Underwater Multi-Vehicle Co-operative Target-Tracking

by

Mei Yi Cheung

S.B, Columbia University (2011)
S.M., Massachusetts Institute of Technology (2013)

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2019

Author ..... Signature redacted
Department of Mechanical Engineering
August 21, 2019

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

Accepted by ..... Signature redacted
Nicolas Hadjiconstantinou
Chair, Department Committee on Graduate Theses

Underwater Multi-Vehicle Co-operative Target-Tracking

by

Mei Yi Cheung

Submitted to the Department of Mechanical Engineering
on August 21, 2019, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

The marine domain is fundamentally challenging for collaborative, multi-robot sensing and tracking operations. Accurate mapping and modeling of an underwater environment is both time consuming and difficult, especially when robots have limited access to a high-quality localization solution such as GPS. Communications over distances greater than one hundred meters necessitates the use of acoustics (acomms), which introduces networking challenges such as limited throughput and bandwidth. Successful execution of a highly dynamic and co-operative task such as target-tracking requires optimization of the information-gathering process, probabilistic inference over disparate, noisy sensor data, and exchange of local information over a costly communication link. This thesis presents two formulations of the co-operative underwater target-tracking task: a dynamic nonlinear sigma-point joint estimator and a non-Gaussian, non-parametric, multi-modal factor graph formulation of SLAM. Within the field of SLAM simultaneous estimation of a robot's state (localization) and modeling of its environment (mapping), there is a wealth of research into approaches using point estimate representations and Gaussian sensor noise. The marine domain presents two challenges not well addressed by the usual formulation: (1) measurements obtained from diverse sensors often require extensive filtering and parametrization to fuse in a Gaussian syntax and (2) acomms between multiple robots significantly limits the amount of local information that can be shared over the network. The non-Gaussian approach presented here utilizes the technique of synthetic aperture sonar (SAS) to relate disparate acoustic measurements in a consistent probabilistic framework. Experimental results under real-world acoustic conditions are gathered onboard our custom-built lightweight and low-cost ASVs in the Charles River, and detailed design specifications are presented for our testbed robots.

Thesis Supervisor: John J. Leonard
Title: Samuel C. Collins Professor of Mechanical and Ocean Engineering

Acknowledgments

Firstly, I would like to thank my thesis advisor, Professor John Leonard, and my committee, Professor Henrik Schmidt and Dr. Franz Hover for their support and guidance. Heartfelt thanks to my friends and colleagues, Dehann, Pedro and Nick for all the support, expertise and cups of coffee along the way, without whom this work would not have been possible. A special thanks to the many, many people who generously volunteered their time and knowledge towards the considerable challenge of putting vehicles in the water and having them work, especially Dr. Michael Benjamin, Misha, Paul and Oscar. Last but not least, my family, and my husband Josh, for always supporting me.

This work was supported by ONR grants N00014-16-1-2628 and N00014-18-1-2832 and NSF award IIS-1318392.

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Algorithms

List of Acronyms

ASV Autonomous Surface Vehicle

AUV Autonomous Underwater Vehicle

CBF Conventional Beamformer

CZT Chirp Z-Transform

DFT Discrete Fourier Transform

DOA Direction-of-Arrival

DVL Doppler Velocity Log

ECEF Earth-centered Earth-fixed

EKF Extended Kalman Filter

FFT Fast Fourier Transform

FSK Frequency-shift Keying

IDFT Inverse Discrete Fourier Transform

IMU Inertial Measurement Unit

KDE Kernel Density Estimation

KF Kalman Filter

KLD Kullback Leibler Divergence

LBL Long Baseline

LCM Lightweight Communications and Marshaling

LTI Linear Time-Invariant

LOO Leave One Out

MAP Maximum a posteriori

MCMC Markov Chain Monte Carlo

MF Matched Filter

ML Maximum Likelihood

MOOS Mission-Oriented Operating Suite

OWTT One Way Travel Time

PHAT Phase Transform

PSD Power Spectral Density

PSK Phase-shift Keying

PWM Pulse-width Modulation

ROS Robot Operating System

RTK Real-time Kinematic

SAS Synthetic Aperture Sonar

SCOUT Surface Craft for Undersea and Oceanographic Testing

SLAM Simultaneous Localization and Mapping

SNR Signal to Noise Ratio

TOF Time-of-Flight

TDOA Time Difference of Arrival

TDMA Time Division Multiple Access

USBL Ultra Short Baseline

WHOI Woods Hole Oceanographic Institution

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 1

Introduction

The importance of the oceans covering most of our planet cannot be overstated. They are our global shared resource; central to the regulation of our climate and weather systems, habitats for a vast number of living species, reservoir of renewable and non-renewable natural resources. Studying the ocean is a challenging open problem. The underwater environment remains expensive, time-consuming and dangerous for humans to operate within. Autonomous robotic systems stand to provide an effective low-risk, low-cost solution to exploring, instrumenting and studying the ocean. Over the last few decades, there have been large advances in surface and underwater robotic systems with increasingly sophisticated navigation, autonomy and computation capabilities. Individual robotic platforms are now commonly commercially available and exploited in many underwater applications. In the fields of ocean science, mapping, archaeology, oil and gas industry, fishing industry, naval operations, etc. these systems have already made a huge impact. Now teams of underwater robots stand to carry out missions of increasing complexity, along with increased challenge. We look to improving the behavior and control of fleets of robotic vehicles, enhancing their collaborative capabilities of sensing, maneuvering and task-completion, so as to effectively tackle the scope of this area of operations.

1.1 Motivation and Problem Statement

At this time, robots have been operating underwater for more than fifty years, and the list of high-profile successes is ever-growing. In 1966, the United States Navy Cable-Controlled Undersea Recovery Vehicle (CURV) recovered a hydrogen bomb from the floor of the Mediterranean Sea. The original cable teleoperated vehicle was equipped only with an array of optical cameras and spotlights for sensing underwater. Nearly fifty years later, in 2010, two REMUS 6000s and ROV Jason Jr. operated by the Advanced Imaging and Visualization Laboratory (AIVL) at Woods Hole Oceanographic Institution (WHOI) returned to the RMS Titanic to map the site, and captured for the first time comprehensive optical and sonar images of the wreck. Robots were undertaking missions underwater that pushed beyond the boundaries of what was humanly achievable. Starting in 1995, the Japan Agency for Marine-Earth Science and Technology (JAMSTEC) operated Kaiko as the first unmanned vehicle to reach the bottom of Challenger Deep and return with biological samples. In 2009, WHOI's Nereus followed suit, descending an unprecedented 6.8 miles down the Mariana Trench in the Pacific Ocean to sample the seafloor.

The success of government-funded vehicles and operations resulted in a surge of investment in the technology from private industry, and the growth of commercially available vehicles and sensors. In 2018, the private company Ocean Infinity signed a contract with the Malaysian Authorities to deploy 8 AUVs to search for the missing airplane flight MH370 that was lost over the Southern Indian Ocean. Although the AUVs were not able to locate the wreckage, by the end of the search over 112,000 km^2 of ocean floor had been covered. The maturation of the industry has seen two contrasting developments side-by-side. First, the increasing sophistication of large high-powered working robots equipped with a full complement of sensors, put to work monitoring and maintaining deep subsea drilling infrastructure or search and salvage operations. For example the work-class Nexxus ROV from Oceaneering, or the Bluefin-21, an AUV with a 21" diameter hull. Since these state-of-the-art robots are expensive to manufacture and operate, particularly with specialized support ship time

being at a premium, vehicle operations tend to be risk-averse. True fleet deployments are rare.

The second development has been the increasing availability of small, low-cost and low-power platforms with a minimal configuration of sensors, intended to be quick and simple to operate, or to be launched in quantity. For example, the VideoRay [190], which is typically equipped with a camera and a sonar, the recent Bluefin SandShark [176], or the Hydroid REMUS M3V, a Micro-AUV only 36" by 4.8" in size. In particular, it is the potential of deploying a fleet of lightweight, low-cost underwater vehicles that motivates this work. One model for this large-scale deployment is the work that has been done deploying global arrays of free-drifting instrumented buoys known as drifters. Since the start of the Global Drifter Program in 1979, the Physical Oceanography Division of National Oceanic and Atmospheric Administration (NOAA) has deployed over 1400 drifters in an array around the world. Similarly, the ARGO program, begun in 2000, has a network of 3800 drifters [144]. Continuous real-time in-situ observations of the ocean for the last few decades have now been made freely available, enabling exciting new work in climatology and ocean modeling ([66], [47],[94]).

Deploying a fleet of autonomous vehicles capable of carrying out a planned mission as opposed to passive drifting is the next logical step, and the advancement of Autonomous Underwater Vehicle (AUV) technology has made this development not only possible but likely. Consider deploying a fleet of AUVs to autonomously track and sample an oil plume, to continuously patrol and monitor a harbor, to operate and maintain at-sea fisheries or track and observe wildlife for many days and months. The possibilities of missions with new capabilities are rich and their inherent challenges remain to be solved.

We propose target-tracking and pursuit as one canonical mission for a team of collaborating underwater vehicles, and describe the task as follows: A group of agents, in a centralized or de-centralized architecture, fuses local and team measurements to estimate their states, the states of their collaborators (if necessary), and the states of one or more targets over the time period of the mission. Each agent should be

able to dynamically plan and navigate to maintain the best possible estimate within the boundaries set up by the specific mission. The tracking and pursuit problem is a canonical problem in many fields, and has been studied extensively in robotics [183]. This task takes on the key challenges of multi-vehicle operations underwater: GPS-denied localization, low-throughput lossy communication channels and sensors that are limited by the fundamental properties of the environment. The following section covers background and prior work in these areas of research.

1.2 Prior Work

This section will present an overview of the wide range of topics unified by underwater autonomy. Most of these topics are not specific to the underwater domain: Vehicle dynamics, control, planning and navigation. Multi-vehicle strategies and operations. The Simultaneous Localization and Mapping (SLAM) framework. Keeping in mind the diverse amount of work that has been done under these broad topics, we will approach specifically with the perspective of the underwater domain. In addition, several topics relate more closely with the underwater domain, in particular acoustic sensing and communications. More detailed literature reviews are also presented in specific chapters. **Chapter 3** reviews field experiments in the context of underwater target pursuit. **Chapter 4** provides more background on the specific implementations of SLAM that are the most closely related.

1.2.1 Vehicle Control and Navigation

A standard suite of navigation sensors for an underwater vehicle includes: a compass or gyrocompass providing magnetic heading, orientation and their respective rates, a Doppler Velocity Log (DVL) providing speed over ground, acoustic positioning systems such as Long Baseline (LBL), pressure depth sensors and an Inertial Measurement Unit (IMU) [89]. Depth can be estimated by applying the hydrostatic equation to a pressure measurement from a pressure sensor, with errors around 0.1m at 6000db of pressure [152]. Obtaining an accurate geo-referenced longitude and lati-

tude is typically more challenging. GPS location fixes, the go-to localization solution for out of water applications, are only available at or near the surface since the electromagnetic spectrum is severely attenuated in water. Although underwater vehicles are usually equipped with GPS receivers, surfacing is typically costly in time and power for an underwater vehicle traveling around 1-2m/s that may be performing a mission at several hundred meters depth.

Odometry-based solutions can be accurate with sophisticated sensors. DVL-aided INS comprising of a DVL and IMU can potentially provide sub-meter navigational accuracy. The Doppler Velocity Log (DVL) is a multi-beam acoustic sensor that provides accurate speed over ground data on the order of centimeters per second. An array of transducers transmit an acoustic pulse and record its reflection off the sea floor. The Doppler (frequency) shift in the returns are proportional to the speed at which vehicle is traveling. The operational range to the seafloor is frequency dependent and inversely proportional to the measurement accuracy. A DVL payload is also relatively large, in the range of $0.2m^2$ and 15kg, and may not be practical for use in smaller AUVs, although there have been recent developments in reducing these constraints [160]. The cost of a DVL unit is in the tens of thousands, and a high-end IMU is in the hundred of thousands, a prohibitive price for a fleet of vehicles. Additionally, inertial and Doppler methods are subject to incremental errors and drift over long periods of time, as they rely on integration over accelerations and velocities. Position drift rates for commercial INS units can be on the order of several kilometers per hour, and positioning error increases without bound as the distance traveled by the vehicle without a position fix increases [95].

Acoustic localization techniques can provide globally-referenced positions without the need to surface ([111],[182]). LBL systems provide networks of acoustic transponders as reference points for vehicles to interrogate and compute their locations from the responses via two-way travel time ranging. The drawback is the time and cost associated with setting up the transponder network, although there has been work in reproducing the LBL network with a more convenient setup, for example, a set of mobile beacons [39]. The Ultra Short Baseline (USBL) technique allows the vehicle

to instead interrogate a single, typically hull mounted transducer with a physical receiver array, and perform a similar two-way travel-time computation. These systems are also limited in the number of vehicles that can be supported over the acoustic communications channel [139].

The motion of underwater vehicles is described by nonlinear equations and control schemes can range from simple PID controllers to sophisticated nonlinear and adaptive methods [57]. Often an inner-outer loop structure performs sufficiently well, where low-level controllers handle in a high rate closed-loop the vehicle's attitude and forward velocities, while a trajectory controller in a slower outer loop plans control actions assuming the vehicle achieves its desired actuation setpoints [110]. Sliding Mode Control (SMC) is a well-known nonlinear control method that is robust to nonlinearities and parameter uncertainties that often arise in an underwater system [75], [16]. Briefly the controller attempts to drive the nonlinear system to a sliding surface in the state space, designed for its stability and performance characteristics, typically passing through the point of zero error, and then attempt to keep the nonlinear system in the neighborhood of the sliding surface.

1.2.2 Acoustic Communications

The capability and limitations of the inter-vehicle communication network is critical to a dynamic and collaborative multi-vehicle task, especially where underwater sensor data is typically sparse or under-constrained. Acoustics is the main wireless underwater channel for moderate to long distances, ranging from a hundred meters to kilometers. Radio frequencies, the ubiquitous ground, air and space carrier, are dramatically attenuated, propagating only meters at frequencies higher than 1kHz [25]. Extremely low frequency (30-300Hz) or long-wave transmissions propagate through seawater and are used for communication by US Navy submarines [74] but are impractical for smaller autonomous vehicles due to power usage and required antenna size. Optical modems have been demonstrated over ranges of 100m, potentially providing high data rates but are fundamentally limited by water quality as light is scattered and absorbed underwater ([52], [48]). Currently, over moderate ranges, acoustics is

the practical method of choice. An upper bound on performance is given to be 40km · kbps as the range-rate product, however this is for the best case scenario of vertical channels in deep water, and shallow-water or horizontal channels are expected to perform less effectively [124].

The speed of sound in water is approximately 1500m/s, and over an operating area that may range in hundreds of meters, results in significant propagation delays. The channel is characterized by path loss that depends on frequency as well as travel distance [164]. High frequencies are attenuated faster from absorption loss as the signal is converted to heat energy. This imposes a limit on the available bandwidth of the channel. Additionally, underwater acoustic channels are wideband or ultra wideband since the ratio of the carrier frequency to the signal bandwidth is small. Reflectivity at the surface or seafloor can result in multi-path interference, although results have been shown in exploiting this problem for a multiuser system [163]. Ambient acoustic noise in the ocean can be modeled empirically from four sources, thermal noise, turbulence, shipping and waves, each in a different frequency band. The resulting noise is not white, and the power spectral density decays at the rate of approximately 18 dB/decade [165]. Physical properties of the water channel such as temperature, salinity and density vary significantly in space and time. This results in a relatively high bit error probability in acoustic signals, usually requiring some form of error correction for high-rate signals. Acoustic transducers cannot simultaneously transmit and receive, typically resulting in the need for a medium access control (MAC) scheme, which is an active area of research for underwater acoustic networks [5]. Additionally, over long ranges, specific propagation paths can be emphasized, creating an effect such as the Sound Fixing and Ranging (SOFAR) channel [115], or suppressed, resulting in shadow zones. The shallow water acoustic channel in particular is affected by extensive time-varying multi-path [27].

1.2.3 Acoustic Localization

Acoustic localization of a source from a mobile robot is an active field of research both above and underwater [138]. Fundamental challenges arising from the acous-

tics are multipath (reverberation), signal interference especially in the presence of multiple sources and the inherent ambiguity of TOA/TDOA methods. There are two main approaches to the localization step, although the methods by which the location estimate is then tracked in time are more varied. Computing Time Difference of Arrival (TDOA) measurements with stereo microphones using Generalized cross-correlation with phase transform (GCC-PHAT) is a widely used approach that is simple implementation-wise and relatively robust [91], [166]. Given a set of TDOA measurements, the location of the source can then be estimated by triangulation (or multilateration), and with some assumptions, a set of simultaneous equations can be solved for approximate closed-form solutions [24].

Steered power response (SRP) or multiple signal classification (MUSIC) algorithms compute the pseudo-likelihood of candidate locations on a grid instead, typically selecting the maximum as the estimate [45]. In [184], the next pose of the robot is computed via dynamic programming so as to minimize the information entropy of the grid. DOA techniques are used in conjunction with Bayes filtering techniques such as Kalman filtering [133], nonlinear Extended Kalman filtering [13], and particle filtering. Conventional beamforming cascaded with a particle filter has been demonstrated for tracking a single source ([181, 191]), and multiple moving sources [179]. [107] modeled the TDOA measurements as conditional likelihoods, resulting in a measurement model with a von Mises distribution, and handling the nonlinear tracking with a particle filter.

We mention here that the localization pipeline used in this thesis is matched filtering, conventional beamforming and non-Gaussian SLAM respectively. The key difference in approaches is the inclusion of non-Gaussian TDOA measurement likelihoods directly in the inference step, informing both the pose of the vehicle and the location of the source simultaneously. This is made possible by a SLAM solver that handles arbitrary measurement distributions. Related SLAM frameworks will be discussed in the following section.

1.2.4 Underwater Autonomy

There four main categories of marine robot are Human-Operated Vehicle (HOV), Remote-Operated vehicle (ROV), Autonomous Surface Vehicle (ASV) and Autonomous Underwater Vehicle (AUV). ROVs can be further differentiated between Intervention-class and Inspection-class vehicles [23]. Intervention or work-class ROVs are large vehicles in the range of 1000-5000kg, intended to perform tasks such as maintenance, inspection or drilling at depths up to 6km ([82], [193]). These are typically equipped with high-powered hydraulic propulsion and manipulation systems and high-performance Inertial Measurement Unit (IMU)s and exterior optical and sonar sensors. As a result, they make use of sophisticated controllers for finely controlled trajectories and manipulation [158]. Inspection-class ROVs are on the order of 5-100kg, and have overall lower power, volume and depth ratings. As the class suggests, they are designed for inspection tasks around underwater structures and ships and may carry accurate navigation systems and high-resolution imaging sensors. Since these vehicles are usually teleoperated, they do not require and are not equipped with mission-level autonomy. Bluefin Robotics' Hovering Autonomous Underwater Vehicle (HAUV) is one notable exception to this rule [178].

