share with its team and how to compute a navigation solution given that information.

Eustice et al. (2011), McPhail and Pebody (2009), and Fallon et al. (2011) consider navigating an AUV given acoustic ranges to a 'server' surface ship using MLE, a nonlinear least mean squares method, and iSAM, respectively. These methods consider each range estimate originating from an independent ship position estimate, which is, in general, only valid if the position is provided via independent GPS observations.

Bahr et al. (2009a) provide an alternative augmented navigation framework allowing an AUV to navigate given ranges to a cooperative navigation aid (CNA) (GPS reference source). This algorithm tracks all potential AUV positions as the intersection of range circles propagated forward by the AUV's dead-reckoning. Morice and Veres (2010) suggests a similar geometric bounding technique to solve for the initial AUV position based on set membership.

Maczka et al. (2007) present an egocentric EKF (equivalent to the egocentric approaches presented in Section 1.3) that is capable of sharing range measurements and position estimates between multiple AUVs. As noted by Roumeliotis and Bekey (2002) and demonstrated by Walls and Eustice (2011), ignoring the correlation that builds between platform estimates given relative observations leads to inconsistency. On the other hand, this approach is trivially tolerant to communication failure.

Diosdado and Ruiz (2007) proposed a decentralized SLAM information filter algorithm for use with AUVs. Each vehicle transmits the information matrix and vector corresponding to its accumulated landmark-based map. This algorithm is similar to the channel filter (Grime et al., 1992) in that it tracks common information in the network in order to maintain consistency.

The centralized extended Kalman filter (CEKF) framework proposed by Webster et al. (2009b, 2012), similar to other centralized delayed-state cooperative navigation solutions (Section 1.3), is particularly well-suited for synchronous-clock acoustic navigation. This post-process method fuses all available information. This work was later decentralized by the authors (Webster et al., 2010, 2013) by exploiting properties of the information filter, but is not feasible in practice without modification as it requires a non-lossy communication channel. This method serves as the basis

Table 1.1 Comparison of various underwater communication modes.

Mode	Frequency	Range	Data Rate
Long-Wave Radio	$1 - 100$ kHz	$6 - 20$ m	1 kbps
Optical	—	100 m	>1 Mbps
Acoustic	8-28kHz	100m-10km	80 bps-5 kbps (burst)

for the work described in Chapter 2.

Fallon et al. (2010a) suggested a bookkeeping method dubbed the 'measurement distribution framework'. Similar to some DDF approaches for terrestrial cooperative navigation, this framework distributes locally obtained map representations throughout a network. Unlike its terrestrial counterparts, only the newest information is transmitted in an effort to reduce the communication cost. Each vehicle tracks the information it has received from other platforms in a measurement table. All contiguous information is then processed to obtain the best available estimate over vehicle TOA and TOL poses. Vehicles have the ability to request information that is known to have been lost (deduced from the measurement table). This method does not practically scale to many vehicles because of communication cost; every GPS and range measurement must be transmitted accompanied by a delta dead-reckoning from the previous observation event, this includes all TOLs and TOAs.

1.5 A Review of Underwater Acoustic Communication

Multiple platform underwater navigation relies on the ability of each platform to communicate with its team. The physical characteristics of the underwater environment limit underwater communication compared to terrestrial or air (Catipovic, 1990; Preisig, 2006; Partan et al., 2007; Otnes et al., 2012). The high rate of absorption of both long-wave radio and optical signals in water has limited their use for communications (see Table 1.1). Acoustic communication improves upon these other modes in terms of range and reliability but still faces several complications.

Acoustic communication suffers from large propagation delays as a result of the slow sound speed—roughly 1500 m/s, five orders of magnitude slower than the speed of light. Bandwidth is also extremely limited when compared to terrestrial radio-based communication networks. A rough limit on the range-rate product for acoustic communication is $40 \text{ km} \cdot \text{kbps}$, which mostly applies to vertical channels in deep water and overestimates performance in shallow-water (Partan et al., 2007). Further physical difficulties faced by acoustic communication include: absorption and spreading losses, waveguide effects, multipath, surface scattering, bubbles, and ambient noise (Partan et al., 2007). In practice, underwater vehicle networks are only able to obtain real-world bandwidth on the order of 100 bps (Murphy, 2012).

Most acoustic transducers cannot simultaneously transmit and receive. Therefore, standard acoustic communication networks partition a transmission periods amongst the network, limiting each vehicle's opportunity to share information. Time division multiple access (TDMA) is used predominantly although frequency-division multiple access (FDMA) and code division multiple access (CDMA) are also used with varying degrees of success. No typical medium access control (MAC) protocol exists, and is usually designed application-specific. The choice of MAC dictates the ability of each vehicle to regularly transmit navigation information.

1.6 A Review of Robot Planning

Planning has generally been considered a separate problem from localization and SLAM and answers the question 'how do I get there?'. In general, deterministic start-to-goal path planning is a well studied problem (Latombe, 1991; Choset et al., 2005; LaValle, 2006). Recently, researchers have begun to consider planning under uncertainty such that the planned path reflects the uncertainty in the map, and may even attempt to improve the map. For range-only cooperative localization, we are interested in computing paths that provide useful localization information to another vehicle instead of simply a collision-free path to a goal state.

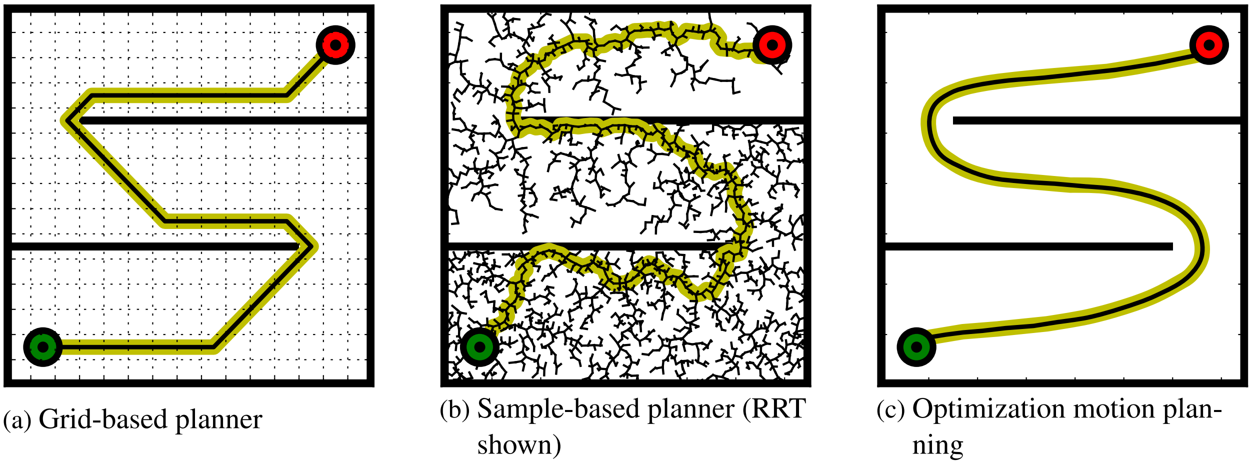
1.6.1 Deterministic Robot Planning

Path planning classically considers computing a collision-free path from a start to goal state given a known configuration space. In early planning work, the objective of planning algorithms was completeness—the ability to find a collision-free path if one exists. These methods, however, did not scale well to high-dimensional configuration spaces and did not explicitly optimize any performance metric. More recent planning algorithms can generally be categorized into three classes: grid or resolution-complete, sampling-based, or optimization motion planning. These methods all assume that the vehicle has perfect state feedback and that all controls will be executed deterministically. Examples of each planner are illustrated in Fig. 1.6.

Resolution-complete planners discretize the environment into grid cells and apply search techniques such as Dijkstra's (Dijkstra, 1959) or A* (Hart et al., 1968) in order to find optimal paths within the grid environment. Dolgov et al. (2010) present an A* approach for driving in unstructured parking lots. Resolution-complete planners are generally intractable in high-dimensional configuration spaces. Using optimal search strategies, resolution-complete planners are also resolution optimal.

Sampling-based motion planning algorithms such as the probabilistic road map (PRM) (Kavraki et al., 1996; Švestka and Overmars, 1997) and the rapidly-exploring random tree (RRT) (Lavalle,

Figure 1.6 Illustration of planning approaches for a holonomic vehicle moving from the start (green) to goal (red).



(a) Grid-based planner

(b) Sample-based planner (RRT shown)

(c) Optimization motion planning

1998; LaValle and Kuffner, 2001) can quickly explore high-dimensional configuration spaces and offer probabilistic guarantees on completeness (completeness with probability one in the limit of infinite samples). Elbanhawi and Simic (2014) provide a comprehensive review of sampling-based techniques. Rickert et al. (2014) noted that probabilistic completeness does not generally reflect the ability of a planner to provide high quality paths. The rapidly-exploring random graph (RRG) algorithm and its derivative RRT* (Karaman and Frazzoli, 2011) provide both probabilistic completeness and asymptotic optimality for a user defined performance metric. Essentially, their proof relies on showing that in the limit, the RRG contains *all* possible paths through the environment.

Optimization motion planning and trajectory optimization have recently become popular due to the local optimality guarantees they provide (Quinlan and Khatib, 1993; Brock and Khatib, 2002; Ratliff et al., 2009; Kalakrishnan et al., 2011; Zucker et al., 2013). Sampling-based planners often include an optimization or 'smoothing' final step. Optimization motion planning can also be performed more rapidly than sampling-based planners, although initialization may be nontrivial.

1.6.2 Robot Planning under Uncertainty

Unfortunately, both control and sensing are nondeterministic, that is, control actions cannot be applied perfectly nor the true state known exactly. Planning under uncertainty has been addressed by planning in the information state or belief state, herein referred to as belief space planning. Optimal paths are constructed over beliefs (distribution over state) as opposed to the underlying state. Belief space planners will often seek paths that gain information (reduce uncertainty) in the state. As with deterministic planning, solutions to belief space planning problems include both sampling-based and optimization approaches.

Sampling-based approaches to belief space planning include: a simple breadth-first-search graph search (Sim and Roy, 2005), roadmap algorithms (Prentice and Roy, 2009; Agha-mohammadi et al., 2014), RRT (van den Berg et al., 2011a), and RRG methods (Bry and Roy, 2011; van den Berg et al., 2011a). These methods do not explicitly perform an optimization step, but are able to find near optimal paths by searching over a set of candidate solutions.

Optimal motion planning with perfect state feedback can be formalized as a Markov decision process (MDP) given a probabilistic state transition model. With imperfect state information, the belief space planning problem can be framed as a partially observable Markov decision process (POMDP). While POMDPs are generally intractable (Kaelbling et al., 1998), several recent methods have assumed parametric belief states (e.g., Gaussian) and are able to compute solutions in polynomial time (Platt et al., 2010; van den Berg et al., 2011b, 2012; Patil et al., 2014; He et al., 2011). These methods make linear Gaussian approximations and rely on iterative linear quadratic Gaussian (LQG) or differential dynamic programming (DDP) to solve.

Active SLAM attempts to plan a path in (partially) unknown environments that trade off between completing some objective and reducing uncertainty in its map. In general, the active SLAM problem can be expressed as a POMDP. Recently, active SLAM has been framed within belief space planning frameworks (Sim et al., 2004; Sim and Roy, 2005; Stachniss et al., 2005; Kollar and Roy, 2008; Kim and Eustice, 2015; Chaves et al., 2014; Indelman et al., 2014, 2015).

1.6.3 Robot Cooperative Control

Several researchers have proposed cooperative adaptive planning strategies in order to optimize some performance metric. For example, Hollinger et al. (2012) present a path planning algorithm for collecting data from a sensor network over the acoustic channel, i.e., data-muling, onboard an AUV to maximize the information collected while minimizing the travel cost. Similarly Burns et al. (2006) and Leonard et al. (2007) propose planning and control methods to maximize communication performance and survey coverage, respectively. Both Lynch et al. (2008) and Peterson and Paley (2013) present cooperative control algorithms to estimate unknown spatial fields using consensus-information filters.

1.6.4 Planning for Cooperative Localization

Planning for range-only cooperative localization has been previously considered by several researchers. Olson et al. (2006a) proposed an algorithm for an AUV to localize a fixed LBL beacon. Charrow et al. (2012) presented an algorithm to localize a mobile target given radio-frequency based range-only observations. Tan et al. (2014) considered planning a path for one AUV in order to localize another AUV along a desired path. Both Bahr et al. (2012) and Seto et al. (2011) considered a similar problem, but did not assume that the desired path was known at planning time.

To the best of our knowledge, no previous work has considered uncertainty in the execution of a path or in the acoustic channel.

1.7 Thesis Outline

Cooperation within multiple platform teams offers the potential to improve performance, e.g., area-coverage and efficiency, while simultaneously providing navigation information to each team member. These are desirable traits for a scientific tool, especially in the underwater community where a vast area is yet unexplored and a high demand for geo-referenced data exists. This thesis considers two problems:

- Given a team of underwater vehicles, we explore how to localize each team member using one-way-travel-time (OWTT)-derived range constraints. This problem addresses the information that must be exchanged amongst the team to produce a consistent navigation estimate (i.e., a solution that does not double-count information).
- Certain geometries of cooperative teams are better suited to localize team members given relative range observations, suggesting that vehicle surveys can be intelligently planned to minimize error growth. In this problem, we consider the effect of relative vehicle trajectories on reducing error across the vehicle network.

The solution domain is extremely constrained given the low bandwidth and generally erratic nature of the acoustic communication channel. Both the estimation and control subproblems must be tolerant to small infrequent communication updates between team members. The proposed problem is to exchange information between members of an underwater communication network in order to augment their navigation with measured relative range and, given an estimation framework, compute minimal navigation error trajectories. Toward this end, we have produced the following contributions:

(i) Developed an exact one-to-many (server-to-client) cooperative localization algorithm.

(ii) Designed an approximate many-to-many (peer-to-peer) scalable localization framework, which computes the best solution given the subset of information available at-hand to each platform.

(iii) Developed a planning strategy for producing minimal navigation error survey trajectories.

1.7.1 Document Roadmap

Each objective and contribution is detailed in the following chapters.

Chapter 2 We show that by expanding the state representation of a vehicle to include historic poses (i.e., using a delayed-state framework), we can explicitly track correlation that develops between platform estimates as a result of sharing relative observations. We provide an efficient method to incrementally transmit the delayed-state pose chain, represented as an MRF, over a faulty communication channel in order to exactly reproduce the centralized solution to the server-to-client localization problem.

Chapter 3 We exploit the factor structure of a pose-graph in order to approximately share navigation information over a faulty communication channel. This method achieves a scalable (albeit approximate) peer-to-peer localization solution, i.e., one in which every vehicle in the network shares its local information and uses information obtained from other vehicles.

Chapter 4 We present a framework to compute a server trajectory that optimally localizes a client vehicle along a desired survey path. To this end, we have incorporated both uncertain sensing and acting into the planning framework, as well as a probabilistic model of the acoustic communication channel. We allow server trajectories to be computed with a receding-horizon approach or an optimization of parameterized trajectories.

Chapter 5 We provide a summary of contributions as well as potential avenues for future work.

Appendix A We review probabilistic graphical models as they relate to efficient distributed solutions to the cooperative localization problem. Specifically, we review Gaussian MRFs and factor graph models.

Appendix B We detail the XY AUV odometry model that results from forward Euler integrating DVL and attitude observations. We also discuss the OWTT range observation model.

Appendix C We summarize specialized hardware and equipment for synchronous-clock acoustic navigation as employed throughout our field trials. We detail the acoustic message specification for each reported localization algorithm.

Chapter 2

One-to-many Localization

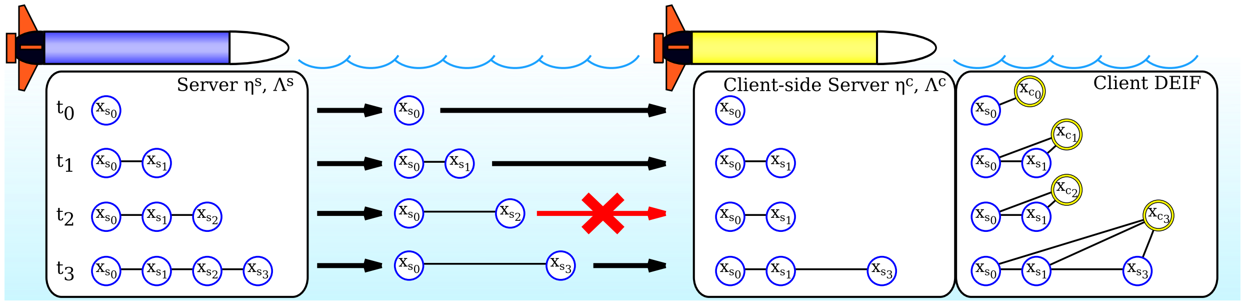
This chapter considers the problem of localizing a client vehicle by measuring its relative range to a server platform[1]. In general, the server-client cooperative localization problem is a distributed state estimation problem. Information sharing and fusion is nontrivial since the server and client state estimates become correlated after incorporating relative range information. Moreover, platforms are restricted by severe communication constraints imposed by the underwater acoustic channel. We leverage properties of the MRF within a 'delayed-state' framework in order to achieve a consistent estimate that can be computed onboard the client.

Underwater vehicles typically rely on fusing DVL body-frame velocities, attitude, and pressure depth observations to compute a dead-reckoned navigation solution. While attitude and depth are well instrumented, there is no easy method to directly observe XY-horizontal position (GPS does not work underwater). In this chapter, we report a novel algorithm enabling server vehicles to cooperatively aid the navigation of subsea client vehicles, and which is tolerant to the packet loss and low-bandwidth that is endemic in underwater acoustic communication networks. Our algorithm is capable of bounding the position error growth of the client vehicles to that of the server vehicles.

Typical bounded-error underwater navigation methods, such as LBL, measure the relative range between the vehicle and fixed reference beacons (Milne, 1983; Whitcomb et al., 1999b). The relative range is typically measured using two-way TOF acoustic broadcasts and assuming a known sound-speed profile. Narrowband LBL beacon networks, however, are limited in their ability to scale to many vehicles because only one vehicle can interrogate the network at a time. Moreover, the range of vehicle operations is limited to the acoustic footprint of the beacon network. In the same vein, USBL systems allow a topside ship to observe the relative range and bearing of a subsea vehicle, but are similarly limited in their ability to scale to large vehicle networks.

[1] Portions of this work have appeared in previously published work including (Walls and Eustice, 2012, 2013, 2014a).

Figure 2.1 Origin state method algorithm overview: the server (blue) fuses its local observations and adds delayed-state poses at each time-of-launch (TOL) (the blue circles). The server uses our novel origin state method to incrementally broadcast its pose-graph in a fault-tolerant way. At the time-of-arrival (TOA) of each received origin state packet, the client (yellow) reconstructs the server pose-graph and updates its estimator to fuse all new information. In this example, although the client misses the server transmission at t_2, the client still reconstructs the server pose-graph after receiving the origin state packet at t_3, which encapsulates all server information accumulated between t_1 and t_3.



The use of synchronous-clock hardware enables a team of vehicles to observe their relative range via the OWTT of narrowband acoustic broadcasts (Eustice et al., 2011). The OWTT relative range is measured between the transmitting vehicle at the TOL and the receiving vehicle at the TOA. Since ranging is passive—all receiving platforms observe relative range from a single broadcast—OWTT networks scale well. The use of OWTT observations to augment vehicle navigation presents several open questions regarding how to share and incorporate information across the network in a communication tolerant.

The underwater acoustic communication channel is severely limited by the physical characteristics of seawater (Partan et al., 2007). Acoustic communication is constrained by high latency and low bandwidth with packet loss often greater than 50%. The underwater acoustic channel has an upper-bound range rate product of $40 \text{ km} \cdot \text{kbps}$. In practice, underwater vehicle networks are only able to obtain real-world bandwidth on the order of 100 bps (Murphy, 2012), which is several orders of magnitude less than terrestrial communication networks. An unacknowledged broadcast protocol is also commonly employed in conjunction with TDMA scheduling, which further limits overall bandwidth by dividing transmission time between platforms in the network. All of these challenges amount to a communication framework that enforces small communication data packets and infrequent updates between vehicles.

A variety of cooperative localization frameworks exist for improving position estimates across a team of robots by sharing local navigation information. Previous methods, however, do not address the severely limited bandwidth and fragility of the underwater acoustic communication channel. In this chapter, we consider a solution to the navigation of a client vehicle aided by a server platform, as depicted in Fig. 2.1. We present the following contributions:

- We present a general algorithm, called the origin state method (OSM), that allows a server to broadcast its pose-graph via a faulty, low bandwidth communication channel that works in real-time for practical, underwater, acoustic networks.

- We use the OSM algorithm to compute a ‘delta information’ over server TOL poses, which serves as input to a decentralized extended information filter (DEIF) algorithm (Webster et al., 2013) that exactly computes the result of the centralized extended information filter (CEIF) onboard the client up to communication delay.

- We provide extensive evaluation over more than $12\ \mathrm{h}$ of field trials demonstrating the ability of the OSM to broadcast a server pose-graph to multiple clients using a small, fixed-bandwidth data packet.

- We provide a novel factor-based interpretation of the ‘delta information’ and discuss how it can be used in real-time in other estimation frameworks such as the nonlinear least-squares iSAM framework (Kaess et al., 2008).

The remainder of this chapter proceeds as follows. Section 2.1 reviews the prior literature within cooperative localization. Section 2.2 summarizes the proposed OSM algorithm. Section 2.3 describes the interface between the OSM and the DEIF. Section 2.4 presents the results of more than $12\ \mathrm{h}$ of field trials performed with multiple AUVs. Finally, Section 2.6 concludes.

2.1 Related Work

Cooperative vehicle networks enable robots with the best navigation sensors to localize robots with poorer position estimates. The goal is to augment each platform’s local sensing with measurement constraints between the vehicles themselves as depicted in Fig. 2.2. Prior literature is discussed below and summarized in Table 2.1.

Simple, real-time algorithms that require minimal bandwidth are within the egocentric class of filters (Fox et al., 2000; Maczka et al., 2007; Vaganay et al., 2004). These algorithms scale by treating each inter-vehicle relative observation as independent and only require the transmitter’s

Figure 2.2 Joint pose-graph (MRF) over server, $\mathbf{x}_{s_i}$, and client, $\mathbf{x}_{c_j}$, poses where $\mathbf{\Theta}$ is a vector including server and client poses, Λ is the information matrix, and $\boldsymbol{\eta}$ is the information vector. Green lines illustrate edges generated by relative range observations between server TOL poses and client TOA poses. In order to construct the full pose-graph, the client must have access to the server’s local information (shown in red).

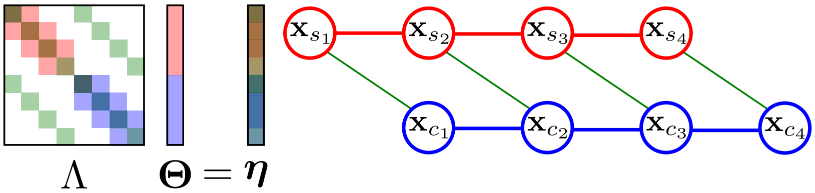


Table 2.1 Summary of prior algorithms for multiple platform navigation, specifically within the context of server-client communication topologies. No previous method is able to reproduce the centralized solution while being tolerant to communication packet loss.

	Literature	**Online/ real-time**	**Packet loss tolerant**	**Bandwidth conservative** [a]	**Reproduces centralized**	**Consistent estimate**	**Comments**
Centralized	Howard et al. (2002)	no	—	—	yes	yes	Centralized MLE.
	Dellaert et al. (2003)	no	—	—	yes	yes	Centralized MLE.
	Webster et al. (2012)	no	—	—	yes	yes	Centralized EKF.
Egocentric	Fox et al. (2000)	yes	yes	yes	no	no	Sampling-based approach.
	Vaganay et al. (2004)	yes	yes	yes	no	no	Moving LBL paradigm.
	Maczka et al. (2007)	yes	yes	yes	no	no	Egocentric KF.
Distributed	Roumeliotis and Bekey (2002)	yes	no	no	yes	yes	Distributed EKF.
	Ribeiro et al. (2006)	yes	no	yes	yes	yes	Quantized innovations.
	Bahr et al. (2009b)	yes	yes	yes	no	yes	Bank of estimators.
	Fallon et al. (2010a)	yes	yes	no	yes	yes	Requires acknowledgements.
	Cunningham et al. (2010, 2012)	yes	yes	no	yes	yes	Transmits reduced pose-graph.
	Kim et al. (2010)	yes	yes	no	yes	yes	Transmits entire server graph.
	Leung et al. (2010)	yes	yes	no	yes	yes	Transmits knowledge sets.
	Webster et al. (2010, 2013)	yes	no	yes	yes	yes	Transmits delta information.
	Nerurkar et al. (2011a)	yes	no	yes	yes	yes	Sign-of-innovations.
	Bailey et al. (2011)	yes	no	yes	yes	yes	Transmits delta information.
	Origin State Method (this chapter)	**yes**	**yes**	**yes**	**yes**	**yes**	Can be used in conjunction with EIF or iSAM.

[a] We consider an algorithm bandwidth conservative if it employs a fixed-bandwidth data packet and does not require acknowledgement.

current position estimate. While completely resistant to communication failure, these methods do not account for the correlation that develops through relative observations between robot estimates, which can lead to inconsistent (i.e., overconfident) estimates (Maczka et al., 2007). The negative consequences of ignoring correlation have been demonstrated by Roumeliotis and Bekey (2002), Bahr et al. (2009b), and Walls and Eustice (2011).

Covariance intersection (Julier and Uhlmann, 1997) can be used to consistently fuse two estimates with unknown correlation. Recently, it has been applied to the cooperative localization problem by Li and Nashashibi (2013) and Carrillo-Arce et al. (2013) to cope with inconsistency in egocentric algorithms. However, previous work requires a full rank (i.e., range and bearing) relative observation. Bahr et al. (2009b) previously noted the challenge of incorporating partial relative pose information in cooperative frameworks with covariance intersection.

Bahr et al. (2009b) and Fallon et al. (2010a) proposed distributed bookkeeping strategies to ensure that information is incorporated in a consistent manner. Each of their approaches requires additional bandwidth or use of acknowledgments. Similarly motivated, Ribeiro et al. (2006) and Nerurkar et al. (2011a) achieve consistency through a noteworthy approach in which they transmit just a single bit per measurement (representing the sign-of-innovations)—yielding an algorithm that closely mirrors the standard Kalman filter. While reducing overall bandwidth, the algorithm requires full packet reception, which is unrealistic for faulty communication channels.

The most general cooperative localization algorithms estimate the full joint distribution over all vehicle poses termed the pose-graph (Fig. 2.2), and can generally be realized through centralized estimators in post-process or high-bandwidth systems in real-time (Howard et al., 2002; Dellaert et al., 2003). Roumeliotis and Bekey (2002) developed a distributed EKF-based method, though it requires moderately high-bandwidth, and two-way information exchange. Cunningham et al. (2010, 2012) and Kim et al. (2010) studied the problem of nonlinear SLAM in a distributed fashion using SAM where each platform *(i)* transmits its full local pose-graph (or a representative subset), *(ii)* collects the local pose-graphs from neighboring platforms, and *(iii)* estimates the full distribution by optimizing over all available graphs. The result is a consistent estimate that matches the centralized estimator solution at the expense of high communication cost, which grows with the size of the local graph. Leung et al. (2010) exploits the Markov property to reduce the required information exchange within a recursive Bayesian filter.

Webster et al. (2012) presented a post-process centralized EKF for synchronous-clock acoustic cooperative localization. They later distributed this centralized filter result exactly (Webster et al., 2013) with the DEIF by leveraging the sparse update properties of the delayed-state information filter. Their solution requires a strict server-client support topology, as the server transmits representative local information (the ‘delta state’ information) to the client where the centralized filter solution is reproduced. Bailey et al. (2011) independently developed an equivalent formulation for sharing locally obtained information, relying on fusion centers to perform relative robot measurement updates. The fusion centers increase complexity, but allow for arbitrary communication topologies. In practice, both of these methods are not practical in the underwater scenario because they require a non-faulty communication channel. We previously reported (Walls and Eustice, 2012) a preliminary method toward alleviating the non-faulty communication constraint in distributing local server information, but which still relied upon a client acknowledgment scheme—limiting scalability to multiple clients.

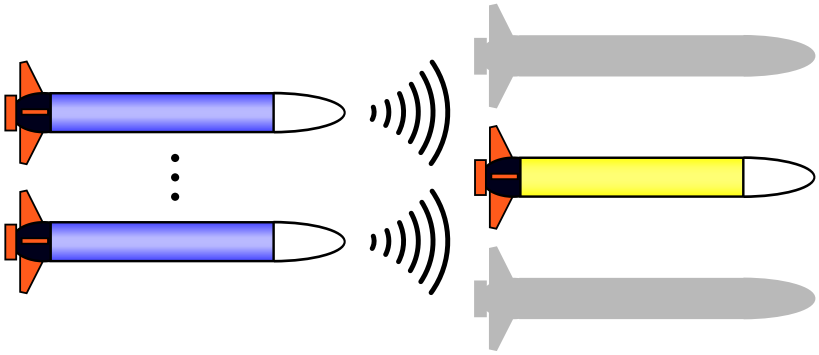
Several other works in acoustic cooperative underwater navigation have emerged for fusing OWTT-based relative ranges in server-client communication topologies (Vaganay et al., 2004; Maczka et al., 2007; McPhail and Pebody, 2009; Bahr et al., 2009a,b; Eustice et al., 2011; Fallon et al., 2010b, 2011). Previously considered 'single-beacon' cooperative localization or the use of CNAs are equivalent to the server-client scenario explored here. Earlier methods, however, generally compromise between offline and consistent (by estimating the full joint distribution), or real-time and inconsistent (by ignoring correlation between relative range observations). Note that online methods that neglect correlation are, in fact, exact when the server positions are actually independent. While many of these algorithms have been validated in post-process using experimental data, only a few have been presented with real-time field trials including Maczka et al. (2007), Fallon et al. (2010b), and Fallon et al. (2010a).

Our work is closest to Webster et al. (2013) and Bailey et al. (2011) in its effort to distribute local data fusion and to leverage the sparsity of the Gaussian information form to compactly broadcast this information. The approach reported herein improves upon previous methods by *(i)* improving network scalability via a generally *passive* origin shifting scheme that eliminates the need for client-side acknowledgments, *(ii)* introducing a recovery packet mechanism that enables clients to enter and leave the network or recover after a long period of communication dropout, and *(iii)* presenting several AUV trials demonstrating our algorithm's ability for real-time underwater navigation.

2.2 Consistent Cooperative Localization

We consider one or more independent server vehicles aiding the navigation of multiple client vehicles, such as is depicted in Fig. 2.3. For the sake of presentation, we refer to a single server vehicle throughout this section, although our algorithm can support multiple (the multiple server scenario is demonstrated in Section 2.4). The client vehicles are able to passively observe their range to the server vehicle during periodic server broadcasts. Each client updates its pose estimate using its local information, the range observations to the servers, and the information broadcast by each server. A centralized estimator, for example Webster et al. (2012), which has access to the

Figure 2.3 Supported communication topology. One or more server vehicles (left, blue) broadcast state information to a network of client vehicles (right, yellow). The OSM algorithm allows each client to reproduce the corresponding centralized estimate. Servers can support many clients in parallel.

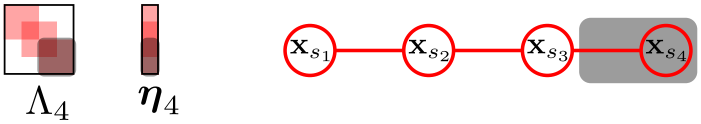


local and relative observations of all vehicles, but is realizable in post-process only, serves as the gold-standard benchmark solution. The centralized solution includes information from all servers and a single client vehicle, but not information from the other client vehicles. Our formulation is able to reproduce this centralized delayed-state filter result onboard each client vehicle in real-time for the server and individual client states.

Relative range observations occur between the server at the time-of-launch (TOL) and each client at the time-of-arrival (TOA) by measuring the one-way-travel-time (OWTT) of an acoustic broadcast. All vehicles synchronize their local clocks to GPS time while at the surface. Low-drift reference clocks enable the vehicles to remain synchronized throughout operation. During our trials we used a SeaScan Inc. temperature compensated crystal oscillator (TXCO), which provides a stable reference pulse with approximately 1 ms drift over 14 h (Eustice et al., 2011). Newer commercially available free-running clocks promise several orders of magnitude improved performance, for example, Symmetricom (2013) provides a 120 mW chip scale atomic clock with less than 1 ms drift over 5000 h ($\sim$$208$ days). Appendix C details the synchronous-clock and acoustic modem hardware.

Previously reported algorithms that are able to compute a consistent estimate track the joint distribution over both client *and* server poses, i.e., the joint pose-graph over the client and server (see Fig. 2.2). Although the client may only be interested in computing its own state estimate, tracking the server's pose-graph allows the client to track correlation between successive relative range observations. In general, the server must transmit its full local pose-graph to the client at the time of each new relative measurement in order for the client to compute this full solution. Since the size of the server pose-graph continually grows, transmitting the full server information is not feasible in a communication constrained domain. OSM supplies a fixed-bandwidth representation of new server poses that allows each client to asynchronously reconstruct the server pose-graph despite high packet loss. Each server broadcast contains all new local information relative to a

Figure 2.4 Server pose-graph example. The gray-shaded area of the information matrix illustrates that process predictions and measurements occurring between the previous TOL, $\mathbf{x}_{s_3}$, and the current TOL, $\mathbf{x}_{s_4}$, contribute additively to a subblock of the information matrix. The information matrix sparsity pattern corresponds to the adjacency matrix of the MRF. The information shown here exactly corresponds to the server information in Fig. 2.2.



server state known by the client, termed the origin state. The client then reconstructs the server pose-graph and can compute the centralized solution. Fig. 2.1 provides a graphical overview of the OSM algorithm.

2.2.1 Information Filter

The OSM algorithm relies upon manipulating a Gaussian distribution parameterized in the information form to efficiently broadcast the server pose-graph, an MRF. We assume that the server state evolves with linear Gaussian noise models. Client process and measurement models, however, can be fully nonlinear. The server MRF will therefore have a known and predictable structure that we can leverage to broadcast it over an unreliable communication channel. Although we may use any estimator that satisfies our assumptions, we employ a delayed-state information filter to initially construct the server pose-graph.

The information filter tracks a Gaussian distribution over its state, $\boldsymbol{x}$, parametrized in the information form; that is $p(\boldsymbol{x}) = \mathcal{N}^{-1}\big(\boldsymbol{x}; \boldsymbol{\eta}, \Lambda\big)$, where the information vector, $\boldsymbol{\eta}$, and matrix, Λ, are related to the mean and covariance by

$$\boldsymbol{\eta} = \Lambda\boldsymbol{\mu} \text{ and } \Lambda = \Sigma^{-1}, \tag{2.1}$$

where $\boldsymbol{\mu}$ and Σ are the mean vector and covariance matrix of $\boldsymbol{x}$, respectively.