AUVs on the other hand, are designed to function for long periods on time underwater with minimal human input. Their original purpose, to survey the seafloor, remains one of their main applications at present. Given the challenges associated with the acoustic communications channel and the high manufacturing and operation costs for a platform, it is not surprising that a majority of underwater missions comprise of simple autonomous behaviors such as lawnmower trajectories. There is tremendous active research effort to provide more sophisticated techniques, and additional related field experiments are outlined in the following section and respective chapters. AUVs vary in size, capabilities and purpose, notably the past few years have seen the introduction of smaller, lightweight and low-cost platforms aided by the increasing miniaturization of compute resources. Vehicles such as the Bluefin Sandshark [176], the ecoSUB [132], the Korea Ocean Research and Development Institute (KO-

RDI) ISiMI [84] or the Hydroid REMUS M3V belong to this new generation of AUVs in the range of 5-15kg and 1-3m in length. The design of these vehicles promotes fleet deployments and multi-vehicle operations, however their payload capacities limit the type and complexity of sensors they can carry. Localization and navigation without the aid of expensive IMUs, DVLs or side-scan sonar but with the possibility of collaborative behaviors and improved computing resources represents an exciting new domain for algorithm development.

AUVs equipped with sonars have been applied to missions as diverse as surveying and imaging the seafloor and marine habitats [156],[67], mine detection and countermeasures [151],[126], ocean feature and plume tracking [53], imaging under arctic ice [185], search and salvage operations [134] and many others. The vast majority of applications are dependent on an AUV's ability to localize itself or other acoustic targets, and in many cases the uncertainty or error in the location estimate translates directly to an overall decrease in mission performance or imaging resolution.

Survey, mapping and inspection correspond to one broad category of applications, with an emphasis on positioning accuracy, coverage and resolution. AUVs have been used to capture high-resolution maps of shipwrecks and archaeological sites [17],[46], marine habitats and coral reefs [155], and geological features [196]. Target or feature tracking and adaptive sampling corresponds to another, where navigation, path-planning and dynamic estimation and control are critical to the mission. AUVs have been integral to dangerous missions such as tracking chemical plumes [101], oil spills, [90] harmful algae blooms [143], as well as oceanographic features such as temperature gradients and fronts [200], [35]. Path-planning and control for tracking or adaptive sampling applications making use of predictive models or estimators is an active area of research. Optimal methods include mixed-integer linear programming [199], [188], model-predictive control [141], shortest-path planners on graphical models [130]. Stochastic methods like rapidly exploring random trees (RRT*) [36] have also been studied.

1.2.5 Multi-Vehicle Operations

We briefly overview the current state of underwater multi-vehicle operations with a specific focus on field experiments, roughly organized by but not limited to the general topics of infrastructure, navigation, dynamic control (formation-flying), target tracking and pursuit.

Various acoustic communications infrastructures intended for multi-vehicle deployments have been presented. Cruz *et al.* described a framework named SUNSET consisting of a MARES AUV, two ASVs and Navigation and Instrumentation buoys [34]. The Underwater Acoustic Network (UAN) project included a network of bottom-fixed acoustic nodes and mobile nodes mounted on Folaga AUVs [22]. In Monterey Bay, the Persistent Littoral Undersea Surveillance Network (PLUSNet) was deployed with the goal of tracking diesel submarines and included gateway buoys, Seaglider AUVs, Bluefin AUVs and fixed bottom nodes [70]. Monterey Bay Research Institute's (MBARI) Controlled, Agile and Novel Observing Network (CANON) fields Dorado and Long-Range AUVs as well as drifters, gliders and wave-gliders [1].

Among the most dynamic missions that have been studied are formation flight and target pursuit, which demand tight coordination and communication to operate at a high closed-loop bandwidth. Fiorelli *et al.* demonstrated formation flying with three gliders in a fixed triangle formation [55]. During the AOSN II sea trials in Monterey Bay in 2003, the gliders performed missions to climb and descend a temperature gradient, as well as follow a propeller-driven Dorado AUV. The centralized mission planner was located on an onshore computer, and gliders had to surface to received updated plans. This was followed by the development of Glider Coordinated Control System (GCCS) which similarly relayed pre-computed trajectories to surfacing gliders, but predicted their underwater motion and locations using ocean models that could be used to update the control laws [123]. Field experiments were conducted with two Slocum gliders in Buzzards Bay [62] and ten gliders in Monterey Bay [97], showing successful co-ordination over very long missions (24 days). Similarly, Das *et al.* demonstrated at the 2010 CANON field trials a Dorado AUV executing a trajec-

tory with reference to a drifter's position, which was transmitted through an Iridium satellite link [41].

Formation flying field trials with leader-follower strategies in generally straight trajectories have been presented in several works. In [12], with a fleet of five AUVs in a centralized architecture where followers reported acoustically to the leader and could be designated for replacement, in [20] with two AUVs of differing capabilities, one that could hover at the surface providing radio communication with the support ship, and the second following a trajectory underwater, and in [161], describing a vehicle following two leaders in a triangle formation at 0.5m/s speed, with a 4s cycle time communicating ranges to each of the leader vehicles.

Recently, Branch *et al.* demonstrated two front-tracking behaviors onboard three types of AUVs, small Iver AUVs, Kongsberg Seagliders and MBARI's Long-Range AUVs [19]. Each vehicle was commanded using the Iridium satellite network during an automated process of surfacing, data transmission and receiving commands from the shore station, for a total cycle time of 5 minutes and 40 seconds not including transit to the surface. Temperature samples were used to estimate the front's position and generate trajectories that transected the front. Experimental results for two AUVs performing co-operative adaptive surveys for detecting internal waves are given in [131]. A collaborative decentralized area coverage behavior for the task of asset protection was demonstrated with the UAN network in [21]. Paull *et al.* integrated adaptive path-planning with probabilistic SLAM-based trajectory estimation to show co-operative area coverage with two vehicles in a shallow-water environments [125].

Clark *et al.* presented a stereo hydrophone system for autonomously tracking marine life, and experimental results tracking a tagged 1-m leopard shark [31]. The hydrophones were set 2.4m apart and towed at least 1m below the surface of the water, providing bearing and signal strength to commercially available fish tags, which were then passed to a particle filter to estimate the location of the shark. Lin *et al.* described a similar stereo-hydrophone system for tracking with two AUVs, with the addition of a leader-follower control system to generate pursuit trajectories [104]. The hydrophones generated bearing and distance measurements to the tag, which were

fused with a particle filter for estimation, while an A* path planner was used for obstacle avoidance and target-pursuit.

1.2.6 Simultaneous Localization and Mapping

SLAM refers to the simultaneous construction of a map of a robot's environment and estimation of its state. Since this field was first introduced in the early 1990's [96], as the need for robust, autonomous localization increased with the capabilities of mobile robot platforms, a vast amount of research has taken place. We present here a general overview of the field, and subsequently a more specific subset of domain and application specific results. For additional details, the reader is referred to more thorough reviews by Durrant-Whyte and Bailey [9], [10], Cadena *et al.* [98] and others, [150], [77]. In particular, the marine domain is an appealing application for SLAM due to the above-mentioned challenges of underwater navigation, the difficulty of obtaining a global reference, underwater feature (landmark) extraction, and the need to effectively fuse measurements obtained from a diverse set of underwater sensors. A good review of the underwater field is given in [127].

A SLAM system is typically described by front-end and back-end components. The front-end system consisted of the models by which sensor data and raw measurements from the robot(s) are abstracted, represented and related to each other, i.e. relative frames, rigid transforms, measurement residuals, graphical models etc. with the purpose of generating a consistent estimation framework from a set of raw data. The back-end, which we also call the solver, performs joint inference over the framework constructed by the front-end, in order to find the best possible estimate for every unknown variable in the problem. The standard SLAM formulation is a front-end describing Gaussian measurement likelihoods, and a back-end Maximum a posteriori (MAP) estimator.

Extended Kalman Filter (EKF)-SLAM is one of the first standard MAP formulations [159]. A standard EKF update is used to propagate an augmented and linearized state vector containing the poses of all relevant robots and landmarks towards an MAP solution. The transition (observation) model comprises of all of the residuals between

the measurements and the predicted state vector. Since this approach requires inverting the large, potentially dense covariance matrix, it can be computationally intensive. Ribas *et al.* described a front-end line-detecting feature from sonar scans used with an EKF pose-graph SLAM back-end and demonstrated experimental results on an AUV traveling in a marina [142]. The computational complexity of the EKF approach was addressed by building and joining sequences of local maps. Recently, Mallios *et al.* [4] showed experimental results for pose-based EKF-SLAM solution in an underwater cave system, cross-registering returns from imaging and profiling sonars to bound the inertial drift of the vehicle. Experimental results have also been demonstrated on features such as reefs [194], seabed objects [171] and bathymetry [54]. EKF methods are extended by Unscented Kalman filter (UKF) [109], Extended Information filter (EIF) [49] and Sparse Extended Information filter (SEIF).

Particle filtering methods are another standard back-end implementation. Dellaert *et al.* demonstrated Monte Carlo localization using non-Gaussian measurement likelihoods [44]. However, for high-dimensional augmented state vectors, particle depletion becomes a problem. Each resampling step represents a loss of information on unlikely modes, which depending on the application may be undesirable. Using a Rao-Blackwellized particle filter is one possible solution. FastSLAM [172] is a state-of-the-art solution, wherein each particle is used to represent a full augmented state vector, and a set of particles tracks a subset of Gaussian parametric solutions to the inference problem. In [73], FastSLAM was demonstrated with an AUV, fusing AHRS, DVL and imaging sonar to improve inertial navigation. Barkby *et al.* introduced BPSLAM, an implementation of FastSLAM in which particles are used to record trajectories and their corresponding bathymetry maps, rather than individual feature estimates [11]. The consistency between new observations and each particle's map can then be used to weight the particle set during resampling.

Finally, Graph-based SLAM is the third current state-of-the-art category of methods, in which a factor graph front-end is used to represent conditional dependencies between variables and measurements at large-scales [92]. A factor graph is a graphical model of the inference problem. Dependent variables are connected to each other only

via nodes known as factors that represent their conditional likelihood functions. Typically, these likelihoods are assumed to be Gaussian, such that the inference problem can be written as a nonlinear least-squares optimization minimizing the error of the measurement residuals. Solving the factor graph refers to computing the marginal posterior state estimates over variables in the graph. If all the likelihoods are assumed Gaussian, solving forward-only over the graph as a Hidden Markov Model (HMM) reproduces the optimal KF solution. Solving over all variables corresponds to the move from a filtering approach to a smoothing approach, in which the entire trajectory is estimated; this problem was first described by Lu and Milios [105]. Many back-end solvers have since been developed. GraphSLAM proposes solving the nonlinear least-squares problem with iterative Gauss-Newton or Levenberg-Marquardt local linearizations [69]. Loopy Belief Propagation by Ranganathan and Dellaert [137] iteratively propagates beliefs directly on the factor graph in the presence of loops on the network.

Kaess *et al.* made the key insight that the least-squares optimization could be solved incrementally on a triangular decomposition of the square root information matrix instead of performing the full QR factorization for every variable at each step, leading to the iSAM algorithm [87]. The square root information matrix is incrementally updated by a series of Givens rotations that modifies only the updated columns and rows, thus recycling the rest of the computation. The limitation of the method was fill-in of the triangular matrix due to loop closures, which required computationally expensive periodic variable reordering. This was addressed in iSAM2 [86] by introducing belief propagation on the Bayes (Junction) Tree. Bayesian networks was introduced by Pearl as graphical models for probabilistic inference [129]. Much like the factor graph, the Bayes network and Bayes Tree described the conditional likelihood between dependent variables, and it was straight-forward to go from one representation to the other. In particular, the Bayes Tree turns a factor graph with loops into a tree with special structure, on which incremental operations can be solved locally without the need for batch re-ordering. We continue the discussion on the Bayes Tree in Chapter 4.

Experimental results with Graph-based SLAM underwater have demonstrated the benefits of the smoothing approach and full trajectory estimation. In [117], range-only SLAM was tested experimentally on a single AUV. The nonlinear least-squares inference problem included TOF measurement residuals and was solved using a standard Gauss-Newton approach. Fallon *et al.* applied the iSAM algorithm on-line to fuse acoustic range-only and side-scan sonar measurements with INS data onboard a REMUS AUV [51]. Single-beam sonar registration without the need for inertial/odometry measurements was demonstrated with iSAM [189], and with forward-looking sonar and pose-graph SLAM in [100]. Acoustic structure from motion (ASFM) [80] presents a factor-graph formulation for fusing odometry and imaging sonar measurements to recover environmental point features in 3D.

Newman *et al.* [118] demonstrated incorporating range, azimuth, and elevation measurements from a SAS system into a constant-time SLAM (CTS) framework with manual data association, while indicating that a graph-theoretical perspective would likely resolve some of the limitations encountered by the CTS method. In contrast to CTS, our approach uses a non-Gaussian factor graph approach, and does not divide the problem into submaps. Instead, the Bayes (Junction) Tree strategy is used to decompose the full graph problem into statistically independent branches, in a mechanism that allows previous computations to be recycled in a near-optimal manner [58, 85, 86]. Additional work specific to our application is discussed in Chapter 4.

1.2.7 Collaborative Localization

The goals of collaborative (collective/co-operative) localization are closely related to that of dynamic multi-vehicle target-tracking and pursuit. The task is to describe an algorithm or set of behaviors by which a collaborating team can improve local navigation at each agent by exchanging certain information. The capabilities of each agent, their operating environment, available sensors and characteristics of the communication channels are the main variables of the problem. This mission was first described by Kurazume *et al.* [93]. The implementation was for a rotating subset of agents to remain stationary, acting as temporary landmarks. Roumeliotis *et al.* extended this

to an online de-centralized KF estimator that fused relative pose information received from collaborating team members in range [146]. As discussed by the authors in [146], this problem can be viewed as a special case of multi-robot SLAM, needing only the inclusion of relative measurements to landmarks in the environment. Subsequently, Mourikis and Roumeliotis showed an upper bound on the expected rate of uncertainty increase that was independent of the topology and accuracy of relative position measurements, and dependent only on the accuracy of local inertial sensors [147], [114]. The KF approach was subsequently extended to an EKF-based implementation [108].

Along similar lines as the development of SLAM solutions, a particle filter approach was demonstrated by Fox *et al.* [173]. As the full joint posterior over all states was impractical to compute, each agent estimated their marginal posterior beliefs with a local particle filter, performing incremental and independent updates for each sensor, odometry or detection input.

Multi-vehicle collaborative and in-the-loop acoustic navigation has been demonstrated in the field. Two-node experiments were presented in [50] with a ship to DVL-aided AUV configuration and in [187] with a pair of AUVs. In each it is assumed nodes are equipped with synchronous clock enabling exchange of One Way Travel Time (OWTT) measurements, and DVL, attitude and depth sensors. Sensor fusion was handled by a maximum likelihood approach, solving a nonlinear least-squares optimization minimizing the measurement log-likelihood with the Levenberg-Marquardt algorithm. Recently, Claus *et al.* demonstrated OWTT closed-loop navigation with three AUVs and a topside transponder located on a support ship [32]. Nonlinear control was handled by an EKF and range measurements were estimated with a particle filter. Vehicles exchanged information with a Time Division Multiple Access (TDMA) cycle time of 60s. Co-operative navigation in the form of collaborating vehicles providing a moving baseline network was presented in [38], and for optimal path-planning given a single beacon AUV in [28]. Visual odometry has also been tested in clear water and compared with a traditional hyperbolic localization method [33], using a network of fixed sensor nodes and two AUVs with differing capabilities, where the visual odometer was shown to have around 5% error of along track distance.

In [8], Bahr *et al.* described the main challenges to co-operative navigation underwater, namely, the Gaussian distribution is not a good description of the acoustic channel and fundamental sensing capabilities. Outliers, reflections, discontinuous properties of the water column all contribute to signal noise that is not normally distributed. Additionally, the measurement distributions are inherently multi-modal and often of lower dimension. An algorithm was proposed that iteratively minimized a cost function based on odometry and detection residuals using the Kullback-Leibler divergence as a distance metric [7]. Experimental results were shown with ASVs and an AUV, and compared to conventional EKF/PF approaches.

1.3 Conclusion

The rapid technological advance of autonomous underwater robotic systems in recent years, especially with regards to lightweight and low-cost platforms, promotes the development of highly dynamic co-operative multi-vehicle missions. Underwater operations have by and large been undertaken with small numbers of vehicles that are highly capable, but also extremely costly to operate. The current push towards using more vehicles with less individual capability runs in contrary to the established paradigm, and demands smarter, more robust autonomy. The underwater environment represents significant challenges in navigation, sensing, and communication that are not seen in other domains. The target-tracking archetype demands the full package of effective navigation, localization, and cooperation between multiple dynamic agents, and this thesis presents two algorithms for collaborative underwater multi-vehicle joint estimation and inference. An autonomous vehicle testbed was designed and built to provide experimental data for all methods presented, addressing the plethora of challenges that real-world conditions present in the field of acoustics. The structure of the remainder of this thesis is as follows:

Chapter 2 details the design and build of a small, low-cost ASV, as well as its applicability to multi-vehicle field experiments at the MIT Sailing Pavilion on the Charles River in Cambridge, MA. The design specifications of the ASV seek to close

replicate the functionality provided by low-cost marine vehicles used for multi-vehicle deployments as mentioned in the problem statement. The following chapters describe the overall capabilities and limitations of the hardware and software package, as well as the specific suite of sensors, most notably the acoustic modems and transducers. Field experiments presented in Chapters. 3 and 5 were conducted using these vehicles. Finally, local bathymetry data and acoustic channel statistics for the shallow-water operational area of the Lower Charles River, collected over the course of several field trials, are shown.

Chapter 3 presents a de-centralized joint estimation and control framework for multiple underwater agents collaborating to track a maneuvering target. In particular, the problem of range-only (TDOA) pursuit in a shallow-water acoustic environment is studied with the goal of characterizing the performance of acoustics-in-the-loop, in which the acoustic modem is used as the measurement as well as the main communications channel. The effects of cycle time, packet loss, and de-centralization on the frequency response of the full integrated system are analyzed, and we demonstrate an implementation of partial information updates that decreases the effective cycle time. Experimental results are presented with three ASVs in the field and four WHOI Micro-Modems forming the acoustic network, showing that at present, an acoustics channel can sustain highly dynamic multi-vehicle missions.

Chapter 4 introduces our formulation of Synthetic Aperture Sonar (SAS) in the context of non-Gaussian SLAM. We detail an acoustic data processing pipeline for waveforms recorded by an underwater vehicle equipped with a single hydrophone, in which TDOA measurements are obtained by matched filtering and DOA measurements are obtained by conventional beamforming. Measurements are subsequently modeled as non-Gaussian probability distributions via kernel density estimation, with a key goal of retaining ambiguous multiple modes without the computational cost of explicit hypothesis tracking. These non-Gaussian likelihoods are then directly formulated in a factor (pose) graph SLAM framework. We summarize the main algorithm of the non-Gaussian mm-iSAM back-end solver. The marginal posterior distributions of every variable in the factor graph can be resolved via approximate belief propa-

gation on a Bayes Tree graphical model. Finally, we formulate a factor that can be used to represent synthetic aperture sonar in a general factor graph framework, comprising of forward (array to beacon) and backward (beacon to array) non-Gaussian measurement likelihoods.

Chapter 5 presents experimental results from solving non-Gaussian SAS-SLAM on field data collected in the shallow-water environment of the Charles River. Three types of missions are considered, characterized by varying the trajectories of the ASV and a stationary, drifting, or maneuvering beacon. Each resulting factor graph implementation is discussed. Performance of the algorithm at simultaneously localizing the beacon and vehicle locations is evaluated against GPS ground truth data. Additionally, we consider the multi-modality of estimated posterior distributions with the Kullback Leibler Divergence (KLD) from a Gaussian distribution as a metric.

Chapter 6 concludes this thesis with a summary of the main contributions, critical analysis of the methods and experiments that have been presented, and last but not least avenues for future work.

Chapter 2

An MIT Experimental Autonomous Surface Vehicle

A fleet of low-cost, lightweight autonomous surface vehicles were designed and built at MIT for field experiments in underwater navigation, control and acoustic communications. In particular, the underwater acoustic channel for both sensing and communications is fundamentally challenging to accurately model, increasing the value of evaluating the performance of our algorithms on real-world field data. This chapter discusses the design decisions, hardware and software architectures that comprise our ASV platform, which is subsequently deployed for the field experiments detailed in Chapter 3 and 5. The ASV's dynamic control, sensing and autonomous capabilities are summarized. RTK-GPS is presented as the ground-truth localization solution enabling objective evaluation of our algorithms' performance. Additionally, we overview presently available robotic systems with comparable design parameters, and show bathymetric and acoustic data characterizing the main operating area in the lower Charles River Basin, as part of a laboratory at the MIT Sailing Pavilion, Cambridge, MA [2].

2.1 Comparable Watercraft

We briefly overview comparable marine systems that have been reported in literature. The main reference vehicle for this ASV is Joseph Curcio's MIT SCOUT kayak [37]. The Surface Craft for Undersea and Oceanographic Testing (SCOUT) was designed and manufactured at MIT in 2004 as an inexpensive robotic platform providing autonomous navigation and control for algorithm research with various underwater payloads and sensors. SCOUT's main design goals, portability, low cost and minimal support infrastructure are shared by this vehicle design. In 2010, a laboratory was installed at the MIT Sailing Pavilion on the Charles River, providing easy access to the water, fixed infrastructure and a dedicated shore station. This meant in part, the SCOUTs designed for the open ocean were larger than necessary for the shallow-water river environment and less maneuverable than desired for experiments in dynamic control. Additionally, more features could be incorporated that took advantage of the dedicated shoreside laboratory space, most notably accurate RTK-GPS for ground truth position information.

Figure 2-1: MIT SCOUT ASV

Figure 2-2: Clearpath Robotics Heron M300 USV

The Kingfisher (M100/M200) and Heron (M300) unmanned surface vehicle series produced by Clearpath Robotics are comparable robotic systems designed for shallow, fresh-water operations. The catamaran-style robot has separate compute and payload bays centered over two pontoons inset with jet propulsion thrusters. A switch away from regular propeller thrusters allows launch and recovery by a single person sliding the vehicle into the water from a flat surface but trades off the ability to op-

erate in saltwater, and fouling is a concern. However, we draw two important design specifications from this vehicle for the ASV. The shallow draught and low thruster power means it cannot tow a WHOI Micro-Modem transducer towfish at sufficient speed and depth. Additionally, the maximum battery size (nominal 2.5hrs at 1m/s constant) and power infrastructure cannot sustain the acoustic modem transmitting at 50W burst.

Figure 2-3: WHOI Jetyak

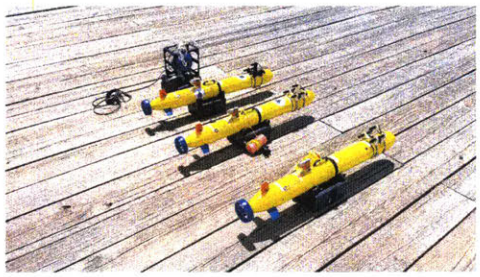

Figure 2-4: Bluefin SandShark

The WHOI Jetyak is a modified Mokai jet-powered kayak with added infrastructure for instrumentation and autonomous operation [88]. It is capable of operating for 8-10 hours at a maximum speed of 3.5-5.5 m/s, and with a draft of less than 20cm, can traverse very shallow waterways. The total cost of replacement for the vehicle was reported to be $15,000, excluding sensor payloads, and including the base $6000 price tag of the unmodified Mokai.

Finally, the Bluefin SandShark was designed specifically as a low-cost research and development AUV to reduce the barrier of entry for multi-vehicle operations [121]. Its smaller hull size makes deployments in operating areas such as the Charles River more practical, but limits the maximum dimensions and power consumption of potential payloads. The autonomy package consists of a Raspberry Pi computer, an Arduino microcontroller and a 9DOF microelectricalmechanical IMU. The vehicle is also equipped with GPS, WiFi modules, and a Microsemi chipscale atomic clock (CSAC). Fischell *et al.* presented experimental results characterizing the autonomous behaviors and acoustic payload performance of this AUV in the same operating region of the Charles River [56].

2.2 Hardware Design

The key design features for the MRG-ASV and the respective design solutions are:

- Stable, seaworthy, low-cost and man-portable platform
- Dynamic capability on the order of a typical medium-sized AUV: REMUS taken as reference (Turning radius 4.6m traveling at 0.8m/s)
- Provide accurate ground-truth states for experimental verification of simulated inertial odometry: Tilt-compensated compass/ RTK-GPS [167]
- Remote and autonomous control: WiFi and Freewave comms/ MOOS/MOOS-IvP software suite [119, 15]
- Acoustic communications: Two acoustic transducer packages (WHOI MicroModem and MIT pi-USBL) [63, 149]

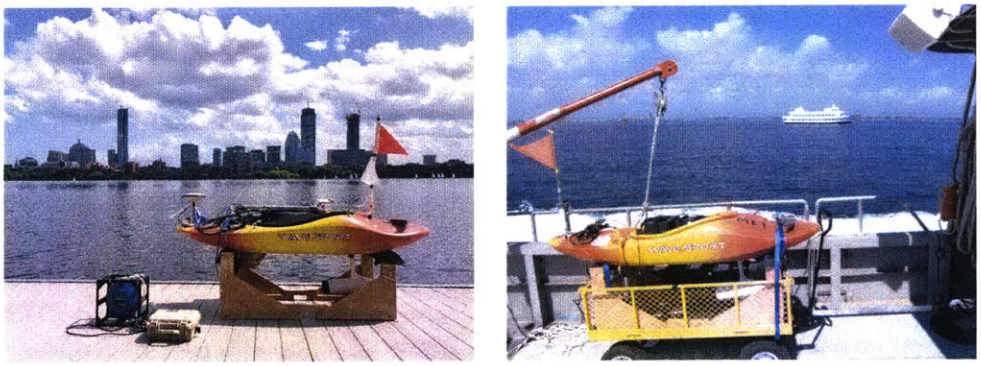

Figure 2-5: MRG-ASV *Nostromo* on the dock of the MIT Sailing Pavilion and in the harbor of Boston

2.2.1 Mechanical Designs and Electronics

The hull is a commercial WaveSport Fuse 35 whitewater kayak. At 178cm by 58cm and 10.5kg, it is small and lightweight, satisfying the requirement of easy transport, launch and recovery. The whitewater hull shape is distinctly different from the traditional kayak hull used by others such as SCOUT, almost completely-flat at the bottom and

sloping up towards either end. The greater volume to length ratio and increased stability (as well interior space) of a wider hull led us to choose the whitewater kayak over a traditional design. Although less efficient for transit, it is highly maneuverable and seaworthy despite the size, having been used on both the Lower Charles River where wind speeds can reach 50mph [3], and in the harbor of Boston.

A Minn-Kota trolling motor rated for 50lb of thrust is mounted on a ServoCity servo at the bow. Smaller thrusters did not provide sufficient thrust when towing a modem towfish. Traditionally the thruster is mounted in the stern of the vehicle, but for several reasons we opted to mount the thruster near the bow of the vehicle. The flat-bottomed hull means that the kayak does not track straight by default, greatly increasing the complexity of control for a stern-mounted thruster. Since the kayaks were also expected to have to station-keep frequently, a bow-mounted thruster provides greater maneuverability and accuracy when doing so. Finally, a bow-mounted thruster means the transducer towfish can be towed astern of the thruster and the vehicle driven forwards without risk of the propeller hitting the cable.

An MS530-1 Mega Servo is used to turn the thruster shaft, which is directly clamped to the output of the servo to minimize slipping. A plastic cylinder housing an o-ring and sleeve bearing is bolted directly to the hole in the hull for the shaft and for water tightness there are both interior and exterior o-ring seals. Both servo and motor controller receive Pulse-width Modulation (PWM) command signals from an Arduino board which in turn receives power and angular commands from control software onboard a Linux gumstix-Duovero (Fig. 2-7). A fixed skeg is mounted at the stern for stabilization in yaw and sway. The vehicle's maximum speed when towing a transducer towfish is roughly 1.6 m/s (3 kts), and over 2 m/s (4 kts) when not towing a modem. The maximum turn angle is a software-limited $\pm 30°$, and the maximum turn rate is a software-limited $\pm 10°/s$. In practice the ASV is able to turn 180° in approximately 3 seconds. The dynamics of the tow-cable are not modeled.

A hardware-based emergency stop switch is located on the front of the kayak and disconnects the battery from the power distribution circuit. Self Power Reset is an additional safety feature provided by the power circuit shown in Fig. 2-6 that

enables the Arduino Microcontroller to power cycle the vehicle autonomously or via RC command. This feature helps the vehicle to recover from temporary loss of power, software or computing failures.

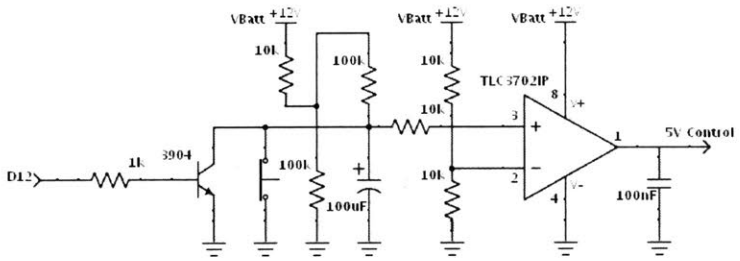

Figure 2-6: MRG-ASV Self Power Reset Circuit

While the vehicle is turned off, the switch is closed, shorting the comparator input to ground and turning off the output. When the vehicle is turned on, the switch is opened. The 100μF capacitor begins to charge from the battery through a $100k\Omega$ resistor. After a few seconds, it raises the input to the comparator enough that the output turns on, powering on the vehicle. During a reset event, or initiated by the Arduino via the transistor, charge is dumped out of the capacitor, switching off the comparator output. Since power is no longer available, the transistor immediately returns to its original state. The capacitor can then recharge directly from the battery, and after a few seconds, switches the comparator's output back to on, repowering the vehicle. The drawbacks of this design are the short delay of a few seconds every time the vehicle turns on, and the idle power draw of around 100μA.

2.2.2 Sensing and Connectivity

An OceanServer tilt-compensated compass provides angular and angular rate data up to 40Hz (10Hz reporting) up to 0.5° nominal heading accuracy. In practice, surface conditions represent a significant source of noise, and the raw compass data is filtered with a first-order low-pass filter of time constant $\tau = 2$ seconds. Variance in heading is empirically estimated at 10deg^2. It is mounted in the bow to isolate interference from other electronics, specifically the Micro-Modem.

An uBlox NEO GPS receiver is used in conjunction with Real-time Kinematic (RTK) positioning provided by the open source package RTKLIB [167] to reduce atmospheric effects, compensate for periods of poor satellite coverage and improve overall positioning accuracy. Our RTK system has a single fixed base station installed above the MIT Sailing Pavilion Laboratory providing a reference server to all mobile field agents. We briefly outline the positioning algorithm below.

RTK positioning uses a standard EKF to optimally estimate the state of an agent given a known reference (base station). The typical mode of operation is fixed-base short baseline without receiver dynamics. Let an agent a be subscribed to a base station b. Let the carrier-phase Φ single difference between satellites i, j for either receiver be $\Phi_i, a - \Phi_j, a = \Phi^a_{ij}$. Let the double difference carrier-phase measurement be $\Phi^a_{ij} - \Phi^b_{ij} = \Phi_{ab,ij}$. Similarly, this notation defines double-differences in pseudorange P and geometric range ρ, expressed in meters. For the sake of notation we omit L1 and L2 carrier-phase frequency dependence except in the cases where they have to be distinguished.

Let m be the number of satellite fixes. Define $\bar{r}^i$ as the position of satellite i in an Earth-centered Earth-fixed (ECEF) frame, $\bar{r}_b$ as the base-station antenna position and $\bar{r}_a$ as the position of the vehicle's antenna. Since the GPS antenna is mounted directly over the vehicle's array mounting point, we will also consider this to be the vehicle's experimental location without co-ordinate transformation. The measurement equations are:

$$\Phi_{(ab,ij)} = \rho_{(ab,ij)} + \lambda(B^i_{ab} - B^j_{ab}) + \epsilon_\Phi \tag{2.1}$$

$$P_{(ab,ij)} = \rho_{(ab,ij)} + \epsilon_P \tag{2.2}$$

$$\rho_{(ab,ij)} = \|\bar{r}_a - \bar{r}^i\| - \|\bar{r}_a - \bar{r}^j\| - \|\bar{r}_b - \bar{r}^i\| + \|\bar{r}_b - \bar{r}^j\| \tag{2.3}$$

where B^i_{ab} represents the single difference of carrier-phase ambiguity. λ is the wavelength of the carrier and ϵ is measurement error. The positioning state vector at time t is $\bar{x}_t$ and its covariance be $\mathbf{P}$, where:

$$\bar{x}_t = \begin{bmatrix} \bar{r_a}^\mathsf{T} & \bar{B_{L1}}^\mathsf{T} & \bar{B_{L2}}^\mathsf{T} \end{bmatrix}^\mathsf{T} \tag{2.4}$$

$$\bar{B} = \begin{bmatrix} B^1_{ab} & B^2_{ab} & \cdots & B^m_{ab} \end{bmatrix}^\mathsf{T} \tag{2.5}$$

The state update of $\bar{x}$ are described by the following Linear Time-Invariant (LTI) system with additive white noise of covariance $\mathbf{Q}$:

$$\hat{x}_{t+1} = \mathbf{A}\hat{x}'_t \tag{2.6}$$

$$\mathbf{P}_{t+1} = \mathbf{A}\mathbf{P}'_t\mathbf{A}^T + \mathbf{Q} \tag{2.7}$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{I}_{3x3} & & \\ & \mathbf{I}_{3x3} & \\ & & \mathbf{I}_{3(m-1)x3(m-1)} \end{bmatrix}, \tag{2.8}$$

$$\mathbf{Q} = \begin{bmatrix} \infty_{3x3} & & \\ & \mathbf{0}_{3x3} & \\ & & \mathbf{0}_{3(m-1)x3(m-1)} \end{bmatrix} \tag{2.9}$$

The update at time t of the state $x(t)$ given a measurement vector y_t is a standard EKF update:

$$\hat{x}'_t = \hat{x}_t + \mathbf{K}_t(y_t - \mathbf{h}(\hat{x}_t) \tag{2.10}$$

$$\mathbf{P}'_t = (\mathbf{I} - \mathbf{K}_t\mathbf{H}(\hat{x}_t)\mathbf{P}_t \tag{2.11}$$

$$\mathbf{K}_t = \left(\mathbf{P}_t\mathbf{H}(\hat{x}_t)(\mathbf{H}(\hat{x}_t)\mathbf{P}_t)^T + \mathbf{R}_t\right)^{-1} \tag{2.12}$$

The measurement model (h) is given by:

$$\mathbf{h} = \begin{bmatrix} \bar{h}^\mathsf{T}_{\Phi,L1} & \bar{h}^\mathsf{T}_{\Phi,L2} & \bar{h}_{P,L1} & \bar{h}^\mathsf{T}_{P,L2} \end{bmatrix}^\mathsf{T} \tag{2.13}$$

$$\bar{h}_\Phi = \begin{bmatrix} \rho_{(ab,12)} + \lambda(B^1_{ab} - B^2_{ab}) \\ \rho_{(ab,13)} + \lambda(B^1_{ab} - B^3_{ab}) \\ \vdots \\ \rho_{(ab,1m)} + \lambda(B^1_{ab} - B^m_{ab}) \end{bmatrix}, \quad \bar{h}_P = \begin{bmatrix} \rho_{(ab,12)} \\ \rho_{(ab,13)} \\ \vdots \\ \rho_{(ab,1m)} \end{bmatrix} \tag{2.14}$$

Define $\bar{e}_i$ as the line of sight vector from the vehicle antenna to satellite i, and σ^2 as the covariance of carrier-phase or pseudorange error, determined by an a priori model. Define the single differencing matrix $\mathbf{D}$:

$$\mathbf{D} = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 \\ 1 & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 1 & 0 & 0 & \cdots & -1 \end{bmatrix} \tag{2.15}$$

Finally, the Jacobian $\mathbf{H}$ and the covariance of measurement error $\mathbf{R}$ are:

$$\mathbf{H} = \begin{bmatrix} -\mathbf{DE} & 0 & \lambda_{L1}\mathbf{D} & 0 \\ -\mathbf{DE} & 0 & 0 & \lambda_{L2}\mathbf{D} \\ -\mathbf{DE} & 0 & 0 & 0 \\ -\mathbf{DE} & 0 & 0 & 0 \end{bmatrix} \tag{2.16}$$

$$\mathbf{E} = \begin{bmatrix} \bar{e}_1^\mathsf{T} & \bar{e}_2^\mathsf{T} & \cdots & \bar{e}_m^\mathsf{T} \end{bmatrix}^\mathsf{T} \tag{2.17}$$

$$\mathbf{R} = \text{diag}\left[\mathbf{DR}_{(\Phi,L1)}\mathbf{D}^\mathsf{T},\ \mathbf{DR}_{(\Phi,L2)}\mathbf{D}^\mathsf{T},\ \mathbf{DR}_{(P,L1)}\mathbf{D}^\mathsf{T},\ \mathbf{DR}_{(P,L2)}\mathbf{D}^\mathsf{T}\right] \tag{2.18}$$

For further RTK hardware and software implementation details, the reader is referred to [168]. WiFi connectivity is required for transmission of the base station reference and therefore RTK accuracy. In the event of temporary loss of connectivity or satellite fix, a local GPS Kalman Filter (KF) estimate is recorded instead, however these events are a negligible portion of experimental time. Additionally, we note that both satellite fix and number of satellites fixed are used heuristically to improve

positioning accuracy by switching positioning modes, where 10 satellites is considered a good fix. Variance in vehicle positioning is empirically estimated at 10^{-4}m^2, and this is the noise covariance used in subsequent experiments.