The single vehicle navigation problem is framed in terms of estimating the joint distribution over a collection of historic poses (i.e., past vehicle states). In this case, the state vector is composed of these historic poses, termed ‘delayed-states’,

$$\boldsymbol{x} = \Big[\mathbf{x}_1^\top, \ldots, \mathbf{x}_{n-1}^\top, \mathbf{x}_n^\top\Big]^\top,$$

corresponding to the distribution $p(\boldsymbol{x})$, i.e., the pose-graph. The information filter state vector grows over time by performing prediction with augmentation. As noted in Eustice et al. (2006a), processes that evolve sequentially with the Markov property, i.e., with distributions that can be factored

$$p(\boldsymbol{x}) = p(\mathbf{x}_n|\mathbf{x}_{n-1}) \cdots p(\mathbf{x}_2|\mathbf{x}_1)p(\mathbf{x}_1),$$

admit a sparse, block tri-diagonal information matrix. This sparsity leads to an update formulation that only affects a small sub-block of the information matrix and vector, as depicted in Fig. 2.4. New odometry inputs and local measurements only modify a block of the information matrix and vector corresponding to the current robot pose and the most recent delayed-state (i.e., only the value of the pose-graph edge between the last delayed-state and the current state is affected). Herein, we assume that new poses are appended onto the state vector. Below we review the filter mechanics originally presented by Eustice et al. (2006a) that lead to these sparse updates.

Process Prediction

Consider a state vector that is composed of a delayed-state and the current vehicle state, $\boldsymbol{x}_n = [\mathbf{x}_d^\top, \mathbf{x}_n^\top]^\top$, with distribution $p(\boldsymbol{x}_n|\mathbf{Z}_n)$ parameterized by the information matrix and vector

$$\Lambda_n = \begin{bmatrix} \Lambda_{d,d} & \Lambda_{d,n} \\ \Lambda_{n,d} & \Lambda_{n,n} \end{bmatrix}$$
$$\boldsymbol{\eta}_n = \begin{bmatrix} \boldsymbol{\eta}_d \\ \boldsymbol{\eta}_n \end{bmatrix},$$

where $\mathbf{Z}_n$ represents the set of observations received up to time index n. The process prediction computes the distribution $p(\boldsymbol{x}_{n+1}|\mathbf{Z}_n)$ given a state transition model

$$\mathbf{x}_{n+1} = f(\mathbf{x}_n, \mathbf{u}_n) + \mathbf{w}_n,$$

where $\mathbf{u}_n$ represents a control input and $\mathbf{w}_n \sim \mathcal{N}\big(0, \mathrm{Q}\big)$ is an independent noise disturbance. The information matrix and vector are propagated

$$\bar{\Lambda}_{n+1} = \begin{bmatrix} \Lambda_{d,d} - \Lambda_{d,n}\Omega^{-1}\Lambda_{n,d} & \Lambda_{d,n}\Omega^{-1}\mathrm{F}^\top\mathrm{Q}^{-1} \\ \mathrm{Q}^{-1}\mathrm{F}\Omega^{-1}\Lambda_{n,d} & \Psi \end{bmatrix}$$
$$\bar{\boldsymbol{\eta}}_{n+1} = \begin{bmatrix} \boldsymbol{\eta}_d - \Lambda_{d,n}\Omega^{-1}\boldsymbol{\eta}^\star \\ \mathrm{Q}^{-1}\mathrm{F}\Omega^{-1}\boldsymbol{\eta}_n - \Psi(f(\boldsymbol{\mu}_n, \mathbf{u}_n) - \mathrm{F}\boldsymbol{\mu}_n) \end{bmatrix},$$

where, for notational convenience,

$$\begin{aligned}\Omega &= \Lambda_{n,n} + \mathrm{F}^\top \mathrm{Q}^{-1}\mathrm{F}\\ \Psi &= (\mathrm{Q} + \mathrm{F}\Lambda_{n,n}^{-1}\mathrm{F}^\top)^{-1}\\ \boldsymbol{\eta}^\star &= \boldsymbol{\eta}_n - \mathrm{F}^\top \mathrm{Q}^{-1}(f(\boldsymbol{\mu}_n, \mathbf{u}_n) - \mathrm{F}\boldsymbol{\mu}_n).\end{aligned}$$

State augmentation appends the newest vehicle state to the state vector while also maintaining the previous state as a delayed-state. In this case, prediction with augmentation simultaneously appends the new state and predicts forward

$$\begin{aligned}\bar{\Lambda}_{n+1} &= \underbrace{\begin{bmatrix}\Lambda_{d,d} & \Lambda_{d,n} & 0\\ \Lambda_{n,d} & \Lambda_{n,n} & 0\\ 0 & 0 & 0\end{bmatrix}}_{\Lambda_n} + \begin{bmatrix}0 & 0 & 0\\ 0 & \mathrm{F}^\top\mathrm{Q}^{-1}\mathrm{F} & -\mathrm{F}^\top\mathrm{Q}^{-1}\\ 0 & -\mathrm{Q}^{-1}\mathrm{F} & \mathrm{Q}^{-1}\end{bmatrix}\\ \bar{\boldsymbol{\eta}}_{n+1} &= \underbrace{\begin{bmatrix}\boldsymbol{\eta}_d\\ \boldsymbol{\eta}_n\\ 0\end{bmatrix}}_{\boldsymbol{\eta}_n} + \begin{bmatrix}0\\ \mathrm{F}^\top\mathrm{Q}^{-1}(f(\boldsymbol{\mu}_n) - \mathrm{F}\boldsymbol{\mu}_n)\\ -\mathrm{Q}^{-1}(f(\boldsymbol{\mu}_n) - \mathrm{F}\boldsymbol{\mu}_n)\end{bmatrix}.\end{aligned}$$

We see here that the information update is sparse, as it only affects the lower block corresponding to the previous and newest states. As mentioned earlier, the cumulative effect is that the information admits a sparse tri-diagonal structure as more delayed-states are added.

Measurement Update

Consider the general nonlinear observation model

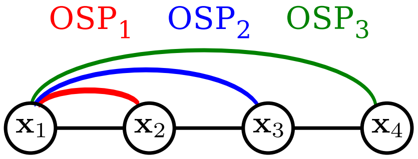
$$\mathbf{z}_{n+1} = h(\mathbf{x}_{n+1}) + \mathbf{v}_{n+1},$$

where $\mathbf{v}_{n+1} \sim \mathcal{N}\big(0, \mathrm{R}\big)$ is an independent noise perturbation. The observation model represents the conditional distribution $p(\mathbf{z}_{n+1}|\mathbf{x}_{n+1})$. We can then compute the distribution over state, $p(\boldsymbol{x}_{n+1}|\mathbf{Z}_{n+1})$, as a simple additive update

$$\begin{aligned}\Lambda_{n+1} &= \bar{\Lambda}_{n+1} + \mathrm{H}^\top\mathrm{R}^{-1}\mathrm{H}\\ \boldsymbol{\eta}_{n+1} &= \bar{\boldsymbol{\eta}}_{n+1} + \mathrm{H}^\top\mathrm{R}^{-1}(\mathbf{z}_{n+1} - h(\bar{\boldsymbol{\mu}}_{n+1}) + \mathrm{H}\bar{\boldsymbol{\mu}}_{n+1}),\end{aligned}$$

where H is the observation Jacobian and $\bar{\boldsymbol{\mu}}_{n+1}$ is the state mean following process prediction. The observation model generally only involves a portion of the state and, therefore, the observation

Figure 2.5 Pose-graph over server TOL poses. The OSM algorithm allows the client to reconstruct the underlying server pose-graph (horizontal lines) from the set of received OSPs (colored arcs). Each OSP encodes the transition from $\mathbf{x}_1$, which represents the origin state.



Jacobian H is sparse. The information update term is defined by the outer product, $\mathrm{H}^\top \mathrm{R}^{-1}\mathrm{H}$, which is then also sparse and only touches blocks of the information matrix corresponding to nonzero elements of the measurement Jacobian.

2.2.2 Origin State Method

Our goal is to broadcast information that allows the client to reconstruct the server pose-graph. However, the server does not have any knowledge of the information that the client has actually received, because communication is broadcast and unacknowledged. The underlying assumption is that the server pose-graph grows as a Markov chain—the standard model for a dynamical system. The Markov chain is a special case of the more general MRF. Loop closures, popular in SLAM literature, are not supported by this model since including them in the pose-graph breaks this chain assumption. Each origin state broadcast, called an origin state packet (OSP), encodes a server transition from the origin state to the current state (Fig. 2.5). The OSP represents the relationship between the origin and current state as their joint marginal distribution, i.e., the two-node pose-graph over the origin and current TOL states. We show here that the client can incrementally reconstruct the server's pose-graph from the sequence of received OSPs.

Server-side Origin State Operation

The server vehicle maintains an information filter, augmenting its state vector with a copy of each TOL state. At the TOL, the server broadcasts an OSP containing the joint marginal information over the current TOL state and a designated previous delayed-state, the origin, as depicted in Fig. 2.5. The index label of the new TOL state and the origin are included in the OSP for reconstruction by the client.

We can partition the set of intermediate server TOL states occurring between the origin state, $\mathbf{x}_o$, and the nth TOL state, $\mathbf{x}_n$, into the set of states received and not received by the client, $\boldsymbol{x}_r$ and $\boldsymbol{x}_{\bar{r}}$, respectively. The server pose-graph at the nth TOL then represents the distribution over the state vector

$$\boldsymbol{x}^s_n = \left[\mathbf{x}_o^\top, \mathbf{x}_{r\cup\bar{r}}^\top, \mathbf{x}_n^\top\right]^\top, \tag{2.2}$$

where the 's' superscript indicates the distribution as tracked by the server. Note that the server has no knowledge of the partition r and $\bar{r}$. This distribution is expressed

$$p^s(\mathbf{x}_o, \mathbf{x}_{r\cup\bar{r}}, \mathbf{x}_n | \mathbf{Z}_n) = \mathcal{N}^{-1}\big(\boldsymbol{x}_n^s; \boldsymbol{\eta}_n^s, \Lambda_n^s\big),$$

$$\Lambda_n^s = \begin{bmatrix} \Lambda_{o,o} & \Lambda_{o,r\cup\bar{r}} & 0 \\ \Lambda_{r\cup\bar{r},o} & \Lambda_{r\cup\bar{r},r\cup\bar{r}} & \Lambda_{r\cup\bar{r},n} \\ 0 & \Lambda_{n,r\cup\bar{r}} & \Lambda_{n,n} \end{bmatrix}, \tag{2.3}$$

$$\boldsymbol{\eta}_n^s = \begin{bmatrix} \boldsymbol{\eta}_o \\ \boldsymbol{\eta}_{r\cup\bar{r}} \\ \boldsymbol{\eta}_n \end{bmatrix}, \tag{2.4}$$

where $\mathbf{Z}_i$ is the set of all observations up to the ith time index. Note that Λ_n^s is block tri-diagonal as per the Markov chain assumption.

Under the OSM, the server computes an OSP at every TOL, which is simply the joint marginal distribution corresponding to the state vector

$$\boldsymbol{x}_{\text{OSP}_n}^s = \Big[\mathbf{x}_o^\top, \mathbf{x}_n^\top\Big]^\top, \tag{2.5}$$

computed via the Schur complement of (2.3) and (2.4):

$$\begin{aligned} p_{\text{OSP}_n}^s(\mathbf{x}_o, \mathbf{x}_n | \mathbf{Z}_n) &= \int_{\mathbf{x}_{r\cup\bar{r}}} p^s(\mathbf{x}_o, \mathbf{x}_{r\cup\bar{r}}, \mathbf{x}_n | \mathbf{Z}_n) d\mathbf{x}_{r\cup\bar{r}} \\ &= \mathcal{N}^{-1}\big(\boldsymbol{x}_{\text{OSP}_n}^s; \boldsymbol{\eta}_{\text{OSP}_n}^s, \Lambda_{\text{OSP}_n}^s\big), \end{aligned}$$

$$\Lambda_{\text{OSP}_n}^s = \begin{bmatrix} \Lambda_{o,o}^s & \Lambda_{o,n}^s \\ \Lambda_{n,o}^s & \Lambda_{n,n}^s \end{bmatrix}, \tag{2.6}$$

$$\boldsymbol{\eta}_{\text{OSP}_n}^s = \begin{bmatrix} \boldsymbol{\eta}_o^s \\ \boldsymbol{\eta}_n^s \end{bmatrix}. \tag{2.7}$$

This formulation allows the server to remain ignorant about states the client has received, r. Moreover, it allows the server to send useful information to multiple clients, where each client has a different set of received server TOL states. In order for the client to reconstruct the server pose-graph, the client must already have the origin state in its representation, (i.e., the origin is a previously received TOL state). The index label of the current TOL, n, and the origin, o_n, are included in the broadcast.

Algorithm 1 Server-side Origin State Method

```
Require: $\Lambda_0, \boldsymbol{\eta}_0$ {initial server belief}
$\Lambda_b, \boldsymbol{\eta}_b, o_b \leftarrow 0$ {backup origin state packet}
loop
    if $k$ is $\mathrm{TOL}_n$ then
        $\Lambda^s_n, \eta^s_n, o_n \leftarrow$ originPacket$(\Lambda_k, \boldsymbol{\eta}_k)$
        if $o_n \neq o_{n-1}$ then
            {origin has been shifted, update backup packet}
            $\Lambda_b, \boldsymbol{\eta}_b, o_b \leftarrow \Lambda^s_{n-1}, \eta^s_{n-1}, o_{n-1}$
        end if
        broadcastOriginPacket$(\Lambda^s_n, \eta^s_n, o_n, \Lambda_b, \boldsymbol{\eta}_b, o_b)$
        if recoveryRequired() then
            broadcastRecovery() {Section 2.2.2}
        end if
        $\Lambda_k, \eta_k \leftarrow$ predictAugment$(\Lambda_k, \boldsymbol{\eta}_k)$
        $n \leftarrow n + 1$
    else
        $\Lambda_k, \boldsymbol{\eta}_k \leftarrow$ predict$(\Lambda_k, \boldsymbol{\eta}_k)$
    end if
    $\Lambda_k, \boldsymbol{\eta}_k \leftarrow$ localMeasUpdate$(\Lambda_k, \boldsymbol{\eta}_k, \mathbf{z}_k)$
    $k \leftarrow k + 1$
end loop
```

Algorithm 1 summarizes the server-side operation. The server simply maintains an information filter over its pose-graph, augmenting its state vector with each new TOL pose. At each TOL the server computes an OSP to broadcast to the client. Origin shifting and recovery is introduced and discussed below.

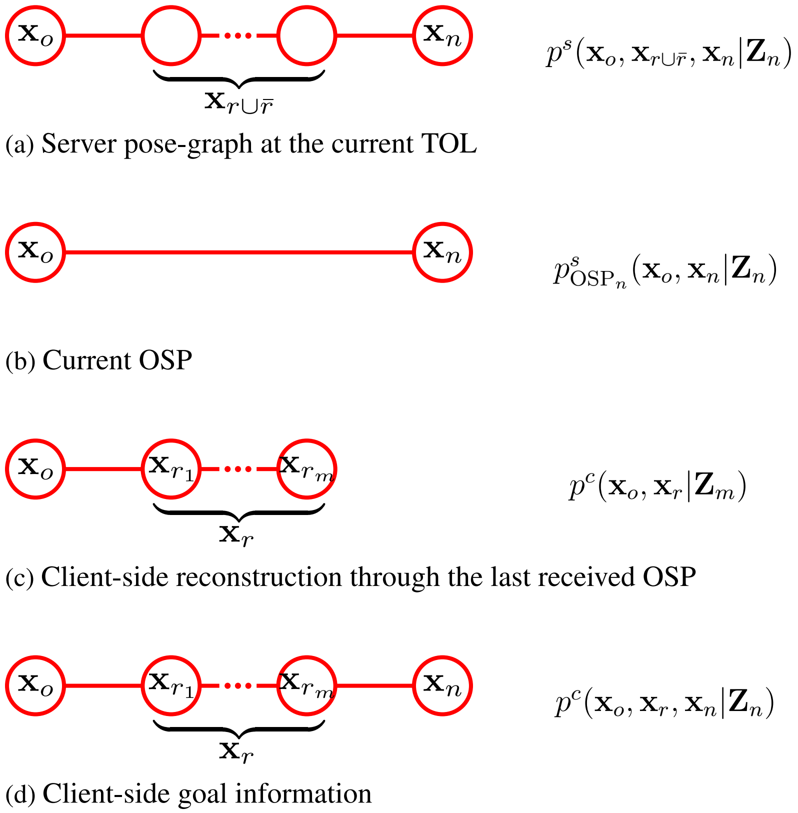
Although, in general, the size of the server pose-graph grows with the addition of each new TOL pose, the dimension of the OSP marginal information matrix and vector is fixed. The OSP dimension is twice the state dimension—therefore, a minimal vehicle state size is desirable to reduce communication packet size.

Client-side Origin State Operation

The client incrementally reconstructs the server pose-graph from the sequence of successfully received OSPs. Before the nth TOL, the client-side version of the server pose-graph reconstruction contains the server states

$$\begin{aligned}\boldsymbol{x}^c_{r_m} &= \left[\mathbf{x}_o^\top, \mathbf{x}_r^\top\right]^\top \\ &= \left[\mathbf{x}_o^\top, \mathbf{x}_{r'}^\top, \mathbf{x}_{r_m}^\top\right]^\top,\end{aligned}$$

Figure 2.6 Illustration of OSM operation. (a) represents the server's pose-graph at the nth TOL. The server broadcasts an OSP (b) to the client. The client has already reconstructed a portion of the server graph, (c), having missed TOL poses, $\mathbf{x}_{\bar{r}}$. The client then reconstructs the server goal information illustrated in (d) by fusion of (b) and (c). (a) and (d) are equivalent with the unreceived TOL states, $\mathbf{x}_{\bar{r}}$ marginalized out.


$\mathbf{x}_o$
$\mathbf{x}_n$
$\mathbf{x}_{r\cup\bar{r}}$
$p^s(\mathbf{x}_o, \mathbf{x}_{r\cup\bar{r}}, \mathbf{x}_n|\mathbf{Z}_n)$
$\mathbf{x}_o$
$\mathbf{x}_n$
$p^s_{\mathrm{OSP}_n}(\mathbf{x}_o, \mathbf{x}_n|\mathbf{Z}_n)$
$\mathbf{x}_o$
$\mathbf{x}_{r_1}$
$\mathbf{x}_{r_m}$
$\mathbf{x}_r$
$p^c(\mathbf{x}_o, \mathbf{x}_r|\mathbf{Z}_m)$
$\mathbf{x}_o$
$\mathbf{x}_{r_1}$
$\mathbf{x}_{r_m}$
$\mathbf{x}_n$
$\mathbf{x}_r$
$p^c(\mathbf{x}_o, \mathbf{x}_r, \mathbf{x}_n|\mathbf{Z}_n)$

(a) Server pose-graph at the current TOL

(b) Current OSP

(c) Client-side reconstruction through the last received OSP

(d) Client-side goal information

where the 'c' superscript indicates the server distribution as tracked by the client, $r = \{r_1, \ldots, r_m\}$ denotes the set of received server TOL indices, r_m is the latest received TOL index, and $r' = \{r_1, \ldots, r_{m-1}\}$ represents other previously received server TOL indices. To simplify notation, we let $m = r_m$ for the remainder of the discussion. Note that the client has no representation for server states corresponding to missed TOL poses, $\mathbf{x}_{\bar{r}}$. This is equivalent to the server's distribution at m with states in $\mathbf{x}_{\bar{r}}$ marginalized out,

$$p^c(\mathbf{x}_o, \mathbf{x}_{r'}, \mathbf{x}_m|\mathbf{Z}_m) = \mathcal{N}^{-1}\big(\boldsymbol{x}^c_m; \boldsymbol{\eta}^c_m, \Lambda^c_m\big),$$

$$\Lambda_m^c = \begin{bmatrix} \Lambda_{o,o} & \Lambda_{o,r'} & 0 \\ \Lambda_{r',o} & \Lambda_{r',r'} & \Lambda_{r',m} \\ 0 & \Lambda_{m,r'} & \Lambda_{m,m} \end{bmatrix}, \tag{2.8}$$

$$\boldsymbol{\eta}_m^c = \begin{bmatrix} \boldsymbol{\eta}_o \\ \boldsymbol{\eta}_{r'} \\ \boldsymbol{\eta}_m \end{bmatrix}. \tag{2.9}$$

After receiving the nth OSP, the client solves the following problem:

Given the information available to the client,

1. $p^c(\mathbf{x}_o, \mathbf{x}_r | \mathbf{Z}_m)$, the client-side server pose-graph up to the last received TOL and
2. $p^s(\mathbf{x}_o, \mathbf{x}_n | \mathbf{Z}_n)$, the new OSP (computed server-side),

Construct the client goal distribution, i.e., the server pose-graph up to time n, as if unreceived TOL states had been marginalized out,

$$p^c(\mathbf{x}_o, \mathbf{x}_r, \mathbf{x}_n | \mathbf{Z}_n) = \mathcal{N}^{-1}\big(\boldsymbol{x}_n^c; \boldsymbol{\eta}_n^c, \Lambda_n^c\big),$$

$$\Lambda_n^c = \begin{bmatrix} \Lambda_{o,o} & \Lambda_{o,r'} & 0 & 0 \\ \Lambda_{r',o} & \Lambda_{r',r'} & \Lambda_{r',m} & 0 \\ 0 & \Lambda_{m,r'} & \boxed{\Lambda_{m,m}} & \boxed{\Lambda_{m,n}} \\ 0 & 0 & \boxed{\Lambda_{n,m}} & \boxed{\Lambda_{n,n}} \end{bmatrix}, \tag{2.10}$$

$$\boldsymbol{\eta}_n^c = \begin{bmatrix} \boldsymbol{\eta}_o \\ \boldsymbol{\eta}_{r'} \\ \boxed{\boldsymbol{\eta}_m} \\ \boxed{\boldsymbol{\eta}_n} \end{bmatrix}. \tag{2.11}$$

The boxed elements in (2.10) and (2.11) indicate unknown values in the desired goal reconstruction while the remaining values are known from (2.8) and (2.9) because of the assumed Markov structure for the server states (block tri-diagonal information matrix). In other words, all new information since the *last received* OSP only affects a small portion of the information matrix and vector. The client-side reconstruction is illustrated in Fig. 2.6.

The client-side reconstruction begins by marginalizing out $\mathbf{x}_r$ from the (partially unknown) goal distribution, (2.10) and (2.11), producing an expression that can be equated to the (known) received nth OSP, (2.6) and (2.7),

$$p^c_{\mathrm{OSP}_n}(\mathbf{x}_o, \mathbf{x}_n|\mathbf{Z}_n) = \int_{\mathbf{x}_r} p^c(\mathbf{x}_o, \mathbf{x}_r, \mathbf{x}_n|\mathbf{Z}_n) d\mathbf{x}_r \tag{2.12}$$

$$\equiv p^s_{\mathrm{OSP}_n}(\mathbf{x}_o, \mathbf{x}_n|\mathbf{Z}_n). \tag{2.13}$$

Marginalization of the goal distribution via the Schur complement leads to a set of linear equations in the unknowns. We proceed with a two-step marginalization procedure. First, we marginalize out r' states from (2.10) and (2.11) via the Schur complement:

$$\begin{bmatrix} \tilde{\Lambda}_{o,o} & \tilde{\Lambda}_{o,m} & 0 \\ \tilde{\Lambda}_{m,o} & \tilde{\Lambda}_{m,m} & \Lambda_{m,n} \\ 0 & \Lambda_{n,m} & \Lambda_{n,n} \end{bmatrix} = \begin{bmatrix} \Lambda_{o,o} & 0 & 0 \\ 0 & \Lambda_{m,m} & \Lambda_{n,m} \\ 0 & \Lambda_{n,m} & \Lambda_{n,n} \end{bmatrix} - \begin{bmatrix} \Lambda_{o,r'} \\ \Lambda_{m,r'} \\ 0 \end{bmatrix} \Lambda^{-1}_{r',r'} \begin{bmatrix} \Lambda_{r',o} & \Lambda_{r',m} & 0 \end{bmatrix} \tag{2.14}$$

$$\begin{bmatrix} \tilde{\boldsymbol{\eta}}_o \\ \tilde{\boldsymbol{\eta}}_m \\ \boldsymbol{\eta}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{\eta}_o \\ \boldsymbol{\eta}_m \\ \boldsymbol{\eta}_n \end{bmatrix} - \begin{bmatrix} \Lambda_{o,r'} \\ \Lambda_{m,r'} \\ 0 \end{bmatrix} \Lambda^{-1}_{r',r'} \boldsymbol{\eta}_{r'}, \tag{2.15}$$

where the tilde decoration indicates block elements that are changed from (2.10) and (2.11). This step results in the following set of expressions,

$$\begin{aligned} \tilde{\Lambda}_{o,o} &= \Lambda_{o,o} - \Lambda_{o,r'}\Lambda^{-1}_{r',r'}\Lambda_{r',o}\,, \\ \tilde{\Lambda}_{m,m} &= \Lambda_{m,m} - \Lambda_{m,r'}\Lambda^{-1}_{r',r'}\Lambda_{r',m}\,, \\ \tilde{\Lambda}_{o,m} &= -\Lambda_{o,r'}\Lambda^{-1}_{r',r'}\Lambda_{r',m}\,, \\ &= \tilde{\Lambda}^{\top}_{m,o}\,, \\ \tilde{\boldsymbol{\eta}}_o &= \boldsymbol{\eta}_o - \Lambda_{o,r'}\Lambda^{-1}_{r',r'}\boldsymbol{\eta}_{r'}\,, \\ \tilde{\boldsymbol{\eta}}_m &= \boldsymbol{\eta}_m - \Lambda_{m,r'}\Lambda^{-1}_{r',r'}\boldsymbol{\eta}_{r'}\,. \end{aligned} \tag{2.16}$$

Second, we marginalize out state m from (2.14) and (2.15) and equate to the server-side OSP, (2.6) and (2.7),

$$\begin{bmatrix} \Lambda^s_{o,o} & \Lambda^s_{o,n} \\ \Lambda^s_{n,o} & \Lambda^s_{n,n} \end{bmatrix} = \begin{bmatrix} \tilde{\Lambda}_{o,o} & 0 \\ 0 & \Lambda_{n,n} \end{bmatrix} - \begin{bmatrix} \tilde{\Lambda}_{o,m} \\ \Lambda_{n,m} \end{bmatrix} \tilde{\Lambda}^{-1}_{m,m} \begin{bmatrix} \tilde{\Lambda}_{m,o} & \Lambda_{m,n} \end{bmatrix} \tag{2.17}$$

$$\begin{bmatrix} \boldsymbol{\eta}^s_o \\ \boldsymbol{\eta}^s_n \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{\eta}}_o \\ \boldsymbol{\eta}_n \end{bmatrix} - \begin{bmatrix} \tilde{\Lambda}_{o,m} \\ \Lambda_{n,m} \end{bmatrix} \tilde{\Lambda}^{-1}_{m,m}\tilde{\boldsymbol{\eta}}_m. \tag{2.18}$$

The unknown values of the goal client-side server reconstruction, boxed in (2.10) and (2.11), are

Algorithm 2 Client-side Origin State Method

```
$\Lambda_0, \boldsymbol{\eta}_0 \leftarrow 0$
**loop**
  **if** $(\Lambda_n^s, \eta_n^s, o_n, \Lambda_b, \eta_b, o_b) \leftarrow$ receivedPacket() **then**
    **if** havePrimaryOriginIndex($o_n$) **then**
      $\Lambda_n, \boldsymbol{\eta}_n \leftarrow$ addOriginPacket($\Lambda_n^s, \eta_n^s, o_n$)
      updateDEIF($\Lambda_n, \boldsymbol{\eta}_n$) {(2.21), (2.22)}
    **else if** haveSecondaryOriginIndex($o_b$) **then**
      $\Lambda_{n-1}, \boldsymbol{\eta}_{n-1} \leftarrow$ addOriginPacket($\Lambda_b, \eta_b, o_b$)
      $\Lambda_n, \boldsymbol{\eta}_n \leftarrow$ addOriginPacket($\Lambda_n^s, \eta_n^s, o_n$)
      updateDEIF($\Lambda_n, \boldsymbol{\eta}_n$) {(2.21), (2.22)}
    **else**
      requestRecovery()
    **end if**
  **end if**
**end loop**
```

then computed by substituting (2.16) into the above equality

$$
\begin{aligned}
\tilde{\Lambda}_{m,m} &= \tilde{\Lambda}_{m,o}\left(\tilde{\Lambda}_{o,o} - \Lambda^s_{o,o}\right)^{-1}\tilde{\Lambda}_{o,m} \,, \\
\Lambda_{m,m} &= \tilde{\Lambda}_{m,m} + \Lambda_{m,r'}\Lambda^{-1}_{r',r'}\Lambda_{r',m} \,, \\
\Lambda_{m,n} &= -\tilde{\Lambda}_{m,m}\tilde{\Lambda}^{-1}_{o,m}\Lambda^s_{o,n} \,, \\
\Lambda_{n,n} &= \Lambda^s_{n,n} + \Lambda_{n,m}\tilde{\Lambda}^{-1}_{m,m}\Lambda_{m,n} \,, \\
\tilde{\boldsymbol{\eta}}_m &= -\tilde{\Lambda}_{m,m}\tilde{\Lambda}^{-1}_{o,m}\left(\tilde{\boldsymbol{\eta}}_o - \boldsymbol{\eta}^s_o\right) \,, \\
\boldsymbol{\eta}_m &= \tilde{\boldsymbol{\eta}}_m + \Lambda_{m,r'}\Lambda^{-1}_{r',r'}\boldsymbol{\eta}_{r'} \,, \\
\boldsymbol{\eta}_n &= \boldsymbol{\eta}^s_n + \Lambda_{n,m}\Lambda^{-1}_{m,m}\tilde{\boldsymbol{\eta}}_m \,.
\end{aligned}
\tag{2.19}
$$

When only one TOL state has been received (i.e., $m = 1$, $r = \{r_1\}$, and $r' = \{\emptyset\}$), the derivation proceeds as above, but with only the second marginalization step.

As an implementation aside, at the TOA of the first received OSP, the client does not need to perform any computation to reconstruct the server pose-graph. The initial OSP is simply the two-node server pose-graph consisting of the server origin state and the current TOL state. (Note that this allows any new clients to immediately enter and join the network, i.e., the network can dynamically resize).

Origin Shifting

The information difference $\left(\tilde{\Lambda}_{o,o} - \Lambda^s_{o,o}\right)$ in (2.19) represents the delta information known about the origin state between the client through the last received OSP, $p^c(\mathbf{x}_o|\mathbf{x}_r, \mathbf{Z}_m)$, and the server at the current TOL, $p^s(\mathbf{x}_o|\mathbf{x}_n, \mathbf{Z}_n)$. This difference approaches machine precision as the time between the origin and new TOL state grows (because little additional smoothing of the origin state occurs after sufficient time). The reconstruction rules require the inversion of this decreasing term, leading to numerical inaccuracies that can cause divergent errors in the reconstruction. A simple solution is to ensure that the origin is periodically shifted forward.

We initially proposed an origin shifting scheme based on acknowledgments from each client (Walls and Eustice, 2012); however, an acknowledgment based scheme does not scale well to many clients. Moreover, numerical instability will continue to plague an acknowledgement driven system if the server does not regularly receive acknowledgments. Instead, we propose a shifting scheme in which the server evaluates a function based upon the numerical stability of the newest OSP—keeping the OSP broadcast *passive* and not requiring any client acknowledgements, such that the method can more easily scale to many client vehicles.

During our real-time experiments (Section 2.4), the server shifted the origin forward using a threshold, T, on the trace of the difference term in (2.19),

$$\mathrm{trace}\big(\tilde{\Lambda}_{o,o} - \Lambda^{s}_{o,o}\big) < \mathrm{T}. \tag{2.20}$$

The trace is only used as a measure to test the numerical stability of the OSP and is tuned to produce an accurate reconstruction. Note that $\tilde{\Lambda}_{o,o}$ is the $\Lambda^{s}_{o,o}$ element from the previous OSP, and is therefore readily available without additional computation.

When the shifting function suggests shifting the origin, the new origin is set to the previous TOL state. The server is now free to marginalize out TOL states preceding the new origin. To help ensure that each client vehicle can maintain a reconstruction of the server pose-graph that contains the origin state, in practice each server broadcast encodes at least two OSPs: the primary OSP encoding the transition from the origin to the current TOL, and a secondary OSP encoding the transition from the previous origin to the current origin. Depending upon the available bandwidth, the server could extend the broadcast to include more than two OSPs to increase tolerance to packet loss. From an implementation standpoint, the server does not need to recompute previous OSPs, it simply stores previously broadcast messages. The server-side origin shifting step and corresponding client behavior are outlined in Algorithm 1 and Algorithm 2, respectively.

After the server shifts the origin forward, the client-side reconstruction will contain server TOL states preceding the new origin. The reconstruction rules (2.19) are unmodified if the client first marginalizes out these earlier server states.

Recovery Packet

Passively shifting the origin limits the robustness of the OSM algorithm. The server can no longer guarantee that the client has received the origin TOL state (or the previous origin state, as described above). If the client vehicle has not received an update in a sufficiently long period of time, it will require a special information packet in order to recover (i.e., reconstruct the current server pose-graph given the state it has already received). Once the client identifies that it has lagged behind, it transmits a recovery request to the server containing the last received TOL index. After receiving a client request, the server computes this special information as an additive ‘delta information’ (discussed in Section 2.3) from the last TOL state that the client has received up to the current origin state. One implementation detail here is that now the server must not marginalize out the oldest TOL states from its pose-graph in order to compute a recovery packet, unless it can guarantee that each client has received a more recent TOL. The full client-side operation is summarized in Algorithm 2.

Recovery requests could limit the scalability of the algorithm because each client vehicle requires a slot in the TDMA schedule to transmit. In our field trials (Section 2.4), recoveries were rarely necessary, so that only a small fraction of the TDMA was reserved.

2.3 Online Distributed Estimation

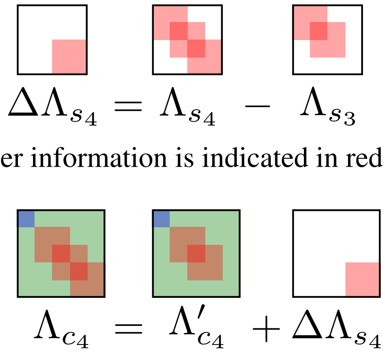
We demonstrate that the incremental reconstruction of the server pose-graph within the OSM framework can be used onboard the client to exactly reproduce the centralized solution to the multiple vehicle localization problem. We couple the OSM algorithm with the DEIF algorithm update (Webster et al., 2013) to compute the client-side state estimate following OWTT range observations. The DEIF algorithm is a method in which a client vehicle can exactly reproduce the centralized delayed-state filter solution for server-client cooperative networks. Essentially, the DEIF provides an efficient way to incorporate the newest server information in a delayed-state framework. The DEIF, as originally proposed, is not real-time practical since it relies on an unrealistic communication assumption (perfect packet reception) to build the server information—this is remedied by the OSM representation. The full operation of the OSM and DEIF is illustrated in Fig. 2.1.

To review the DEIF, the server vehicle maintains an information filter to fuse its local measurements, augmenting its state vector with each TOL position. As with the OSM, the server state evolves as a Markov process. Each ‘delta information’ encompasses all the local information that the server has gained between TOLs, computed as

$$\begin{aligned} \Delta\Lambda_{s_n} &= \Lambda_{s_n} - \Lambda_{s_{n-1}}, \\ \Delta\boldsymbol{\eta}_{s_n} &= \boldsymbol{\eta}_{s_n} - \boldsymbol{\eta}_{s_{n-1}}, \end{aligned} \tag{2.21}$$

where the operation conforms for the dimensionality difference and the ‘s’ subscript indicates the

Figure 2.7 Illustration of the DEIF update step. The client adds the server delta information to exactly reconstruct the centralized information.


$\Delta\Lambda_{s_4} = \Lambda_{s_4} - \Lambda_{s_3}$
$\Lambda_{c_4} = \Lambda'_{c_4} + \Delta\Lambda_{s_4}$

(a) Server-side DEIF operation. Server information is indicated in red.