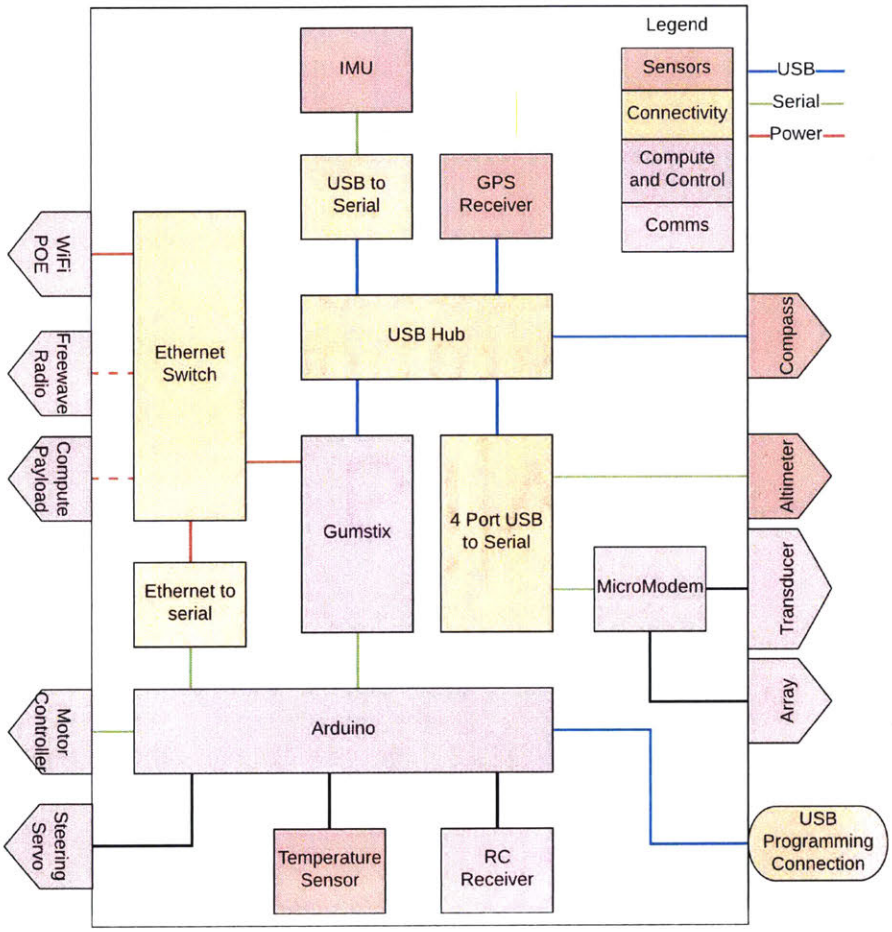

Figure 2-7: Schematic of Electronic System Components and Architecture

Radio frequency (RF) communication with the vehicle is provided by two radios. A 2.4 GHz Bullet M from Ubiquiti provides typical WiFi functionality and bandwidth. Good performance is observed out to 600 meters using an omni-directional antenna (7 dBi) on the vehicle and a sector antenna on shore (12 dBi). Additionally, a longer range communication option is provided by Freewave FGR2-PE 900 MHz ethernet radio. The radios are lower bandwidth (115 Kbps shared among all active radios)

but much longer range, with good performance up to 1km away depending on surface conditions. A full list of component specifications is included in Appendix A.

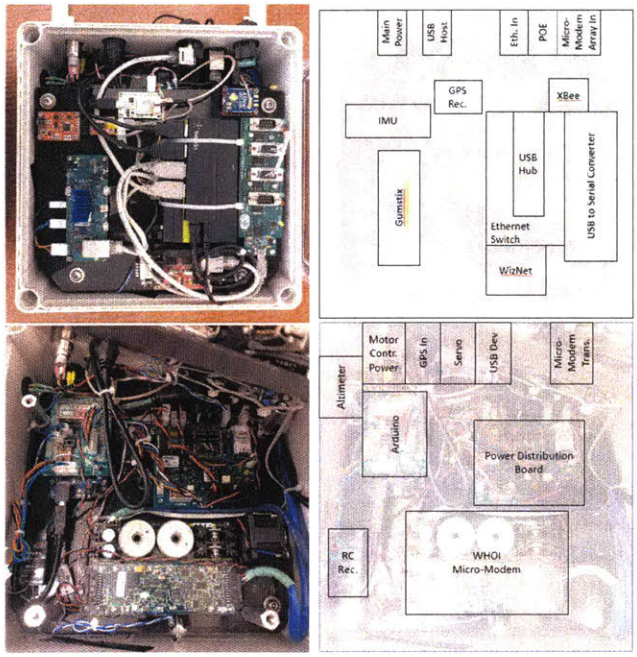

Figure 2-8: MRG-ASV Main Electronics Box (Top and Bottom Layers) Annotated.

Fig. 2-7 presents a schematic view of the ASV's system components, while Fig. 2-8 shows the actual layout in the waterproof box mounted in the seat of the ASV. The ASV is additionally able to mount a payload compute box. The payload compute can run its own MOOS community and interface with the community run on the gumstix. It is powered by the ASV's onboard battery.

2.2.3 Acoustic Modems and Transducers

Two acoustic source and transducer setups, depicted in Fig. 2-9, are used with the MRG-ASV. The WHOI Micro-Modem with a single transducer towfish and four-element array is able to transmit and receive encoded information packets (used in Chapter 3). The pyramidal hydrophone array which only receives is used in conjunction with a Lubell acoustic source for transmitted user-defined waveforms (used in Chapter 5).

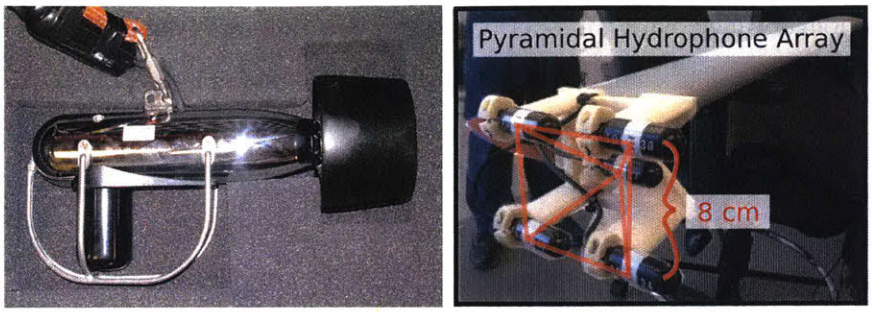

Figure 2-9: Two acoustic systems: WHOI MicroModem Transducer Towfish [63] and piUSBL pyramidal hydrophone array [149]

The WHOI Micro-Modem is a commercially available and widely used acoustic modem [63]. The WH-BT2 transducer towfish is capable of transmitting and receiving. It is towed by the kayak at a depth of 1.5m on a flexible cable, simulating underwater acoustic communications with the experimental benefits of ground-truth RTK-GPS and RF connectivity. This is the maximum depth possible while also ensuring that the vehicle does not run aground on the shallow shelf of the Charles River, so as to reduce the acoustic interference from surface effects. For additional robustness to multipath and packet loss, a four-element HTI-96 MIN array is towed alongside the cable.

Several data packet types differing in error-coding schemes, lengths and data capacities are provided by the Micro-Modem. All Micro-Modem packet types are sent at 25kHz and an acoustic source level of 190 dB rel μPa. Packet types with the Micro-Modem 1 are not flexible, and six types are available with different pairs of coding type and data capacity, potentially allowing selecting a tradeoff between robustness and bitrate. Although encoding, processing and decoding of the binary data into packets is handled exclusively by the Micro-Modem, we present briefly the types of packets used in our field experiments and reasoning for their usage. With all packets one-way travel-time ranging is calculated assuming a time-synchronized source, with onboard clocks synchronized using standard network time protocol via WiFi connectivity pre-launch. In Chapter 3 we use two packet types, the mini-packet and the Rate 0 packets. Both packets are modulated by frequency shift keying (FSK). The

mini-packet is the most robust to packet loss Statistics for acoustic communications in the operational area are provided in Section 2.4. Chapter 3 presents field experiments utilizing the Micro-Modem

Chapter 5 presents field experiments with a pyramidal hydrophone array (receive only) [116]. The array is mounted off the stern of the ASV at a depth of 1.5m. Fig. 2-9 shows the in-water elements of both acoustic systems. A Lubell Labs underwater speaker is used as a transmit source for this array (Fig. 2-5 Left). The element positions of the stern-mounted receiver array are specified (in meters):

$$\mathbf{P} = \begin{bmatrix} 0 & 0 & 0.05657 \\ 0.04 & 0.04 & 0 \\ -0.04 & 0.04 & 0 \\ -0.04 & -0.04 & 0 \\ 0.04 & -0.04 & 0 \end{bmatrix} \tag{2.19}$$

2.3 Autonomy and Operations Software

Mission-Oriented Operating Suite (MOOS) is a lightweight communications architecture built in C++ [119]. Comparable open-source software available include Robot Operating System (ROS) and Lightweight Communications and Marshaling (LCM), however MOOS is convenient for its suite of available marine behaviors provided by MOOS-IvP autonomy software [15]. Fig. 2-10 shows a high-level schematic of the software architecture. The primary communication principle of MOOS is the elimination of peer to peer communication between applications. Instead, a database called the MOOSdb maintains every variable in the operating system to be shared. Apps update these variables according to their functionality asynchronously, and are always pushed the most recent variables they maintain a subscription for.

For example, the GPS driver updates the vehicle's x-y position in the database at a rate of 10Hz, as well as a set of status messages (no. of fixed satellites etc.). The mission-level behavioral driver determines, based on the most recent mission-level command, the appropriate low-level thrust and heading setpoints and communicates

an update to the MOOSdb. This update is immediately served to the drivers for low-level control, which then compute control updates based on the current vehicle position and speed, and translates these into Microcontroller instructions. Every vehicle, and the shoreside station, maintains a separate database. Unless otherwise noted, vehicles perform their missions autonomously.

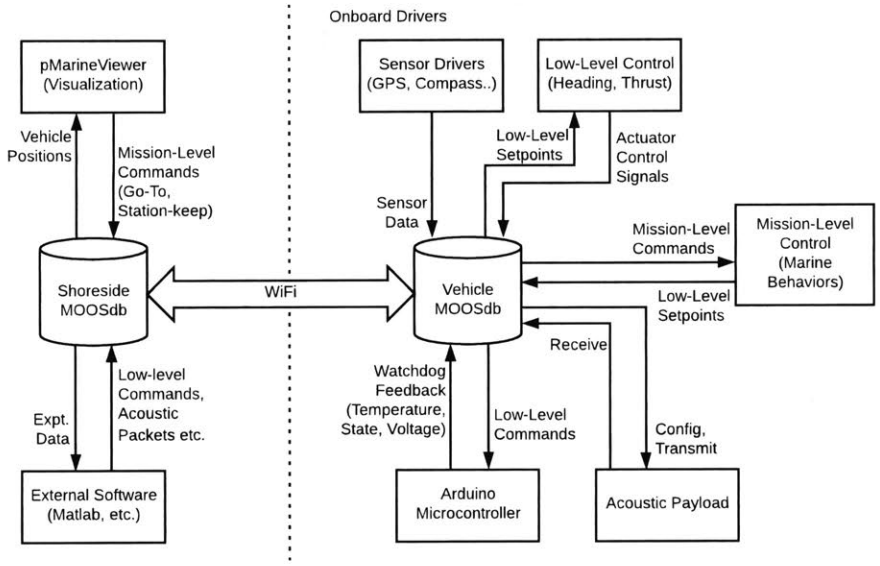

Figure 2-10: MOOS/MOOS-IvP Shoreside and Onboard Vehicle software architecture for the MRG-ASV. The MOOSDB database architecture stores all relevant global variables and provides communication between low-level control and mission-level behaviors.

2.3.1 Autonomous Marine Behaviors

The marine behaviors of waypoint transit with trackline following and station-keeping are implemented within the MOOS-IvP autonomy framework. Integrating these two behaviors with a mission-level state-machine is used to build more complex autonomy instructions. Unlike a typical AUV, the ASV has a very tight turning radius but is prone to side-slip due to the shape of its hull.

The MOOS trackline controller reduces crosstrack error by steering the vehicle towards a point on the trackline (the straight line connecting the vehicle's previous waypoint to its next waypoint) a constant lead distance $d_l = 15$ meters ahead. The vehicle drives towards its leadpoint l_d along the trackline, otherwise it drives directly towards the waypoint when it is less than d_l away. This creates a feedback loop where the vehicle turns more towards the trackline as crosstrack error increases, thereby driving down the error. Once the vehicle enters the capture radius (4m), it is considered to have arrived and transitions to a station-keeping behavior. Fig. 2-11 depicts these behaviors in terms of their algorithmic parameters. Holding a stationary position is typically very difficult for the thrusters/actuators of most AUV/ASVs, and station-keep or loiter behaviors are used instead. Within the inner radius (5m) around the station-keep point, the vehicle is considered close enough and does not drive. Within the outer radius (10m) around the point, the vehicle drives at a limited speed back towards the point. Beyond the outer radius, the vehicle is then free to drive at its maximum transit speed.

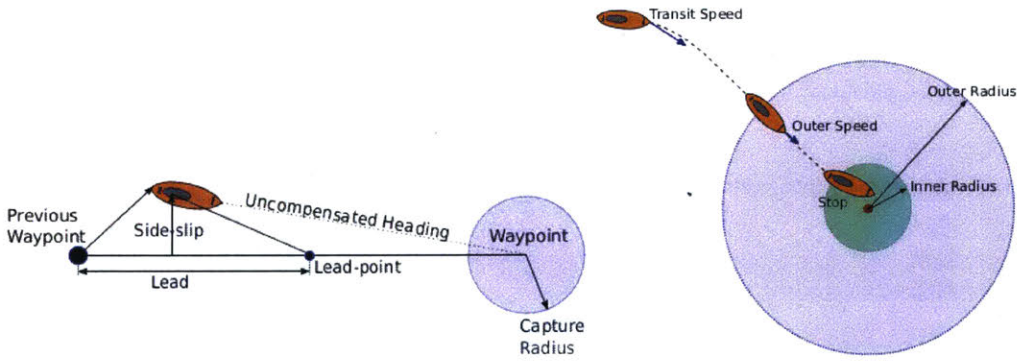

Figure 2-11: Waypoint, Trackline and Stationkeep Autonomous Behaviors

Control commands are typically issued as waypoints. Depending on the mission, waypoint and station-keep behaviors are integrated with a state machine to construct a trajectory. Low-level PID controllers are relied on to drive thrust and heading actuators to their setpoints at a rate that is significantly faster than the trajectory-following dynamics. Low-level setpoints and their respective actuator controls are heading (servo angle) and thrust (motor power). The heading controller is a tuned

PID with a rise time of 4s and 30% overshoot for closed-loop control.

2.4 Operations in the Charles River

All field experiments were performed on the Lower Charles River between the Harvard and Longfellow bridges (Fig. 2-12 and 2-14), off the MIT Sailing Pavilion. A total of three ASVs, *Nostromo*, *Silvana* and *Kestrel* were manufactured and are considered functionally identical for the purposes of our field trials. Additionally, a mobile station, *Icarus* was manufactured, consisting of the software architecture of a typical ASV, but externally equipped with only RTK-GPS, WiFi and acoustic transducers. Typically this mobile station serves as an acoustic beacon with ground-truth location data that can be carried on a motorboat. The Charles River is a challenging shallow-water acoustic environment with depths ranging from 2-12m, hard boundary surfaces (seawalls) on both banks and a soft, muddy base.