(b) Client-side DEIF operation. Local client information and range information are shown in blue and green, respectively.

server's information. The delta information is illustrated in Fig. 2.4 and Fig. 2.7. Delta information packets can be conceptually considered as expressing a transition on the server pose-graph from the previous TOL state to the current TOL state. The DEIF design insightfully noted that the delta information updates computed by the server are equivalent to the delta information updates within the CEIF.

The client-side DEIF is driven by its local measurement updates and periodic (assumed non-lossy) delta information packets from the server vehicle, which the fault-tolerant OSM algorithm provides. The client-side DEIF tracks the current client state in addition to the set of server TOL states

$$\boldsymbol{x}_n = \left[\mathbf{x}_{c_n}^\top, \mathbf{x}_{s_1}^\top, \ldots, \mathbf{x}_{s_n}^\top\right]^\top,$$

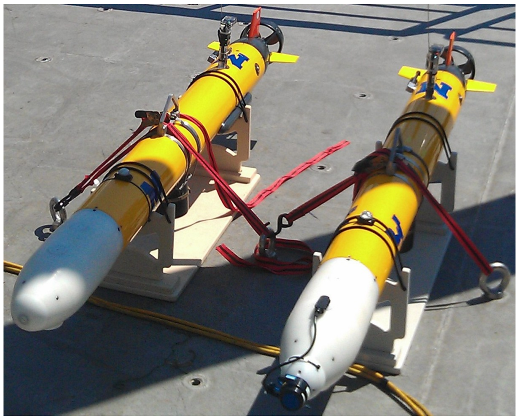
where the 'c' subscript indicates the client state. Upon packet reception, the client vehicle simply adds the delta information into its information filter

$$\begin{aligned} \Lambda_{c_n} &= \Lambda'_{c_n} + \Delta\Lambda_{s_n}, \\ \boldsymbol{\eta}_{c_n} &= \boldsymbol{\eta}'_{c_n} + \Delta\boldsymbol{\eta}_{s_n}, \end{aligned} \tag{2.22}$$

where information includes both server TOL states and the current client state. Following the subsequent relative range measurement update, the client-side filter matches the corresponding centralized filter exactly up to communication delay. The delta information addition is illustrated in Fig. 2.7. The client is not required to maintain the full set of server TOL poses in its state vector. Full details of the algorithm as well as extensive comparitve results are provided in Webster et al. (2013).

The DEIF requires that the client receive each delta information (2.21) broadcast by the server in order to reconstruct the server pose-graph. We use the OSM to reconstruct the server pose-graph

Figure 2.8 Ocean-Server, Inc. Iver2 AUVs used in the field experiments.



onboard the client. The client vehicle is then able compute the delta information to the DEIF, and therefore locally reconstruct the CEIF.

2.4 Field Trials

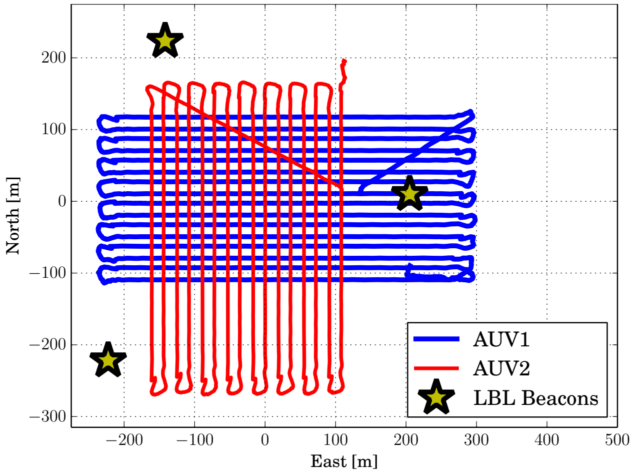
Seven field trials spanning more than $12\ \mathrm{h}$ of operation with a single server and two client vehicles were performed over August 18–20, 2013, at the University of Michigan Biological Station (UMBS). These trials demonstrate the OSM algorithm's ability to incrementally broadcast and reconstruct the server pose-graph, compute the centralized solution, and fuse range-only constraints in a multiple vehicle framework. In addition to the single server-client topology, we provide post-process results demonstrating a two-server support network.

2.4.1 Experimental Setup

We fielded two Ocean-Server, Inc. Iver2 AUVs, designated AUV1 and AUV2, (Fig. 2.8). Each AUV is outfitted with an advanced dead-reckoning sensor suite outlined in Appendix C. Throughout our experiments, AUV1 acts as the server, aiding AUV2, which is considered to be the client. AUV1 is the only vehicle that observes and fuses GPS when at the surface. To demonstrate the ability of our OSM algorithm to support multiple client vehicles, we also consider a topside support ship (with only GPS reported velocity and heading for input) as a client vehicle. All vehicles were outfitted with a Woods Hole Oceanographic Institution (WHOI) Micro-modem and co-processor board capable of encoding multiple frame, higher bandwidth phase-shift keying (PSK) data packets (Freitag et al., 2005a,b).

For baseline comparision of our OWTT navigation, we deployed a three-beacon $25\ \mathrm{kHz}$ LBL network to independently measure underwater vehicle position. Fig. 2.9 depicts the LBL beacon

Figure 2.9 Experiment B setup with three LBL beacon locations.



locations used during our field trials, which were positioned to provide good triangulation observability over the client vehicle survey area (the beacon locations were moored and held fixed for all experiments). Each vehicle interrogated the LBL network roughly twice per minute, resulting in a maximum of six range constraints per minute. We recorded two-way LBL, along with GPS position fixes at the surface, for all vehicles for ground-truth comparison (none of the client vehicles used these measurements during the real-time experiments).

2.4.2 Vehicle State Description

Since AUV attitude and depth are both instrumented with small bounded error, we focus on world-frame XY horizontal position estimation. By broadcasting pressure depth with each acoustic packet, OWTT $3\mathrm{D}$ range measurements, $z_{3\mathrm{D}}$, can be projected into the horizontal plane, z_r, according to the method outlined in Appendix B. Moreover, we are motivated to maintain a minimal state size because of the limited acoustic channel capacity.

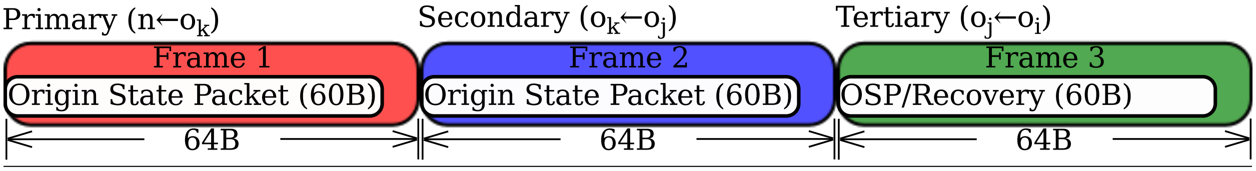
The state estimator on each vehicle tracks a state vector composed of its horizontal position, $\mathbf{x}_k = [x_k, y_k]^\top$. The state is time-propagated using an odometry-driven plant model,

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{u}_{k+1},$$

where $\mathbf{u}_{k+1}$ is the delta odometry measurement. The odometry input and corresponding input covariance, Q_{k+1}, are obtained by Euler integrating DVL and attitude heading reference system (AHRS) measurements and performing a first-order covariance estimate as described in Appendix B. In the case of the topside surface craft, world-frame velocity is integrated from GPS reported speed and track direction.

For the server vehicle, GPS reported x, y observations at vehicle surfacings are treated as linear

Figure 2.10 Acoustic message composition. Each PSK Rate 1 and Rate 2 Micro-modem message contains three 64-byte frames. We use the first two frames to hold two origin state packets and the third frame is reserved for recovery packets, or an additional origin state packet if no recovery has been requested.



observations of state. OWTT measurements, z_r, provide a range between the server TOL position and the client TOA position, with nonlinear observation model:

$$z_r = \left\| \mathbf{x}_{s_{\text{TOL}}} - \mathbf{x}_{c_{\text{TOA}}} \right\|_2 + v,$$

where $v \sim \mathcal{N}(0, \sigma_r^2)$ represents the range measurement noise. In our experiments, we used $\sigma_r = 30\ \mathrm{cm}$, to account for noise in both TOF observation and depth.

2.4.3 Acoustic Communication Considerations

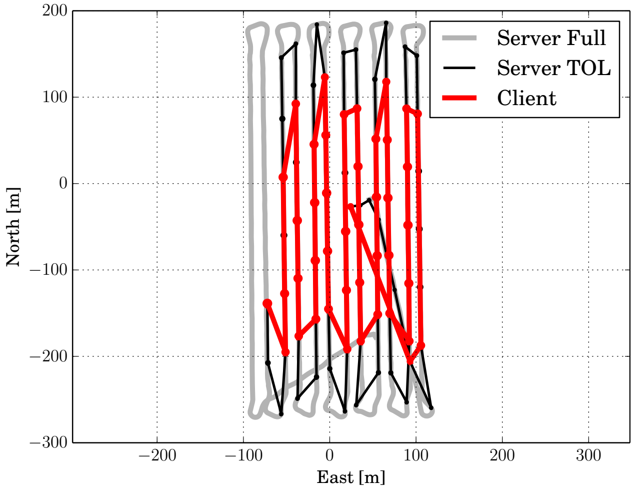
Each origin state packet requires $60\ \mathrm{bytes}$ to encode. Each double precision element of the origin state information is rounded to a precision of 10^{-5} to reduce the packet size. Moreover, since the information matrix is symmetric, only the upper diagonal elements are transmitted. The full message specification is detailed in Appendix C. Both Micro-modem PSK Rate 1 and Rate 2 messages allow the user to broadcast three 64 byte frames (Fig. 2.10). We fill the first two frames of Rate 1 and Rate 2 messages with the primary and secondary OSPs as discussed in Section 2.2.2. If a client vehicle has requested a recovery packet, we transmit the custom recovery packet in the remaining third frame, so that normal operation continues for vehicles that do not require a recovery step, otherwise we broadcast a tertiary OSP. Acoustic packets were encoded using dynamic compact control language (DCCL) (Schneider and Schmidt, 2010a) and transmitted using the Goby-acomms library (Schneider and Schmidt, 2012).

We employed a fixed TDMA cycle, whereby all vehicles were assigned a communication time slot. The server vehicle broadcast an OSP roughly once per minute, while the client transmitted a single data packet over the same time window, used to monitor vehicle health state and to place

Table 2.2 Acoustic statistics across Experiments A–G

	A	B	C	D	E	F	G
Server:AUV1 broadcast bps	25.6	25.1	22.4	23.0	23.0	22.8	23.2
Client:AUV2 Reception rate	47.9%	43.2%	32.9%	32.1%	53.0%	34.4%	33.7%
Client:Topside Reception rate	46.1%	55.9%	62.6%	57.1%	60.2%	50.4%	39.9%

Figure 2.11 Pose-graph reconstruction example from Experiment G. Light gray trajectory depicts the server pose-graph over all poses, while the black pose-graph represents server TOL states. The thick red line represents the client-side reconstructed server pose-graph (as if missed server TOL poses had been marginalized out). The two pose-graphs are equal up to communication round-off errors.

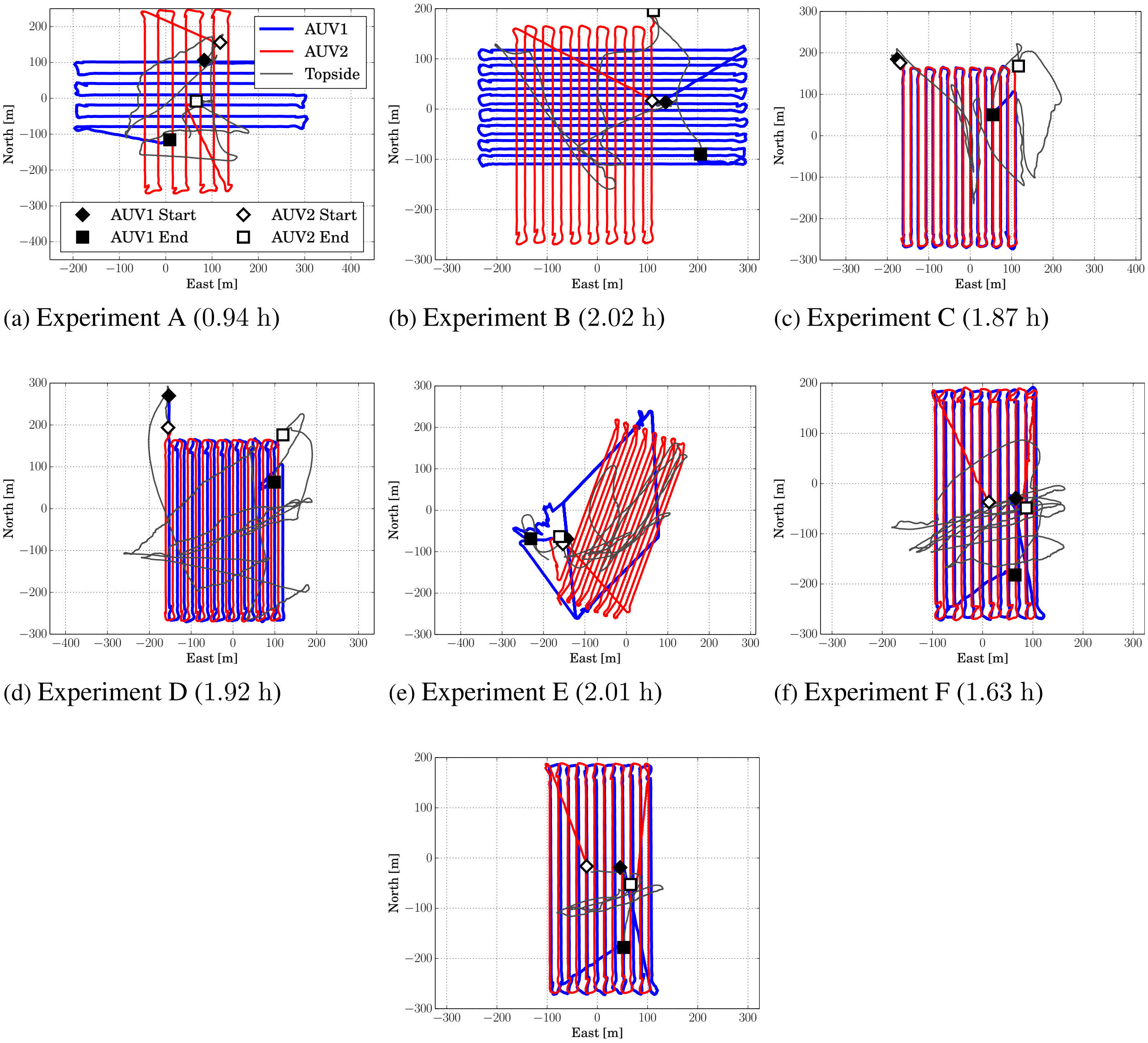


recovery requests when needed. As noted in Table 2.2, the average server throughput used for navigation data was ∼25 bps. The average client-side navigation packet reception rates across experiments varied between 32% and 63%. Our origin state method allowed each client to reliably reconstruct the server pose-graph despite the small bandwidth allotment and low reception rates.

2.4.4 Results

Fig. 2.12 summarizes the relative vehicle trajectories over the seven individual field trials. The relative server-client geometries between AUV1 and AUV2 were purposely varied between the different experiments while the topside surface craft had no control and drifted around the survey site, occasionally motoring to stay within the site boundary. During Experiments A and B, the server and client swam on orthogonal lawn-mower trajectories, Experiments C and D, the server and client swam along the same lawn-mower trajectory with the server following at a fixed distance, Experiment E, the server encircled the client via a bounding diamond-shaped trajectory, while during Experiments F and G, the server-client swam along the same lawn-mower trajectory, beginning at different boundaries of the survey area. As seen in Table 2.2, the different relative geometries and conditions led to varied communication reception performance ranging from 32–63% throughput. During the course of our experiments, each vehicle swam at fixed depth with AUV1 holding a depth of 8 m, AUV2 at 10 m (apart from prescribed surface intervals for GPS ground-truth), and the topside transducer was suspended at 10 m depth.

Figure 2.12 Summary of the seven field trials used for experimental evaluation. (a)–(g) Topdown view of the 3-node vehicle trajectories with total trial time indicated in each subcaption. AUV2 and Topside acted as clients while AUV1 performed the server role. (a) AUV1 shifted the origin forward at an accelerated rate in order to artificially induce recovery requests. (b) AUV1 and AUV2 followed orthogonal lawnmower surveys. (c),(d) AUV1 followed AUV2 along the same lawnmower survey. (e) AUV1 maintained a diamond box path bounding AUV2's lawnmower survey. (f),(g) AUV1 and AUV2 followed similar lawnmower surveys, beginning from opposite ends of the survey area, i.e., AUV1 began on the East boundary while AUV2 began on the West boundary.



(a) Experiment A (0.94 h)

(b) Experiment B (2.02 h)

(c) Experiment C (1.87 h)

(d) Experiment D (1.92 h)

(e) Experiment E (2.01 h)

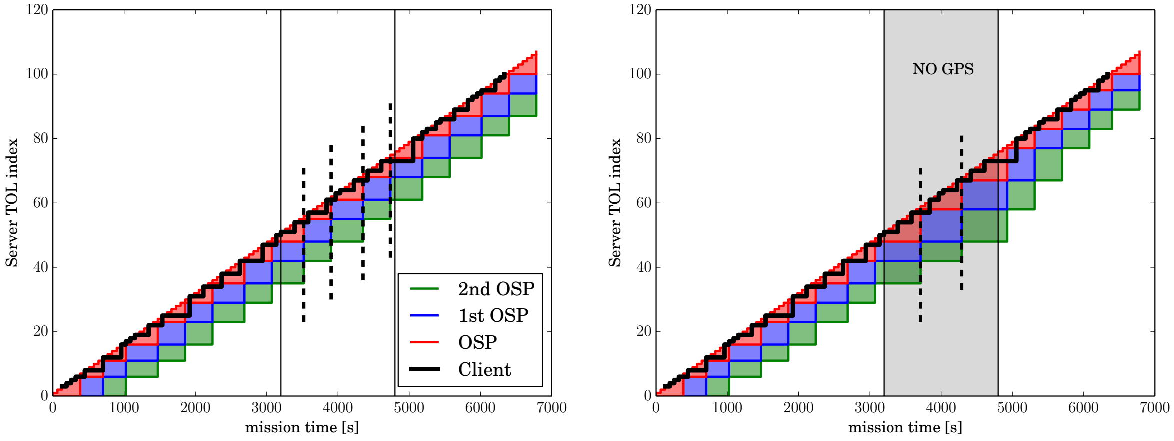
(f) Experiment F (1.63 h)

(g) Experiment G (1.65 h)

Client-side Server Pose-graph Reconstruction

The client was able to accurately reconstruct the server pose-graph using the OSM framework throughout all of our field trials. Fig. 2.11 illustrates an example 'true' server pose-graph with the client-side reconstruction overlaid. Note that the client-side reconstruction does not contain all of

Figure 2.13 Server OSP TOL indices and the corresponding index received by the client during Experiment D. The shaded red/green/blue regions represent the OSPs broadcast by the server (color coded according to Fig. 2.10), while the thick black line plots the client's latest received server index. Large steps in the server indices indicate an origin shift. (a) illustrates the real-time server origin shifting, while (b) demonstrates how the OSM algorithm adapts to varying conditions, waiting longer before shifting the origin when GPS is cut out. Note that the real-time server shifted the origin four times during the same region whereas the post-process origin only shifted twice (indicated by dashed lines).



(a) GPS available during surface intervals

(b) No GPS during shaded region

the server TOL poses. The client's version of the server pose-graph, however, is equivalent to the server's as if the TOL states corresponding to dropped messages had been marginalized out. As shown in Fig. 2.14a, each client is able to reconstruct the server pose-graph with small error, on average to the order of 10^{-5} m. Moreover, this error is attributed to the communication round-off of the OSP (using the origin-shifting scheme with full precision OSPs in post-process, both mean and max reconstruction error is on the order of 10^{-12} m).

Server Origin Shifting

During Experiment A, the server shifted the origin forward at an artificially increased rate (every other TOL) in order to force a client recovery request. During this trial, AUV2 and Topside received one and three recovery packets, respectively, after transmitting requests to the server. These packets were successfully integrated into the server pose-graph reconstruction.

Throughout the remaining 11 h of field trials showcasing normal operation, neither client required a recovery packet. Moreover, during normal operation (Experiments B–G), both clients used a total of 554 Frame 1 OSPs, 62 Frame 2 OSPs, and only 2 Frame 3 OSPs. The server used a trace threshold $T = 5 \cdot 10^{-2}$.

The client will require a secondary (Frame 2) OSP at most once per server origin shift, because the secondary packet 'catches' the client up to the current origin. We expect the client to occasionally require a secondary packet following an origin shift because of the high likelihood of missing any single transmission. However, the client would need to lose communication with the server over an entire period between origin shifts in order to require the tertiary (Frame 3) OSP. In order to require a recovery packet, communication would have to drop out over at least two origin shift periods, depending on the number of broadcast redundant OSPs.

The rate at which the server will shift the origin (based on the trace metric in (2.20)) depends on the server noise models and available measurements. Absolute position observations, e.g., GPS, and noisy process models will reduce correlation between the current state and the origin state, so that subsequent measurements will not influence the origin as much. Therefore, after receiving absolute position observations or sufficient time given a noisy process model, the server will be forced to shift the origin. Fig. 2.13 illustrates the server's origin shifting during Experiment D. The server surfaced at regular intervals, receiving several GPS observations. In post-process, we cut out GPS measurements over a nearly $30\ \mathrm{min}$ window. In this case, the server shifted the origin forward less frequently (twice as opposed to four times as seen in Fig. 2.13b). This demonstrates the ability of the OSM algorithm to automatically adapt origin shifting to varying measurement availability.

Client-side DEIF

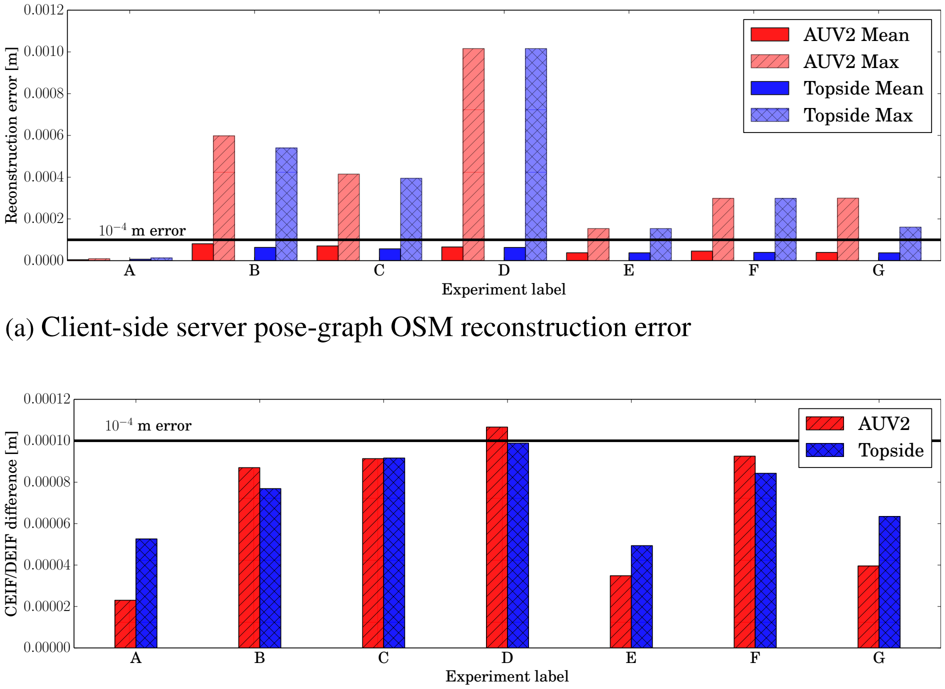
Each client employed a DEIF to integrate server and client information with relative range observations. The state estimate of the post-process CEIF agrees with our real-time DEIF with differences commensurate with those reported by Webster et al. (2013) (on the order of $10^{-5}\ \mathrm{m}$) and on par with the errors observed in the server pose-graph reconstruction (see Fig. 2.14b). The two estimates are equal at the TOA of each OSP. The estimates may vary in between TOAs because the CEIF is able to incorporate server information the instant it is received, while the DEIF must wait to incorporate server information until the OSP is received (i.e., up to communication delay).

Multiple Server Implementation

It is a relatively simple extension to move from a single to multiple independent server implementation as is depicted in Fig. 2.3. In this case, the client vehicle reconstructs the local pose-graph from each server and fuses relative range constraints. Range constraints are still measured by the client to each server. The client-side DEIF reproduces the centralized filter (up to communication delay). Communication delay will cause the linearization point used for range observations to differ between the client-based and centralized results. This is because one (or both) of the servers will have made observations that the client has not yet received.

We tested the two server scenario in post-process using the field collected sensor data from Experiment C. In this trial, the Topside vehicle acted as the client with AUV1 and AUV2 acting

Figure 2.14 Summarized reconstruction and estimation error across all seven field trials (A–G). (a) The server pose-graph reconstruction error is computed as the norm of the difference between server TOL poses received on the client and their actual value computed on the server. The maximum error for any single pose remains below 1 mm for all trials, while the mean error is on the order 10^{-5} m. (b) Difference in the CEIF versus the DEIF client trajectory estimate at the TOA. Each bar represents the mean norm difference between state estimates for each estimator. Note that the client-side DEIF is able to reproduce the centralized CEIF estimate in real-time to high accuracy.



(a) Client-side server pose-graph OSM reconstruction error

(b) Client-side decentralized estimate (DEIF) versus centralized benchmark (CEIF)

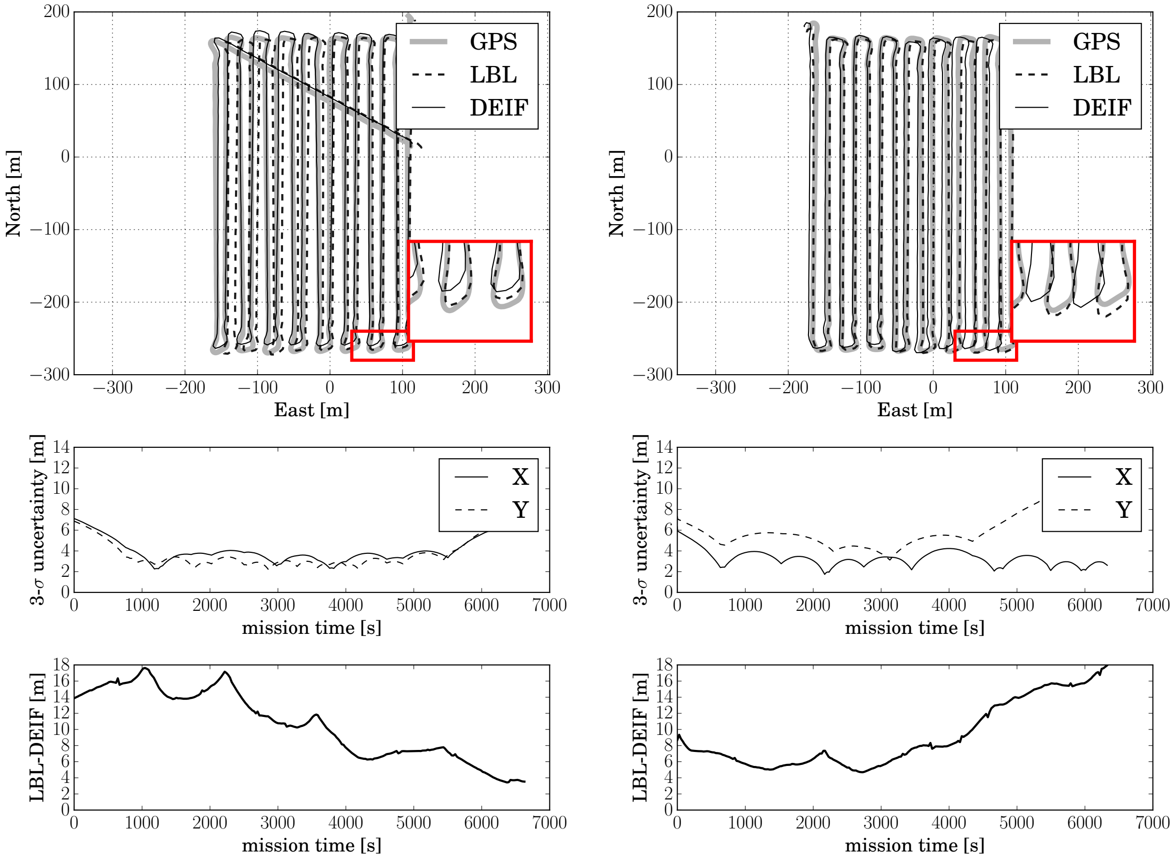
as servers. The Topside-based DEIF solution differs with the corresponding CEIF on average by 4.5 cm. We can attribute this difference primarily to a difference in range observation linearization point.

Performance Baseline

OWTT navigation allows AUVs to navigate subsea for extended periods of time with bounded error. Bounded error navigation is usually achieved with the use of networks of stationary acoustic beacons, e.g., LBL or USBL. The cost of deploying, surveying, and later recovering an LBL system is high with respect to time, however. Here, we discuss the ability of OWTT relative ranging frameworks to approach the accuracy of LBL systems without the operational and equipment overhead. The accuracy of the post-process CEIF has previously been extensively compared to an LBL navigation solution by Webster et al. (2012).

Fig. 2.15 compares both LBL and OWTT based DEIF navigation solutions to a baseline trajectory computed by fusing GPS and odometry measurements. Since each range observation only

Figure 2.15 OWTT and LBL navigation comparison to a GPS baseline. OWTT compares well with LBL in (a), while OWTT has a larger error in (b) (outlined in red in the lower right corner). The server/client relative geometries (Fig. 2.12), however, greatly influence this result. The middle axis in each subfigure plots the x, y uncertainty estimated by the DEIF, showing that in (b) y, East, uncertainty is not as well bounded. Finally, the lower axis shows the norm difference between OWTT and LBL solutions.



(a) Experiment B: good server-client geometry

(b) Experiment C: poor server-client geometry

adds information in a single direction, the relative geometry between server and client is of utmost importance. Experiment B (Fig. 2.15a) intuitively has an informative relative geometry as the server continually crosses over the clients path, helping to bound error in both x, y directions. Experiment C (Fig. 2.15b), however, does not have such a useful relative geometry, as the server largely remains behind the client along the North–South direction. Indeed, we see that the server is able to well bound uncertainty along the East–West direction during Experiment B, but not as well in Experiment C. Moreover, due to an informative relative server trajectory, the client DEIF closely reproduces the full LBL solution in Experiment B. This is further evidenced by the norm difference between LBL and DEIF solutions (Fig. 2.15), which illustrates that the difference between solutions decreases in Experiment B but increases with mission time in Experiment C.

Each client vehicle received at most one OWTT relative range constraint over each one minute TDMA window, while potentially receiving six LBL ranges over the same period. Using OWTT navigation, however, we can achieve bounded error on the same order as LBL. This solution is only enhanced by selecting appropriate server relative geometries. Advantageously, our method also scales to many client vehicles, whereas the update rate for narrowband LBL decreases with new clients. Optimal relative positioning of server vehicles has received recent attention (Tan et al., 2014; Bahr et al., 2012), and is the subject of Chapter 4.

2.5 Discussion

The OSM with the DEIF constitutes a smoothing algorithm built upon a delayed-state information filter. While an extended information filter is sub-optimal for nonlinear estimation, our solution compares well to nonlinear smoothing algorithms such as iSAM (Kaess et al., 2008). The delayed-state information filter is equivalent to a nonlinear least-squares approach with the exception that measurement constraints are only linearized once upon their initialization. This difference is small in our implementation as all odometry constraints are linear, and infrequent range measurements account for the only nonlinearities.

The OSM algorithm reproduces the linear server information up to communication round-off errors onboard the client. The delta information (2.21) computed from the client-side reconstructed pose-graph is a linear factor over the previously received and current server TOL states. This factor contains all information due to observations occurring between the TOLs including process predictions, odometry, and absolute position measurements such as GPS. Fig. 2.16, for example, illustrates a delta factor as summarizing all information that occurs between server TOL poses.

We can compute a factor to represent the delta information using the concept of generic linear constraints (GLCs) proposed by Carlevaris-Bianco and Eustice (2013). First, we compute the

Figure 2.16 Server factor graph example. The above factor graph shows the full server factor graph. Shaded nodes and edges indicate constraints occurring between the previous and current TOL. The lower graph illustrates the reconstructed client-side factor graph. The shaded delta state factor induces an equivalent factor potential as the original server-side factors.

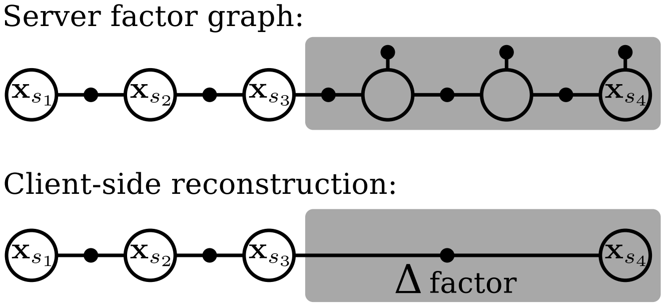
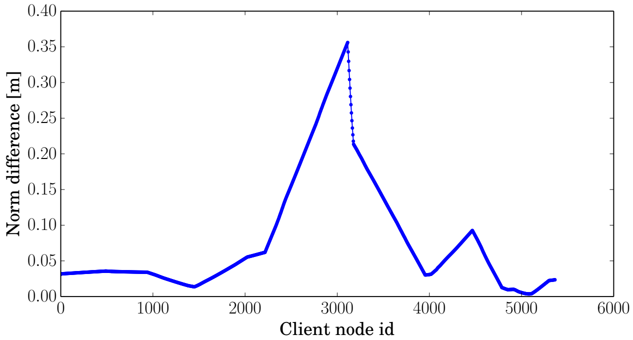


Figure 2.17 Difference in estimated pose between the DEIF algorithm and the nonlinear least-squares iSAM during Experiment F. The mean and maximum differences are 0.079 m and 0.35 m, respectively.



eigen-decomposition of the delta information

$$\Delta\Lambda = \mathrm{UDU}^\top,$$

where $\Delta\Lambda \in \mathbb{R}^{p\times p}$, $q = \mathrm{rank}(\Delta\Lambda)$, $\mathrm{U} \in \mathbb{R}^{p\times q}$, and $\mathrm{D} \in \mathbb{R}^{q\times q}$ is a diagonal matrix. Using the decomposition, we define the delta factor as a Gaussian observation with mean and covariance

$$\mathbf{z}_\Delta = \mathrm{D}^{\frac{1}{2}}\mathrm{U}^\top \boldsymbol{x} \tag{2.23}$$

$$\Sigma_\Delta = \mathrm{I}_{q\times q}, \tag{2.24}$$

where $\boldsymbol{x} = [\mathbf{x}_{s_{n-1}}^\top, \mathbf{x}_{s_n}^\top]^\top$ is the state vector over the previous and current server TOL poses. This factorization (while not unique) produces the same information over the server TOL poses as the additive delta information block.

iSAM is equivalent to the DEIF with the exception that measurement constraints may be relinearized after their initialization. This difference is small in our implementation as all odometry constraints are linear, and infrequent range measurements account for the only nonlinearities.

We implemented the factor representation for delta information, (2.23) and (2.24), to provide server information to the client. Fig. 2.17 plots the difference between the DEIF and iSAM for Experiment F. The small difference in DEIF and iSAM estimates is due to differing linearization points. The ability to relinearize range constraints may have additional benefits over the DEIF when a good linearization point is initially not available, for example, when the client dead-reckoned error has grown too large.