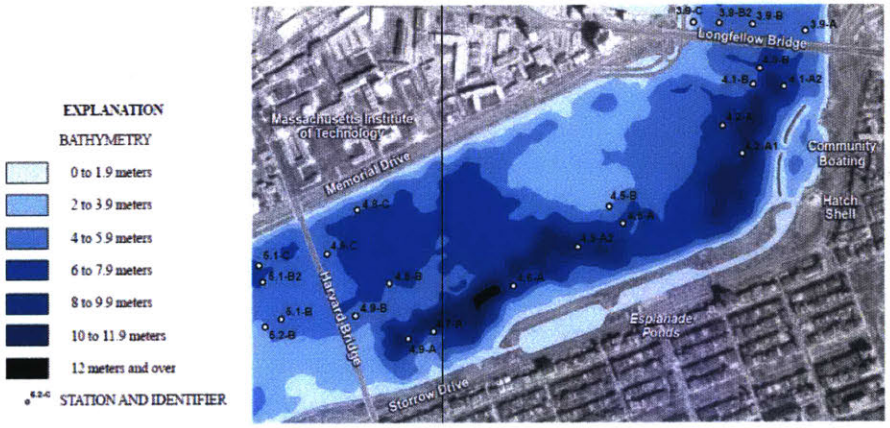

Figure 2-12: Operational Area of the Charles River in front of the MIT Sailing Pavilion, between Harvard Bridge and Longfellow Bridge, USGU Bathymetry map and orthophoto map base by MassGIS [177]

A bathymetry map of the operational area is composed from altimeter data over several lawnmower missions and presented in Fig. 2-13 with depth scale, and Fig. 2-14 over a recent satellite image. The water level in the Lower Charles is controlled by locks and varies seasonally. The experimental data agrees with the survey map. A very shallow shelf towards the east prevents missions with towed transducers. There

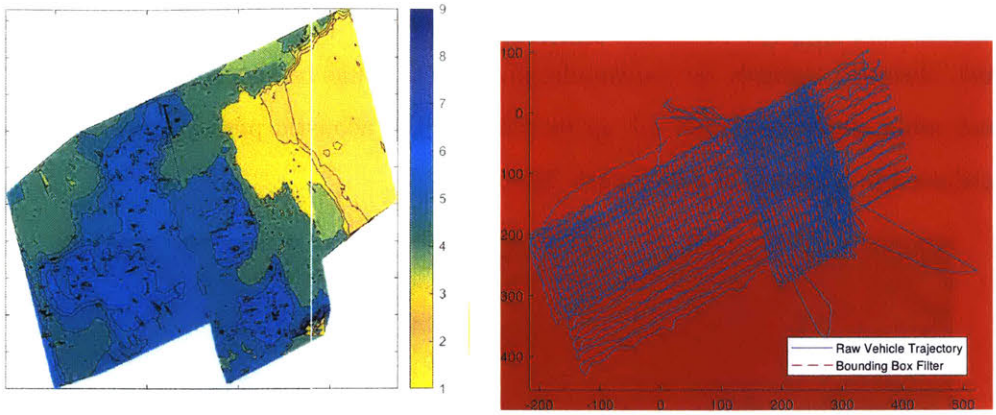

Figure 2-13: (*Left:*) Bathymetry contour map generated from two experimental trials (*Right:*) Vehicle trajectories with respective bounding boxes

is a deep trench towards the southern esplanade that functions as a shadow zone for most acoustic transmissions.

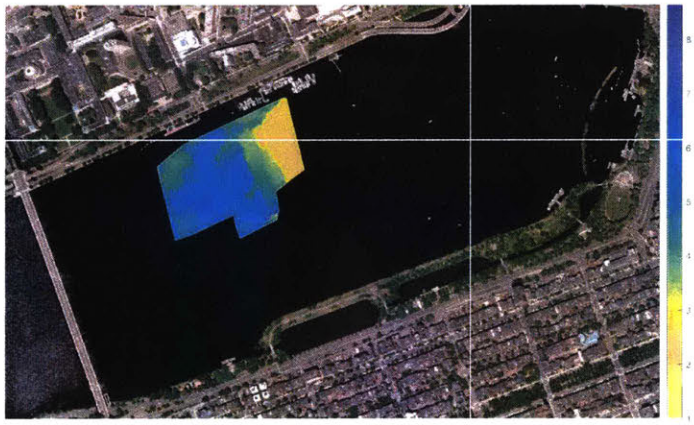

Figure 2-14: Operational Area Composite Bathymetry map

For the WHOI Micro-Modem that transmits at high frequency (25kHz)and power (50W burst), multipath effects are the dominant source of packet loss in the shallow-water environment. Performance of the channel varies widely in space and time, and with weather conditions on the surface [26]. Fig. 2-15 shows SNR statistics reported by the Micro-Modem over several preliminary missions in the direction of either Harvard or Longfellow bridge, up to the maximum connectivity range of the ASV. SNR and packet loss rates were observed to vary spatially and temporally over

the length of several hours and several tens of meters. In some locations, such as a trench observed towards the esplanade, no packets could be successfully received. A round-robin experiment was set up to evaluate the acoustic performance at 9 sites distributed in a square of side length 300m.

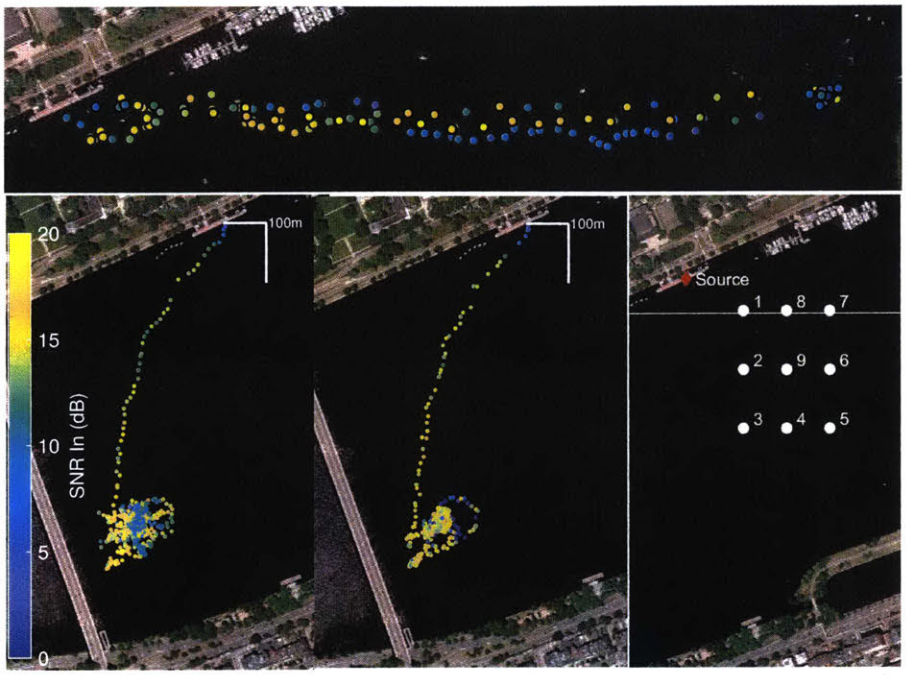

Figure 2-15: WHOI Micro-Modem SNR Statistics from ranges 0 to 700 meters. (*Top:*) Towards Longfellow Bridge (*Bottom:*) Towards Harvard Bridge on different experimental days and at 9 sites distributed in the nominal operating area.

During preliminary field trials, we found that noise from the trolling motor was significantly affecting the modem's performance, as shown in Fig. 2-16. As a result, a default behavior was implemented in software to switch the motor off just prior to scheduled transmits or receipts. Additionally, the Micro-Modem was isolated from the motor driver's power distribution circuit.

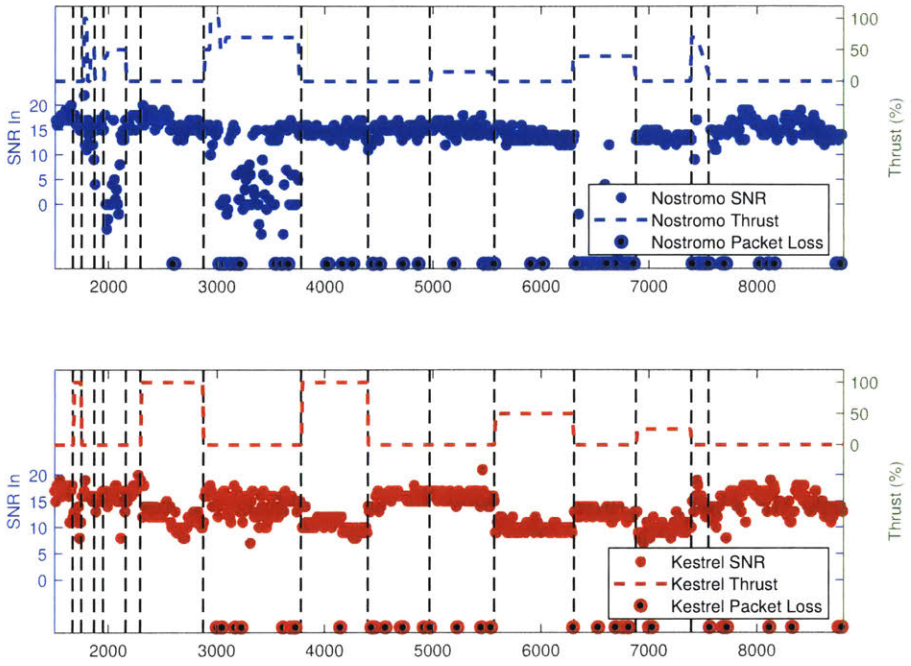

Figure 2-16: Receive SNR reported by the Micro-Modem, plotted against packet loss (shown as points) and current Thrust for two vehicles.

Fig. 2-17 presents receive SNR reported by the Micro-Modem for a round-robin type mission. One modem was mounted fixed at the dock and the other was towed by an ASV visiting each of nine sites in a fixed sequence. Five two-way transmissions of a filled FSK0 packet were sent each time a site was visited. We note that SNR is reported by the Micro-Modem if a packet was detected even if it was not successfully decoded. Missed or corrupted packets are denoted by a grey background line.

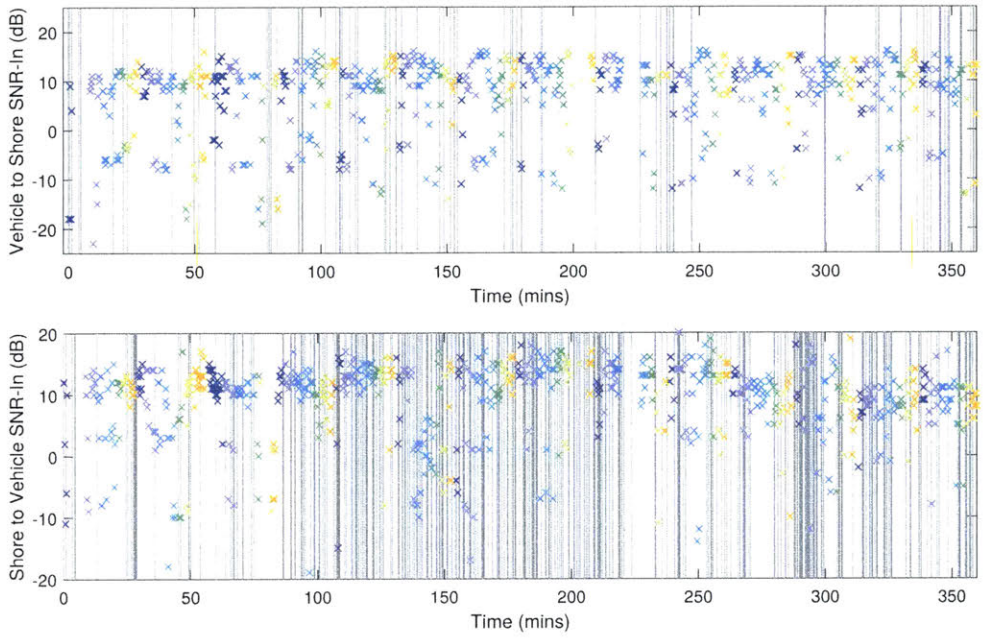

Figure 2-17: Receive SNR reported by the Micro-Modem for a two-way link. (*Top*:) Vehicle towards the shore. (*Bottom:*) Shore towards vehicle, recorded at nine sites (color-coded) in a round-robin loop. Five two-way no-ack transmissions were sent per round-robin visit. Grey lines indicate missed packets.

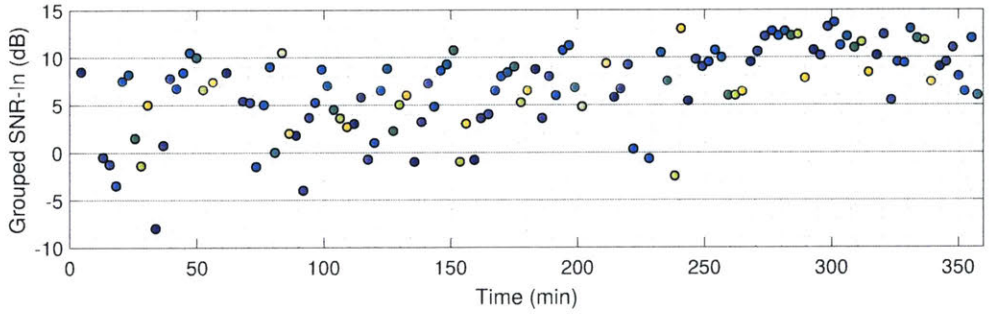

Figure 2-18: Grouped averaged Receive SNR reported by the Micro-Modem for two-way communication between Shore and Vehicle, recorded at nine sites (color-coded). Successful packets are grouped by site with at maximum five two-way no-ack transmissions per grouped datapoint.

In this instance, one direction of the channel (towards the shore) performed noticeably better than the other. For each visit to a single site, SNR was grouped and averaged, as shown in Figs. 2-18 and 2-19, and the packet loss rate was calculated. We observe that the overall performance of the channel varies significantly in time and space, with packet loss rates for higher bitrate packets as high as 50%. The river environment also includes potentially random and unmodeled sources of noise such as passing boats.

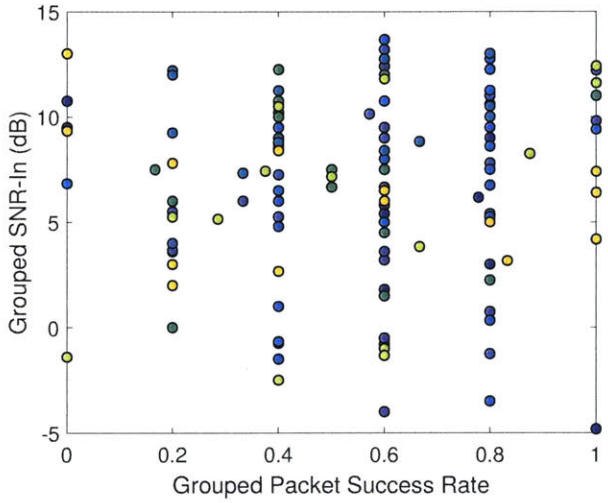

Figure 2-19: Grouped averaged Receive SNR reported by the Micro-Modem for the FSK0 packet, color-coded by site and plotted against overall Packet Success rate, with at maximum five two-way no-ack transmissions per grouped datapoint.

Acoustic statistics for the piUSBL pyramidal array and Lubell acoustic source are presented in Chapter 4 together with a discussion of the data processing techniques applied to the recorded waveforms. Experimental characterization of the SNR received at this array are presented in Figs. 4-2 and 4-3. Additional details on this acoustic payload can be found in [148]. The main sources of noise were found to be low frequency acoustic noise from surface effects such as wind and waves, and high frequency electronic noise on the power distributions circuit. Due to hardware constraints, the DAQ payload could not be isolated from the main power distribution circuit, however we show that processing with a matched filter significantly eliminates this source of noise.

2.5 Conclusion

This chapter has detailed the design of low-cost, lightweight ASVs fully capable of autonomous dynamic control, high-level mission behaviors and supporting a variety of sensors and acoustic payloads. Towards the overarching goal of developing and demonstrating highly dynamic multi-vehicle algorithms in the field, these platforms provide a combination of real-world underwater acoustic conditions and data and ground-truth via RTK-GPS. Preliminary characterization of the operating area in the Charles River were presented in the form of a bathymetry map and modem SNR statistics. In addition to the experiments presented in this thesis, we are happy to report the use of these vehicles in other work related to the underwater and acoustic domains [140], [128].

Chapter 3

De-centralized Multi-Vehicle Dynamic Pursuit with Acoustic Communications

In this chapter, we present a joint estimation and control framework for multiple de-centralized underwater agents collaborating to track a maneuvering target. In particular, the problem of range-only (TDOA) pursuit in a shallow-water acoustic environment is studied with the goal of characterizing the performance of acoustics-in-the-loop, in which the acoustic modem is used as the measurement as well as the main communications channel. The fundamental limits of the UWB channel means without sophisticated multiplexing techniques, there exists a trade-off in the control space of frequency and quantization of information updates against the measurement. The effects of cycle time, packet loss and de-centralization on the frequency response of the full integrated system is analyzed, and we demonstrate an implementation of partial information updates that decreases the effective cycle time. Experimental results are presented with three ASVs in the field and four WHOI Micro-Modems as the acoustic network, showing that at present, highly dynamic multi-vehicle missions are sustainable by the acoustics channel.

3.1 Motivation and Prior Work

Our thesis is the advent of small, light-weight and low-cost marine robotic systems in recent years has provided the necessary platform to undertake truly dynamic multi-vehicle collaborative missions underwater. As a small step towards developing advanced algorithms for this purpose, this chapter studies the de-centralized multi-vehicle estimation problem over an acoustic network of mobile agents. The general pursuit problem has been extensively studied on land, in the air and at sea. Search and pursuit-evasion games, also known as one-sided search and adversarial search, are canonical missions with a broad range of algorithms and solutions [30], [183]. Underwater, the mission of tracking and pursuing a marine animal or oceanographic feature such as a temperature gradient or front fits into this set of problems.

Specifically, strategies may be generally categorized over the environment of the search, the pursuer dynamics and the target dynamics. This chapter in particular addresses estimation of the target's dynamics. The environment may be modeled as discrete or continuous. The former has been studied in the context of cops and robbers games on a graph for many years [120], while the latter has been studied in the context of a lion-and-man game on the open plane [6], in which the purpose is typically to identify the conditions under which the game can be 'won' by the pursuers. On mobile robotic systems, the problem extends to include continuous control, navigation and path-planning. Estimation and modeling of the target or evader has long been a main area of research [102], while the constraints of the agent communication network such as packet loss and scheduling has received less attention [112]. De-centralized approaches often indirectly address communication constraints by minimizing the exchange of information between agents [29]. Multi-robot localization with kalman filters distributed to local agents was studied by [146]. Optimal path-planning to maximize the rate of information collection by the mobile agents is an NP-Hard problem that has been variously studied with approximate methods such as LP-relaxation and finite-horizon planning [201], [78]. In the ocean, the presence of strong flow fields and limited vehicle dynamic capability have been

addressed by a level-set method for time and energy optimal path-planning [99].

Above-water, there are a number of experimental results with multiple collaborating robots tracking a target. In [103], tracking with range-only data was shown with a team of mobile robots collaborating to plan paths that would increase the accuracy of a particle-filter estimator. A particle-filter based experiment with four robots was demonstrated to locate an unknown intruder by collaboratively distributing the search space [113]. Hausman *et al.* evaluated the performance of EKF and measure of uncertainty nonlinear joint estimation algorithms onboard UAVs for one and two robot missions in target-tracking and kidnapping [71]. Experimental works related to pursuit in the marine environment include tracking a tagged leopard shark in extremely shallow water with an array 2.4m in length mounted on an AUV [31]. A real shark was tagged with an acoustic beacon as the target, although it only moved approximately 200m in the 48 minutes reported. Bean *et al.* demonstrated range-only leader-follower formation flight with WHOI Micro-Modem mini-packets [12], while a similar mission was reported in [20] with DSPComm modems and asymmetric vehicles operating at 0.5m/s and 3m/s respectively. Cruz *et al.* study a two-way centralized link in which mobile followers communicate their positions to a controller which then transmits back control commands [34]. The agents travel relatively slowly, around 10cm/s, and the cycle time is around 20s. A vehicle following two leaders in a triangle formation was demonstrated experimentally, with ranges around 15m, speeds around 0.5m/s and a 4 second cycle time [161].