2.6 Chapter Summary

This chapter presents the OSM for server-client cooperative localization over an extremely faulty and bandwidth-limited communication channel. Moreover, the OSM algorithm allows a client vehicle to exactly reproduce the centralized estimate—therefore guaranteeing that the estimate is consistent. We validated this distributed estimation algorithm over more than $12\ \mathrm{h}$ of field trials deploying three vehicles. We demonstrated how our algorithm is adaptable to support both multiple clients and multiple servers, albeit while maintaining strict server-client communication.

Chapter 3

Peer-to-peer Localization

In the previous chapter, we presented a method for server-client cooperative localization that exploited the structure of the underlying Gaussian MRF in order to distribute inference. The OSM algorithm allowed the client vehicle to reconstruct exactly the corresponding centralized estimator. However, OSM requires an unwieldy recovery mechanism because the server pose-graph reconstruction is limited by machine numerical precision. Moreover, OSM requires that the server distribution grows according to a Markov chain. In this chapter, we present an algorithm that is similar in spirit, but exploits the structure of the underlying factor graph[1]. By exploiting properties of the constituent factor types, we derive an algorithm that is completely tolerant to communication failure (i.e., *never* requires a recovery packet) and is extensible to other graph-based factors. Each vehicle reconstructs a consistent approximation to the full centralized estimator. This method, while approximate, more readily scales to larger vehicle networks toward peer-to-peer cooperative localization.

We employ approximate marginalization techniques to enable vehicles to share an informative subset of composed factors. Approximate marginalization has recently become a popular tool within the SLAM community for reducing graph complexity, e.g., (Vial et al., 2011; Carlevaris-Bianco et al., 2014; Huang et al., 2013; Mazuran et al., 2014). In this work, we exploit similar tools in order to efficiently distribute locally obtained information within a cooperative localization framework.

The specific contributions of this chapter include:

- We present a cooperative localization algorithm for communication-limited networks that is completely passive (i.e., does not rely on acknowledgments) and employs a fixed bandwidth data packet.
- We approximate the true locally constructed factor graph using approximate marginalization

[1] Portions of this work appear in (Walls et al., 2015b).

techniques to produce a structure that can be more easily broadcast.

- We provide a comparative evaluation of our algorithm and a performance summary from a three vehicle field deployment including two AUVs and a topside ship.

3.1 Related Work

Within cooperative localization frameworks, teams of communicating vehicles localize each other by sharing local sensor data and observing relative vehicle pose. Early work explored sharing information among the team and performing local data fusion (Kurazume et al., 1994; Fox et al., 2000; Roumeliotis and Bekey, 2002; Howard et al., 2002; Dellaert et al., 2003).

More recent methods proposed framing cooperative localization as a multiple vehicle graph-based SLAM problem (Kim et al., 2010; Cunningham et al., 2013). These methods implicitly handle correlation that develops between vehicle estimates when relative information is shared; our proposed method falls within this category. The centralized estimator for OWTT cooperative localization shares a similar structure to a pose-graph SLAM formulation with known data association.

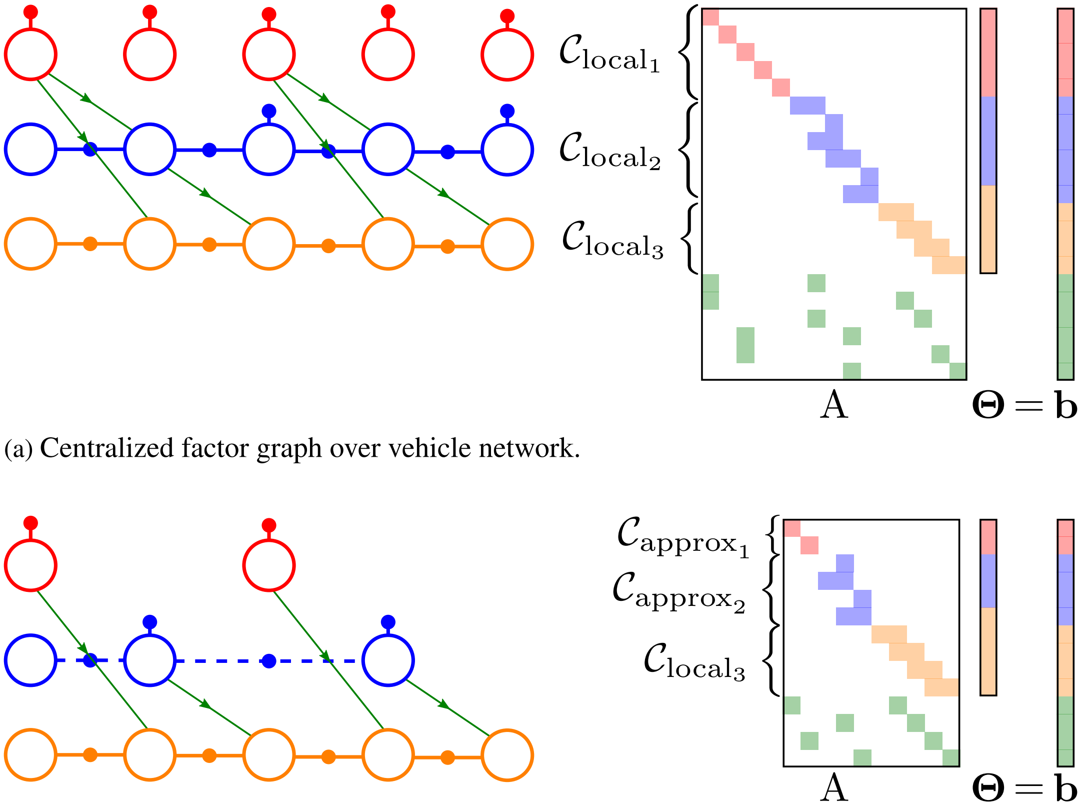
Leung et al. (2010) presented a decentralized cooperative localization algorithm capable of reproducing the centralized estimate in the presence of dropped communication. The bandwidth required for transmitting knowledge sets, however, may exceed the acoustic channel capacity during periods of lost communication connectivity. Nerurkar and Roumeliotis (2013) similarly targeted cooperative localization in bandwidth constrained scenarios, but require constant communication.

Many other researchers in the underwater domain have considered fusing relative range measurements derived from observing the TOF of acoustic broadcasts. Previous work (Vaganay et al., 2004; Maczka et al., 2007; Bahr et al., 2009a,b; Fallon et al., 2010a; Eustice et al., 2011; Webster et al., 2012, 2013; Walls and Eustice, 2014a; Paull et al., 2014) has considered the limitations of the acoustic channel and the difficulties of fusing range-only observations.

The complexity of graph-based estimation is largely dependent on the size (i.e., number of variable nodes) and sparsity of the graph (i.e., number of edges). Graph sparsification methods have been introduced (Vial et al., 2011; Carlevaris-Bianco et al., 2014; Huang et al., 2013; Mazuran et al., 2014) to both reduce the number of variables in the graph and increase the sparsity without greatly affecting the solution. We take advantage of approximate marginalization techniques—used in graph sparsification to reduce the number of variables—to refactor a local factor graph over a smaller set of pose variable nodes to reduce the communication requirement.

The work proposed here is closest to Fallon et al. (2010a) and the OSM presented in Chapter 2. Within OSM, we exploited the structure of the information matrix associated with the underyling

Figure 3.1 Example cooperative localization factor graph—empty circles represent variable pose nodes, solid dots are odometry and prior factors, and arrows illustrate range-only factors and the direction of communication. In this example, red represents a topside ship with access only to GPS, while blue and orange represent AUVs. Take note that the dashed blue factors represent a new set of approximate factors distinct from the original in (a).


$\mathcal{C}_{\text{local}_1}$
$\mathcal{C}_{\text{local}_2}$
$\mathcal{C}_{\text{local}_3}$
A
$\mathbf{\Theta} = \mathbf{b}$
$\mathcal{C}_{\text{approx}_1}$
$\mathcal{C}_{\text{approx}_2}$
$\mathcal{C}_{\text{local}_3}$
A
$\mathbf{\Theta} = \mathbf{b}$

(a) Centralized factor graph over vehicle network.

(b) Factor graph over vehicle network constructed onboard the orange vehicle by our algorithm.

Gaussian MRF to cope with unpredictable packet loss. Unfortunately, the algorithm requires periodically shifting an origin pose, requires a recovery mechanism for pathological communication failures, and does not seamlessly allow for a vehicle with no odometry (e.g., a surface vehicle with only GPS) to participate because of its Markov chain assumption. Fallon et al. (2010a) proposed constructing a graph-based estimator on each vehicle by broadcasting individual factors to the team. Their method transmits each factor over a large set of pose nodes and requires data requests when communication is lost. In contrast, we exploit the factor structure of the local graph to communicate information within a completely passive framework. Paull et al. (2014) independently arrived at a similar solution discussed in Section 3.4 and Section 3.5.

3.2 Problem Statement

We formulate cooperative localization within a factor graph framework (discussed in Appendix A). We show that the underlying structure of the centralized factor graph consists of factors that represent local vehicle observations and factors that represent relative vehicle measurements. We will first review the factor graph formulation for a single vehicle and then expand to the full vehicle network.

In the single vehicle setting, the factor graph approach is a smoothing algorithm that estimates the entire trajectory of the vehicle. Recall that a factor graph is a bipartite graph with pose (variable) nodes and factor (measurement) nodes representing the joint distribution over the unknown poses. The ith vehicle graph represents the joint distribution over its N poses, $\mathbf{X}_i = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$, as

$$p(\mathbf{X}_i|\mathbf{Z}_i) \propto p(\mathbf{x}_1) \prod_i p(\mathbf{z}_{\text{odo}_i}|\mathbf{x}_i, \mathbf{x}_{i-1}) \prod_j p(\mathbf{z}_{\text{prior}_j}|\mathbf{x}_j), \tag{3.1}$$

where $\mathbf{Z}_i$ represents the set of observations and we assume each vehicle has access to its initial belief $p(\mathbf{x}_1)$. The graph structure is a chain as we only consider unary 'prior' factors, $\mathbf{z}_{\text{prior}}$, (e.g., GPS) and pairwise sequential 'odometry' factors, $\mathbf{z}_{\text{odo}}$, (e.g., integrated velocity). We assume that pose variable nodes represent rigid-body position and orientation. An odometry factor is the observed rigid-body transformation between two poses.

For convenience, we define a 'link', $\mathcal{L}_i = (\mathbf{z}_{\text{odo}_i}, \mathbf{z}_{\text{prior}_i})$, associated with the ith pose node, $\mathbf{x}_i$, as a 2-tuple containing the odometry factor to the previous pose node and a prior factor. Note that each link need not have both an odometry and prior factor, for example the initial link only contains a prior factor. The local chain is the set of links that represent the vehicle trajectory corresponding to (3.1), $\mathcal{C}_{\text{local}} = \{\mathcal{L}_i\}_{i=1}^N$. In Fig. 3.1a, each uniformly colored subgraph represents a local chain.

We can construct the posterior distribution over the entire M vehicle network (i.e., all vehicle poses), $\{\mathbf{X}_1, \ldots, \mathbf{X}_M\}$,

$$p(\mathbf{X}_1, \ldots, \mathbf{X}_M|\mathbf{Z}_1, \ldots, \mathbf{Z}_M, \mathbf{Z}_r) \propto \prod_{i=1}^M \underbrace{p(\mathbf{X}_i|\mathbf{Z}_i)}_{\mathcal{C}_{\text{local}_i}} \prod_k \underbrace{p(\mathbf{z}_k|\mathbf{x}_{i_k}, \mathbf{x}_{j_k})}_{\text{relative factors}}, \tag{3.2}$$

where $\mathbf{Z}_r$ is the set of relative vehicle constraints and each $\mathbf{z}_k$ represents a constraint between poses on vehicles i_k and j_k. In this chapter, $\mathbf{z}_k$ is a OWTT range constraint between a transmitting vehicle's TOL pose and a receiving vehicle's TOA pose. The factor graph for a three vehicle network is illustrated in Fig. 3.1a. The centralized MAP estimate for each vehicle is obtained

$$\begin{aligned}\mathbf{X}_i^* &= \underset{\{\mathbf{X}_1,\ldots,\mathbf{X}_M\}}{\operatorname{argmax}} \; p(\mathbf{X}_1,\ldots,\mathbf{X}_M|\mathbf{Z}_1,\ldots,\mathbf{Z}_M,\mathbf{Z}_r) \\ &= \underset{\{\mathbf{X}_1,\ldots,\mathbf{X}_M\}}{\operatorname{argmin}} \; -\log p(\mathbf{X}_1,\ldots,\mathbf{X}_M|\mathbf{Z}_1,\ldots,\mathbf{Z}_M,\mathbf{Z}_r), \end{aligned} \tag{3.3}$$

which results in a nonlinear least-squares problem for Gaussian noise models (Dellaert and Kaess, 2006).

The full joint distribution (3.2) can be factored as a product of each vehicle's local chain and the relative vehicle factors. Therefore, in order to construct (and perform inference on) the full factor graph, the ith vehicle must have access to the set of local factors from all other vehicles, $\{\mathcal{C}_{\text{local}_j}\}_{j\neq i}$, and the set of all relative vehicle factors. Sharing this information, however, is nontrivial due to the limitations of the acoustic communication channel.

In this chapter, we present an algorithm for each vehicle to communicate an *approximation* of its local chain. In turn, each vehicle in the network can construct a graph consisting of its local chain, the set of approximate chains received from other vehicles, and the set of relative vehicle measurements that it observes locally, i.e., range constraints for which it has measured the OWTT (a subset of all ranges) as in Fig. 3.1b. We will show that our approximation produces an accurate representation that allows communication over a faulty and bandwidth-limited channel, and that using a subset of relative vehicle measurements still provides significant improvement over dead-reckoned navigation.

3.3 Cooperative Localization

The key requirement for cooperative localization is the ability to share locally obtained observations, contained here within a local chain. Below, we develop an algorithm for broadcasting a local chain across a faulty low-bandwidth communication channel. Our approach is completely passive, in other words, the information each vehicle broadcasts is independent of the rest of the network—a desirable property for an unacknowledged channel. In this section, we outline an effective strategy to *(i)* share a local chain (Algorithm 3) and *(ii)* use the set of received local chains and observed relative range observations to compute a navigation estimate (Algorithm 4), as illustrated in Fig. 3.1.

A local chain consists of odometry and prior factors. We first show that we can leverage the composition operation over odometry factors to represent each new odometry factor as a transformation relative to an 'origin' (Section 3.3.1). A corresponding decomposition operator is performed onboard the receiving vehicle in order to recover the original factor. Moreover, both of

Algorithm 3 Local chain distribution.

```
initialize($\mathcal{C}_{\text{local}}, \mathcal{C}_{\text{approx}}$) {Prior factor on initial pose.}
while is_running() do
    add_new_link($\mathcal{C}_{\text{local}}$) {New odometry and prior factors.}
    if $t$ is TOL then
        append_approx($\mathcal{C}_{\text{local}}, \mathcal{C}_{\text{approx}}$) {Section 3.3.2.}
        $\mathbf{L}_K$ = choose_links($\mathcal{C}_{\text{approx}}$) {Section 3.3.3.}
        $\mathbf{L}'_K$ = compose_links($\mathbf{L}_K$) {Section 3.3.1.}
        broadcast_links($\mathbf{L}'_K$)
    end if
end while
```

these operations are tolerant to communication failure, so that every received odometry factor can be used to reconstruct a portion of the local chain.

We compose the odometry factor of each link, $\mathcal{L}_i$, with all previous odometry factors to obtain the broadcast link, $\mathcal{L}'_i$. A receiving vehicle reconstructs the transmitter's chain consisting of decomposed odometry factors and prior factors contained in received links. If a broadcast is missed, upon the next successful reception the receiving vehicle can still reconstruct the chain *without* the prior factor in the missed link. We rebroadcast links with prior factors to help ensure an accurate chain reconstruction, meaning that at each TOL we broadcast a set of links, $\mathbf{L}$.

Since we only use the set of locally observed range measurements, each vehicle need only share TOL pose nodes. However, a local chain, $\mathcal{C}_{\text{local}}$, consists of many additional pose nodes. To reduce the communication burden, we employ approximate marginalization to compute an approximate local chain $\mathcal{C}_{\text{approx}}$ that only contains odometry and prior factors over TOL pose nodes (Section 3.3.2).

The ith receiving vehicle is able to reconstruct the broadcast local chains by decomposing broadcast odometry factors with odometry factors already received. Using the set of reconstructed approximate local chains $\{\mathcal{C}_{\text{approx}_j}\}_{j \neq i}$, its own local chain $\mathcal{C}_{\text{local}_i}$, and the set of observed OWTT range constraints, the vehicle can construct and solve a batch nonlinear least-squares problem (3.3) to estimate its smoothed trajectory.

3.3.1 Odometry Composition

In this section, we show that composition of rigid-body transformations is reversible (decomposition). Moreover, it is equivalent to marginalization over the odometry chain—that is, the marginal information induced by the the full set of odometry factors is equal to the information induced by the composed odometry factor. We leverage this property to broadcast each link $\mathcal{L}_i \in \mathcal{C}_{\text{approx}}$ over an unreliable communication channel.

Composed Odometry Factor

Rigid-body transformation (e.g., odometry) observations are full rank relative-pose constraints expressed as

$$\mathbf{z}_{ij} = h_{ij}(\mathbf{x}_i, \mathbf{x}_j) + \mathbf{w}_{ij} \tag{3.4}$$

$$= \ominus\mathbf{x}_i \oplus \mathbf{x}_j + \mathbf{w}_{ij}, \tag{3.5}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are two poses with respect to the same reference frame, $\mathbf{w}_{ij} \sim \mathcal{N}\big(\mathbf{0}, \mathrm{R}_{ij}\big)$ is an independent additive noise vector, and $\oplus$ and $\ominus$ represent generalized compounding and inverse operators. These operators are simple addition and subtraction for poses in $\mathbb{R}^d$ and are nonlinear functions defined for poses on $SE(2)$ and $SE(3)$ (Smith et al., 1986).

Per Smith et al. (1986), a composite observation $\mathbf{z}_{ik}$ is computed through the sequence

$$\mathbf{z}_{ik} = g(\mathbf{z}_{ij}, \mathbf{z}_{jk}) \tag{3.6}$$

$$= \mathbf{z}_{ij} \oplus \mathbf{z}_{jk}. \tag{3.7}$$

The composite relation $\mathbf{z}_{ik}$ is a random variable. A first-order covariance approximation is computed through

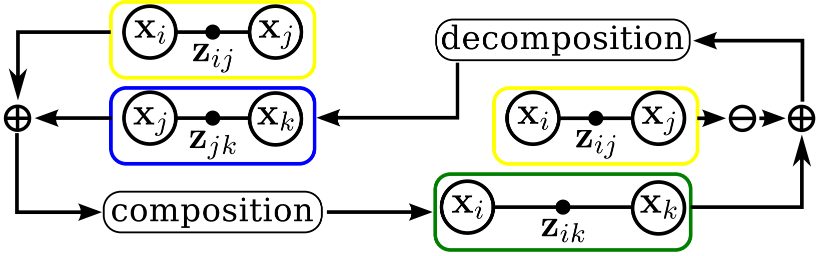
$$\mathbf{z}_{ik} \approx g\big|_{\mathbf{z}_{ij},\mathbf{z}_{jk}} + \mathrm{J}_{1\oplus}\delta\mathbf{z}_{ij} + \mathrm{J}_{2\oplus}\delta\mathbf{z}_{jk} \tag{3.8}$$

$$\sim \mathcal{N}\big(\mathbf{z}_{ij} \oplus \mathbf{z}_{jk}, \mathrm{J}_{1\oplus}\mathrm{R}_{ij}\mathrm{J}_{1\oplus}^\top + \mathrm{J}_{2\oplus}\mathrm{R}_{jk}\mathrm{J}_{2\oplus}^\top\big), \tag{3.9}$$

where $\mathrm{J} = \partial g / \partial(\mathbf{z}_{ij}, \mathbf{z}_{jk}) = [\mathrm{J}_{1\oplus}, \mathrm{J}_{2\oplus}]$.

Suppose we are given the composed observations $(\mathbf{z}_{ij}, \mathrm{R}_{ij})$ and $(\mathbf{z}_{ik}, \mathrm{R}_{ik})$. It follows directly from (3.9) that we can invert this operation to recover the transformation between $\mathbf{x}_j$ and $\mathbf{x}_k$ and

Figure 3.2 A sequence of odometry factors can be composed into a single factor. Likewise, a composed factor can be decomposed given another factor.



its covariance as

$$\mathbf{z}_{jk} = \ominus\mathbf{z}_{ij} \oplus \mathbf{z}_{ik} \tag{3.10}$$

$$\mathrm{R}_{jk} = \mathrm{J}_{2\oplus}^{-1}(\mathrm{R}_{ik} - \mathrm{J}_{1\oplus}\mathrm{R}_{ij}\mathrm{J}_{1\oplus}^{\top})\mathrm{J}_{2\oplus}^{-\top}, \tag{3.11}$$

where $\mathrm{J}_{2\oplus}$ is guaranteed to be nonsingular by the nature of the compounding operation—a full-rank rigid-body transformation. We call this operation, 'decomposition'. Composition and decomposition are illustrated in Fig. 3.2.

In our field trials, each pose is parameterized by the vehicle's XY horizontal position, i.e., $\mathbf{x}_i \in \mathbb{R}^2$. In this case, $\mathrm{J}_{1\oplus} = \mathrm{J}_{2\oplus} = \mathrm{I}$, and composition and decomposition reduce to simple addition and subtraction, respectively. Paull et al. (2014) introduced an equivalent composition/decomposition operation applicable in only $\mathbb{R}^2$.

Any link in the local chain can be communicated by broadcasting its composed odometry to an 'origin' pose (in practice, we use the local coordinate frame origin) as in line 7 of Algorithm 3. The receiving vehicle simply decomposes the odometry factor with the set of already received links as in line 5 of Algorithm 4. In this manner, we can exactly distribute the odometry backbone of a local graph. The subset of links each vehicle has received is unknown to the broadcasting vehicles, however, composition/decomposition still allows a receiver to reconstruct a complete chain over received links.

Composed Factor Equivalence

Here, we elaborate on the claim that the information induced by the composed odometry factor is equivalent to the marginal information induced by the full set of odometry factors. The result of this claim is that the composed odometry factor adds *exactly* the same information as the original factors, i.e., no information is either gained or lost. Without loss of generality, we consider two odometry factors: $\mathbf{z}_{ij}$ between poses $\mathbf{x}_i$ and $\mathbf{x}_j$, and $\mathbf{z}_{jk}$ between poses $\mathbf{x}_j$ and $\mathbf{x}_k$. The composed factor, $\mathbf{z}_{ik}$, is computed through (3.9).

The marginal information induced by $\mathbf{z}_{ij}$ and $\mathbf{z}_{jk}$ is computed by constructing the full potential and then marginalizing over the intermediate pose node $\mathbf{x}_j$, i.e.,

$$p(\mathbf{x}_i, \mathbf{x}_k | \mathbf{z}_{ij}, \mathbf{z}_{jk}) = \int_{\mathbf{x}_j} p(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k | \mathbf{z}_{ij}, \mathbf{z}_{jk}) d\mathbf{x}_j.$$

First, the information induced over the set of involved pose variable nodes, $\{\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k\}$, is obtained

$$
\begin{aligned}
\Lambda_{ijk} &= \mathrm{J}_{ijk}^\top \mathrm{R}_{ijk}^{-1} \mathrm{J}_{ijk} \\
&= \begin{bmatrix} \mathrm{J}_{1\oplus}\mathrm{J}_\ominus & \mathrm{J}_{2\oplus} & 0 \\ 0 & \mathrm{J}_{1\oplus}\mathrm{J}_\ominus & \mathrm{J}_{2\oplus} \end{bmatrix}^\top \begin{bmatrix} \mathrm{R}_{ij} & \\ & \mathrm{R}_{jk} \end{bmatrix}^{-1} \begin{bmatrix} \mathrm{J}_{1\oplus}\mathrm{J}_\ominus & \mathrm{J}_{2\oplus} & 0 \\ 0 & \mathrm{J}_{1\oplus}\mathrm{J}_\ominus & \mathrm{J}_{2\oplus} \end{bmatrix} \\
&= \begin{bmatrix} \mathrm{J}_\ominus^\top \mathrm{J}_{1\oplus}^\top \mathrm{R}_{ij}^{-1} \mathrm{J}_{1\oplus}\mathrm{J}_\ominus & \mathrm{J}_{1\oplus}^\top \mathrm{J}_\ominus^\top \mathrm{R}_{ij}^{-1} \mathrm{J}_{2\oplus} & \\ \mathrm{J}_{2\oplus}^\top \mathrm{R}_{ij}^{-1} \mathrm{J}_{1\oplus}\mathrm{J}_\ominus & \mathrm{J}_{2\oplus}^\top \mathrm{R}_{ij}^{-1} \mathrm{J}_{2\oplus} + \mathrm{J}_\ominus^\top \mathrm{J}_{1\oplus}^\top \mathrm{R}_{jk}^{-1} \mathrm{J}_{1\oplus}\mathrm{J}_\ominus & \mathrm{J}_\ominus^\top \mathrm{J}_{1\oplus}^\top \mathrm{R}_{jk}^{-1} \mathrm{J}_{2\oplus} \\ & \mathrm{J}_{2\oplus}^\top \mathrm{R}_{jk}^{-1} \mathrm{J}_{1\oplus}\mathrm{J}_\ominus & \mathrm{J}_{2\oplus}^\top \mathrm{R}_{jk}^{-1} \mathrm{J}_{2\oplus} \end{bmatrix}.
\end{aligned}
$$

The marginal information, Λ_{ik} is then computed by the Schur complement. For poses in $\mathbb{R}^d$, $\mathrm{J}_{1\oplus} = \mathrm{J}_{2\oplus} = -\mathrm{J}_\ominus = \mathrm{I}$ and the marginal information induced by the original factors is

$$
\Lambda_{ik} = \begin{bmatrix} (\mathrm{R}_{ij} + \mathrm{R}_{jk})^{-1} & -(\mathrm{R}_{ij} + \mathrm{R}_{jk})^{-1} \\ -(\mathrm{R}_{ij} + \mathrm{R}_{jk})^{-1} & (\mathrm{R}_{ij} + \mathrm{R}_{jk})^{-1} \end{bmatrix}. \tag{3.12}
$$

The information induced by the composed factor $\mathbf{z}_{ik}$ is easily computed

$$
\begin{aligned}
\Lambda'_{ik} &= \mathrm{J}^\top \mathrm{R}_{jk}^{-1} \mathrm{J} \\
&= \begin{bmatrix} \mathrm{J}_{1\oplus}\mathrm{J}_\ominus & \mathrm{J}_{2\oplus} \end{bmatrix}^\top \mathrm{R}_{jk}^{-1} \begin{bmatrix} \mathrm{J}_{1\oplus}\mathrm{J}_\ominus & \mathrm{J}_{2\oplus} \end{bmatrix} \\
&= \begin{bmatrix} \mathrm{J}_\ominus^\top \mathrm{J}_{1\oplus}^\top \mathrm{R}_{ik}^{-1} \mathrm{J}_{1\oplus}\mathrm{J}_\ominus & \mathrm{J}_\ominus^\top \mathrm{J}_{1\oplus}^\top \mathrm{R}_{ik}^{-1} \mathrm{J}_{2\oplus} \\ \mathrm{J}_{2\oplus}^\top \mathrm{R}_{ik}^{-1} \mathrm{J}_{1\oplus}\mathrm{J}_\ominus & \mathrm{J}_{2\oplus}^\top \mathrm{R}_{ik}^{-1} \mathrm{J}_{2\oplus} \end{bmatrix}
\end{aligned} \tag{3.13}
$$

where R_{ik} is determined from (3.9). Again, for poses in $\mathbb{R}^d$, we can compute the induced information

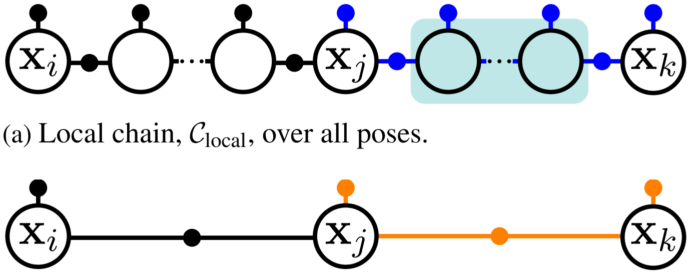
$$
\Lambda'_{ik} = \begin{bmatrix} \mathrm{R}_{ik}^{-1} & -\mathrm{R}_{ik}^{-1} \\ -\mathrm{R}_{ik}^{-1} & \mathrm{R}_{ik}^{-1} \end{bmatrix} = \begin{bmatrix} (\mathrm{R}_{ij} + \mathrm{R}_{jk})^{-1} & -(\mathrm{R}_{ij} + \mathrm{R}_{jk})^{-1} \\ -(\mathrm{R}_{ij} + \mathrm{R}_{jk})^{-1} & (\mathrm{R}_{ij} + \mathrm{R}_{jk})^{-1} \end{bmatrix}. \tag{3.14}
$$

The information (3.14) is equivalent to the marginal information induced by the original factors (3.13), i.e., $\Lambda_{ik} = \Lambda'_{ik}$. This equivalence can also be verifed for poses on $SE(2)$ and $SE(3)$.

3.3.2 Local Chain Approximation

A local chain, $\mathcal{C}_{\text{local}}$, will include both unary prior factors and pairwise sequential odometry factors over the full set of pose nodes. Ideally, the local chain would only include pose nodes at each TOL—the only poses involved in relative ranging events observed by other vehicles—to reduce the size of the chain that must be shared. In this section, we detail a method to incrementally

Figure 3.3 At the new TOL, $\mathbf{x}_k$, we compute a set of factors (orange) that best represents the distribution induced by the original factors (blue) since the last TOL, $\mathbf{x}_j$, with intermediate nodes (shaded) marginalized out.



(a) Local chain, $\mathcal{C}_{\text{local}}$, over all poses.

(b) Approximate local chain, $\mathcal{C}_{\text{approx}}$, over TOL poses.

approximate the full local chain with a chain only over TOL pose nodes, $\mathcal{C}_{\text{approx}}$, as illustrated in Fig. 3.3.

We refer to the marginal distribution over TOL pose nodes computed from $\mathcal{C}_{\text{local}}$ as the target distribution. The marginal cannot be exactly represented by a set of prior and odometry factors (a requirement for communication by odometry composition, see Section 3.3.1). However, we can compute the set of factors that most closely models this target distribution. The general framework for representing a Gaussian distribution using a desired nonlinear factor set is outlined by Mazuran et al. (2014).

We incrementally construct the approximate local chain by 'refactoring' the full local chain between TOLs as in Fig. 3.3. The links contained between the last TOL and the current TOL induce a marginal distribution over the TOL nodes $p(\mathbf{x}_j, \mathbf{x}_k) = \mathcal{N}\big(\boldsymbol{\mu}, \Sigma\big) = \mathcal{N}^{-1}\big(\boldsymbol{\eta}, \Lambda\big)$, the target distribution.

We compute a single odometry factor, $\mathbf{z}_{jk} = h_{\text{odo}}(\mathbf{x}_j, \mathbf{x}_k) + \mathbf{w}_{jk}$, and a prior over each node, $\mathbf{z}_j = h_{\text{prior}}(\mathbf{x}_j) + \mathbf{w}_j$ and $\mathbf{z}_k = h_{\text{prior}}(\mathbf{x}_k) + \mathbf{w}_k$, that most closely induce the target distribution, $p(\mathbf{x}_j, \mathbf{x}_k)$. Each factor is perturbed by independent zero-mean Gaussian noise with covariance given by R_{jk}, R_j and R_k. Let the distribution induced by the approximate factors be

$$\begin{aligned} q(\mathbf{x}_j, \mathbf{x}_k) &= \mathcal{N}^{-1}\big(\boldsymbol{\eta}', \Lambda'\big) && (3.15) \\ \boldsymbol{\eta}' &= \mathrm{J}^\top \mathrm{R}^{-1} \left(\mathbf{Z} - h(\mathbf{x}_i, \mathbf{x}_j)\right) && (3.16) \\ \Lambda' &= \mathrm{J}^\top \mathrm{R}^{-1} \mathrm{J} && (3.17) \\ \mathbf{Z} &= [\mathbf{z}_{jk}^\top, \mathbf{z}_j^\top, \mathbf{z}_k^\top]^\top \\ \mathrm{R} &= \operatorname{diag}(\mathrm{R}_{jk}, \mathrm{R}_j, \mathrm{R}_k) \\ \mathrm{J} &= \begin{bmatrix} \partial h_{\text{odo}}/\partial \mathbf{x}_j & \partial h_{\text{odo}}/\partial \mathbf{x}_j \\ \partial h_{\text{prior}}/\partial \mathbf{x}_j & \\ & \partial h_{\text{prior}}/\partial \mathbf{x}_k \end{bmatrix}, \end{aligned}$$

where $\mathbf{Z}$ represents the stacked observation vector, J is the stacked measurement Jacobian, and R is the block-diagonal stacked measurement covariance. We compute $\mathbf{Z}$ and R to minimize the Kullback-Leibler divergence (KLD) between $p(\mathbf{x}_j, \mathbf{x}_k)$ and $q(\mathbf{x}_j, \mathbf{x}_k)$

$$D_{\mathrm{KL}}\left(p(\mathbf{x}_j, \mathbf{x}_k) || q(\mathbf{x}_j, \mathbf{x}_k)\right) = \frac{1}{2}\left(\mathrm{tr}(\Lambda'\Sigma) + (\boldsymbol{\mu}' - \boldsymbol{\mu})^\top \Lambda (\boldsymbol{\mu}' - \boldsymbol{\mu}) - k + \ln \frac{\det \Sigma'}{\det \Sigma}\right), \quad (3.18)$$

where k is the dimension of the stacked state, i.e., $\dim(\boldsymbol{\mu})$. The KLD is minimized when $\mathbf{Z}$ is the expected observation evaluated at the mean, i.e., $\mathbf{Z} = h(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j)$ such that $\boldsymbol{\mu}' = \boldsymbol{\mu}$. R is then computed from (3.18)

$$\mathrm{R}^* = \underset{\mathrm{R} \in \mathrm{S}^{++}}{\mathrm{argmin}} \, \mathrm{tr}\left(\mathrm{J}^\top \mathrm{R}^{-1} \mathrm{J} \Sigma\right) - \ln \det \left(\mathrm{J}^\top \mathrm{R} \mathrm{J}\right), \quad (3.19)$$

where S^{++} is the cone of symmetric positive definite matrices (with appropriate sparsity structure). As shown in Mazuran et al. (2014), the above optimization problem is convex and can be solved efficiently.

We have not enforced a consistency constraint on R, which would guarantee that the approximate factors are not overconfident. The consistency constraint can be expressed as a linear matrix inequality, $\Sigma' - \Sigma \succeq 0$, as in Carlevaris-Bianco and Eustice (2014). Although an explicit consistency constraint was not employed here, the approximation produced a consistent estimate in all field trials.

In general, Λ will not be full rank and we cannot compute approximate factors as in (3.19). This will only occur if no prior is available over the time window $i \to j$ (as is often true for AUVs). In this case, however, we can *exactly* represent the target information with a composed odometry factor (see Section 3.3.1) and no prior factors. Similarly, if no odometry is available over $i \to j$ (e.g., for a topside vehicle with only GPS), then the TOL poses are independent and a prior may be computed to exactly represent the target distribution without an odometry factor.

At the TOL, after computing new approximate link factors in $\mathcal{C}_{\text{approx}}$, a vehicle broadcasts the new TOL link and k additional links to rebroadcast, $\mathbf{L}$, as in Algorithm 3. The size of k is predetermined depending on the available bandwidth.

3.3.3 Rebroadcasting Priors

$\mathcal{C}_{\text{approx}}$ is reconstructed from the set of received links and therefore may not contain all prior factors. We use some additional bandwidth to rebroadcast useful or informative links with prior factors. Below, we consider finding the set of prior factors that are most informative about the current vehicle pose so that receiving vehicles are able to best use their observed range measurement.

A simple method is to broadcast the k most recent prior factors (k-last in Section 3.4). Intuitively, the most recent prior factors may be most informative about the transmitting vehicle's current pose. However, as a counterpoint, consider a vehicle with perfect (deterministic) odometry. Then, the best prior factors would be the factors with lowest uncertainty, not the most recent. For this reason, we develop a method to identify a more informative set of prior factors based on the expected set of factors already received by the network (k-best in Section 3.4).

Since each prior factor is broadcast over a lossy and unacknowledged channel, we assume that the ith transmission is lost with probability r_i. If the jth link has been broadcast m_j times, then the probability that it has been received is

$$p_{\text{rec}_j} = 1 - \prod_{i=1}^{m_j} r_i. \tag{3.20}$$

The results presented in Section 3.4 assume that r_i is a fixed parameter; however, r_i could incorporate additional knowledge, or even be a learned parameter. We employ a similar Bernoulli reception model in Chapter 4.

We use a mutual information objective for determining the utility of a prior factor. Mutual information indicates the uncertainty reduction in the current pose of the reconstructed chain, $\mathbf{x}_N$, by adding knowledge of a prior factor. Let $\mathbf{L}_K = \{\mathcal{L}_{i_1}, \ldots, \mathcal{L}_{i_k}\} \subset \{\mathcal{L}_1, \ldots, \mathcal{L}_N\}$ be the set of k links that we will broadcast. The optimal set then satisfies

$$\begin{aligned} \mathbf{L}_K^* &= \arg\max_{\mathbf{L}_K} \mathrm{E}\left[\mathrm{MI}[\mathbf{x}_N | \mathbf{L}_K]\right] \\ &= \arg\max_{\mathbf{L}_K} \mathrm{E}\left[\log\det \Lambda_{\text{recon}}(\mathbf{x}_N)\right] \\ \Lambda_{\text{recon}} &= \underbrace{\Lambda_{\text{odo}}}_{\text{all odo factors}} + \sum_j p_{\text{rec}_j} \underbrace{\Lambda_{\text{prior}_j}}_{\text{each prior}}, \end{aligned}$$

where $\Lambda_{\text{recon}}(\mathbf{x}_N)$ represents the marginal information of $\mathbf{x}_N$ in the reconstructed chain computed from Λ_{recon} via the Schur complement. In general, this is a difficult combinatorial optimization problem. Instead, we follow a simpler approach. We evaluate the objective for each link and then greedily select the single link, $\mathcal{L}_i$, that maximizes the objective, increment m_i, and remove $\mathcal{L}_i$ from the set of potential links to broadcast. We repeat the process of evaluating each link and greedily selecting the best k times.

3.3.4 Batch Estimation

As mentioned previously, the gold-standard centralized estimator computes the MAP configuration over *all* vehicle poses (3.3). The ith vehicle, however, only has access to its local chain, $\mathcal{C}_{\text{local}_i}$, and

Algorithm 4 Cooperative localization

```
Require: $\mathcal{C}_{\text{local}}$ {Local chain}
C = ∅ {Set of received chains.}
$\mathbf{Z}_r = \emptyset$ {Set of received range observations.}
while is_running() do
    if $\mathbf{L}_{\text{rec}}, \mathbf{z}_r, i$ = received_broadcast() then
        add_links($\mathbf{C}[i], \mathbf{L}_{\text{rec}}$) {Links from vehicle $i$.}
        $\mathbf{Z}_r = \mathbf{Z}_r \cup \mathbf{z}_r$
        solve_batch($\mathcal{C}_{\text{local}}, \mathbf{C}, \mathbf{Z}_r$)
    end if
end while
```

the set of reconstructed chains from broadcast links. Let $\mathbf{I}$ be the set of vehicle indices for which ith vehicle has reconstructed a chain. The full posterior (3.2) is approximated by

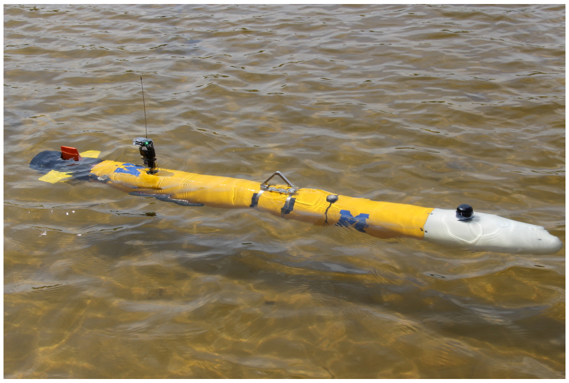
$$p(\mathbf{X}_i, \mathbf{X}_{j\in\mathbf{I}}|\mathbf{Z}_i, \mathbf{Z}_{j\in\mathbf{I}}, \mathbf{Z}_r) \propto \underbrace{p(\mathbf{X}_i|\mathbf{Z}_i)}_{\mathcal{C}_{\text{local}_i}} \prod_{j\in\mathbf{I}} \underbrace{p(\mathbf{X}_j|\mathbf{Z}_j)}_{\mathcal{C}_{\text{approx}_j}} \prod_k p(\mathbf{z}_k|\mathbf{x}_{i_k}, \mathbf{x}_{j_k}). \tag{3.21}$$

as illustrated in Fig. 3.1b. Note that the posterior is only computed over local vehicle poses and the set of received broadcast TOL poses and that only the set of locally obtained relative pose constraints is used. The vehicle can then employ any graph-based solver (e.g., square-root SAM) to compute a solution (e.g., see Appendix A). Algorithm 4 reviews the estimation procedure performed by each vehicle.

3.4 Field Trials

For validation, we fielded two Ocean-Server Inc. Iver2 AUVs, termed AUV-A and AUV-B (see Fig. 3.4), and a topside support ship. The AUVs are each equipped with an advanced dead-

Figure 3.4 One of the Iver2 AUVs used in the field trials.



reckoning sensor suite. Each AUV observed GPS during intermittent surface intervals. The topside support ship had access only to GPS (no measured odometry). All vehicles used the WHOI Micro-modem and co-processor board with a synchronous-clock reference for inter vehicle communication and ranging. The AUV hardware is detailed in Appendix C. Experiments were conducted July 25–27, 2014, at the UMBS.

Each vehicle pose was parameterized by its XY horizontal position. GPS observations were transformed to a local coordinate frame and constitute a full-rank linear observation. We assumed that the GPS accuracy was constant and therefore we assumed noise with a fixed standard deviation of $5 \mathrm{~m}$. The AUV odometry, $\mathbf{z}_{ij}$, and corresponding covariance, R_{ij}, were computed by Euler integrating DVL and AHRS and performing a first-order covariance approximation (see Appendix B.1).

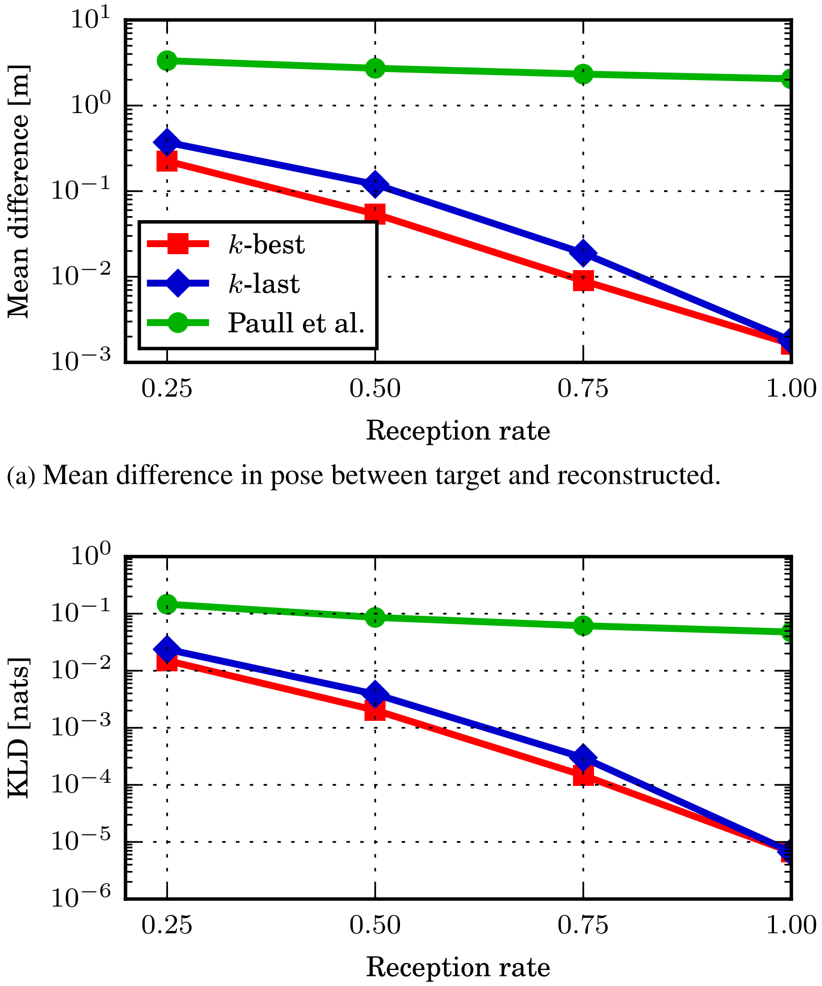
Observed OWTT ranges represent a 3D *slant* range. Since depth is measured with bounded error, we are able to project ranges into the 2D horizontal plane and use the resulting *pseudo* ranges within our estimation framework (Appendix B.2). We used a fixed OWTT measurement uncertainty since the relative depth between each vehicle's acoustic transducer was small and relatively constant.

Vehicles broadcast navigation messages roughly once per minute according to a fixed TDMA schedule. The proposed composition/decomposition allowed a coarse quantization (factors were rounded to $1 \mathrm{~cm}$). Each navigation packet (i.e., set of links $\mathbf{L}$) required a single $64 \mathrm{~B}$ frame of either a Micro-modem Rate 1 or Rate 2 data packet. The previously detailed OSM (Chapter 2) minimally requires two $64 \mathrm{~B}$ frames, leading to a 50% reduction. The full message specification is detailed in Appendix C.

We employed our k-last prior factor selection strategy during real-time experiments (for $k = 2$). In post-process, we computed the prior factors that would have been broadcast with our k-best strategy.

We also implemented the algorithm proposed by Paull et al. (2014) for comparison. Their algorithm similarly broadcasts each new TOL pose as a composed factor. There are two large differences in our approach: *(i)* they transmit prior factors directly from the full local chain separately, i.e., they must broadcast each prior link in addition to TOL links, and *(ii)*, they also broadcast received range factors and their corresponding TOA links. The second difference effectively allows for bidirectional ranging at the cost of packets that grow linearly in the size of the network (a TOA pose is broadcast for each vehicle in the network). During the three vehicle post-process evaluation, their algorithm broadcast more links requiring greater bandwidth (4 composed odometry factors, 2 range factors, and a prior factor compared to 3 composed odometry factors and 2 prior factors for ours).

Figure 3.5 Local graph reconstruction quality for k-last, k-best, and Paull et al. (2014) for various reception rates averaged over 100 trials for six local chains. Note log scale on ordinate axis.



(a) Mean difference in pose between target and reconstructed.

(b) KLD between target distribution and reconstructed.

3.4.1 Local graph reconstruction

Here we demonstrate the quality of the local chain reconstruction under varying channel conditions. We ran three trials ranging from $60 - 90$ min to collect navigation observations and broadcast event times for each of our two AUVs resulting in six local chains. During the trials, each AUV had intermittent access to GPS during brief surface intervals. In post-process, we computed and sampled the set of navigation packets at different reception rates and reconstructed the local chain. For each reception rate, we artificially sampled different sets of received messages 100 times. We then compared the reconstructed local chain to the target chain (the marginal full chain over received TOL pose nodes). Fig. 3.5 compares the local chain reconstruction for prior factor selection methods, k-last and k-best, as well as the method proposed by Paull et al. (2014).

The k-best selection strategy outperforms k-last for both mean reconstruction error and KLD

Table 3.1 Acoustic communication reception rates between pairs of vehicles across all trials.

	Trial 1	Trial 2	Trial 3
AUV-A $\leftrightarrow$ AUV-B	57.0%	62.9%	37.6%
AUV-A $\leftrightarrow$ Topside	82.8%	74.9%	52.9%
AUV-B $\leftrightarrow$ Topside	84.8%	78.3%	86.6%

by a factor of two or three in most trials. At 100% reception rate (no dropped packets) the full approximate chain was reconstructed and is therefore equivalent for k-best and k-last. Moreover, the full approximate chain closely represents the full local chain. While k-best outperforms k-last, k-last has a computational advantage incurring essentially no cost to choose the last several priors.

Our proposed algorithm improves upon Paull et al.'s method by several orders of magnitude in most trials. Our method is able to include more prior information into the broadcast links, and therefore leads to a better reconstruction.

As mentioned in Section 3.3.1, composition is equivalent to marginalization. Therefore, when no priors are present in the local chain, all compared algorithms are exactly equal to the target distribution (up to quantization errors).

3.4.2 Cooperative localization

We executed three multiple vehicle trials to demonstrate the ability of our algorithm to provide useful navigation information to an AUV. We varied the number of vehicles and AUV access to GPS in each trial to demonstrate a variety of practical applications. Acoustic reception rates varied between 37–86% up to $500\ \mathrm{m}$ relative range as summarized in Table 3.1. Results are summarized in Fig. 3.6. Fig. 3.2 reports each distributed algorithm's ability to produce the centralized uncertainty estimate onboard AUV-B. Although we only show results for AUV-B, note that all vehicles compute a local reconstruction of the centralized estimator.

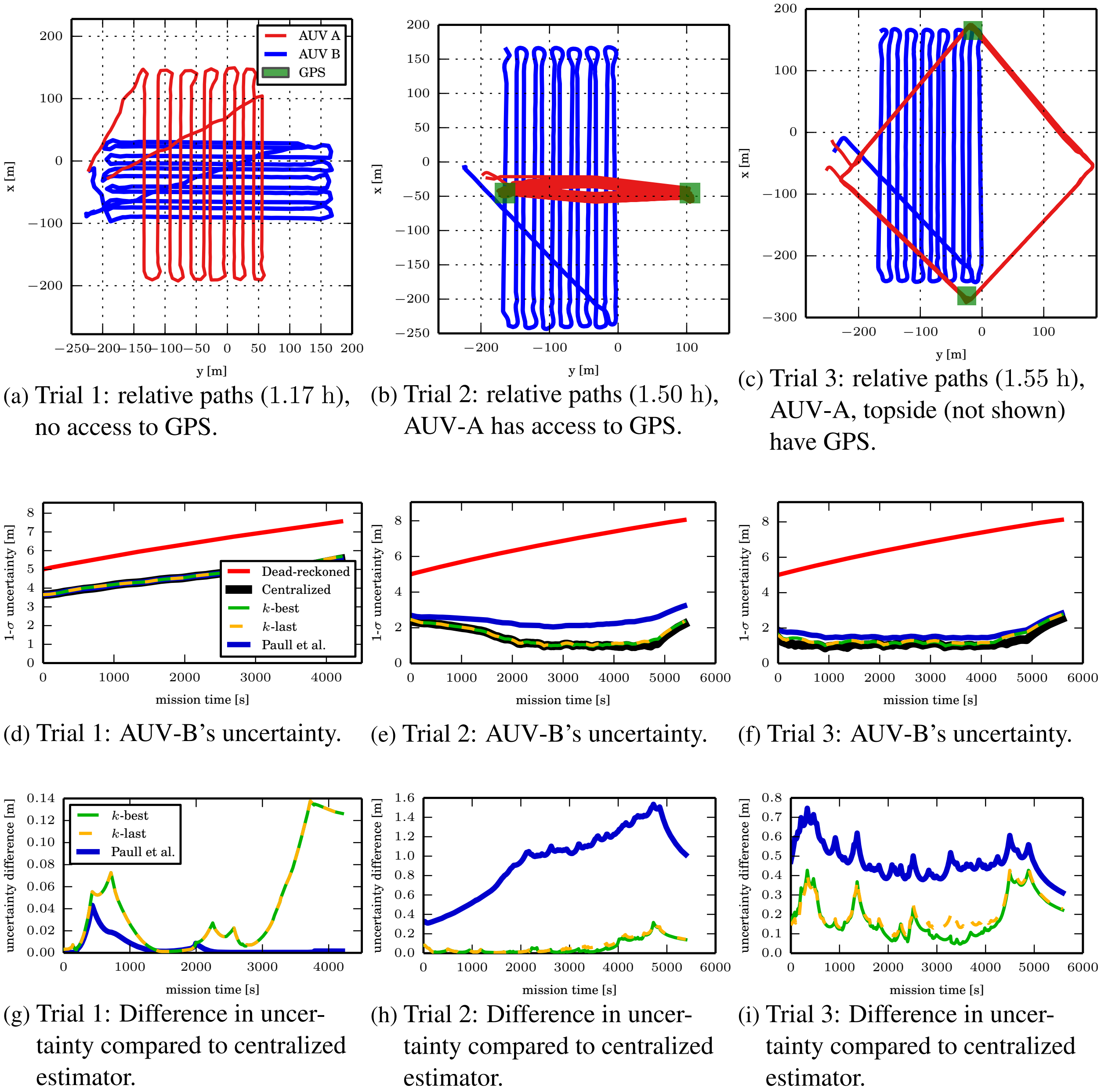
Trial 1

AUV-A and AUV-B performed overlapping orthogonal lawnmower surveys (see Fig. 3.6a). Both vehicles had only dead-reckoned navigation available. Since there are no GPS priors, our local graph reconstruction is equivalent to Paull et al. (2014). Moreover, the local chain reconstruction

Table 3.2 Average 1-σ uncertainty difference compared to the centralized estimator across all trials.

	Trial 1 [m]	Trial 2 [m]	Trial 3 [m]
Paull et al. (2014)	**0.005**	0.981	0.463
Proposed (k-last)	0.039	0.086	0.212
Proposed (k-best)	0.039	**0.074**	**0.192**

Figure 3.6 Summary of field trials and performance comparison. (a–c) shows each vehicle trajectory. (d–f) plots the smoothed uncertainty in each AUV-B pose computed as the fourth root of the determinant of the pose marginal covariance. AUV-A access to GPS is indicated by green boxes.



(a) Trial 1: relative paths (1.17 h), no access to GPS.

(b) Trial 2: relative paths (1.50 h), AUV-A has access to GPS.

(c) Trial 3: relative paths (1.55 h), AUV-A, topside (not shown) have GPS.

(d) Trial 1: AUV-B's uncertainty.

(e) Trial 2: AUV-B's uncertainty.

(f) Trial 3: AUV-B's uncertainty.

(g) Trial 1: Difference in uncertainty compared to centralized estimator.

(h) Trial 2: Difference in uncertainty compared to centralized estimator.

(i) Trial 3: Difference in uncertainty compared to centralized estimator.

for all algorithms was exact. k-last and k-best produced the same solution since there are no priors to rebroadcast. Since Paull et al. (2014) incorporated range constraints between both vehicles, their algorithm computed a solution that more closely matches the centralized result (see Fig. 3.2). Their algorithm did not exactly reproduce the centralized result because AUV-B did not receive all range factors observed by AUV-A. Although we only used the local subset of range factors, we were still able to benefit from relative range observations and the difference compared to the centralized estimator is small.

Trial 2

AUV-A executed a narrow diamond trajectory over AUV-B's lawnmower survey. Both vehicles had dead-reckoned navigation available, but AUV-A also intermittently received GPS during short surface intervals (see Fig. 3.6b). Fig. 3.6e plots the pose uncertainty onboard AUV-B. Using our proposed algorithm, AUV-B was able to accurately approximate AUV-A's pose-graph, including the many prior factors. k-best here performed slightly better than k-last because of its ability to more accurately reproduce AUV-A's local chain. Although Paull et al. (2014) were able to incorporate range information in both directions, our reconstruction of AUV-A was more accurate, leading to a more confident estimate. Due to the diminishing returns of information provided by the range observations, our algorithm was able to closely reproduce the centralized solution despite only using a subset of the range observations.

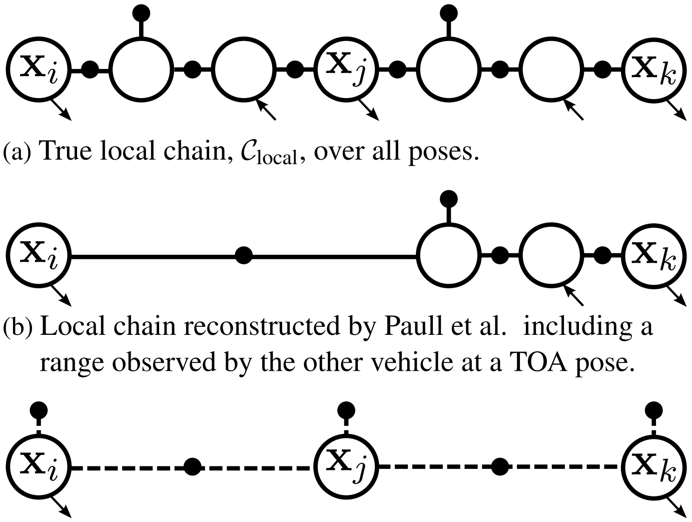
Trial 3

A topside vehicle with constant GPS access supported AUV-A (with intermittent GPS) and AUV-B (see Fig. 3.6c). The factor graph produced here is illustrated in Fig. 3.1. AUV-A followed a large diamond over AUV-B's lawnmower survey while the topside vehicle drifted above the survey area. In this case, the centralized estimator used ranges between all three vehicles. Our method only used ranges between the local platform and the other vehicles. Paull et al. (2014), however, included ranges in both directions and could also use ranges between other vehicles, but only if the TOL and TOA poses in each local chain had been received. Once again, our reconstruction of AUV-A's chain is more informative. As shown in Fig. 3.2, we have received the bulk of the benefit in terms of uncertainty reduction from the local set of range observations and achieved a smaller estimate uncertainty lower than Paull et al. (2014).

Summary

In all field trials, the presented factor composition method produces an accurate representation of the centralized estimator. Moreover, both k-best and k-last methods produced a consistent estimate—they were never more confident than the centralized estimator. By applying an approximate marginalization procedure, we can broadcast a small yet informative set of factors. The results also demonstrate that the local set of range observations is capable of bounding uncertainty growth.

Figure 3.7 Comparison of reconstructed local chains by Paull et al. (b) and our method (c) as if the data transmitted at the second TOL, $\mathbf{x}_j$, was not received. Arrows away from the chain indicate TOLs, arrows toward the chain indicate TOAs.



(a) True local chain, $\mathcal{C}_{\text{local}}$, over all poses.

(b) Local chain reconstructed by Paull et al. including a range observed by the other vehicle at a TOA pose.

(c) Approximate local chain, $\mathcal{C}_{\text{approx}}$, reconstructed by our method, k-last, with $k \geq 1$.

3.5 Discussion

Our proposed algorithm is similar in some respects to that independently introduced by Paull et al. (2014), however, we believe there are many beneficial differences illustrated in Fig. 3.7. The most marked improvement in our algorithm is its ability to broadcast a local chain by using an accurate approximation that includes a more informative set of prior factors. The compact approximate chain contains fewer links to broadcast. While we currently only incorporate ranges measured locally, this allows for a fixed bandwidth data packet, regardless of the size of the vehicle network. We can use the additional bandwidth to rebroadcast previous informative links and for other practical purposes (e.g., command, control, vehicle health). Paull et al. (2014) are able to incorporate inter-vehicle ranges, but only when the receiving vehicle has received all involved poses. Both methods do not support cascaded communication topologies (one vehicle supports another, which in turn supports another). Finally, the local subset of range observations is sufficient for accurate navigation in many practical situations (as evidenced in Section 3.4).

3.6 Chapter Summary

Accurate localization extends the capacity of AUVs to perform ocean science. OWTT underwater cooperative localization promises improved navigation for AUVs over larger area and time scales without additional infrastructure. We exploit the structure of the composition operation over odometry factors and an accurate approximation of the local chain (factor graph) to share locally observed sensor data across a fragile communication channel. That is, any pose node in the chain can be represented with its composed odometry link. We then use the collection of received chains and observed relative pose measurements to compute an improved navigation solution. The proposed method scales to many vehicles in that each vehicle in the network can simultaneously act as server and client. Compared to the previous server-client architecture introduced in Chapter 2, the proposed method is more tolerant to communication failure as a recovery mechanism is not required.

Our proposed method can also find application in the broader robotics community where limited communication is an operational factor, e.g., search and rescue. Avenues for future work include extending our chain approximation method to include information received from other vehicles. This would allow better localization performance in certain communication topologies, for example, cascaded networks.

Chapter 4

Planning for Cooperative Localization

In this chapter, we assume that a distributed estimation framework for server-client localization (such as those developed in Chapter 2 and Chapter 3) is available and examine how the quality of the client estimate varies with the vehicle network geometry[1]. We then use this to plan a server vehicle path for cooperative localization. We assume that the client mission plan, i.e., desired trajectory, is available at planning time. Our goal is to compute a practical server trajectory that minimizes the predicted client uncertainty through OWTT support.

Within OWTT cooperative localization there is an inherent trade-off between communication fidelity and localization accuracy—increasing horizontal relative range diminishes the probability of successful communication, but improves horizontal position estimation, as illustrated in Fig. 4.1. Therefore, planning serves to benefit from modeling the communication channel. Belief space planning represents a principled approach to balancing standard path-planning objectives (e.g., minimum distance) with motion, sensing, and, in this chapter, communication uncertainty. Resulting paths seek information in order to reduce state uncertainty and successfully meet a task-specific objective.

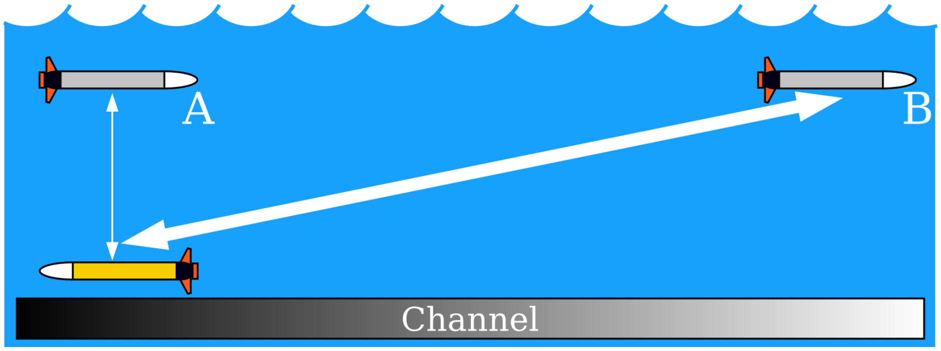
Belief space planning solves a more general POMDP problem by making a linear Gaussian assumption. Most belief space planning approaches, however, ignore that sensor measurements may not be obtained, for example, because of a lossy communication channel. We introduce a model of the underwater acoustic communication channel into belief space planning that allows a trajectory optimization algorithm to find server paths that provide useful navigation information and are also likely to demonstrate successful communication (and hence provide informative relative range observations).

This chapter provides the following contributions:

- We propose a belief space planning framework for server-client cooperative localization with

[1] Portions of this work have previously appeared in (Walls and Eustice, 2014b; Walls et al., 2015a).

Figure 4.1 Cooperative localization example: a server vehicle 'A' or 'B' can provide a relative range constraint (white arrows) to the client vehicle (yellow). 'A' is more likely to make the range observation as evidenced by the channel gradient (darker means higher probability of making a measurement). 'B', however, provides more information in the horizontal plane (as indicated by the thicker arrow). Our proposed planning framework balances these objectives.



a known client mission plan.

- We exploit a modified observation likelihood function that allows us to model randomness in obtaining measurements. We then employ this modified observation function to model the underwater acoustic channel.
- We propose an optimization algorithm for computing locally optimal receding-horizon control actions or path parameters that represent practical AUV trajectories.

Additionally, we provide a channel model that is representative of data collected during AUV field trials. We then use the channel model to demonstrate our planning framework in several simulated scenarios.

4.1 Related Work

Path planning problems typically seek a feasible path from start to goal positions within a configuration space. When state is not known deterministically, planning can be formalized within a POMDP framework (Kaelbling et al., 1998). In this case, objective functions are defined over the space of potential distributions over state, called the belief space. Planning methods must cope with uncertainty in motion, sensing, and the surrounding environment. Due to the high dimensionality of the belief spaces of real-world problems, solutions to POMDP instances must typically be approximated.

Several prior approaches to belief space planning (Prentice and Roy, 2009; Agha-mohammadi et al., 2014; Bry and Roy, 2011) attempted to solve belief space planning problems with sampling-based planners. They addressed uncertainty in sensing, motion, or both. These methods involve constructing candidate trajectories through the underlying state space by random sampling, then searching over candidate solutions. These algorithms find the best solution within the space of candidate trajectories but do not explicitly optimize the trajectory.

Recently, optimization motion planning has been applied to belief space planning (Platt et al., 2010; van den Berg et al., 2012; Patil et al., 2014). This chapters builds off of these methods in order to optimize over both open-loop control actions and parameterized trajectories. Several methods (Platt et al., 2010; Patil et al., 2014) assume maximum-likelihood observations in order to achieve rapid replanning. Van den Berg et al. (2012) drop the maximum-likelihood assumption at the cost of greater computational expense, but planning is performed off-line and a control policy is executed at run time.

The related area of active SLAM research has focused on computing paths that lead to lower uncertainty maps. Stachniss et al. (2005) presented an early particle-based approach to make informative decisions under uncertainty, integrating localization, mapping, and exploration. Sim and Roy (2005) proposed a breadth-first search procedure for minimizing an A-optimal objective on vehicle uncertainty. Recently, Indelman et al. (2015) merged optimization-based belief space planning within active SLAM. These authors additionally model unknown observation events (similar in spirit to the channel model presented here), but employ a simplification that results in a deterministic state covariance.

The multiple vehicle coordination problem is well studied (Kumar et al., 2004), however, navigation for *localization* remains an open challenge. Martinez and Bullo (2006) proposed a heuristic coordination algorithm for target-tracking with a range-only sensor. Charrow et al. (2014) similarly presented a target-localization control policy using range-only sensors. The most relevant approaches to the underwater range-only localization were presented by German et al. (2012), Tan et al. (2014), and Bahr et al. (2012). These prior works, however, do not include motion and sensing uncertainty and, in some cases, consider only discrete action spaces. Moreover, no prior work has modeled the communication channel, although several authors have imposed a heuristic maximum communication range, e.g., Tan et al. (2014). We propose a principled method to incorporate channel behavior during planning.

4.2 Belief Space Planning

The general planning problem we address in this chapter is an instance of a POMDP, since both sensing and motion are stochastic. We first state our problem instance within a Gaussian belief space (Section 4.2.1) and derive the belief evolution (Section 4.2.2). Then, we modify the belief representation to handle nondeterministic observation events (Section 4.2.3). Finally, we develop a planning framework that leverages this modified belief representation (Section 4.2.4 and Section 4.2.5).

4.2.1 Gaussian Belief Space Planning

Consider a system with state $\mathbf{x}_k$ at time step k. The state transition model captures uncertainty in executing an action,

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k), \tag{4.1}$$

where $\mathbf{u}_k$ is the control input, and $\mathbf{w}_k \sim p(\mathbf{w}_k|\mathbf{x}_k, \mathbf{u}_k)$ is an independent noise disturbance with known density, assumed here to be zero-mean Gaussian with covariance matrix M_k. This transition model equivalently encodes the transition distribution $p(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathbf{u}_k) = \mathcal{N}\big(f(\mathbf{x}_k, \mathbf{u}_k, 0), \mathrm{M}_k\big)$.

Observations are also random variables and modeled as

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{v}_k), \tag{4.2}$$

where $\mathbf{v}_k \sim p(\mathbf{v}_k|\mathbf{x}_k)$ is an independent noise perturbation with known density. We assume that the observation noise is additive and drawn from a zero-mean Gaussian distribution with covariance matrix N_k. The observation model is then represented by the likelihood function $p(\mathbf{z}_k|\mathbf{x}_k) = \mathcal{N}\big(h(\mathbf{x}_k, 0), \mathrm{N}_k\big)$.

The belief $\mathbf{b}_k$ is the posterior distribution over state given all observations and control inputs up to time k, i.e.,

$$\mathbf{b}_k = p(\mathbf{x}_k|\mathbf{Z}_{1:k}, \mathbf{U}_{1:k-1}), \tag{4.3}$$

where $\mathbf{Z}_{1:k} = \{\mathbf{z}_i\}_{i=1}^{k}$ is the set of observations and $\mathbf{U}_{1:k-1} = \{\mathbf{u}_i\}_{i=1}^{k-1}$ is the set of all control inputs. In Section 4.2.3, we modify the observation and belief definitions to include unknown observation events.

In this work, we consider a general quadratic cost function comprised of the sum of a terminal cost and stage costs over the beliefs and controls through time-horizon L

$$\min_{\mathbf{B}_{1:L}, \mathbf{U}_{1:L-1}} \mathrm{E}\left[\mathbf{b}_L^\top \mathrm{Q}_L \mathbf{b}_L + \sum_{k=1}^{L-1} \mathbf{b}_k^\top \mathrm{Q}_k \mathbf{b}_k + \mathbf{u}_k^\top \mathrm{R}_k \mathbf{u}_k\right], \tag{4.4}$$

where $\mathbf{B}_{1:L} = \{\mathbf{b}_k\}_{k=1}^{L}$ is the set of belief states that satisfy the relationship in (4.3), and the expectation is taken with respect to the random observations, $\mathbf{Z}_{1:L}$. The cost weight matrices Q_k and R_k are positive semi-definite and positive definite, respectively. Quadratic (or approximately quadratic) cost functions are commonly employed in belief space planning problems for the relative ease with which they can be minimized.

Computing the optimal solution to a cost function over belief states is generally intractable because of the high dimensionality of the space of all beliefs. Under the assumption of additive Gaussian noise and locally linear models, however, a belief is then approximately Gaussian and can be parameterized by the state mean $\hat{\mathbf{x}}_k$ and vectorized covariance matrix Σ_k,

$$\mathbf{b}_k = \left[\hat{\mathbf{x}}_k^\top, \mathrm{vec}(\Sigma_k)^\top\right]^\top. \tag{4.5}$$

Note that the $\mathrm{vec}(\,\cdot\,)$ operation exploits the symmetry in Σ_k. This Gaussian parameterization enables tractable optimization.

Our strategy closely follows previous methods for deriving an analytical belief evolution and optimizing within that framework (He et al., 2011; van den Berg et al., 2012). We proceed in two steps. First, we derive the belief dynamics, which define the belief transition distribution $p(\mathbf{b}_{k+1}|\mathbf{b}_k, \mathbf{u}_k)$. Second, we use the predictive belief dynamics to minimize a cost function over beliefs and controls (4.4). Our approach differs from the aforementioned works with the introduction of a channel model and uses an alternative optimization framework that allows simple and practical parametric trajectories to be considered.

4.2.2 Belief Dynamics

The belief dynamics detail the evolution of beliefs given a control sequence. For discrete control spaces, we could enumerate all possible sequences. However, brute force enumeration may be intractable for even moderate horizon lengths. For continuous control spaces, enumeration is not possible. Below we show that we can analytically compute the distribution over beliefs.