In earlier work [140], we studied upper performance limits for a two-vehicle system pursuing a target; the experiment involved acoustic range-only lateration via interleaved acoustic communications. This chapter reports on time-difference-of-arrival (TDOA) pursuit with three vehicles. Here, the target is not time-synced to the pursuit system, hence opening the way toward high-speed pursuit of non-cooperative targets, i.e. those with identifiable signatures but arbitrary transmission timing. We report experimental results for both centralized and decentralized communication structures. In general, TDOA measurements with three or more receivers give rise to hyperbolic functions that intersect at one or more points of origin. Hyperbolic

estimation is a non-linear estimation problem that can be solved with approximate closed-form solutions in some instances [24],[65], alternatively linear [197] and non-linear optimization methods [198] have also been applied. In this work, a non-linear sigma-point filter is used for estimation as it is well-suited to handling packet losses as well as scheduling delays in measurement updates.

3.2 De-centralized Range-Only Estimation

The behavior of a maneuvering ocean target, i.e. a ship or an animal, can be represented by the following general-purpose nonholonomic 2D dynamic model:

$$\begin{aligned} \dot{\phi}_d &= Q \\ \dot{x}_d &= V\cos(\phi) \\ \dot{y}_d &= V\sin(\phi) \end{aligned} \tag{3.1}$$

where the assumption is that the target travels at a constant along-track speed V with stochastic noise-driven zero-mean turning rate $Q(t)$. Let $[x_d\ ,\ y_d]$ be the target's 2D Cartesian position co-ordinates, ϕ_d be the heading angle, and $\dot{\phi}_d$ be the heading turn rate. The target's estimated state vector is then $\hat{s}(t) = [\hat{\phi}_d(t)\ ,\ \hat{x}_d(t)\ ,\ \hat{y}_d(t)]$. For convenience of notation, let the estimated target position vector be $\hat{p}(t) = [\hat{x}_d(t)\ ,\ \hat{y}_d(t)]$. Then, the system of differential equations represented by Eqn. 3.1 is the nonlinear motion model describing the target's state $s(t) = g(s_{t-1,\cdots,0})$. Although more precise ship motion models are available in the literature, we will assume for the mission that the characteristics of the target are largely unknown.

A nonlinear Hermite-Gauss Quadrature sigma-point filter, also known as an unscented filter, is used to estimate the target's location with partial or full sets of TDOA measurements from each collaborator [83]. The principle of the sigma-point filter is that a set of discretely sampled sigma points can be used to parametrize the mean and covariance of a state without the need for linearization, such that the resulting first and second statistical moments can be computed when the sample points

are propagated through a nonlinear system. Sigma points are typically generated by computing the following $2N + 1$ weighted points for an N dimensional state vector with mean $\bar{x}$ and covariance $\mathbf{P}_{xx}$, given by:

$$\mathbf{X}_0 = \bar{x} \qquad w_o = \frac{k}{N+k} \tag{3.2}$$

$$\mathbf{X}_i = \bar{x} + \left[\sqrt{(N+k)\mathbf{P}_{xx}}\right]_i \qquad w_i = \frac{k}{2(N+k)} \tag{3.3}$$

$$\mathbf{X}_{i+N} = \bar{x} - \left[\sqrt{(N+k)\mathbf{P}_{xx}}\right]_i \qquad w_{i+N} = \frac{k}{2(N+k)} \tag{3.4}$$

where $k \in \Re$ is a tuning parameter, $[\,\cdot\,]_i$ indicates the ith row or column of a matrix, and w_i is the weight associated with the ith point. The weights are selected such that the mean and covariance of the state represented by the sigma points are always recovered by the conventional first and second moments:

$$\mu = \sum_i w_i \mathbf{X}_i \tag{3.5}$$

$$\sigma^2 = \sum_i w_i (\mathbf{X}_i - \mu)^2 \tag{3.6}$$

The sigma points are then propagated through the nonlinear system $g(s_t)$, and the mean and covariance can be found by:

$$\mathbf{X}(t+1 \mid t) = g\left(\mathbf{X}(t)\right) \tag{3.7}$$

$$\hat{x}(t+1 \mid t) = \sum_i w_i \mathbf{X}(t+1 \mid t) \tag{3.8}$$

$$\mathbf{P}(t+1 \mid t) = \sum_i w_i \left[\mathbf{X}(t+1 \mid t) - \hat{x}(t+1 \mid t)\right] \times \left[\mathbf{X}(t+1 \mid t) - \hat{x}(t+1 \mid t)\right]^{\mathsf{T}} \tag{3.9}$$

Hermite (Gaussian) quadrature sigma-points are used in our implementation, generating $3N$ sigma-points given by the following abscissas and weights:

$$\begin{aligned} \bar{x} &= \begin{bmatrix} -1.224 & 0 & 1.224 \end{bmatrix} \\ w &= \begin{bmatrix} 0.2954 & 1.1816 & 0.2954 \end{bmatrix} \end{aligned} \tag{3.10}$$

The mean and covariance priors, and finally the state vector can then be updated with measurement information using a standard Kalman Filter innovation step.

$$\begin{aligned} K(t) =& P(t \mid t-1) H_k^{\mathsf{T}}(t) \times \\ & (H_k(t) P(t \mid t-1) H_k^{\mathsf{T}} + \mathbf{R})^{-1} \end{aligned} \tag{3.11}$$

$$\hat{s}(t) = \hat{s}(t \mid t-1) + K(t)(z(t) - \hat{z}(t)) \tag{3.12}$$

$$P(t) = (I - K(t) H_k(t)) P(t \mid t-1) \tag{3.13}$$

where $\mathbf{R}$ is the diagonal matrix with diagonal elements R corresponding to the measurement noise covariance in the respective dimensions. Experiments were performed with surface vehicles for the benefit of RTK GPS ground-truth position measurements, thus the system is formulated in the 2D plane. The estimator can easily be extended to 3D by re-deriving the observation gradient, however at least one additional tracking vehicle is required.

A full information estimate is computed as soon as the entire communication cycle is complete. In the 2D three vehicle mission, partial and full corresponds to two measurements (one communication) and three measurements (two communications) respectively. If the ranging transmission is lost, or a vehicle does not receive any additional measurements, we use a standard approach of setting the noise of the lost measurement to infinity [157]. We define the full update for the case of 3 agents. Let the three TOA measurements be m_i, m_j, m_k. If a full estimate is made, the state estimate of the partial is also updated. In both cases, the estimator gain matrix K and covariance matrix P are then updated by typical Kalman Filter equations, where $P(t \mid t-1)$ represents the *a priori* covariance.

$$\hat{z}(t) = \begin{bmatrix} r_i(t) - r_j(t) \\ r_j(t) - r_k(t) \end{bmatrix}$$

$$z(t) = c \begin{bmatrix} m_i(t) - m_j(t) \\ m_j(t) - m_k(t) \end{bmatrix}$$

$$H_k(t) = \left[\begin{pmatrix} 0 \\ (r_i(t)(\hat{x}_d(t \mid t-1) - x_i(t)))^{-1} \\ \quad -(r_j(t)(\hat{x}_d(t \mid t-1) - x_j(t)))^{-1} \\ (r_i(t)(\hat{y}_d(t \mid t-1) - y_i(t)))^{-1} \\ \quad -(r_j(t)(\hat{y}_d(t \mid t-1) - y_j(t)))^{-1} \end{pmatrix} \right. , \tag{3.14}$$
$$\left. \begin{pmatrix} 0 \\ (r_j(t)(\hat{x}_d(t \mid t-1) - x_j(t)))^{-1} \\ \quad -(r_k(t)(\hat{x}_d(t \mid t-1) - x_k(t)))^{-1} \\ (r_j(t)(\hat{y}_d(t \mid t-1) - y_j(t)))^{-1} \\ \quad -(r_k(t)(\hat{y}_d(t \mid t-1) - y_k(t)))^{-1} \end{pmatrix} \right]$$

In addition to the full information update, a partial information estimate is computed and acted upon as soon as any two time-of-arrival (TOA) measurements are available at any vehicle. This decreases the effective cycle time and provides robustness in the event of packet loss of any one transmission in the cycle. Given TOA measurements m_i, m_j from vehicles i, j, where $i \neq j$, we compute the observed change in target location z and the observation gradient H_k for innovation. Let $v_i = [x_i\ ,\ y_i]$ as the vector of Cartesian position co-ordinates for vehicle i. Partial estimates are generated prior to and separately from full information estimates. A zero first row is present in the observation matrix to be consistent with the definition of the state vector, although the target's heading is assumed to be unobserved as the acoustic measurement is range-only.

$$r_i(t) = \|\hat{p}(t \mid t-1) - v_i(t)\|_2 \tag{3.15}$$

$$r_j(t) = \|\hat{p}(t \mid t-1) - v_j(t)\|_2 \tag{3.16}$$

$$\hat{z}(t) = r_i(t) - r_j(t) \tag{3.17}$$

$$z(t) = c(m_i(t) - m_j(t)) \tag{3.18}$$

$$H_k(t) = \begin{bmatrix} 0 \\ \Big((r_i(t)(\hat{x}_d(t \mid t-1) - x_i(t)))^{-1} \\ \quad -(r_j(t)(\hat{x}_d(t \mid t-1) - x_j(t)))^{-1}\Big) \\ \Big((r_i(t)(\hat{y}_d(t \mid t-1) - y_i(t)))^{-1} \\ \quad -(r_j(t)(\hat{y}_d(t \mid t-1) - y_j(t)))^{-1}\Big) \end{bmatrix} \tag{3.19}$$

Fig. 3-1 shows two general block diagram representations of centralized and decentralized feedback systems with lossy communication channels in the loop. A decentralized system means each agent computes its local estimate and control action independently of collaborators as soon as the measurements are received. Agents may take action at different times and have asymmetric estimates. A centralized system architecture assumes measurements are communicated from local agents (followers) to a single controller (leader) which then computes and communicates the control actions back through the same lossy channel.

The advantages of a centralized controller are that control actions are computed based on a single consensus estimate, whereas decentralized systems may diverge from consensus over time. External information such as human in-the-loop decision-making or large-scale data assimilation and cloud computation may be easily integrated. The disadvantages of the centralized architecture is the additional control message that has to be sent, and over the acoustic channel, susceptibility to packet loss and two-way travel times. The consideration of acknowledgments (acks) for the control packet to ensure networked stability may also result in a Byzantine General's problem.

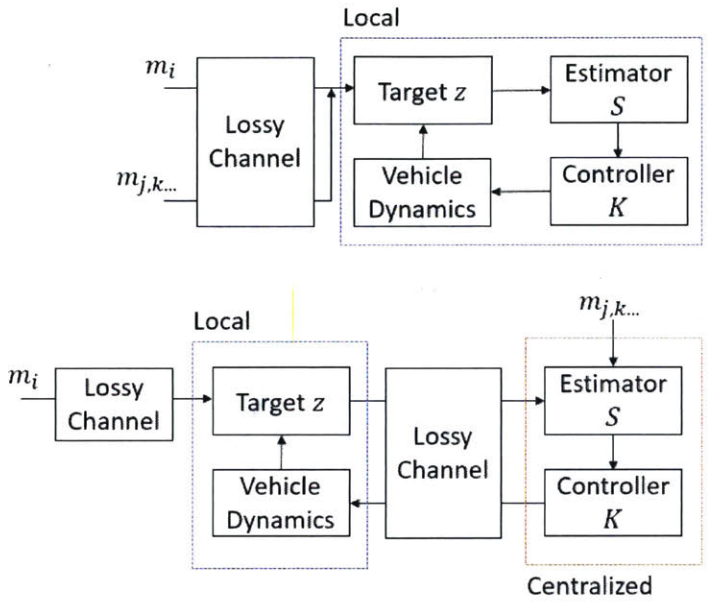

Figure 3-1: General Block Diagrams for Feedback Systems with Lossy Communication Channels in the Loop. (*Top:*) Decentralized system. (*Bottom:*) Centralized system.

Previous work with a centralized controller has shown that the overall cycle time has a significant impact on tracking performance when the control action must be communicated over the acoustic link [140].

For multi-vehicle acoustic communications, time-division multiple access (TDMA) scheduling is commonly used to prevent collisions and interference. Due to the TDMA schedule, each vehicle updates its information state at different times, and this inequality is only emphasized when more agents are required. In this work we show that a decentralized control system can effectively decrease the overall cycle time and improve the system's dynamic response as it enables each vehicle to take action immediately upon receiving enough information. Further, we allow each vehicle to compute an estimate based on only partial information and start taking action even earlier in the cycle.

Fig. 3-2 shows the slots in the TDMA schedule. The control cycle is as follows: The target broadcasts a non-time-synced acoustic message, and the vehicles compute TOA measurements. Follower vehicles broadcast their TOA measurements and po-

sitions during their designated TDMA slot. Every message heard is concatenated and repeated by the receiver for increased robustness to packet loss without affecting packet transmission characteristics. Each vehicle computes a partial update of the target's location as soon as possible, and begins driving toward a waypoint determined by the vehicle formation relative to the target's new position. All vehicles compute a full update at the end of the cycle and generate new waypoint commands, which are given priority over partial updates.

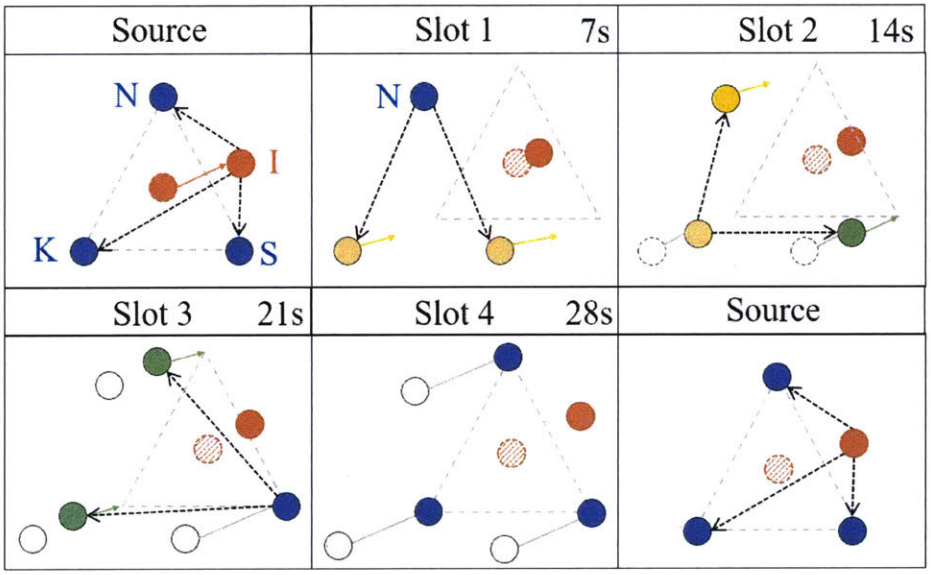

Figure 3-2: Decentralized TDMA schedule by slot timing for field experiments with three mobile agents. The acoustic source I is shown at the center of an equilateral triangle formation. Agents N, K, S make a partial state update (amber state) when two TOA data packets are available and a full state update (green state) with all three data packets. Within the total cycle time, the source does not transmit again.

The control action is generated by assuming that agents seek to maintain the same formation relative to the target. This is a simplification of the general path-planning problem since the TDMA cycle time between waypoint updates is kept as short as possible. Low-level dynamics of the vehicle such as heading control and thrust control are handled by the controllers as described in Chapter 2.

The TDMA schedule is pre-determined and keyed off a target transmission, that is vehicles only transmit in their designated slot if they have received new information

acoustically this cycle (Fig. 3-3). In the event the ranging transmission from the acoustic source is lost, vehicles continue towards or station-keep at their previous waypoints. Experimentally, the target to programmed to broadcast approximately every 28 seconds based on its local clock time. Each slot is 7s long, allowing 5s for transmission and 2s of guard time. Vehicles make updates and take action as soon as enough information is available. It is clear from the scheduling that the vehicle transmitting last is able to make a full update first, and the time between partial updates by each vehicle also varies. One heuristic to prevent this implicit bias could be to implement a meta-schedule where vehicles cycle positions in the schedule every so often. For field experiments, we designate each vehicle by their name and their order of transmission in the TDMA schedule: Nostromo (*N1*), Silvana (*S2*) and Kestrel (*K3*). The source is designated Icarus (*I*).

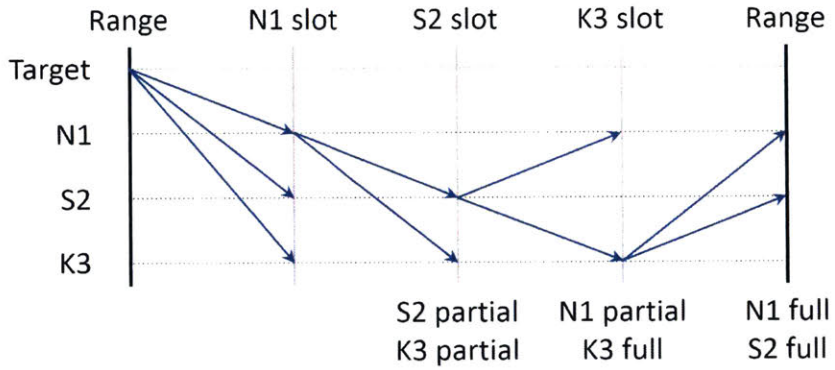

Figure 3-3: Diagram of TDMA schedule for decentralized control architecture showing the broadcast transmission slot for each vehicle. Acoustic transmissions are shown with arrows. Vehicle updates are shown below the appropriate slot in the cycle assuming no packet losses occur.

In addition, we implement a centralized controller with the same sigma-point filter estimator and messaging scheme. This control architecture could be preferred if the controller was also a fusion center for outside information, if some vehicles were out of communication with others, or if vehicles had different computational abilities. However, the centralized controller may be less robust to packet losses. In this implementation, estimation and control are computed and broadcast acoustically from a

centralized shore station fusing information from three pursuers. The TDMA schedule is adapted to include an additional fifth slot dedicated to the acoustic transmission of a control command (Fig. 3-4), bringing the total cycle time to approximately 35 seconds. The period of the acoustic ranging transmission is adjusted to match. As in the decentralized schedule, each vehicle broadcasts all the information it has acquired the current cycle. The centralized estimator computes a full update and transmits a multiplexed control command acoustically in the last slot.