A belief $\mathbf{b}_k$ represents the posterior distribution over state. The belief dynamics are computed via a recursive Bayes filter

$$\begin{aligned}\mathbf{b}_{k+1} &= p(\mathbf{x}_{k+1}|\mathbf{Z}_{1:k+1}, \mathbf{U}_{1:k}) \\ &= \eta p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1})\underbrace{p(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathbf{u}_k, \mathbf{Z}_{1:k})}_{\bar{\mathbf{b}}_{k+1}},\end{aligned} \tag{4.6}$$

where $\bar{\mathbf{b}}_{k+1}$ is the predicted belief given $\mathbf{b}_k$ and $\mathbf{u}_k$ and η is a normalization constant.

The belief dynamics can be computed in closed form within an EKF under the assumption of Gaussian noise models and approximately locally linear transition and observation functions. The

EKF update given the current belief follows

$$\begin{aligned}
\hat{\mathbf{x}}_{k+1} &= f(\hat{\mathbf{x}}_k, \mathbf{u}_k, 0) + \mathrm{K}\Big(\mathbf{z}_{k+1} - h\big(f(\hat{\mathbf{x}}_k, \mathbf{u}_k, 0), 0\big)\Big) && (4.7)\\
\Sigma_{k+1} &= (\mathrm{I} - \mathrm{KH})\bar{\Sigma} && (4.8)\\
\mathrm{F} &= \frac{\partial}{\partial \mathbf{x}} f(\hat{\mathbf{x}}_k, \mathbf{u}_k, 0)\\
\mathrm{H} &= \frac{\partial}{\partial \mathbf{x}} h(f(\hat{\mathbf{x}}_k, \mathbf{u}_k, 0), 0))\\
\bar{\Sigma} &= \mathrm{F}\Sigma_k \mathrm{F}^\top + \mathrm{M}_k\\
\mathrm{K} &= \bar{\Sigma}\mathrm{H}^\top(\mathrm{H}\bar{\Sigma}\mathrm{H}^\top + \mathrm{N}_k)^{-1}.
\end{aligned}$$

The mean update (4.7) is random in $\mathbf{z}$, while the covariance update (4.8) is deterministic. Moreover, the distribution over the belief mean is Gaussian since $\mathbf{z}$ is Gaussian. The distribution over beliefs—the belief dynamics—can therefore be represented by a Gaussian process

$$\begin{aligned}
\mathbf{b}_{k+1} &= g(\mathbf{b}_k, \mathbf{u}_k) + \mathrm{W}(\mathbf{b}_k, \mathbf{u}_k)\mathbf{w}_k && (4.9)\\
g(\mathbf{b}_k, \mathbf{u}_k) &= \begin{bmatrix} f(\hat{\mathbf{x}}_k, \mathbf{u}_k, 0) \\ \mathrm{vec}(\bar{\Sigma} - \mathrm{KH}\bar{\Sigma}) \end{bmatrix}\\
\mathrm{W}(\mathbf{b}_k, \mathbf{u}_k) &= \begin{bmatrix} \sqrt{\mathrm{KH}\bar{\Sigma}} \\ 0 \end{bmatrix},
\end{aligned}$$

where $\mathbf{w}_k \sim \mathcal{N}\big(0, \mathrm{I}\big)$ and $\sqrt{\cdot}$ is a suitable matrix square-root factor. The sequence of distributions over beliefs computed through the belief dynamics is illustrated in Fig. 4.2a.

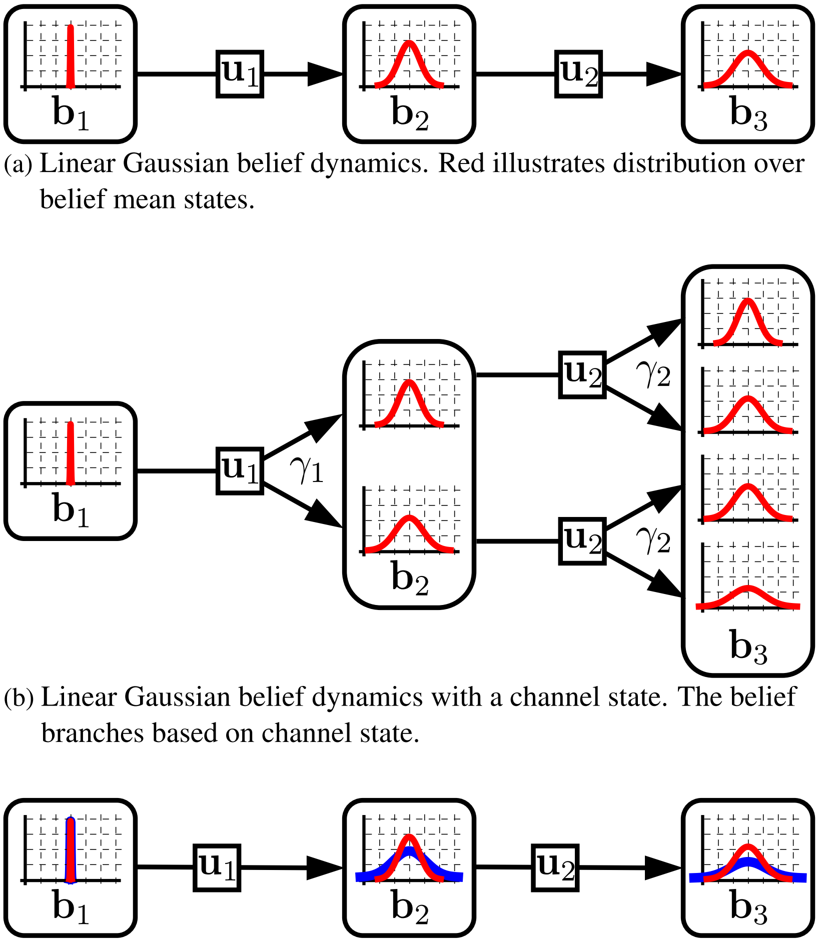
4.2.3 Belief Dynamics with Uncertain Channel State

Prior belief space planning methods (as above) assume that the set of future observation events is known. Whether an observation is made or not, however, is often not known before execution time. For example, the set of range observations that occur in an underwater acoustic network depends on the (unknown) set of successful acoustic transmissions.

To address this, we introduce a binary channel state variable, $\gamma_k \sim$ Bernoulli(λ_k), which models the event that an observation is received at the kth time index. Sinopoli et al. (2004) introduced an equivalent model for studying convergence properties of the EKF error covariance in sensor networks with fading channels. We modify the observation model (4.2) by conditioning on the channel state variable 'switch'

$$p(\mathbf{z}_k|\mathbf{x}_k, \gamma_k) = \mathcal{N}\big(h(\mathbf{x}_k, 0), \gamma_k \mathrm{N}_k + (1 - \gamma_k)\sigma^2 \mathrm{I}\big), \qquad (4.10)$$

Figure 4.2 Illustration of belief dynamics. (a) represents standard Gaussian belief dynamics, resulting in a distribution over possible mean states. Covariance in this case is deterministic. (b) depicts the belief dynamics evolution including an unknown channel state—the belief state branches for every possible channel state. Given the sequence of channel states, the belief dynamics are equivalent to (a). (c) shows our approximation of (b), which can be tractably computed.



(a) Linear Gaussian belief dynamics. Red illustrates distribution over belief mean states.

(b) Linear Gaussian belief dynamics with a channel state. The belief branches based on channel state.

(c) Approximate linear Gaussian belief dynamics with a channel state. Blue represents a distribution over covariance.

where we take the limit as $\sigma^2 \to \infty$. If the observation is received ($\gamma_k = 1$), the revised model is exactly (4.2). When the observation is not received ($\gamma_k = 0$), the observation has infinite uncertainty or, equivalently, zero information. Note that the observation model is again represented by a Gaussian distribution.

The belief now represents the posterior distribution over state given all channel states in addition to observations and control inputs

$$\mathbf{b}_k = p(\mathbf{x}_k|\mathbf{\Gamma}_{1:k}, \mathbf{Z}_{1:k}, \mathbf{U}_{1:k-1}), \tag{4.11}$$

where $\mathbf{\Gamma}_{1:k} = \{\gamma_i\}_{i=1}^k$. By leveraging the Markov property of the state transition model, the belief dynamics can be computed with a recursive Bayes filter

$$\begin{aligned}\mathbf{b}_{k+1} &= p(\mathbf{x}_{k+1}|\mathbf{\Gamma}_{1:k+1}, \mathbf{Z}_{1:k+1}, \mathbf{U}_{1:k}) \\ &= \eta p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1}, \gamma_{k+1}) p(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathbf{u}_k, \mathbf{\Gamma}_{1:k}, \mathbf{Z}_{1:k}),\end{aligned} \tag{4.12}$$

where η is a normalization constant and we assume that the set of channel states is independent of the state, i.e., $p(\gamma_i|\mathbf{x}_i) = p(\gamma_i)$. While the channel state may certainly depend on the underlying state, we have assumed independence to simplify the belief dynamics and so that the belief is not informed by whether an observation is received or not—the channel model is only used at planning time and is uninformative during execution.

We track the evolution of belief states (4.12) with an EKF. We evaluate the state prediction and measurement update in the limit by substituting the observation model (4.10) into the Bayes filter (4.12) and taking the limit as $\sigma^2 \to \infty$. The EKF update given the current belief closely follows the standard EKF update

$$\hat{\mathbf{x}}_{k+1} = f(\hat{\mathbf{x}}_k, \mathbf{u}_k, 0) + \gamma \mathrm{K}\Big(\mathbf{z} - h\big(f(\hat{\mathbf{x}}_k, \mathbf{u}_k, 0), 0\big)\Big) \tag{4.13}$$

$$\begin{aligned}\Sigma_{k+1} &= (\mathrm{I} - \gamma \mathrm{K}\mathrm{H})\bar{\Sigma} \qquad (4.14) \\ \mathrm{F} &= \frac{\partial}{\partial \mathbf{x}} f(\hat{\mathbf{x}}_k, \mathbf{u}_k, 0) \\ \mathrm{H} &= \frac{\partial}{\partial \mathbf{x}} h(f(\hat{\mathbf{x}}_k, \mathbf{u}_k, 0), 0)) \\ \bar{\Sigma} &= \mathrm{F}\Sigma_k\mathrm{F}^\top + \mathrm{M}_k \\ \mathrm{K} &= \bar{\Sigma}\mathrm{H}^\top(\mathrm{H}\bar{\Sigma}\mathrm{H}^\top + \mathrm{N}_k)^{-1},\end{aligned}$$

where we have dropped the subscripts on $\mathbf{z}_{k+1}$ and γ_{k+1} for brevity. Sinopoli et al. (2004) present the equivalent update, although derived by first constructing the joint distribution over predicted state and observation and then conditioning on the observation. Within the update equations, the channel state variable multiplies the Kalman gain, resulting in an intuitive behavior—the standard Kalman update when the measurement is present, and an uncorrected process prediction otherwise.

The belief dynamics defined within the EKF update are random in the current observation and channel state, $\mathbf{z}$ and γ, respectively. For tractable planning, we approximate the belief state transition with a Gaussian transition model

$$\mathbf{b}_{k+1} = g(\mathbf{b}_k, \mathbf{u}_k) + \mathrm{W}(\mathbf{b}_k, \mathbf{u}_k)\mathbf{w}_k, \tag{4.15}$$

where $\mathbf{w}_k \sim \mathcal{N}\big(0, \mathrm{I}\big)$. Since γ is not Gaussian, we compute the above expression by moment matching, i.e., the first two moments of (4.15) are set to that of (4.13)–(4.14),

$$
\begin{aligned}
g(\mathbf{b}_k, \mathbf{u}_k) &= \mathrm{E}[\mathbf{b}_{k+1}] = \mathrm{E}\begin{bmatrix} \hat{\mathbf{x}}_{k+1} \\ \mathrm{vec}(\Sigma_{k+1}) \end{bmatrix} \qquad (4.16)\\
&= \begin{bmatrix} f(\hat{\mathbf{x}}_k, \mathbf{u}_k, 0) \\ \mathrm{vec}(\bar{\Sigma} - \lambda \mathrm{KH}\bar{\Sigma}) \end{bmatrix} \\
\mathrm{W}(\mathbf{b}_k, \mathbf{u}_k) &= \sqrt{\mathrm{Var}[\mathbf{b}_{k+1}]} = \sqrt{\mathrm{Var}\begin{bmatrix} \hat{\mathbf{x}}_{k+1} \\ \mathrm{vec}(\Sigma_{k+1}) \end{bmatrix}} \qquad (4.17)\\
&= \begin{bmatrix} \sqrt{\lambda \mathrm{KH}\bar{\Sigma}} & \\ & \sqrt{\lambda(1-\lambda)\mathbf{v}\mathbf{v}^\top} \end{bmatrix} \\
\mathbf{v} &= \mathrm{vec}(\mathrm{KH}\bar{\Sigma}),
\end{aligned}
$$

where $\mathrm{E}[\gamma] = \lambda$. The evolution of beliefs is illustrated in Fig. 4.2b and Fig. 4.2c.

The belief dynamics formulation derived in Section 4.2.2 follows a similar form, although does not consider random observation events. Without the channel state variable, the portion of the belief state vector corresponding to the state covariance matrix is deterministic, i.e., no random variables appear in the EKF covariance update when observation events are known. Here, the covariance depends on γ, and is therefore random. When an observation is guaranteed to be obtained, however, $\lambda = 1$ and (4.15) is equivalent to (4.9).

The belief dynamics (4.15) represent the predictive transition of a belief state with random observation events. Kim and Eustice (2013) and Indelman et al. (2015) both considered the related problem of planning with unknown loop-closure events within active SLAM. Both parameterize Gaussian beliefs in the information (inverse covariance) form for which the measurement update is a simple additive step. Indelman et al. (2015) (approximately) marginalize over γ in the posterior, as opposed to taking the expectation with respect to γ, resulting in a deterministic information (covariance) matrix. It is also worth noting that the belief dynamics are not invariant to belief parameterization. In the information form, as in (Kim and Eustice, 2013; Indelman et al., 2015), the measurement information is scaled by λ, whereas, here, the Kalman gain is scaled instead.

4.2.4 Belief Space Optimization Motion Planning

We use the belief dynamics defined in (4.15) to compute a locally optimal solution to the cost function defined by (4.4). Previously, van den Berg et al. (2012) solved a similar objective using dynamic programming to obtain a control policy mapping belief state to control action. Approaching the optimization as a batch process instead allows us to easily optimize over path parame-

Algorithm 5 Belief space trajectory optimization.

```
Require: $\mathbf{b}_1, \bar{\mathbf{U}}^{(0)}$ {initial belief and nominal control}
while not converged do
    $\bar{\mathbf{b}}_{k+1}^{(i)}, \mathrm{A}_k, \mathrm{B}_k, \mathrm{W}_k = \texttt{execute}(\bar{\mathbf{b}}_k^{(i)}, \bar{\mathbf{u}}_k^{(i)})$ for all $k$
    $\mathrm{G}, \mathrm{J}, \mathrm{W} = \texttt{construct\_system}(\{\bar{\mathbf{b}}_k, \mathrm{A}_k, \mathrm{B}_k, \mathrm{W}_k\})$
    $\delta\mathbf{U}^* = \texttt{solve\_objective}(\mathrm{Q}, \mathrm{R}, \mathrm{G}, \mathrm{J}, \mathrm{W})$
    $\bar{\mathbf{U}}^{(i+1)} = \texttt{line\_search}(\bar{\mathbf{U}}^{(i)}, \delta\mathbf{U}^*)$
end while
return $\bar{\mathbf{U}}^{(i)}$
```

ters assuming a known path-following controller—desirable for planning practical trajectories for AUVs.

The cost function (4.4) can be expressed in batch as a function of the stacked vectors of beliefs and controls

$$\min_{\mathbf{B},\mathbf{U}} \; \mathrm{E}\left[\mathbf{B}^\top \mathrm{Q} \mathbf{B} + \mathbf{U}^\top \mathrm{R} \mathbf{U}\right], \tag{4.18}$$

where we have dropped the subscripts on $\mathbf{B}_{1:L}$ and $\mathbf{U}_{1:L-1}$, $\mathrm{Q} = \mathrm{diag}(\mathrm{Q}_1, \ldots, \mathrm{Q}_L)$, and $\mathrm{R} = \mathrm{diag}(\mathrm{R}_1, \ldots, \mathrm{R}_{L-1})$. We take a sequential approach—compute the set of beliefs as a function of control actions, substitute into the objective function, and solve for the minimizing set of control inputs.

We first linearize each belief transition around a nominal trajectory $\bar{\mathbf{B}}_{1:L}$ and control sequence $\bar{\mathbf{U}}_{1:L-1}$

$$\delta\mathbf{b}_{k+1} \approx \mathrm{A}_k \delta\mathbf{b}_k + \mathrm{B}_k \delta\mathbf{u}_k + \mathrm{W}_k \mathbf{w}_k, \tag{4.19}$$

where $\delta\mathbf{b}_k = \mathbf{b}_k - \bar{\mathbf{b}}_k$, $\delta\mathbf{u}_k = \mathbf{u}_k - \bar{\mathbf{u}}_k$, and A_k and B_k are the Jacobians of $g(\mathbf{b}_k, \mathbf{u}_k)$ with respect to $\mathbf{b}_k$ and $\mathbf{u}_k$, respectively. Note that $\bar{\mathbf{B}}_{1:L}$ is obtained by executing $g(\,\cdot\,)$ along the nominal control trajectory given $\mathbf{b}_1$. We also linearize $\mathrm{W}(\mathbf{b}_k, \mathbf{u}_k)$ such that the ith column can be written

$$\mathrm{W}_k^{(i)} \approx \mathrm{F}_k^{(i)} \delta\mathbf{b}_k + \mathrm{G}_k^{(i)} \delta\mathbf{u}_k + \bar{\mathrm{W}}_k^{(i)} \tag{4.20}$$

and $\bar{\mathrm{W}}_k^{(i)} = \mathrm{W}^{(i)}(\bar{\mathbf{b}}_k, \bar{\mathbf{u}}_k)$. Note that W_k is a function of the channel model with parameter λ_k. In Section 4.4 we discuss the form of λ_k, which depends on the state. During planning, we evaluate λ_k about the nominal trajectory.

We can write the stacked vectors of belief state and control deviations as

$$\delta\mathbf{B} = [\delta\mathbf{b}_1^\top, \dots, \delta\mathbf{b}_L^\top]^\top \tag{4.21}$$

$$\delta\mathbf{U} = [\delta\mathbf{u}_1^\top, \dots, \delta\mathbf{u}_{L-1}^\top]^\top. \tag{4.22}$$

The batch belief dynamics are then expressed by concatenating (4.19) over the time horizon

$$\begin{aligned}\delta\mathbf{B} &= \mathrm{A}\delta\mathbf{B} + \mathrm{B}\delta\mathbf{U} + \mathrm{W}\mathbf{w}\\ \delta\mathbf{B} &= \underbrace{(\mathrm{I}-\mathrm{A})^{-1}\mathrm{B}}_{\mathrm{G}}\,\delta\mathbf{U} + \underbrace{(\mathrm{I}-\mathrm{A})^{-1}\mathrm{W}}_{\mathrm{J}}\,\mathbf{w},\end{aligned} \tag{4.23}$$

where $\mathbf{w}$ is the stacked vector of noise perturbations $\mathbf{w}_k$ and the stacked Jacobians are defined

$$\mathrm{A} = \begin{bmatrix} 0 & & & \\ \mathrm{A}_1 & 0 & & \\ 0 & \ddots & \ddots & \\ & 0 & \mathrm{A}_{L-1} & 0 \end{bmatrix} \tag{4.24}$$

$$\mathrm{B} = \begin{bmatrix} 0 & & \\ \mathrm{B}_1 & 0 & \\ 0 & \ddots & \ddots \\ & 0 & \mathrm{B}_{L-1} \end{bmatrix} \tag{4.25}$$

$$\mathrm{W} = \begin{bmatrix} 0 & & \\ \mathrm{W}_1 & 0 & \\ 0 & \ddots & \ddots \\ & 0 & \mathrm{W}_{L-1} \end{bmatrix}. \tag{4.26}$$

We substitute the batch belief dynamics (4.23) into the cost function (4.18) and evaluate the expectation to obtain

$$\begin{aligned}&\min_{\delta\mathbf{U}} \left[(\mathrm{G}\delta\mathbf{U} + \bar{\mathbf{B}})^\top\mathrm{Q}(\mathrm{G}\delta\mathbf{U} + \bar{\mathbf{B}}) + (\delta\mathbf{U} + \bar{\mathbf{U}})^\top\mathrm{R}(\delta\mathbf{U} + \bar{\mathbf{U}}) + \mathrm{tr}(\mathrm{J}^\top\mathrm{Q}\mathrm{J})\right]\\ =&\min_{\delta\mathbf{U}} \left[(\mathrm{G}\delta\mathbf{U} + \bar{\mathbf{B}})^\top\mathrm{Q}(\mathrm{G}\delta\mathbf{U} + \bar{\mathbf{B}}) + (\delta\mathbf{U} + \bar{\mathbf{U}})^\top\mathrm{R}(\delta\mathbf{U} + \bar{\mathbf{U}})\right.\\ &\left.+ \textstyle\sum_i \big((\mathrm{F}^{(i)}\mathrm{G} + \mathrm{G}^{(i)})\delta\mathbf{U} + \bar{\mathrm{W}}^{(i)}\big)^\top(\mathrm{I}-\mathrm{A})^{-\top}\mathrm{Q}(\mathrm{I}-\mathrm{A})^{-1}\big((\mathrm{F}^{(i)}\mathrm{G} + \mathrm{G}^{(i)})\delta\mathbf{U} + \bar{\mathrm{W}}^{(i)}\big)\right].\end{aligned}$$

This cost function is quadratic in the control update term, $\delta\mathbf{U}$, and can therefore be minimized by taking the derivative of the cost and setting to zero

$$\delta \mathbf{U}^* = -\mathrm{D}^{-1}\mathbf{E}, \tag{4.27}$$

where

$$\begin{aligned}
&\mathrm{D} = \mathrm{G}^\top \mathrm{Q}\mathrm{G} + \mathrm{R} + \textstyle\sum_i (\mathrm{F}^{(i)}\mathrm{G} + \mathrm{G}^{(i)})^\top \mathrm{C}(\mathrm{F}^{(i)}\mathrm{G} + \mathrm{G}^{(i)})\\
&\mathbf{E} = \mathrm{G}^\top \mathrm{Q}\delta\mathbf{B} + \textstyle\sum_i (\mathrm{F}^{(i)}\mathrm{G} + \mathrm{G}^{(i)})^\top \mathrm{C}\bar{\mathrm{W}}^{(i)}\\
&\mathrm{C} = (\mathrm{I} - \mathrm{A})^{-\top}\mathrm{Q}(\mathrm{I} - \mathrm{A})^{-1}.
\end{aligned}$$

We leverage the sparsity structure of $\mathrm{W}^{(i)}$ to efficiently compute the summation terms. Note that the Hessian is positive definite by construction because of the cost weight matrices. The updated control vector is computed by adding the update to the control at the current iteration

$$\mathbf{U}^{(j+1)} = \mathbf{U}^{(j)} + \epsilon\delta\mathbf{U}^*, \tag{4.28}$$

where ϵ defines the step length based on a simple back-tracking line search. The control update step is repeated until a convergence criteria is satisfied. The full algorithm is outlined in Algorithm 5.

4.2.5 Parameterized Trajectory Optimization

In the previous section, we described a method to compute a sequence of open-loop control actions over a finite horizon. He et al. (2011) proposed planning over macro actions (a collection of sequential actions) to reduce planning complexity, and allow planning over a longer time horizon. Planning over path parameters, similar to macro actions, has also been studied by both Sim et al. (2004) and Kollar and Roy (2008). Here, we apply an optimization over parameterized paths for planning.

For AUV field trials, we desire simple, practical trajectories in order to easily monitor vehicle health and progress. To address this, we assume that each vehicle employs a feedback path-following controller and optimize over parameters that define a trajectory. For example, a center position and radius define a circular path. The control is then a (known) function of the estimated vehicle state and path parameter vector $\boldsymbol{\theta}$. The state transition model (4.1) can be expressed given the controller and parameter vector as

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k(\hat{\mathrm{x}}_k, \boldsymbol{\theta}), \mathbf{w}_k).$$

We iteratively update locally optimal path parameters in the same way control actions were computed. In this case, Jacobians are computed with respect to the path parameters instead of control actions. Finally, we penalize parameter weights (instead of control) within the cost function to ensure a positive definite Hessian. The number of parameters to specify a path is, in general, much smaller than the number of control actions. Therefore, there is also significant computational savings in optimizing over path parameters.

4.3 Planning for Cooperative Localization

Here, we detail the instantiation of the belief space planning algorithm developed in Section 4.2 to plan a server trajectory to localize a client vehicle. We require that the nominal client trajectory and path-following controller are available at planning time. We further assume that the client vehicle is able to reconstruct the centralized server-client estimator, as with the OSM (Chapter 2), and that communication latency is negligible.

The state space system is represented by the stacked server and client vehicle states $\mathbf{x}_k = [\mathbf{x}_{s_k}^\top, \mathbf{x}_{c_k}^\top]^\top$. The state space dynamics of each vehicle are independent and written

$$\begin{bmatrix} \mathbf{x}_{s_{k+1}} \\ \mathbf{x}_{c_{k+1}} \end{bmatrix} = \begin{bmatrix} f_s\big(\mathbf{x}_{s_k}, \mathbf{u}_{s_k}, \mathbf{w}_{s_k}\big) \\ f_c\big(\mathbf{x}_{c_k}, \mathbf{u}_{c_k}, \mathbf{w}_{c_k}\big) \end{bmatrix}, \tag{4.29}$$

where $\mathbf{u}_{s_k}$ represents the only decision variable control action within the planning problem since $\mathbf{u}_{c_k}$ is computed by the client's path-following controller and a given mission plan.

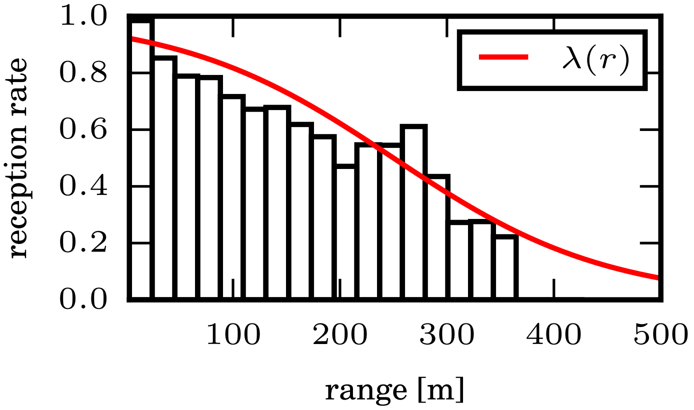
Relative range constraints constitute a nonlinear observation over the server-client state

$$z_{r_k} = \Big\| \mathbf{x}_{s_k} - \mathbf{x}_{c_k} \Big\|_2 + w_{r_k}, \tag{4.30}$$

where $w_{r_k} \sim \mathcal{N}\big(0, \sigma_k^2\big)$. In general, range observations will be a 3D slant range measurement (see Appendix B.2 for more details). Within a linear estimator, such as the EKF, range observations add information in one direction: along the vector between the server and client vehicle positions. For underwater navigation, depth is usually well instrumented, so we seek to minimize horizontal position errors. Therefore, the range measurement utility is highest when the server and client vehicles are far apart, i.e., the vector of added information is approximately parallel to the horizontal plane.

In all experiments presented in this chapter, we use a cost function that penalizes control action and client vehicle uncertainty. Unlike many planning problems, there is no desired terminal state for the server. We can define a quadratic stage cost penalizing client uncertainty fitting the form of

Figure 4.3 Reception rate data and channel state model. The histogram represents the frequency of successfully received acoustic broadcasts. The channel model, λ, illustrates (4.31) tuned to the environment.



the objective in (4.4):

$$
\begin{aligned}
\operatorname{tr}(\Sigma_{c_k}) &= \mathbf{m}^\top \mathbf{b}_k \\
\operatorname{tr}^2(\Sigma_{c_k}) &= \mathbf{b}_k^\top \mathbf{m}\mathbf{m}^\top \mathbf{b}_k \\
&= \mathbf{b}_k^\top Q \mathbf{b}_k,
\end{aligned}
$$

where $\mathbf{m}^\top$ is a row vector that sums the diagonal elements of the covariance matrix corresponding to the client vehicle from the belief state.

4.4 Numerical Simulations

We first present an empirical channel model informed by reception statistics collected during AUV field trials. We then validate our planning framework through several simulated scenarios where we employ this channel model for planning.

4.4.1 Empirical Channel Model

In this section, we review how we compute the distribution over channel states $p(\gamma_k = 1) = \lambda_k$. Physics-based and empirical models exist for studying communication error in underwater channels. One such empirical channel model proposed by Stojanovic (2007) was used by Hollinger et al. (2015) for studying multiple vehicle coordination. Within the model, the probability of successful transmission is written as a function of the transmit frequency, power, and environmental conditions. In this chapter, we are interested in showing how a channel model can be integrated into the planning framework. As such, we employed a simplified model where the probability of successful reception falls off as a sigmoid with the horizontal range, $r = \left\|\mathbf{x}_s - \mathbf{x}_c\right\|_2$, between server and client

$$\begin{aligned} \lambda_k(r) &= p(\gamma_k = 1) \\ &= \left[1 + \exp\left\{ \frac{1}{\tau}(r - \bar{r}) \right\} \right]^{-1}, \end{aligned} \tag{4.31}$$

where $\bar{r}$ is the range at which reception probability is $\frac{1}{2}$ and τ is the length scale that defines the rate at which reception decreases.

We manually fit the parameters of this simple model to data collected over the course of several field trials. We fielded two Iver2 AUVs and a surface ship. Each vehicle was equipped with a WHOI Micro-modem and co-processor board (Freitag et al., 2005b), $25\ \mathrm{kHz}$ BTech Acoustics 2RCL transducer, and a synchronous-clock reference. Each vehicle periodically transmitted Micro-modem Rate 1 and 2 data packets which each consist of three $64\ \mathrm{B}$ PSK encoded data frames. The vehicle hardware and software systems are reviewed in Appendix C.

Over the course of three trials, we collected over 3000 range observations. The trials were conducted in shallow water (roughly $20\ \mathrm{m}$) with all transducers suspended at approximately $10\ \mathrm{m}$. While the true inter-vehicle range is unknown, we estimated range from the filtered online vehicle position estimates in post process. The inter-vehicle range did not exceed $400\ \mathrm{m}$. Fig. 4.3 shows the channel model (4.31) overlaid on a histogram of reception frequencies for varying relative range, where the channel model matches the characteristic behavior of the true channel.

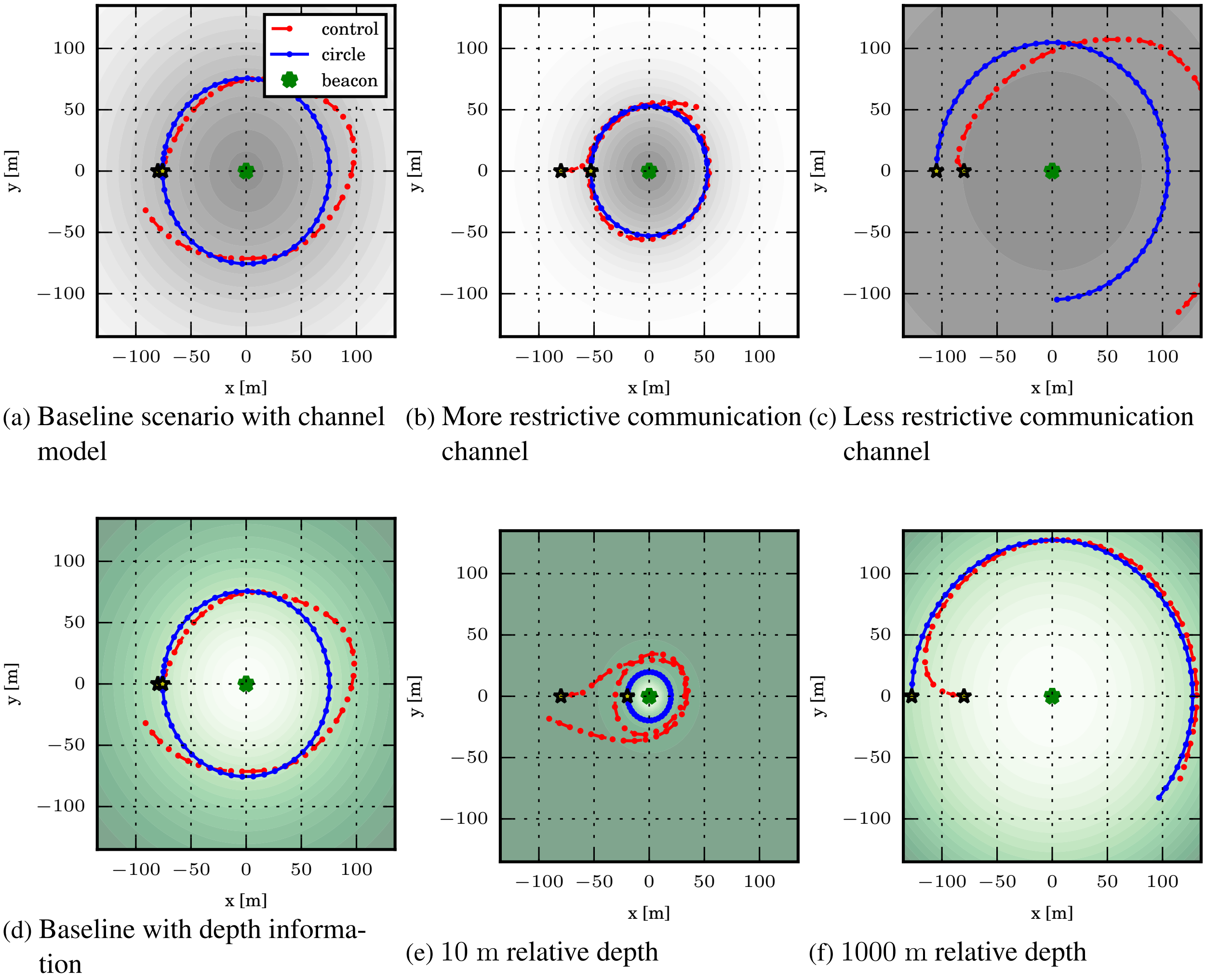
4.4.2 Simulation: Static Beacon Localization

Our first set of simulations helps illustrate the planning problem for a simple scenario—compute a server trajectory to localize a static beacon. This is a special case of server-client localization in which the client vehicle is fixed, and, hence, does not accrue any additional uncertainty over time. This problem is akin to surveying LBL network beacons. Jakuba et al. (2008) recommend that the topside ship circle the beacons at a fixed radius (up to water depth), providing a heuristic to balance position information and communication.

We assume that the server vehicle follows a simple unicycle model. The server translates a fixed distance, Δ, for each timestep with control defined by a change in heading, u_k, perturbed by zero-mean independent Gaussian noise w_k. The server state is parameterized by its XY horizontal position and heading θ:

$$\begin{bmatrix} x_{s_{k+1}} \\ y_{s_{k+1}} \\ \theta_{s_{k+1}} \end{bmatrix} = \begin{bmatrix} x_{s_k} + \Delta\cos(\theta_{s_k} + u_k + w_k) \\ y_{s_k} + \Delta\sin(\theta_{s_k} + u_k + w_k) \\ \theta_{s_k} + u_k + w_k \end{bmatrix}.$$

Figure 4.4 Example planning for static beacon localization. Top row shows planned trajectories with the underlying channel model using the baseline relative depth (color gradient in top row represents communication channel quality, darker is better). Bottom row plots range information using the baseline channel model (color gradient in bottom row represents range information, darker is better).



(a) Baseline scenario with channel model (b) More restrictive communication channel (c) Less restrictive communication channel

(d) Baseline with depth information (e) 10 m relative depth (f) 1000 m relative depth

The beacon (client) state includes simply its XY position. Following each motion step, the server vehicle observes its own position and orientation and the range to the beacon (depending on the channel state).

We optimized over open-loop control actions and parameterized circular paths. For the optimal control trajectories, the server vehicle was initialized in the same position for each trial. For the parametric circular trajectories, the initial server pose is a function of the radius. Fig. 4.4 illustrates planned server trajectories for varying channel model and relative depth parameters over 50 planning steps. This scenario illustrates the tradeoff between a good communication channel or informative range observations. This tradeoff is endemic to all range-only cooperative underwater planning. The top row in Fig. 4.4 demonstrates that our planning framework responds to varying channel parameters. The bottom row shows that information added by each range measurement changes as a function of relative depth between the server and beacon. As the relative depth increases, the server must move farther away in order to localize the beacon.

These simulations also help to illustrate the utility of parameterized trajectories, provided that the parameters are well informed by experience. For this simplified scenario, the planned control and parameterized circle paths do not differ significantly and agree with heuristic LBL localization methods.

4.4.3 Simulation: Server-Client Localization

In the second set of simulated trials, we consider two mobile AUVs—a server and client—and show that the planning framework scales to realistic problem sizes. Each vehicle's state is parameterized by its XY horizontal position and heading θ. We assume a constant forward velocity so that over a fixed time step, the change in position is Δ. The state dynamics follow a simplified bicycle model with steering angle control input u_k:

$$\begin{bmatrix} x_{s_{k+1}} \\ y_{s_{k+1}} \\ \theta_{s_{k+1}} \end{bmatrix} = \begin{bmatrix} x_{s_k} + (\Delta + w_{\Delta_k}) \cos(\theta_{s_k}) \\ y_{s_k} + (\Delta + w_{\Delta_k}) \sin(\theta_{s_k}) \\ \theta_{s_k} + \alpha(\Delta + w_{\Delta_k}) \tan(u_k + w_{u_k}) \end{bmatrix},$$

where α is the reciprocal 'wheel-base' length and $w_{\Delta_k}$and w_{u_k} represent Gaussian noise perturbations in the step length and control, respectively.

We are provided the client survey trajectory at planning time. The estimated survey time is 76 min and is represented by roughly 5000 state transitions. The relative vehicle depth is 100 m and the channel model parameters are $\bar{r} = 250 \text{ m}$ and $\tau = 125 \text{ m}$. By optimizing over 'diamond' path parameters (center position, width, height), we reduce the complexity of the problem. Each vehicle relies on a pure-pursuit path-following controller. Fig. 4.5 depicts the planned trajectory for each vehicle.

We also compared the optimized path to a naïvely planned path, which we have employed in several previous cooperative localization field trials. We set the parameters of the naïve path to be centered on the client survey area and span the length and width of the survey area. Table 4.1 summarizes the performance of each path averaged over 100 simulated trials. Overall, the optimized path receives more OWTTs per trial than the naïvely planned trajectory. Moreover, the client uncertainty (indicated by the determinant and trace in the table) is far lower for the optimized path. The optimized server path remains closer to the center of the client survey, and thus is able to more reliably make OWTT observations.

Figure 4.5 Optimized (red) and naïvely planned (green) server trajectories.

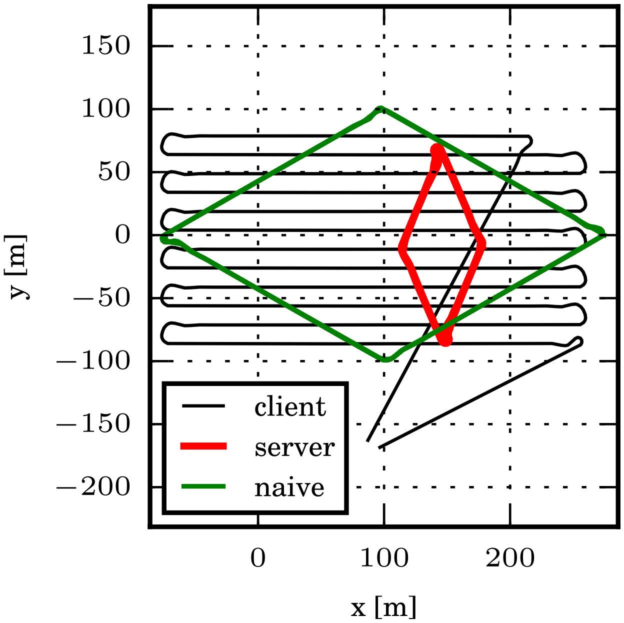


Table 4.1 Average performance comparing an optimized path and a naïvely planned path over 100 simulated trials.

METHOD	OWTTS RECEIVED	$\lvert\Sigma_{c_k}\rvert$	$\mathrm{tr}(\Sigma_{c_k})$
Proposed (optimized)	67.6	0.40	69.56
Naïvely planned	62.9	1.60	402.34

4.5 Chapter Summary

Cooperative underwater localization enables teams of vehicles to exploit range observations to improve their navigation estimates. The quality of the estimate, however, depends on the relative position of the vehicles. Therefore, planning relative paths can benefit online navigation performance. We have presented a probabilistic channel model that represents randomness in observation events. The channel model holds potential utility in other applications involving unknown measurement acquisition. We then integrated the model into a belief space planning framework that optimizes either open-loop control actions or path parameters in order to produce practical AUV trajectories.

Chapter 5

Conclusions and Future Work

AUVs extend the abilities of ocean researchers to observe the world in a way not previously possible—at greater depths and over larger spatial and time scales. Accurate underwater localization is a key competency for safe operation and to precisely geo-reference scientific data. Cooperative localization has recently been enabled by more prevalent underwater communication systems, but faces several design challenges in transitioning to the underwater domain.

Underwater cooperative localization provides an infrastructure-free means of accurate bounded-error position estimation while subsea. The design of such systems must address how to share information consistently and how to communicate such information over a lossy and low-bandwidth underwater acoustic communication channel. Prior to this thesis, estimator consistency had been addressed by *expanding* the state representation to explicitly track correlated information, i.e., by using delayed state or pose-graph models. While providing a solution to the problem of consistency, a larger state representation did not necessarily lend itself to communication over a bandwidth limited channel. Additionally, the estimate quality produced by range-only cooperative localization is strongly influenced by the network geometry. No previous literature has addressed planning informative relative trajectories that considers both the quality of information and the fidelity of the acoustic channel.

5.1 Contributions

The specific contributions of this thesis include:

- We proposed the OSM (Chapter 2) for cooperative server-client localization. The DEIF, which noted that local platform information updates take a sparse form in a pose-graph framework, served as the basis for this work. We showed that we could further exploit the structure of the underlying pose-graph in order to broadcast the server information to the client. We also provided experimental evaluation for the first-ever real-time exact OWTT cooperative localization.

- In Chapter 3, we presented an algorithm for peer-to-peer cooperative localization. We borrowed ideas from the OSM in order to effectively share local information that scales to many vehicles. While the OSM leveraged the structure of the MRF, here, we exploited the factor-level structure of the local information. We showed that odometry factors can be composed and then decomposed for efficient communication. We also applied approximate marginalization to reduce the number of pose nodes that must be shared. We provided a comparative experimental evaluation that validated our approach and showed improvement when compared to a similar method.

- We proposed an motion planning algorithm for server-client localization that includes a probabilistic channel model. The algorithm is based on prior work in belief space planning methods, which provide a solution to more general POMDPs. Solving the POMDP is made tractable by applying linear Gaussian techniques. We model acoustic communication success as a spatially-varying Bernoulli random variable, termed the channel state variable. We modify the measurement observation model to include the channel state variable, and then derive the corresponding EKF for such a model. The EKF propagation equations define the so-called belief dynamics, which provides a predictive model of client state and uncertainty given a server path. Finally, we optimize over server path parameters to obtain locally optimally server paths.

5.2 Future Work

While this thesis has provided novel contributions toward underwater cooperative localization, we recognize room for improvement. Further compelling areas of research are described below.

Cooperative Localization

- The methodologies reported in this thesis enable platforms to share their local navigation information. However, there is no mechanism for sharing information from the wider network, which would be useful for cascaded communication topologies, for example. For cascaded network topologies (or any tree topology), we propose approximate marginalization (as in Chapter 3) to include parent information as an additional factor for children platforms.

- We have only focused on handling direct vehicle-to-vehicle range constraints as derived from OWTT observation. Based on our factor-based localization framework, vehicles could potentially share additional loop-closure or other informative factors, e.g., from a visual SLAM system.

- Roumeliotis and Rekleitis (2004) examined theoretical bounds on uncertainty growth within cooperative networks as a function of network size. Their analysis places strong assumptions on the sensors available to each vehicle and only considers unconstrained communication. As future work, we propose extending their analysis to graph-based cooperative estimation frameworks with unreliable communication as in Chapter 3. This analysis would indicate expected localization performance for underwater vehicle cooperative networks, which could be useful for multi-vehicle survey design.

Planning for Localization

- Optimizing over path parameters is similar to using macro actions (He et al., 2011). It would be interesting to explore a richer set of actions (only diamond trajectories employed in Chapter 4) that could be sequentially combined, yet still easily computed.

- Our planning methodology addressed planning for a single server vehicle given a single client trajectory. A possible avenue for future work would be to simultaneously consider multiple server and client vehicles. Scaling to multiple vehicles is nontrivial since planning algorithm complexity is super linear in the number of vehicles.

- Informative planning for range-only localization can be accomplished using heuristic planners, e.g., German et al. (2012) proposed following the centroid of the client's remaining survey area. These heuristic planners can perform quite well in practice by capturing the intuitive notion that localization quality is tied to varying the relative angle between vehicles—in order to bound uncertainty in all directions—and were designed, no doubt, from user experience. In the same vein, learning algorithms may be capable of selecting good support trajectories given past performance. A learned behavior may have computational benefits that enable rapid replanning or may scale to many vehicles.

Appendix A

Probabilistic Graphical Models

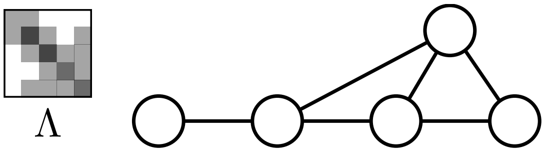
The connection between solutions to linear systems and graphical models has been long studied (Parter, 1961; Rose, 1972; Kschischang et al., 2001; Loeliger, 2004; Cowell et al., 2007) and helps to reveal the underlying structure in the SLAM problem. Fig. 1.2 illustrates the connection between the information matrix and the Markov random field (MRF) and the measurement Jacobian and the factor graph. Understanding the relationship between the SLAM problem and graph representation has led to the development of the current state-of-the-art.

Probabilistic graphical models capture conditional independence relationships within the joint distribution over a set of random variables. The joint distribution is often more simply expressed as a product over conditionally independent subsets. A random variable A is conditionally independent of C given B, i.e., $A \perp\!\!\!\perp C \mid B$, if $p(A \mid B, C) = p(A \mid B)$. Below, we provide a brief overview of the two graphical models with Gaussian noise models referenced in this work.

A.1 Gaussian Markov Random Fields

An MRF is a multivariate random variable $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ that can be represented by an undirected graphical model $G = (V, E)$ where V and E are the vertex (node) and edge sets, respectively. Two unique nodes $\mathbf{x}_i$ and $\mathbf{x}_j$ for $i, j \in V$ are adjacent if they share an edge, $\{i, j\} \in E$. The neigbors of $\mathbf{x}_i$ are the nodes with which $\mathbf{x}_i$ is adjacent, i.e., $\mathcal{N}(i) = \{\mathbf{x}_j | \{i, j\} \in E, \forall j \in V\}$.

Figure A.1 Illustration of a Gaussian MRF. The information matrix Λ corresponds exactly to the adjacency matrix of the graph.



Graph edges encode conditional relationships within the distribution—a variable node is independent of the rest of the graph given its neighbors,

$$\begin{aligned} p(\mathbf{x}_i|\mathbf{X}) &= p(\mathbf{x}_i|\mathbf{x}_{j\in\mathcal{N}(i)}, \mathbf{x}_{j\notin\mathcal{N}(i)}) \\ &= p(\mathbf{x}_i|\mathbf{x}_{j\in\mathcal{N}(i)}). \end{aligned}$$

A special case of MRFs are graphs that admit a clique (complete subgraph) factorization, i.e., the joint density can be factored

$$p(\mathbf{X}) = \prod_{C\in\mathcal{C}} \phi_C(\mathbf{X}_C), \tag{A.1}$$

where $\phi(\,\cdot\,)$ is termed a factor potential.

A multivariate Gaussian distribution is parameterized by its mean and covariance, or, in the natural form, by its information vector and matrix

$$\mathbf{X} \sim \mathcal{N}\big(\boldsymbol{\mu}, \Sigma\big) = \mathcal{N}^{-1}\big(\boldsymbol{\eta}, \Lambda\big).$$

The information matrix, Λ, is exactly the adjacency matrix of G corresponding to the MRF. More specifically, off-diagonal block entries of the information matrix are zero for pairs of vertices that do not share an edge

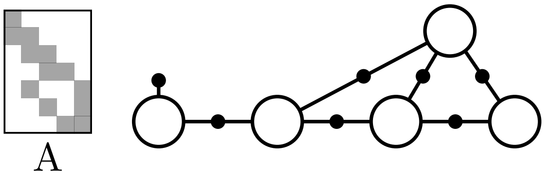
$$\{i, j\} \notin E \implies \Lambda_{ij} = 0. \tag{A.2}$$

This property is illustrated in Fig. A.1.

A.2 Factor Graphs

The factor graph is a bipartite graphical model consisting of variable nodes and factor nodes. Within the SLAM problem, factors or constraints typically represent observations connecting unknown pose or landmark variables. Like the MRF, the factor graph reveals the underlying factor-

Figure A.2 Illustration of a Gaussian graph. The measurement Jacobian A directly corresponds to the factor graph.



ization of the joint distribution

$$p(\mathbf{X}) \propto \prod_i \phi_i(\mathbf{X}_i) \tag{A.3}$$

where $\phi_i(\,\cdot\,)$ represents the the potential induced by the ith factor node over the set of variable nodes, $\mathbf{X}_i$.

The factor graph shares a direct correspondence with the measurement Jacobian—each factor corresponds to block row within the measurement Jacobian.

Appendix B

Vehicle and Observation Models

In this appendix, we review the 2D horizontal-plane odometry model used in the experiments presented in this dissertation. We also describe and discuss the use of 2D pseudo range observations derived from 3D slant range observations.

B.1 DVL Odometry Model

In this section, we outline the procedure to compute odometry updates given noisy DVL and heading data. We employ a first-order covariance approximation as developed by Eustice et al. (2011).

The vehicle horizontal-plane odometry (displacement) in the world frame between time indices i and j is computed

$$\Delta \mathbf{x}_{ij} = \int_{t_i}^{t_j} \mathrm{R}(\psi(t)) \mathbf{u}(t) dt, \tag{B.1}$$

where $\mathbf{u}(t) = [u(t), v(t)]^\top$ is the vector body-frame velocity (surge and sway), $\psi(t)$ is the vehicle heading, and negligible roll and pitch is assumed. The body to world rotation matrix is defined

$$\mathrm{R}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}. \tag{B.2}$$

We can approximate (B.1) given discrete heading and velocity samples by forward Euler numerical integration

$$\Delta \mathbf{x}_{ij} = \Delta t \sum_k \mathrm{R}(\psi_k) \mathbf{u}_k, \tag{B.3}$$

with sample index k and sample period Δt.

DVL velocity and heading observations are nondeterministic. We model the random noise perturbations by independent additive zero-mean Gaussian noise

$$\begin{aligned} \mathbf{z}_{\mathbf{u}_k} &= \mathbf{u}_k + \mathbf{w}_{u_k}, & \mathbf{w}_{u_k} &\sim \mathcal{N}\big(0, \mathrm{R}_{u_k}\big) \\ z_{\psi_k} &= \psi_k + w_{\psi_k}, & w_{\psi_k} &\sim \mathcal{N}\big(0, \sigma^2_{\psi_k}\big). \end{aligned}$$

The observed odometry is then computed as the summation of noisy heading and velocity measurements

$$\begin{aligned} \mathbf{z}_{ij} &= \Delta t \sum_k \mathrm{R}(z_{\psi_k}) \mathbf{z}_{\mathbf{u}_k} \\ &= \Delta t \sum_k \mathrm{R}(\psi_k + w_{\psi_k})(\mathbf{u}_k + \mathbf{w}_{\mathbf{u}_k}). \end{aligned} \tag{B.4}$$

To approximate odometry observation statistics, we first compute the first-order Taylor series expansion of the rotation matrix in (B.4)

$$\begin{aligned} \mathrm{R}(\psi_k + w_{\psi_k}) &\approx \begin{bmatrix} \cos\psi_k - w_{\psi_k}\sin\psi_k & -\sin\psi_k - w_{\psi_k}\cos\psi_k \\ \sin\psi_k + w_{\psi_k}\cos\psi_k & \cos\psi_k - w_{\psi_k}\sin\psi_k \end{bmatrix} \\ &= \underbrace{\begin{bmatrix} \cos\psi_k & -\sin\psi_k \\ \sin\psi_k & \cos\psi_k \end{bmatrix}}_{\mathrm{R}(\psi_k)} - w_{\psi_k} \underbrace{\begin{bmatrix} \sin\psi_k & \cos\psi_k \\ -\cos\psi_k & \sin\psi_k \end{bmatrix}}_{\mathrm{R}'(\psi_k)}. \end{aligned} \tag{B.5}$$

We then substitute (B.5) into (B.4)

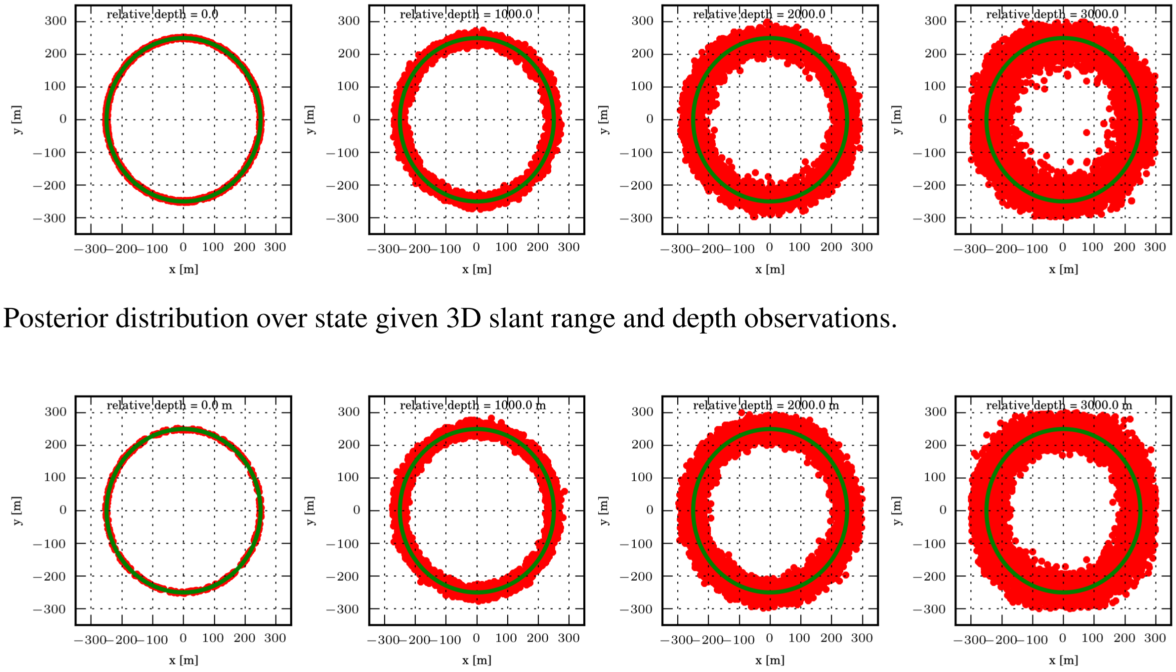
$$\mathbf{z}_{ij} \approx \Delta t \sum_k \left(\mathrm{R}(\psi_k) - w_{\psi_k}\mathrm{R}'(\psi_k)\right)\left(\mathbf{u}_k + \mathbf{w}_{\mathbf{u}_k}\right). \tag{B.6}$$

Finally, the approximate odometry observations statistics are determined

$$\begin{aligned} \mathrm{E}[\mathbf{z}_{ij}] &= \Delta t \sum_k \mathrm{R}(\psi_k)\mathbf{u}_k \\ &= \Delta x_{ij} && \text{(B.7)} \\ \mathrm{R}_{ij} &= \mathrm{E}\left[(\mathbf{z}_{ij} - \mathrm{E}[\mathbf{z}_{ij}])(\mathbf{z}_{ij} - \mathrm{E}[\mathbf{z}_{ij}])^\top\right] \\ &= \Delta t^2 \sum_k \left(\mathrm{R}(\psi_k)\mathrm{R}_{\mathbf{u}_k}\mathrm{R}(\psi_k)^\top + \sigma^2_{\psi_k}\mathrm{R}'(\psi_k)\left(\mathbf{u}_k\mathbf{u}_k^\top + \mathrm{R}_{\mathbf{u}_k}\right)\mathrm{R}'(\psi_k)^\top\right) && \text{(B.8)} \end{aligned}$$

where R_{ij} is the approximate odometry noise covariance.

Figure B.1 Posterior distribution over client state given a single range to a beacon at the origin and relative depth. The horizontal range (green circle) is held constant while the relative depth is increased from left to right. Note that the distribution becomes more uncertain (less informative) as relative depth increases.



(a) Posterior distribution over state given 3D slant range and depth observations.

(b) Approximate posterior distribution over state given 2D pseudo range and depth observations.

B.2 OWTT Observation Model

Many estimation and planning frameworks for OWTT localization project 3D slant ranges into the 2D horizontal plane to obtain a set of *pseudo* ranges used with a fixed measurement covariance. Here, we give a formulation for the pseudo range as a function of the measured slant range and relative depth between the server and client and provide a first-order covariance approximation. Fig. B.1 illustrates that increasing the relative depth between vehicles reduces the horizontal plane information.

We model a slant range between 3D client and server states $\mathbf{x}_c, \mathbf{x}_s \in \mathbb{R}^3$ as

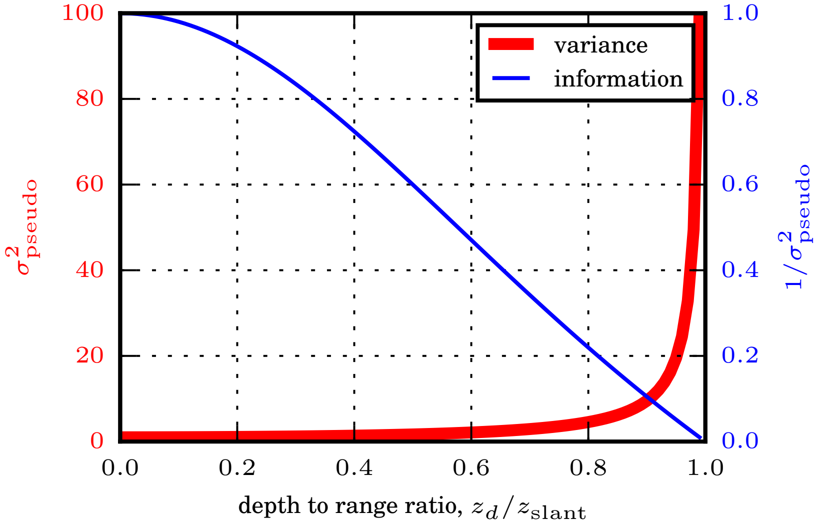
$$z_{\text{slant}} = \left\| \mathbf{x}_s - \mathbf{x}_c \right\|_2 + w_{\text{slant}} \tag{B.9}$$

where $w_{\text{slant}} \sim \mathcal{N}\left(0, \sigma^2_{\text{slant}}\right)$. We assume that we have access to a relative depth observation

$$z_d = z_s - z_c + w_d,$$

where z_s and z_c are the depth components of $\mathbf{x}_s$ and $\mathbf{x}_c$, respectively, and $w_d \sim \mathcal{N}\left(0, \sigma^2_d\right)$. Slant

Figure B.2 Pseudo range variance σ^2_{pseudo} and information $1/\sigma^2_{\text{pseudo}}$ for increasing depth to range ratio. As the relative depth increases relative to the slant range, the variance increases corresponding to a decrease in information in the horizontal plane.



ranges are projected into the horizontal plane given the relative depth

$$\begin{aligned} z_{\text{pseudo}} &= \left(z^2_{\text{slant}} - z^2_d\right)^{1/2} \\ &= \left(\left(||\mathbf{x}_s - \mathbf{x}_c||_2 + w_{\text{slant}}\right)^2 - \left(z_s - z_c + w_d\right)^2\right)^{1/2}, \end{aligned}$$

for $z_d < z_{\text{slant}}$. We can model z_{pseudo} as a Gaussian random variable with variance approximated through the first-order Taylor expansion, i.e.

$$\sigma^2_{\text{pseudo}} = \frac{z^2_{\text{slant}}\sigma^2_{\text{slant}} + z^2_d\sigma^2_d}{z^2_{\text{slant}} - z^2_d}. \tag{B.10}$$

The variance is *not* a simple weighted combination of the depth and slant range variances, nor is it a fixed value. The variance explodes as $z_d \to z_{\text{slant}}$, regardless of the depth and slant range uncertainties. Linearized range observations add information along the vector between the server and client positions. As $z_d \to z_{\text{slant}}$, almost all information is added along the depth direction, and little information is added in the horizontal plane. This relationship is illustrated in Fig. B.2.

Appendix C

Field Equipment

C.1 PeRL Iver AUVs

The Perceptual Robotics Laboratory (PeRL) maintains two AUVs—modified Ocean-Server, Inc. Iver2 AUVs, Iver28 and Iver31. Fig. C.1 showes a schematic overview of the custom Iver2 configuration.

C.1.1 Hardware description

Each AUV includes a typical advanced dead-reckoning sensor suite as tabulated in Table C.1 and reported in Brown et al. (2008, 2009). The AUVs measure body-frame velocities with a $600\ \mathrm{kHz}$ RDI DVL, attitude with a Microstrain 3DM-GX25-AHRS, and depth with a Desert Star Systems SSP-1 digital pressure sensor. Additionally, each vehicle is outfitted wit a WHOI Micro-modem and co-processor board Freitag et al. (2005b) and $25\ \mathrm{kHz}$ BTech Acoustics 2RCL transducer.

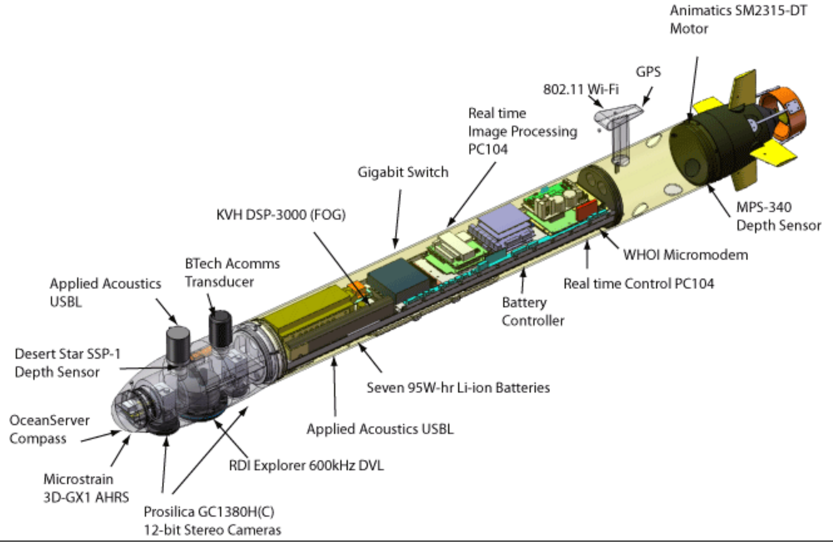
To calibrate the Microstrain heading bias, we apply a nonlinear bias correction to the raw magnetic compass reading θ,

$$\delta\theta(\theta) = A + B\sin\theta + C\cos\theta + D\sin 2\theta + E\cos 2\theta.$$

Table C.1 Iver Navigation sensors: Sampling frequency and noise characteristics

Iver Sensors	**State Variable**	**Frequency**	**Noise (1-σ)**
Microstrain	ϕ, θ, ψ	25 Hz	2°
DVL	$\dot{x}, \dot{y}$	3 Hz	$5\ \mathrm{cm/s}$
pressure depth	z	2 Hz	$10\ \mathrm{cm}$
acoustic modem	slant range	every $\sim$30 sec	$1\ \mathrm{m}$
GPS	x, y	1 Hz	$3\ \mathrm{m}$
LBL	x, y	every $\sim$30 sec	$1\ \mathrm{m}$

Figure C.1 Schematic Iver2 configuration.



The bias correction is a simple sum of harmonic functions fit during a special calibration mission.

C.1.2 Software description

All vehicles run a robotic operating system, perls, developed by PeRL. The modular and extensible software architecture allows vehicles to log command, control, and sensor data real-time and then later playback for new development and debugging. The operating system also contains myriad tools for real-time visualization and control. Bingham et al. (2011) provide further details.

C.2 Synchronous-Clock Acoustic Communication

A pulse per second (PPS) pulse is used to trigger acoustic transmission events with the WHOI Micro-modem (Freitag et al., 2005b). Subsea, a PPSBoard (Eustice et al., 2006b) provides a stable precision clock signal, see Fig. C.2. This time reference allows the Micro-modem to measure the TOA within $125\ \mu\mathrm{s}$ corresponding to $18.75\ \mathrm{cm}$ at $1500\ \mathrm{m/s}$ sound speed. The topside node uses a GPS timeserver with timing accuracy better than $10\ \mu\mathrm{s}$.

Several communication frameworks exist for multiple underwater vehicles operations. Stokey et al. (2005) define the widely used compact control language (CCL) message set, which uses fixed bit-fields to encode messages. Schneider and Schmidt (2010a) provides improved marshalling capabilities with DCCL, which allows for dynamically configurable messages. Schneider and Schmidt (2010b, 2012) wrap the marshalling capabilities of DCCL with a MAC library and lower level modem drivers. Webster et al. (2009a) provide a framework specifically for OWTT navigation. Finally, the CAPTURE framework couples the low-level modem control with a routing

Figure C.2 The PPSBoard provides a stable clock reference while subsea. (Eustice et al., 2011)

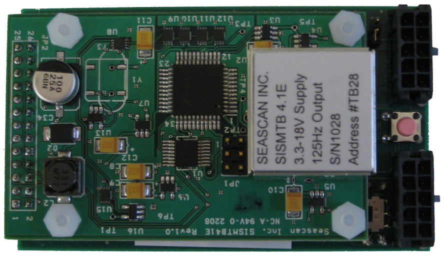


Table C.2 WHOI Micro-modem data packets

Packet	Burst rate	Frames/packet	Frame size	Packet length
PSK R6	490 bps	6	32 B	2.95 s
PSK R5	5388 bps	8	256 B	3.04 s
PSK R4	1301 bps	2	256 B	3.15 s
PSK R3	1223 bps	2	256 B	3.35 s
PSK R2	520 bps	3	64 B	2.95 s
PSK R1	498 bps	3	64 B	3.08 s
FSK R0	80 bps	1	32 B	3.90 s

scheme (Murphy, 2012; Murphy et al., 2014).

All acoustic modem driver-level and MAC scheduling and data marshalling is handled by the `goby-acomms` suite (Schneider and Schmidt, 2012) using DCCL messages with a custom interface to the `perls` operating system. A simple fixed TDMA schedule was employed. Each vehicle was scheduled to broadcast a data packet (either frequency-shift keying (FSK) or PSK, see Table C.2), a mini packet (13 bit data packet), or query the LBL network.

The acoustic message specification for the OSM (Chapter 2) is detailed in Table C.3. Note that server state is parameterized by its XY position, so that each origin state packet information vector and matrix are of dimension four (marginal distribution over origin and newest TOL states). In pracice, the server vehicle broadcasts additional origin state packets, so that each transmit event includes at least two Micro-modem 64 B data frames.

Table C.3 OSM acoustic message specification

Message Field	Range	Precision	Maximum Size
Depth	$0 - 300$ m	1	12 bits
Origin state packet:			
Information vector element	$-10^4 - 10^4$	5	31 bits
Information matrix element	$-10^2 - 10^2$	7	31 bits
Origin TOL index	$0 - 10^3$	–	10 bits
New TOL index	$0 - 10^3$	–	10 bits
Total			60 B

Table C.4 Peer-to-peer acoustic message specification

Message Field	**Min value**	**Max value**	**Precision**	**Maximum Size**
Depth	0 m	-100 m	1	10 bits
Per factor (2 factors per link):				
TOL index	0	10^3	–	10 bits
Measurement vector element	-10^3	10^3	1	15 bits
Covariance element	0	500	1	13 bits
Cross covariance element	-500	500	1	14 bits
Total (3 links)				64 B

The acoustic message specification for broadcasting links for peer-to-peer navigation (Chapter 3) is reviewed in Table C.4. A link is a two-tuple representing two factors: the odometry factor to the previous link and a prior factor. Each factor has equal dimension. We were able to specify the range and resolution of each factor so that three links could be broadcast within a single 64 B data frame.

Bibliography

Agarwal, S., Snavely, N., Seitz, S. M., and Szeliski, R. (2010). Bundle adjustment in the large. In *Proceedings of the European Conference on Computer Vision*, volume 6312, pages 29–42, Crete, Greece.

Agha-mohammadi, A., Chakravorty, S., and Amato, N. M. (2014). FIRM: Sampling-based feedback motion-planning under motion uncertainty and imperfect measurements. *International Journal of Robotics Research*, 33(2):268–304.

Anderson, B. D., Yu, C., Fidan, B., and Hendrickx, J. M. (2008). Rigid graph control autonomous formations. *IEEE Control Systems Magazine*, pages 48–63.

Andersson, L. A. and Nygards, J. (2008). C-SAM: Multi-robot SLAM using square root information smoothing. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2798–2805, Pasadena, CA.

Aragues, R., Cortes, J., and Sagues, C. (2012). Distributed consensus on robot networks for dynamically merging feature-based maps. *IEEE Transactions on Robotics*, 28(4):840–854.

Aspnes, J., Eren, T., Goldenberg, D., Morse, A., Whiteley, W., Yang, Y., Anderson, B., and Belhumeur, P. (2006). A theory of network localization. *IEEE Transactions on Mobile Computing*, 5(12):1663–1678.

Baccou, P. and Jouvencel, B. (2003). Simulation results, post-processing experimentations and comparisons results for navigation, homing and multiple vehicles operations with a new positioning method. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 811–817, Las Vegas, NV.

Bahr, A., Leonard, J. J., and Fallon, M. F. (2009a). Cooperative localization for autonomous underwater vehicles. *International Journal of Robotics Research*, 28(6):714–728.

Bahr, A., Leonard, J. J., and Martinoli, A. (2012). Dynamic positioning of beacon vehicles for cooperative underwater navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3760–3767, Vilamoura, Portugal.

Bahr, A., Walter, M. R., and Leonard, J. J. (2009b). Consistent cooperative localization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3415–3422, Kobe, Japan.

Bailey, T., Bryson, M., Mu, H., Vial, J., McCalman, L., and Durrant-Whyte, H. (2011). Decentralised cooperative localisation for heterogeneous teams of mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2859–2865, Shanghai, China.