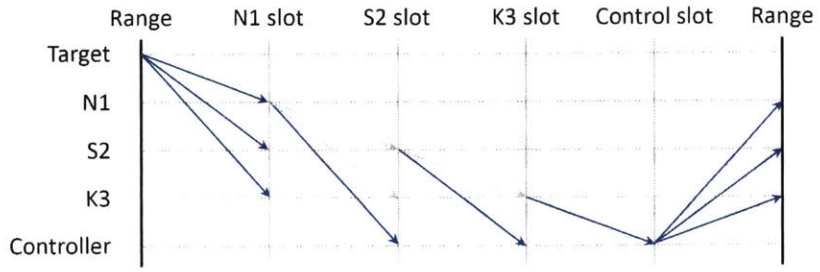

Figure 3-4: Diagram of TDMA schedule for centralized control architecture. An additional slot is introduced for the transmission of control commands from a shore station. Grey arrows represent redundant information paths.

No acknowledgments are sent for either control architecture. One possible implementation to avoid adding a TDMA slot is to designate one of the pursuers as the centralized controller and broadcast control commands in its designated slot instead. This comparison would investigate the benefits of pursuers taking partial actions independently (as the overall communication cycle is of the same length), while the centralized scheme we have chosen investigates the importance of shortening the overall communication cycle.

3.3 Field Experiments

Field experiments were performed with nonholonomic physical and virtual targets. The physical target consisted of a motorboat towing a Micro-Modem and driven approximately randomly at an average speed of 1m/s. The parameters of the motion model were set to be $V = 1$ m/s and heading turning rate $Q = 0.01$ to match this

behavior. The target's Micro-Modem is not time-synced and experiences variations in transmission time of 200ms on average. Trajectories of the virtual target were generated using by the differential system of Eqn. 3.1. We estimate the covariance of the measurement noise to be $R = 0.04$ based on prior characterization of Micro-Modem ranging capabilities [63],[38],

We report results for field experiments conducted with both the centralized and decentralized controller. The decentralized controller was tested with three formations (Fig. 3-5); 'inner' with the target at the center of an equilateral triangle, 'outer' with the target outside the triangle, and 'line' with the vehicles in a staggered line perpendicular to the target. The average distance from each vehicle to the target (a measure of 'formation length') was 90m, 90m, and 75m respectively. Formations are considered to measure the target with high fidelity in the sense of achieving good horizontal dilution of precision (HDOP) properties. Two 'inner' and 'outer' formations were chosen for good resolution of the target and for comparison, the 'line' formation for poor resolution.

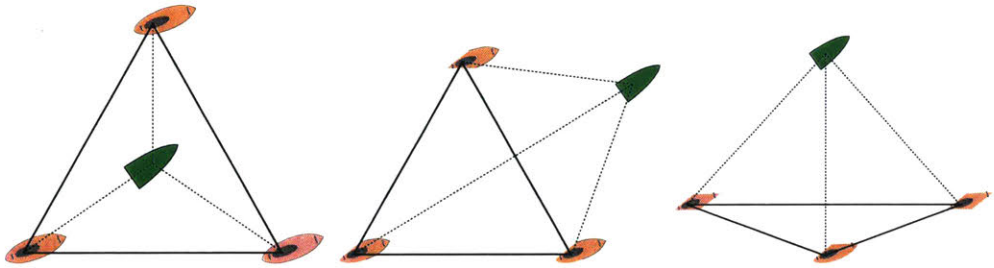

Figure 3-5: Three-agent Formations for target-tracking and pursuit field experiments. Target is shown in green. (*Left:*) Inner formation. (*Middle:*) Outer formation. (*Right:*) Line formation.

The trajectories taken by the target acoustic source for each experiment trial are shown in Fig. 3-6. Trials for the de-centralized architecture are shown at left. Trajectories were made by a nonholonomic physical target, a human-driven motorboat turning psuedo-randomly at a constant forward speed of 1.2m/s. The 'line' formation shown in red is a shorter mission as its performance is expected to be poor. For the centralized architecture, due to the hardware constraints of including an additional modem for communicating the control packet, virtual trajectories were generated

from the stochastic turning rate model, similar to that of the physical target. The centralized controller was tested with two formations, 'inner' and 'outer' and two randomly generated virtual trajectories. For field experiments with virtual trajectories, only the location of the target and the range measurement were simulated, all acoustic transmissions and vehicle motion were equivalent to the decentralized case, with commands being sent over the acoustic channel in the field.

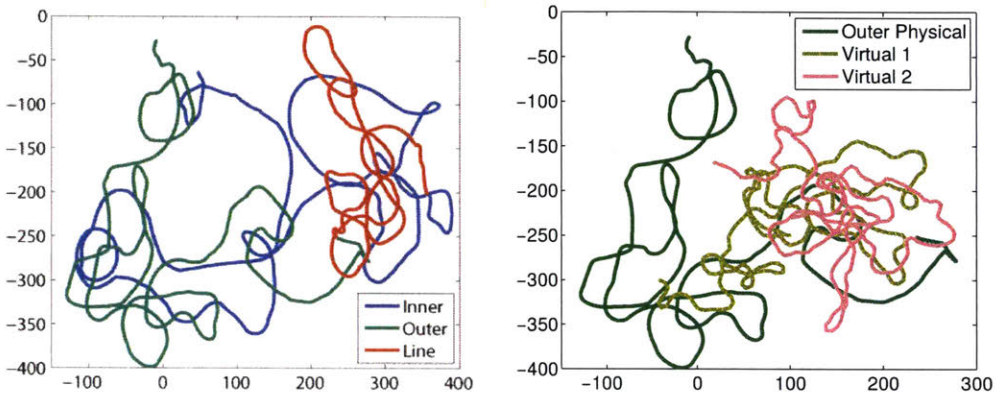

Figure 3-6: Field trials trajectories of acoustic source. (*Left:*) Trajectories shown for three formations with a physical target, a human-driven motorboat. (*Right:*) Two target trajectories simulated with stochastic heading rate model, plotted with the 'outer' formation trial with a physical target for comparison.

Changes in the vehicles' formation over the mission is a intuitive visual representation of the performance of the estimator. Given an ideal estimator, the formation maintained by the agents should always be the desired formation relative to the maneuvering target. The dynamic capabilities (speed and turning rate) of the ASVs are better than the target's. For clarity, formations are shown only every 10 TDMA cycles in Fig. 3-7. Although the de-centralized estimator does not explicitly reach a consensus on the target's state estimate, we observe that the vehicles in general maintain the desired formation. As in the right plot, the de-centralized estimator is also robust to changes in formation. The 'line' formation, which was expected to demonstrate poor performance, is shown in Fig. 3-7. We observe in Fig. 3-7 that the formation of the vehicles occasionally converges to a straight line, for which the sigma-point estimator is unable to resolve target movement in the y-direction.

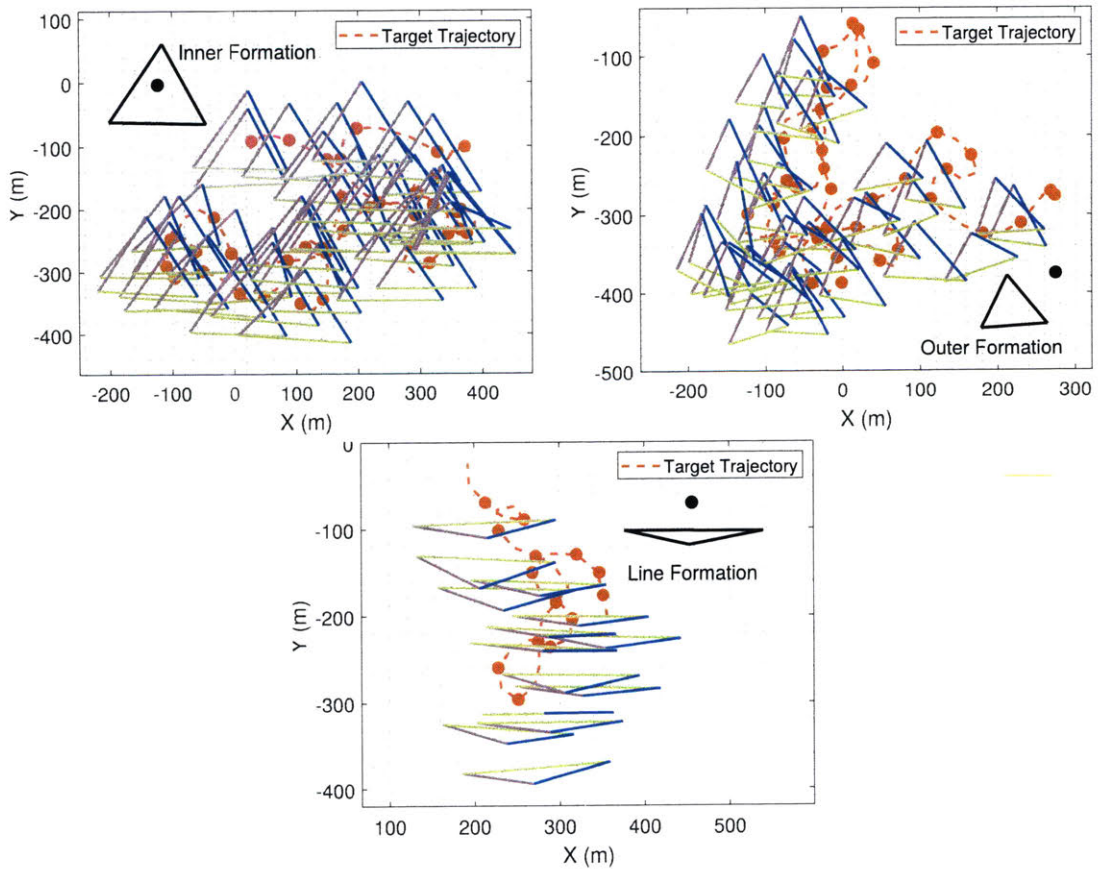

Figure 3-7: Field trials formation of three tracking vehicles, shown every 10 cycles. The target is shown in orange and the desired formation is shown in black. (*Left:*) Inner formation. (*Right:*) Outer formation. (*Middle:*) Line formation.

Similarly, formations for the centralized architecture are shown in Fig. 3-8. The centralized controller's estimate of the target is shown in black, while the vehicle's location at that time step is shown in pink, noting that the controller is estimating the vehicle's location approximately 30 seconds prior.

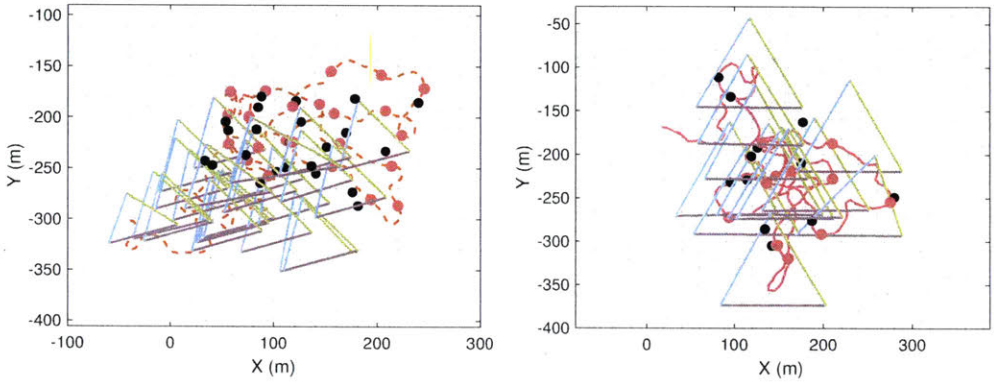

Figure 3-8: Field trials formation of three tracking vehicles for a centralized estimator. Vehicle ground truth is shown in pink and central estimate is shown in black. (*Left:*) Outer formation. (*Right:*) Inner formation.

3.4 Experimental Results

The performance of both system architectures based on the three formations shown in Fig. 3-5 is analyzed by two metrics: 2-norm error in the estimate state of the target and frequency analysis of the control system's empirical transfer function. Fig. 3-9 shows the 2-norm of the tracking error $\|\hat{s} - s\|$ for full information estimates only by the de-centralized controller. Table 3.1 presents the corresponding means of the tracking error.

Table 3.1: Mean Tracking Error by Formation Type

Mission	Nostromo (m)	Silvana (m)	Kestrel (m)
Inner	28.0	27.2	8.61
Outer	30.1	28.9	17.9
Line	61.2	61.3	60.8

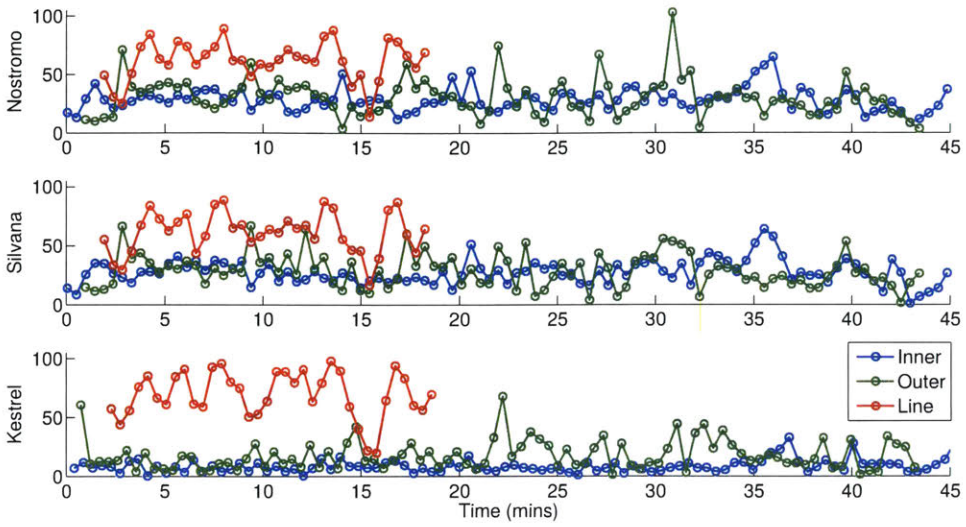

Figure 3-9: Tracking Error for decentralized controller architecture by formation and vehicle. Only the error for full information estimates are shown.

The de-centralized nonlinear sigma point filter on three agents consistently tracks the target with nearly constant error for the full length of the mission. The tracking error is bounded within 30% of the formation length for the 'Outer' and 'Inner' formations, i.e. the target always stays on the interior or exterior of the formation depending on where it started. The system does not go unstable despite vehicle deviations from a fixed formation and differing estimates without consensus-keeping between vehicles. For the 'Line' formation, the tracking error is within 80-90% of the formation length, keeping in mind that tracking performance was expected to be worse for this 'poor' formation, since the vehicles' TOA measurements contain very little information on the target's motion in the y-direction. Even so, the system did not go unstable and estimators did not completely lose the target.

For the 'Inner' and 'Outer' formations, ASV *Kestrel* has a significantly lower mean tracking error than the other two. This is because since it is the last in the cycle to transmit and the first to make a full information estimate, it has an effective cycle time that is 7 seconds shorter. An equally poor estimate in the 'Line' formation supports the conclusion that the error in the 'Line' formation is not a result of update delays due to cycle time.

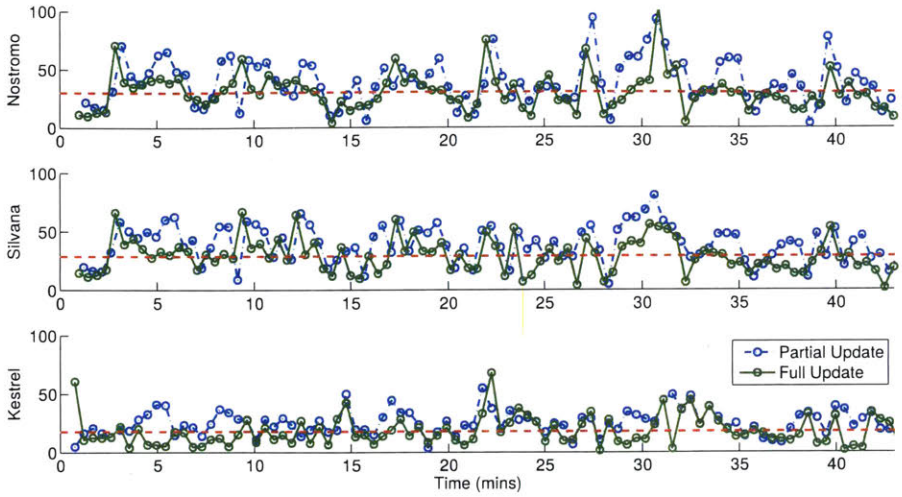

Figure 3-10: Full and Partial information Estimator tracking error for the 'Outer' formation mission by vehicle. Partial information estimates allow vehicles to take action as soon as possible, a critical benefit when TDMA schedules are long. Mean tracking error for full information estimates are shown in red.

Each vehicle leverages partial information updates in order to decrease its effective cycle time. With the decentralized controller, vehicles can make estimates and take action as soon as partial information is available. Tracking error by mission time for both full and partial information updates, and for the inner and outer formations, are shown in Figs. 3-10 and 3-11. For each vehicle, the time delay between the partial update and the full update is $\{21, 14, 7\}$ seconds respectively. The mean tracking error is plotted in red. The tracking error of the partial information updates is not significantly higher than for the full information update.

Table. 3.2 presents the mean tracking error of the estimates made with partial information for each vehicle and formation type. We observe that with the inner formation, the last vehicle in the cycle does not show as significant of improvement in tracking error as for the full information update, however in the outer formation it has a comparable improvement. It can be seen from Fig. 3-7 that the vehicles did not maintain the desired outer formation type a significant amount of the time. Since the measurement function assumes a fixed formation, deviation from that desired position is incurred as error in the target's location.