Bailey, T., Julier, S., and Agamennoni, G. (2012). On conservative fusion of information with unknown non-Gaussian dependence. In *Proceedings of the International Conference on Information Fusion*, pages 1876–1883.

Bar-Shalom, Y., Li, X. R., and Kirubarajan, T. (2001). *Estimation with applications to tracking and navigation*. John Wiley & Sons, Inc., New York.

Barooah, P., Machado da Silva, N., and Hespanha, J. P. (2006). Distributed optimal estimation from relative measurements for localization and time synchronization. In *Proceedings of the IEEE International Conference on Distributed Computing in Sensor Systems*, pages 266–281, San Francisco, CA.

Bellingham, J. G. and Rajan, K. (2007). Robotics in remote and hostile environments. *Science*, 318(5853):1098–1102.

Bertsekas, D. P. and Tsitsiklis, J. N. (1997). *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific.

Bingham, B. S., Walls, J. M., and Eustice, R. M. (2011). Development of a flexible command and control software architecture. *Marine Technology Society Journal*, 45(3):25–36.

Bosse, M., Newman, P., Leonard, J., and Teller, S. (2004). Simultaneous localization and map building in large-scale cyclic environments using the atlas framework. *International Journal of Robotics Research*, 23(12):1113–1139.

Bowen, A. D., Yoerger, D. R., Taylor, C., McCabe, R., Howland, J., Gomez-Ibanez, D., Kinsey, J. C., Heintz, M., McDonald, G., Peters, D. B., Bailey, J., Bors, E., Shank, T., Whitcomb, L. L., Martin, S. C., Webster, S. E., Jakuba, M. V., Fletcher, B., Young, C., Buescher, J., Fryer, P., and Hulme, S. (2009). Field trials of the Nereus hybrid underwater robotic vehicle in the Challenger Deep of the Mariana Trench. In *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, pages 1–10, Biloxi, MS.

Brock, O. and Khatib, O. (2002). Elastic strips: A framework for motion generation in human environments. *International Journal of Robotics Research*, 21(12):1031–1052.

Brown, H., Kim, A., and Eustice, R. M. (2008). Development of a multi-AUV SLAM testbed at the University of Michigan. In *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, pages 1–6, Quebec City, Quebec, Canada.

Brown, H. C., Kim, A., and Eustice, R. M. (2009). An overview of autonomous underwater vehicle research and testbed at PeRL. *Marine Technology Society Journal*, 43(2):33–47.

Bry, A. and Roy, N. (2011). Rapidly-exploring random belief trees for motion planning under uncertainty. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 723–730, Shanghai, China.

Burns, B., Brock, O., and Levine, B. (2006). Autonomous enhancement of disruption tolerant networks. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2105 –2110, Orlando, FL.

Carlevaris-Bianco, N. and Eustice, R. M. (2013). Generic factor-based node marginalization and edge sparsification for pose-graph SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 5728–5735, Karlsruhe, Germany.

Carlevaris-Bianco, N. and Eustice, R. M. (2014). Learning visual feature descriptors for dynamic lighting conditions. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2769–2776, Chicago, IL.

Carlevaris-Bianco, N., Kaess, M., and Eustice, R. M. (2014). Generic node removal for factor-graph SLAM. *IEEE Transactions on Robotics*, 30(6):1371–1385.

Carrillo-Arce, L., Nerurkar, E. D., Gordillo, J., and Roumeliotis, S. I. (2013). Decentralized multi-robot cooperative localization using covariance intersection. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1412–1417, Tokyo, Japan.

Catipovic, J. A. (1990). Performance limitations in underwater acoustic telemetry. *IEEE Journal of Oceanic Engineering*, 15(3):205–216.

Charrow, B., Michael, N., and Kumar, V. (2012). Cooperative multi-robot estimation and control for radio source localization. In *Proceedings of the International Symposium on Experimental Robotics*, Québec, Canada.

Charrow, B., Michael, N., and Kumar, V. (2014). Cooperative multi-robot estimation and control for radio source localization. *International Journal of Robotics Research*, 33(4):569–580.

Chaves, S., Kim, A., and Eustice, R. (2014). Opportunistic sampling-based planning for active visual SLAM. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3073–3080, Chicago, IL.

Chen, C. (1984). *Linear System Theory and Design*. Oxford series in electrical and computer engineering. Oxford University Press.

Choset, H., Lynch, K. M., Hutchinson, S., Kantor, G. A., Burgard, W., Kavraki, L. E., and Thrun, S. (2005). *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, Cambridge, MA.

Cowell, R., Dawid, P., Lauritzen, S., and Spiegelhalter, D. (2007). *Probabilistic Networks and Expert Systems: Exact Computational Methods for Bayesian Networks*. Information Science and Statistics. Springer.

Cunningham, A., Indelman, V., and Dellaert, F. (2013). DDF-SAM 2.0: Consistent distributed smoothing and mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 5220–5227, Karlsruhe, Germany.

Cunningham, A., Paluri, M., and Dellaert, F. (2010). DDF-SAM: Fully distributed SLAM using constrained factor graphs. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3025–3030, Taipei, Taiwan.

Cunningham, A., Wurm, K. M., Burgard, W., and Dellaert, F. (2012). Fully distributed scalable smoothing and mapping with robust multi-robot data association. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1093 –1100, Saint Paul, MN.

Dellaert, F., Alegre, F., and Martinson, E. B. (2003). Intrinsic localization and mapping with 2 applications: Diffusion mapping and marco polo localization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2344–2349, Taipei, Taiwan.

Dellaert, F., Carlson, J., Ila, V., Ni, K., and Thorpe, C. E. (2010). Subgraph-preconditioned conjugate gradients for large scale SLAM. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2566–2571, Taipei, Taiwan.

Dellaert, F. and Kaess, M. (2006). Square root SAM. *International Journal of Robotics Research*, 25(12):1181–1204.

Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271.

Diosdado, J. V. and Ruiz, I. T. (2007). Decentralised simultaneous localisation and mapping for AUVs. *Proceedings of the IEEE OCEANS-Europe Conference and Exhibition*, pages 1–6.

Djugash, J., Singh, S., Kantor, G., and Zhang, W. (2006). Range-only SLAM for robots operating cooperatively with sensor networks. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2078–2084, Orlando, FL.

Dolgov, D., Thrun, S., Montemerlo, M., and Diebel, J. (2010). Path planning for autonomous vehicles in unknown semi-structured environments. *International Journal of Robotics Research*, 29(5):485–501.

Doucet, A., de Freitas, N., and Gordon, N. (2001). *Sequential Monte Carlo Methods in Practice*. Statistics for Engineering and Information Science. Springer-Verlag, New York.

Duckett, T., Marsland, S., and Shapiro, J. (2000). Learning globally consistent maps by relaxation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3841–3846, San Francisco, CA.

Duckett, T., Marsland, S., and Shapiro, J. (2002). Fast, on-line learning of globally consistent maps. *IEEE Transactions on Robotics and Automation*, 12(3):287–300.

Elbanhawi, M. and Simic, M. (2014). Sampling-based robot motion planning: A review. *IEEE Access*, 2:56–77.

Eustice, R. M., Singh, H., and Leonard, J. J. (2006a). Exactly sparse delayed-state filters for view-based SLAM. *IEEE Transactions on Robotics*, 22(6):1100–1114.

Eustice, R. M., Singh, H., and Whitcomb, L. L. (2011). Synchronous-clock one-way-travel-time acoustic navigation for underwater vehicles. *Journal of Field Robotics*, 28(1):121–136.

Eustice, R. M., Whitcomb, L. L., Singh, H., and Grund, M. (2006b). Recent advances in synchronous-clock one-way-travel-time acoustic navigation. In *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, pages 1–6, Boston, MA.

Fallon, M. F., Kaess, M., Johannsson, H., and Leonard, J. J. (2011). Efficient AUV navigation fusing acoustic ranging and side-scan sonar. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2398–2405, Shanghai, China.

Fallon, M. F., Papadopoulis, G., and Leonard, J. J. (2010a). A measurement distribution framework for cooperative navigation using multiple AUVs. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4256–4263, Anchorage, AK.

Fallon, M. F., Papadopoulos, G., Leonard, J. J., and Patrikalakis, N. M. (2010b). Cooperative AUV navigation using a single maneuvering surface craft. *International Journal of Robotics Research*, 29(12):1461–1474.

Folkesson, J. and Christensen, H. (2004). Graphical SLAM—A self-correcting map. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 383–390, Barcelona, Spain.

Fox, D., Burgard, W., Kruppa, H., and Thrun, S. (2000). A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots*, 8(3):325–344.

Freitag, L., Grund, M., Partan, J., Ball, K., Singh, S., and Koski, P. (2005a). Multi-band acoustic modem for the communications and navigation aid AUV. In *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, pages 1080–1085, Washington, D.C.

Freitag, L., Grund, M., Singh, S., Partan, J., Koski, P., and Ball, K. (2005b). The WHOI micro-modem: An acoustic communications and navigation system for multiple platforms. In *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, pages 1086–1092, Washington, D.C.

Frese, U., Larsson, P., and Duckett, T. (2005). A multilevel relaxation algorithm for simultaneous localization and mapping. *IEEE Transactions on Robotics*, 21(2):196–207.

Gadre, A. S. and Stilwell, D. J. (2005). A complete solution to underwater navigation in the presence of unknown currents based on range measurements from a single location. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1420–1425, Edmonton, Canada.

German, C. R., Jakuba, M. V., Kinsey, J. C., Partan, J., Suman, S., Belani, A., and Yoerger, D. R. (2012). A long term vision for long-range ship-free deep ocean operations: Persistent presence through coordination of autonomous surface vehicles and autonomous underwater vehicles. In *Proceedings of the IEEE/OES Autonomous Underwater Vehicles Conference*, pages 1–7, Southampton, UK.

Grime, S., Durrant-Whyte, H. F., and Ho, P. (1992). Communication in decentralized data-fusion systems. In *Proceedings of the American Control Conference*, pages 3299–3303, Chicago, IL.

Gutmann, J. S. and Konolige, K. (1999). Incremental mapping of large cyclic environments. In *Proceedings of the IEEE Symposium on Computational Intelligence in Robotics Automation*, pages 318–325, Monterey, CA.

Hart, P., Nilsson, N., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107.

He, R., Brunskill, E., and Roy, N. (2011). Efficient planning under uncertainty with macro-actions. *Journal of Artificial Intelligence Research*, 40:523–570.

Hendrickson, B. (1995). The molecule problem: Exploiting structure in global optimization. *SIAM Journal on Optimization*, 5(4):835–857.

Hermann, R. and Krener, A. (1977). Nonlinear controllability and observability. *IEEE Transactions on Automatic Control*, 22(5):728–740.

Hollinger, G., Yerramalli, S., Singh, S., Mitra, U., and Sukhatme, G. (2015). Distributed data fusion for multirobot search. *IEEE Transactions on Robotics*, 31(1):55–66.

Hollinger, G. A., Choudhary, S., Qarabaqi, P., Murphy, C., Mitra, U., Sukhatme, G. S., Stojanovic, M., Singh, H., and Hover, F. (2012). Underwater data collection using robotic sensor networks. *IEEE Journal on Selected Areas in Communications*, 30(5):899–911.

Howard, A., Matari, M. J., and Sukhatme, G. (2001). Relaxation on a mesh: A formalism for generalized localization. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1055–1060, Maui, HI.

Howard, A., Matari, M. J., and Sukhatme, G. (2003). Localization for mobile robot teams: A distributed MLE approach. *Experimental Robotics VIII*, pages 146–155.

Howard, A., Matari, M. J., and Sukhatme, G. S. (2002). Localization for mobile robot teams using maximum likelihood estimation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 434–439, EPFL, Switzerland.

Huang, G., Kaess, M., and Leonard, J. (2013). Consistent sparsification for graph optimization. In *Proceedings of the European Conference on Mobile Robots*, pages 150–157, Barcelona, Spain.

Hunt, M., Marquet, W., Moller, D., Peal, K., Smith, W., and Spindel, R. (1974). An acoustic navigation system. Technical Report WHOI-74-6, Woods Hole Oceanographic Institution.

Indelman, V., Carlone, L., and Dellaert, F. (2014). Planning under uncertainty in the continuous domain: a generalized belief space approach. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 6763–6770, Hong Kong, China.

Indelman, V., Carlone, L., and Dellaert, F. (2015). Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments. *International Journal of Robotics Research*. In press.

Jacobs, D. J. and Hendrickson, B. (1997). An algorithm for two-dimensional rigidity percolation: The pebble game. *Journal of Computational Physics*, 137(2):346–365.

Jakuba, M. V., Roman, C. N., Singh, H., Murphy, C., Kunz, C., Willis, C., Sato, T., and Sohn, R. A. (2008). Field report: Long-baseline acoustic navigation for under-ice AUV operations. *Journal of Field Robotics*, 25(11-12):861–879.

Jian, Y.-D., Balcan, D., and Dellaert, F. (2012). Generalized subgraph preconditioners for large-scale bundle adjustment. In Dellaert, F., Frahm, J.-M., Pollefeys, M., Leal-Taix, L., and Rosenhahn, B., editors, *Outdoor and Large-Scale Real-World Scene Analysis*, volume 7474 of *Lecture Notes in Computer Science*, pages 131–150. Springer Berlin Heidelberg.

Jian, Y.-D., Balcan, D., Panageas, I., Tetali, P., and Dellaert, F. (2013). Support-theoretic subgraph preconditioners for large-scale SLAM. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 9–16, Tokyo, Japan.

Julier, S. and Uhlmann, J. (1997). A non-divergent estimation algorithm in the presence of unknown correlations. In *Proceedings of the American Control Conference*, volume 4, pages 2369–2373, Albuquerque, NM.

Julier, S. and Uhlmann, J. (2001). A counter example to the theory of simultaneous localization and map building. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 4, pages 4238–4243, Seoul, Korea.

Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1–2):99–134.

Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J. J., and Dellaert, F. (2012). iSAM2: Incremental smoothing and mapping using the Bayes tree. *International Journal of Robotics Research*, 31(2):216–235.

Kaess, M., Ranganathan, A., and Dellaert, F. (2008). iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6):1365–1378.

Kalakrishnan, M., Chitta, S., Theodorou, E., Pastor, P., and Schaal, S. (2011). STOMP: Stochastic trajectory optimization for motion planning. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4569–4574, Shanghai, China.

Kantor, G. and Singh, S. (2002). Preliminary results in range-only localization and mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1818–1823, Washington, D.C.

Karaman, S. and Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *International Journal of Robotics Research*, 30(7):846–894.

Kavraki, L., Svestka, P., Latombe, J.-C., and Overmars, M. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566 –580.

Kim, A. and Eustice, R. M. (2013). Perception-driven navigation: Active visual SLAM for robotic area coverage. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3196–3203, Karlsruhe, Germany.

Kim, A. and Eustice, R. M. (2015). Active visual SLAM for robotic area coverage: Theory and experiment. *International Journal of Robotics Research*, 34(4-5):457–475.

Kim, B., Kaess, M., Fletcher, L., Leonard, J., Bachrach, A., Roy, N., and Teller, S. (2010). Multiple relative pose graphs for robust cooperative mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3185–3192, Anchorage, AK.

Kinsey, J. C., Eustice, R. M., and Whitcomb, L. L. (2006). Underwater vehicle navigation: Recent advances and new challenges. In *IFAC Conference Manoeuvring and Control Marine Craft*, Lisbon, Portugal.

Kollar, T. and Roy, N. (2008). Trajectory optimization using reinforcement learning for map exploration. *International Journal of Robotics Research*, 27(2):175–196.

Konolige, K. (2004). Large-scale map-making. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 457–463, San Jose, CA.

Kschischang, F. R., Frey, B. J., and Loeliger, H.-A. (2001). Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519.

Kumar, V., Leonard, N., and Morse, A. S. (2004). *Cooperative Control*, volume 309 of *Lecture Notes in Control and Information Sciences*. Springer Verlag.

Kummerle, R., Grisetti, G., Strasdat, H., Konolige, K., and Burgard, W. (2011). G2o: A general framework for graph optimization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3607–3613, Shanghai, China.

Kunz, C., Murphy, C., Singh, H., Pontbriand, C., Sohn, R. A., Singh, S., Sato, T., Roman, C., Nakamura, K.-i., Jakuba, M., Eustice, R., Camilli, R., and Bailey, J. (2009). Toward extraplanetary under-ice exploration: Robotic steps in the Arctic. *Journal of Field Robotics*, 26(4):411–429.

Kurazume, R., Nagata, S., and Hirose, S. (1994). Cooperative positioning with multiple robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1250–1257, San Diego, CA.

Latombe, J.-C. (1991). *Robot Motion Planning*. Kluwer Academic Publishers, Norwell, MA, USA.

Lavalle, S. M. (1998). Rapidly-exploring random trees: A new tool for path planning. Technical Report 98–11, Department of Computer Science, Iowa State University.

LaValle, S. M. (2006). *Planning Algorithms*. Cambridge University Press.

LaValle, S. M. and Kuffner, J. J. (2001). Randomized kinodynamic planning. *International Journal of Robotics Research*, 20(5):378–400.

Leonard, J. J. and Durrant-Whyte, H. F. (1991). Simultaneous map building and localization for an autonomous mobile robot. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1442–1447, Osaka, Japan.

Leonard, N. E., Paley, D., Lekien, F., Sepulchre, R., Fratantoni, D., and Davis, R. (2007). Collective motion, sensor networks, and ocean sampling. *Proceedings of the IEEE*, 95(1):48–74.

Leung, K., Barfoot, T., and Liu, H. (2010). Decentralized localization of sparsely-communicating robot networks: A centralized-equivalent approach. *IEEE Transactions on Robotics*, 26(1):62–77.

Li, H. and Nashashibi, F. (2013). Cooperative multi-vehicle localization using split covariance intersection filter. *IEEE Intelligent Transportation Systems Magazine*, 5(2):33–44.

Loeliger, H.-a. (2004). An introduction to factor graphs. *IEEE Signal Processing Magazine*, pages 1–31.

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

Lynch, K., Schwartz, I., Yang, P., and Freeman, R. (2008). Decentralized environmental modeling by mobile sensor networks. *IEEE Transactions on Robotics*, 24(3):710–724.

Maczka, D. K., Gadre, A. S., and Stilwell, D. J. (2007). Implementation of a cooperative navigation algorithm on a platoon of autonomous underwater vehicles. In *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, pages 1–6, Vancouver, BC.

Mahler, R. P. S. (2000). Optimal/robust distributed data fusion: A unified approach. In *SPIE Int. Symp. Aero./Def. Sensing, Simulation and Cont.*, volume 4052, pages 128–138.

Martinez, S. and Bullo, F. (2006). Optimal sensor placement and motion coordination for target tracking. *Automatica*, 42(4):661–668.

Mazuran, M., Tipaldi, G. D., Spinello, L., and Burgard, W. (2014). Nonlinear graph sparsification for SLAM. In *Proceedings of the Robotics: Science & Systems Conference*, Berkeley, CA.

McPhail, S. and Pebody, M. (2009). Range-only positioning of a deep-diving autonomous underwater vehicle from a surface ship. *IEEE Journal of Oceanic Engineering*, 34(4):669–677.

Milne, P. H. (1983). *Underwater acoustic positioning systems*. Gulf Publishing Company, Houston.

Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. (2002). FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 593–598, Edmonton, Canada.

Morice, C. and Veres, S. (2010). Geometric bounding techniques for underwater localisation using range-only sensors. *Journal of System Controls Engineering*, 224(15):1–9.

Mu, H., Bailey, T., Thompson, P., and Durrant-Whyte, H. (2011). Decentralised solutions to the cooperative multi-platform navigation problem. *IEEE Transactions on Aerospace and Electronic Systems*, 47(2):1433–1449.

Murphy, C. A. (2012). *Progressively Communicating Rich Telemetry from Autonomous Underwater Vehicles via Relays*. PhD thesis, Massachusetts Institute of Technology / Woods Hole Oceaonographic Institution Joint Program, Department of Electrical Engineering and Computer Science.

Murphy, C. A., Walls, J. M., Schneider, T., Eustice, R. M., Stojanovic, M., and Singh, H. (2014). CAPTURE: a communications architecture for progressive transmission via underwater relays with eavesdropping. *IEEE Journal of Oceanic Engineering*, 39(1):120–130.

Nerurkar, E. and Roumeliotis, S. (2013). A communication-bandwidth-aware hybrid estimation framework for multi-robot cooperative localization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1418–1425, Tokyo,Japan.

Nerurkar, E. D. and Roumeliotis, S. I. (2008). Distributed MAP estimation algorithm for cooperative localization of multiple autonomous robotic systems. Technical report, Deptartment of Computer Science & Engineering, University of Minnesota.

Nerurkar, E. D., Roumeliotis, S. I., and Martinelli, A. (2009). Distributed maximum a posteriori estimation for multi-robot cooperative localization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1402–1409, Kobe, Japan.

Nerurkar, E. D., Zhou, K. X., and Roumeliotis, S. I. (2011a). A hybrid estimation framework for cooperative localization under communication constraints. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 502–509, San Francisco, CA.

Nerurkar, E. D., Zhou, K. X., and Roumeliotis, S. I. (2011b). Hybrid estimation framework of multi-robot cooperative localization using quantized measurements. Technical report, Dept. of Comp. Sci. & Eng., University of Minnesota.

Newman, P. and Leonard, J. J. (2003). Pure range-only sub-sea SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1921–1926, Taipei, Taiwan.

Newman, P., Leonard, J. J., Tardos, J. D., and Neira, J. (2002). Explore and return: Experimental validation of real-time concurrent mapping and localization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1802–1809, Washington, D.C.

Ni, K., Steedly, D., and Dellaert, F. (2007). Tectonic SAM: Exact, out-of-core, submap-based SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1678–1685, Roma, Italy.

Olfati-Saber, R. (2007). Distributed kalman filtering for sensor networks. In *Proceedings of the IEEE Conference on Decision and Control*, pages 5492–5498, New Orleans, LA.

Olson, E., Leonard, J. J., and Teller, S. (2006a). Robust range-only beacon localization. *IEEE Journal of Oceanic Engineering*, 31(4):949–958.

Olson, E. B., Leonard, J. J., and Teller, S. (2006b). Fast iterative alignment of pose graphs with poor initial estimates. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2262–2269, Orlando, FL.

Otnes, R., Asterjadhi, A., Casari, P., Goetz, M., Husøy, T., Nissen, I., Rimstad, K., van Walree, P., and Zorzi, M. (2012). *Underwater Acoustic Networking Techniques*. SpringerBriefs in Electrical and Computer Engineering. Springer.

Ozog, P., Carlevaris-Bianco, N., Kim, A., and Eustice, R. M. (2015). Long-term mapping techniques for ship hull inspection and surveillance using an autonomous underwater vehicle. *Journal of Field Robotics*. In Press.

Partan, J., Kurose, J., and Levine, B. N. (2007). A survey of practical issues in underwater networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 11(4):23–33.

Parter, S. (1961). The use of linear graphs in Gauss elimination. *SIAM review*, 3(2):119–130.

Patil, S., Kahn, G., Laskey, M., Schulman, J., Goldberg, K., and Abbeel, P. (2014). Scaling up Gaussian belief space planning through covariance-free trajectory optimization and automatic differentiation. In *Proceedings of the International Workshop on the Algorithmic Foundations of Robotics*, Istanbul, Turkey.

Patwari, N., Ash, J., Kyperountas, S., Hero, A., Moses, R., and Correal, N. (2005). Locating the nodes: cooperative localization in wireless sensor networks. *IEEE Signal Processing Magazine*, 22(4):54–69.

Paull, L., Seto, M., and Leonard, J. J. (2014). Decentralized cooperative trajectory estimation for autonomous underwater vehicles. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 184–191, Chicago, IL.

Peterson, C. K. and Paley, D. A. (2013). Distributed estimation for motion coordination in an unknown spatially varying flowfield. *AIAA Journal on Guidance, Control, and Dynamics*, 36(3):894–898.

Platt, R., Tedrake, R., Kaelbling, L., and Lozano-Perez, T. (2010). Belief space planning assuming maximum likelihood observations. In *Proceedings of the Robotics: Science & Systems Conference*, Zaragoza, Spain.

Polok, L., Ila, V., Solony, M., Smrz, P., and Zemcik, P. (2013). Incremental block cholesky factorization for nonlinear least squares in robotics. In *Proceedings of the Robotics: Science & Systems Conference*, Berlin, Germany.

Preisig, J. (2006). Acoustic propagation considerations for underwater acoustic communications network development. In *Proceedings of the ACM WUWNet*, pages 1–5, New York, NY, USA. ACM.

Prentice, S. and Roy, N. (2009). The belief roadmap: Efficient planning in belief space by factoring the covariance. *International Journal of Robotics Research*, 28(11-12):1448–1465.

Quinlan, S. and Khatib, O. (1993). Elastic bands: connecting path planning and control. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 802–807, Atlanta, GA.

Ratliff, N., Zucker, M., Bagnell, J., and Srinivasa, S. (2009). CHOMP: Gradient optimization techniques for efficient motion planning. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 489–494, Kobe, Japan.

RD Instruments (1996). Acoustic Doppler Current Profiler, principles of operation: A practical primer. Technical report, RD Instruments, San Diego, CA.

Rekleitis, I. M., Dudek, G., and Milios, E. E. (2000). Multi-robot collaboration for robust exploration. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3164–3169, San Francisco, CA.

Ren, W., Beard, R. W., and Atkins, E. M. (2005). A survey of consensus problems in multi-agent coordination. In *Proceedings of the American Control Conference*, volume 3, pages 1859–1864, Portland, OR.

Ribeiro, A., Giannakis, G. B., and Roumeliotis, S. I. (2006). SOI-KF: Distributed Kalman filtering with low-cost communications using the sign of innovations. *IEEE Transactions on Signal Processing*, 54(12):4782–4795.

Rickert, M., Sieverling, A., and Brock, O. (2014). Balancing exploration and exploitation in sampling-based motion planning. *IEEE Transactions on Robotics*, 30(6):1305–1317.

Rose, D. J. (1972). A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations. In Read, R. C., editor, *Graphy theory and computing*, pages 183–217. Academic Press, New York.

Roumeliotis, S. I. and Bekey, G. A. (2002). Distributed multirobot localization. *IEEE Transactions on Robotics and Automation*, 18(5):781–795.

Roumeliotis, S. I. and Rekleitis, I. M. (2004). Propagation of uncertainty in cooperative multirobot localization. *Autonomous Robots*, 17(1):41–54.

Scherbatyuk, A. P. (1995). The AUV positioning using ranges from one transponder LBL. In *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, volume 3, pages 1620–1623, San Diego, CA.

Schneider, T. and Schmidt, H. (2010a). The dynamic compact control language: A compact marshalling scheme for acoustic communications. In *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, pages 1–10, Sydney, Australia.

Schneider, T. and Schmidt, H. (2010b). Unified command and control for heterogeneous marine sensing networks. *Journal of Field Robotics*, 27(6):876–889.

Schneider, T. and Schmidt, H. (2012). Goby-acomms version 2: Extensible marshalling, queueing, and link-layer interfacing for acoustic telemetry. In *IFAC Conference Manoeuvring and Control Marine Craft*, Arenzano, Italy.

Seto, M. L., Hudson, J., and Pan, Y. (2011). Three-dimensional path-planning for a communications and navigation aid working cooperatively with autonomous underwater vehicles. In *Proceedings of the International Conference on Autonomous Intelligent Systems*, pages 51–62, Burnaby, BC, Canada.

Sim, R., Dudek, G., and Roy, N. (2004). Online control policy optimization for minimizing map uncertainty during exploration. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1758–1763, New Orleans, LA.

Sim, R. and Roy, N. (2005). Global A-optimal robot exploration in SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 661–666, Barcelona, Spain.

Sinopoli, B., Schenato, L., Franceschetti, M., Poolla, K., Jordan, M., and Sastry, S. (2004). Kalman filtering with intermittent observations. *IEEE Transactions on Automatic Control*, 49(9):1453–1464.

Smith, R., Self, M., and Cheeseman, P. (1986). Estimating uncertain spatial relationships in robotics. In *Proceedings of Uncertainty in AI*, pages 267–288. Elsevier.

Song, T. L. (1999). Observability of target tracking with range-only measurements. *IEEE Journal of Oceanic Engineering*, 24(3):383–387.

Stachniss, C., Grisetti, G., and Burgard, W. (2005). Information gain-based exploration using Rao-Blackwellized particle filters. In *Proceedings of the Robotics: Science & Systems Conference*, Cambridge, MA.

Stojanovic, M. (2007). On the relationship between capacity and distance in an underwater acoustic communication channel. *SIGMOBILE Mob. Comput. Commun. Rev.*, 11(4):34–43.

Stokey, R. P., Freitag, L. E., and Grund, M. D. (2005). A compact control language for AUV acoustic communication. In *Proceedings of the IEEE OCEANS-Europe Conference and Exhibition*, pages 1133–1137, Brest, France.

Symmetricom (2013). SA.45s CSAC—Chip Scale Atomic Clock. Specification sheet and documentations available at http://www.symmetricom.com/products/frequency-references/chip-scale-atomic-clock-csac/SA.45s-CSAC/.

Tan, Y. T., Gao, R., and Chitre, M. (2014). Cooperative path planning for range-only localization using a single moving beacon. *IEEE Journal of Oceanic Engineering*, 39(2):371–385.

Thrun, S. and Liu, Y. (2003). Multi-robot SLAM with sparse extended information filers. In *Proceedings of the International Symposium on Robotics Research*, pages 254–266.

Thrun, S., Liu, Y., Koller, D., Ng, A. Y., Ghahramani, Z., and Durrant-Whyte, H. (2004). Simultaneous localization and mapping with sparse extended information filters. *International Journal of Robotics Research*, 23(7-8):693–716.

Thrun, S. and Montemerlo, M. (2006). The graph SLAM algorithm with applications to large-scale mapping of urban structures. *International Journal of Robotics Research*, 25(5-6):403–429.

Trawny, N., Roumeliotis, S. I., and Giannakis, G. B. (2009). Cooperative multi-robot localization under communication constraints. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4394–4400, Kobe, Japan.

Vaganay, J., Leonard, J. J., Curcio, J. A., and Wilcox, J. S. (2004). Experimental validation of the moving long base-line navigation concept. In *Proceedings of the IEEE/OES Autonomous Underwater Vehicles Conference*, pages 59–65, Sebasco, ME.

van den Berg, J., Abbeel, P., and Goldberg, K. (2011a). LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information. *International Journal of Robotics Research*, 30(7):895–913.

van den Berg, J., Patil, S., and Alterovitz, R. (2011b). Motion planning under uncertainty using differential dynamic programming in belief space. In *Proceedings of the International Symposium on Robotics Research*, pages 1–16, Flagstaff, AZ.

van den Berg, J., Patil, S., and Alterovitz, R. (2012). Motion planning under uncertainty using iterative local optimization in belief space. *International Journal of Robotics Research*, 31(11):1263–1278.

Vial, J., Durrant-Whyte, H., and Bailey, T. (2011). Conservative sparsification for efficient and consistent approximate estimation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 886–893, San Francisco, CA.

Vickery, K. (1998). Acoustic positioning systems: A practical overview of current systems. In *Proceedings of the IEEE/OES Autonomous Underwater Vehicles Conference*, pages 5–17, Cambridge, MA.

Švestka, P. and Overmars, M. H. (1997). Motion planning for carlike robots using a probabilistic learning approach. *International Journal of Robotics Research*, 16(2):119–143.

Walls, J. M., Chaves, S. M., Galceran, E., and Eustice, R. M. (2015a). Belief space planning for underwater cooperative localization. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Hamburg, Germany. Submitted, Under Review.

Walls, J. M., Cunningham, A. G., and Eustice, R. M. (2015b). Cooperative localization by factor composition over faulty low-bandwidth communication channels. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Seattle, WA. To Appear.

Walls, J. M. and Eustice, R. M. (2011). Experimental comparison of synchronous-clock cooperative acoustic navigation algorithms. In *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, pages 1–7, Kona, HI, USA.

Walls, J. M. and Eustice, R. M. (2012). An origin state method for lossy synchronous-clock acoustic navigation. In *IFAC Workshop on Navigation, Guidance, Control Underwater Vehicles*, pages 56–62, Porto, Portugal.

Walls, J. M. and Eustice, R. M. (2013). An exact decentralized cooperative navigation algorithm for acoustically networked underwater vehicles with robustness to faulty communication: Theory and experiment. In *Proceedings of the Robotics: Science & Systems Conference*, Berlin, Germany.

Walls, J. M. and Eustice, R. M. (2014a). An origin state method for communication constrained cooperative localization with robustness to packet loss. *International Journal of Robotics Research*, 33(9):1191–1208.

Walls, J. M. and Eustice, R. M. (2014b). Toward informative planning for cooperative underwater localization. In *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, pages 1–7, St. John's, Newfoundland, CA.

Walter, M. R., Eustice, R. M., and Leonard, J. J. (2007). Exactly sparse extended information filters for feature-based SLAM. *International Journal of Robotics Research*, 26(4):335–359.

Webster, S. E., Eustice, R. M., Murphy, C., Singh, H., and Whitcomb, L. L. (2009a). Toward a platform-independent acoustic communications and navigation system for underwater vehicles. In *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, pages 1–7, Biloxi, MS.

Webster, S. E., Eustice, R. M., Singh, H., and Whitcomb, L. L. (2009b). Preliminary deep water results in single-beacon one-way-travel-time acoustic navigation for underwater vehicles. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2053–2060, St. Louis, MO.

Webster, S. E., Eustice, R. M., Singh, H., and Whitcomb, L. L. (2012). Advances in single-beacon one-way-travel-time acoustic navigation for underwater vehicles. *International Journal of Robotics Research*, 31(8):935–950.

Webster, S. E., Walls, J. M., Eustice, R. M., and Whitcomb, L. L. (2013). Decentralized extended information filter for single-beacon cooperative acoustic navigation: Theory and experiments. *IEEE Transactions on Robotics*, 29(4):957–974.

Webster, S. E., Whitcomb, L. L., and Eustice, R. M. (2010). Preliminary results in decentralized estimation for single-beacon acoustic underwater navigation. In *Proceedings of the Robotics: Science & Systems Conference*, Zaragoza, Spain.

Whitcomb, L., Jakuba, M., Kinsey, J., Martin, S., Webster, S., Howland, J., Taylor, C., Gomez-Ibanez, D., and Yoerger, D. (2010). Navigation and control of the nereus hybrid underwater vehicle for global ocean science to 10,903 m depth: Preliminary results. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 594–600, Anchorage, AK.

Whitcomb, L., Yoerger, D., and Singh, H. (1999a). Advances in doppler-based navigation of underwater robotic vehicles. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 399–406, Detroit, MI.

Whitcomb, L. L., Yoerger, D. R., and Singh, H. (1999b). Combined doppler/LBL based navigation of underwater vehicles. In *Proceedings of the International Symposium on Unmanned Untethered Submersible Technology*, pages 1–7, Durham, NH.

Williams, S. and Mahon, I. (2004). Simultaneous localisation and mapping on the great barrier reef. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1771–1776, New Orleans, LA.

Yoerger, D. R., Jakuba, M., Bradley, a. M., and Bingham, B. (2007). Techniques for deep sea near bottom survey using an autonomous underwater vehicle. *International Journal of Robotics Research*, 26(1):41–54.

Zhou, X. S. and Roumeliotis, S. I. (2008). Robot-to-robot relative pose estimation from range measurements. *IEEE Transactions on Robotics*, 24(6):1379–1393.

Zucker, M., Ratliff, N., Dragan, A. D., Pivtoraiko, M., Klingensmith, M., Dellin, C. M., Bagnell, J. A., and Srinivasa, S. S. (2013). CHOMP: Covariant hamiltonian optimization for motion planning. *International Journal of Robotics Research*, 32(9-10):1164–1193.