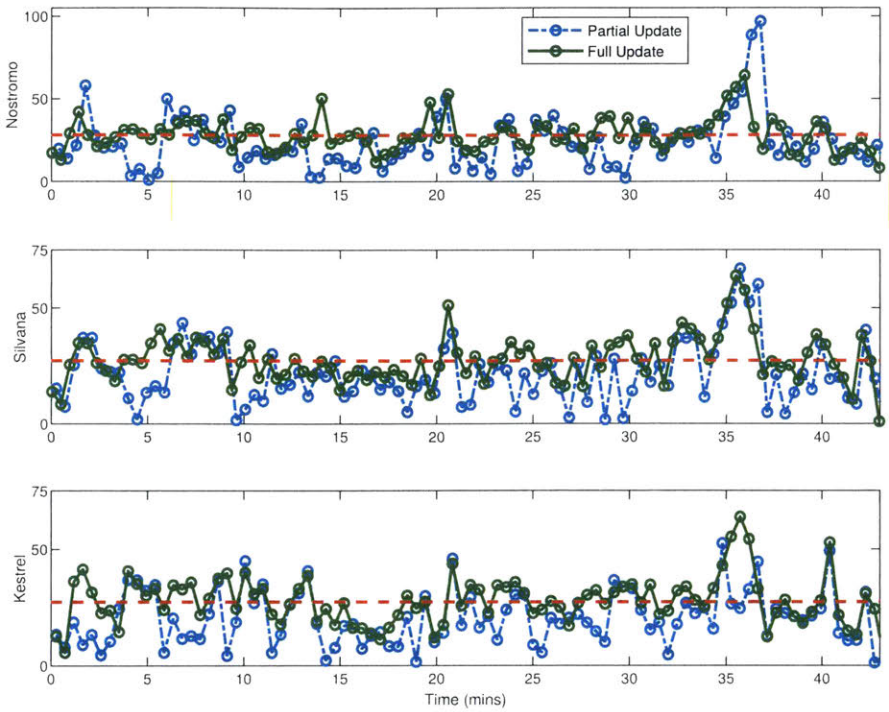

Figure 3-11: Partial and Full information Estimator tracking error for the 'Inner' formation mission by vehicle. Mean tracking error for full information estimates are shown in red.

Table 3.2: Mean Partial Information Tracking Error by Formation Type

Mission	Nostromo (m)	Silvana (m)	Kestrel (m)
Inner Partial	23.3	21.3	19.9
Outer Partial	38.6	39.0	24.9
Line Partial	64.2	65.8	62.3

All three missions were characterized by very low packet loss rates (~3%) for the WHOI Micro-Modem's 32-byte FSK0 packet. Table 3.3 shows packet losses as a percentage of total transmissions sent for the ranging message only, while packet loss rates on vehicle transmissions were similarly less than 3%.

Table 3.3: Ranging Packet Loss Percentages

Mission	Nostromo	Silvana	Kestrel
Inner	3.0	1.0	0.0
Outer	2.2	2.2	1.0
Line	2.8	0.0	0.0

In order to analyze the de-centralized controller's dynamic performance, we compute a direct FFT-based empirical transfer function from the motion of the target (rad/s) to x-y averaged 2-norm estimator error for each vehicle in Fig. 3-12. The spectra shown have been smoothed by a 7-point centered moving average. The approximate break frequency for most of the estimators is 0.047 rad/s, relatively close to half the theoretical Nyquist rate for a 27-second cycle. We observe for the 'Outer' formation that vehicle *Kestrel* has a significantly higher approximate break frequency as it is sampling one slot faster.

Similarly, the mean tracking error for three missions with a centralized controller is shown in Fig. 3-13. Two missions were conducted with an 'inner' formation and one with an 'outer' formation. Transmission loss in any slot is shown as dots by mission color, and the average packet loss rate was less than 3%. Control commands were sent acoustically using a stationary Micro-Modem deployed at the MIT Sailing Pavilion. Formation lengths, vehicle speeds and the parameters of the target model were unchanged.

The performance of the centralized controller is significantly worse, with tracking error growing without bound over time, i.e. the estimator is unstable. The extended cycle time, lack of partial updates and slower vehicle response means the vehicles cannot keep up with the same turning rate and forward speed of the target. The estimator is unable to accurately predict the target's location although we observe

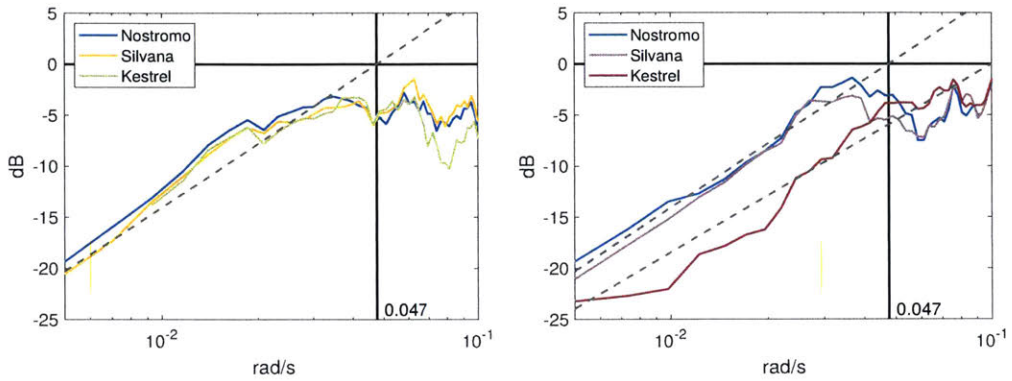

Figure 3-12: Empirical transfer function from target motion to full information estimator error. Functions for the decentralized controller are shown by vehicle for the 'Outer' and 'Inner' mission formations. Spectra have been smoothed by 7-point centered moving average. An axis is shown at the approximate break frequency of 0.047 rad/s.

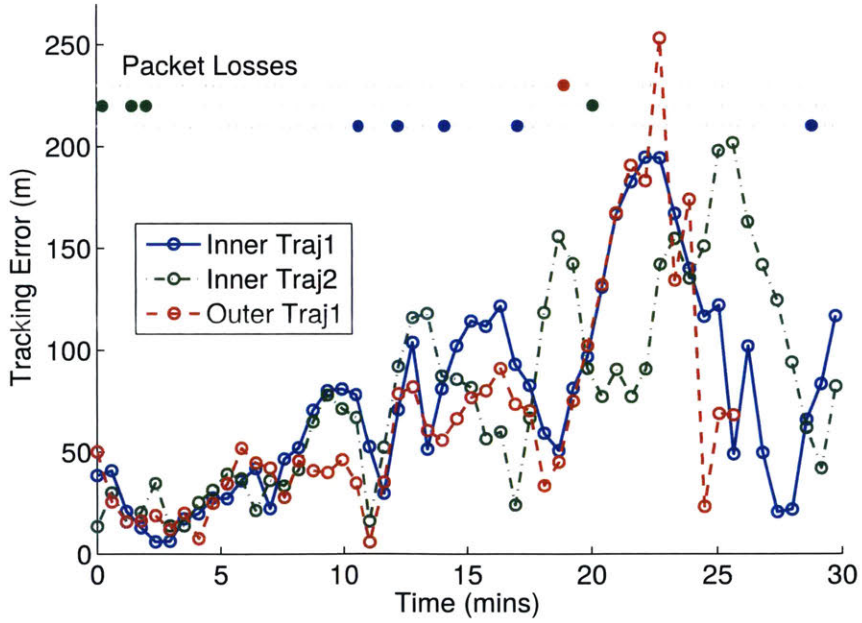

Figure 3-13: Mean X-Y tracking error for centralized controller by mission type. Transmission losses are shown as dots by mission color at top.

from Fig. 3-8 that formations were maintained. The increase in schedule timing by one slot has resulted a severe trade-off in the dynamic capability of the estimator.

Here we comment that it is also possible to have a centralized architecture with the same overall TDMA schedule time. Let one of the vehicles be designated as the leader. The leader performs the estimation and control update locally once it has received two measurements in addition to its own, and uses its TDMA slot for the control command. In the case of very low packet losses, this architecture is essentially comparable with the de-centralized approach, except that no vehicles are able to make partial information updates.

3.5 Conclusion

In this chapter a de-centralized sigma-point estimator was formulated for multi-vehicle range-only underwater target-tracking and pursuit. We showed that this estimation and control framework is capable of tracking fast-moving highly dynamic targets in the presence of TDMA cycle times around 30s, packet loss, and formation errors. Experimental results were presented for three ASVs in the Charles River ranging and communicating on WHOI Micro-Modems and a pre-defined TDMA schedule.

In particular, partial information updates and earlier full information updates were leveraged to decrease the effective cycle time, allowing vehicles to start traveling sooner so as to maintain their fixed formation around the target. A similar centralized controller architecture was presented with the addition of a single TDMA slot, and shown to be unable to track a target with the same maneuvering characteristics as the estimation system went unstable. In particular, these field experiments demonstrate that highly dynamic co-ordinated decentralized control in the underwater environment is viable using lossy acoustic communications.

In a filtering type approach, the TDMA cycle time determines the upper bound on dynamic performance of the joint system. This result, while reasonable, is unsatisfying, as the approach does not scale well to larger fleets if a TDMA approach is used for networking. Possible solutions include multiplexing of the channel, or moving to

the use of even smaller, faster information packets, although those methods represent their own trade-offs in terms of bitrate and packet loss.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 4

Synthetic Aperture Sonar with Non-Gaussian Factor Graph SLAM

Underwater acoustic beamforming typically uses phased arrays which consist of a fixed set of physically connected transducers that receive energy at varying times, depending on the geometry and travel speed of transmission. Increasing the physical aperture of the array can provide finer spatial resolution but is often constrained by the platform the array is mounted on. Synthetic aperture techniques operate by considering the array output over a segment of the platform's trajectory, effectively creating a larger virtual array with increased aperture and thus resolution. In addition, distributing the beamforming task over a network of platforms (vehicles) rather than one large expensive sensor increases resilience and exploration rate, necessitating an accurate relative localization, mapping, and planning solution.

Conventional synthetic beamforming operations resolve the location of features in view by assuming the locations of transducer poses to be fixed and true generally using some method of accurate odometry[14]. The inverse problem of resolving the transducer pose positions given a static scene assumption is more difficult. We believe that this work is the first to leverage single hydrophone acoustic waveforms in a common simultaneous localization and mapping solution, via the synthetic aperture process, to improve the accuracy of the vehicle's pose estimate without prior knowledge of scene.

4.1 Overview of Approach

4.1.1 Synthetic Aperture Sonar

Synthetic aperture radar (SAR) is a well-established technique. The underwater equivalent, synthetic aperture sonar (SAS), faces many challenges unique to the underwater and acoustic environment [135], but as we will establish, simultaneous localization and mapping (SLAM) methods can tackle these difficulties by providing high quality navigational information to an underwater vehicle using low-cost sensors. Conventionally, SAS is a method of processing sonar data in order to produce images of significantly higher azimuthal resolution than the physical array would normally permit. If a single receiver (hydrophone) or array travels in a straight line at a known and constant speed, the sampled acoustic data at each position in time can be seen as having being collected from sequential elements of a much larger, synthetic array. If the receiver's position and velocity are known, the sampled signals can be phase-shifted into coherence, forming an image from the synthetic array.

The accuracy of SAS systems relies heavily on the positioning accuracy of the platform, typically slow-moving, very stable or ship-mounted for high-resolution imaging or object detection purposes [72]. AUVs can receive GPS localization when surfaced, but must rely on other methods while underwater. Unaided inertial navigation is highly susceptible to drift, and acoustic positioning systems such as ultra-short baseline (USBL) or long baseline (LBL) are typically expensive and/or require ship support. Doppler velocity logs (DVL) are accurate but costly, and might require specific flying altitudes. Given the increasing affordability of small light-weight underwater vehicles [121], we formulate the problem using low-cost hydrophone sensors.

4.1.2 Problem Statement

The main algorithm contribution of this Chapter is the incorporation of Synthetic Aperture Sonar (SAS) via conventional beamforming (CBF) into a common factor graph inference operation, with two immediate consequences: *(i)* recorded acoustic

waveforms made by the vehicle in the field are used to simultaneously estimate the position of the acoustic source and each of the receiver positions associated with the synthetic array, and *(ii)* data-fusion inference with other data sources is performed by non-Gaussian smoothing using a factor graph and Bayes tree formulation. The construction of a SAS-factor that takes recorded acoustic waveforms and generates probability distributions over range and look-angle to an unknown beacon are detailed in Section 4.2, and its use in a non-Gaussian factor graph formulation of the underwater target-tracking problem is covered in Section 4.4. Finally, the optimization algorithm used to perform non-parametric inference over the factor graph with multiple SAS-factors is detailed in Section 4.3.

4.2 Acoustic Processing

This section details how we construct from a dataset of recorded acoustic waveforms f_i, over $i \in 1...m$ synthetic array elements distributed in space and time, approximate non-parametric probability distributions over the positions of the acoustic source and/or the array receivers. The following section will cover integrating the SAS-factor in a general factor graph framework and solving the factor graph.

We first describe the data processing pipeline that generates discrete probability distributions over range and look-angle to the beacon by matched filtering and conventional beamforming. Noisy acoustic data is filtered and re-sampled before beamforming to improve the accuracy of the beamformer. All data presented in plots is experimental (see Section. 5.1 for a description of the field trials), unless specifically noted as synthetic. Subsequently we detail the process of 'forward' convolution from the array to the beacon (conventional beamforming), and 'backward' convolution from the beacon to the array (leave one out cross-correlation).

4.2.1 Matched Filter

The raw and noisy acoustic data is filtered to maximize the accuracy of the beamformer, and to provide a TDOA measurement. If the waveform transmitted by the

source is known, then the Matched Filter (MF) is an optimal LTI filter that maximizes the SNR of a known signal in noise-corrupted data [175]. As an example, the mission of tracking a marine animal with an attached pinger (sharks) or via its calls and clicks (whales and dolphins). The assumption of a known transmission is not overly constraining as a signal can be also be identified from recorded data [122].

Let $s(t) = a(t)\cos\omega_o t + \mu(t)$ be a known signal where $a(t)$ is the amplitude modulation, $\mu(t)$ is the angle modulation and ω_o is the carrier angular frequency (presented experimentally in Fig. 5-5). The filter that is matched to $s(t)$ is defined by the impulse response:

$$h(\tau) = ks(\Delta - \tau) \tag{4.1}$$

where k and Δ are arbitrary constants. A measurement $x(t) = s(t) + n(t)$ is made of the signal, where $n(t)$ is additive, stationary white noise. We note the assumption of white noise does not accurately reflect the characteristics of ambient acoustic noise in the ocean which has a significant low frequency component and decays at approximately -18dB/kHz (see Fig. 4-1). In results from our field trials on the Charles River, the measured signal is pre-whitened with a Phase Transform (PHAT) as an accurate model for the ambient noise in the experimental area is assumed not to be known in advance, and surface conditions on the Charles can change rapidly. However, this method does not preclude cascading a different pre-whitening filter with the matched filter if the characteristics of the noise is known in advance, or applying a model-based matched filter as presented in [76].

We derive the optimality of the matched filter as follows, let signal $s(t)$ be of finite duration such that its total energy E is not infinite. Define its spectrum, or Fourier Transform, and the inverse Fourier Transform to be:

$$S(j2\pi f) = \int_{-\infty}^{\infty} s(t)e^{-j2\pi ft}dt \tag{4.2}$$

$$s(t) = \int_{-\infty}^{\infty} S(j2\pi f)e^{j2\pi ft}df \tag{4.3}$$

Let $\omega = 2\pi f$. The signal $s(t)$ and noise $n(t)$ components of the filtered output

can be computed as a convolution of the matched filter with the measurement:

$$y(t) = x(t) * h(t) = (s(t) + n(t)) * h(t) \tag{4.4}$$

$$= s(t) * h(t) + n(t) * h(t) \tag{4.5}$$

$$:= y_s(t) + y_n(t) \tag{4.6}$$

$$= \int_{-\infty}^{\infty} s(\tau)h(t-\tau)d\tau + \int_{-\infty}^{\infty} n(\tau)h(t-\tau)d\tau \tag{4.7}$$

Since convolution is multiplication in the frequency domain, we can write the output signal in terms of its inverse Fourier Transform:

$$Y_s(j\omega) = S(j\omega)H(j\omega) \tag{4.8}$$

$$\therefore y_s(t) = \int_{-\infty}^{\infty} Y_s(j\omega)e^{j\omega t}d\omega \tag{4.9}$$

$$= \int_{-\infty}^{\infty} S(j\omega)H(j\omega)e^{j\omega t}d\omega \tag{4.10}$$

Similarly, for the noise component,

$$Y_n(j\omega) = N(j\omega)H(j\omega) \tag{4.11}$$

$$\therefore y_n(t) = \int_{-\infty}^{\infty} Y_n(j\omega)e^{j\omega t}d\omega \tag{4.12}$$

$$= \int_{-\infty}^{\infty} N(j\omega)H(j\omega)e^{j\omega t}d\omega \tag{4.13}$$

The average SNR of the filtered output at time $t > 0$ is written:

$$\rho(t) := \frac{\langle y_s^2(t)\rangle}{\langle y_n^2(t)\rangle} \tag{4.14}$$

$$= \frac{1}{2\pi}\frac{\left[\int_{-\infty}^{\infty} S(j\omega)H(j\omega)e^{j\omega t}d\omega\right]^2}{\left[\int_{-\infty}^{\infty} N(j\omega)H(j\omega)e^{j\omega t}d\omega\right]^2} \tag{4.15}$$

To evaluate this expression we observe that for the denominator, if the noise is stationary and white as we previously assumed, the power density of the signal $n(t)$ can be written $N_o/2\,|H(j\omega)|^2$, where $N_o/2$ is a constant. For the numerator, we

observe that the integral over our given signal $s(t)$ will be real and apply the following Cauchy-Schwarz inequality[1]

$$\left| \int f(x)g(x)dx \right|^2 \leq \int |f(x)|^2 dx \int |g(x)|^2 dx \tag{4.16}$$

where,

$$f(x) \sim S(j\omega)e^{j\omega t} \tag{4.17}$$

$$g(x) \sim H(j\omega) \tag{4.18}$$

$$\therefore \left[\int_{-\infty}^{\infty} S(j\omega)H(j\omega)e^{j\omega t}d\omega \right]^2 \leq \int_{-\infty}^{\infty} |S(j\omega)e^{j\omega t}|^2 d\omega \int_{-\infty}^{\infty} |H(j\omega)|^2 d\omega \tag{4.19}$$

Finally, the power attributed to the filter $\int_{-\infty}^{\infty} |H(j\omega)|^2 d\omega$ drops out, leaving:

$$\rho \leq \frac{2}{N_o} \int_{-\infty}^{\infty} |S(j\omega)e^{j\omega t}|^2 d\omega \tag{4.20}$$

$$\rho \leq \frac{2E}{N_o} \tag{4.21}$$

where E is the total energy in the signal. By inspection, it is clear that the SNR of the filtered signal (Eqn. 4.21) achieves its highest bound or maximum when the equality holds in Eqn. 4.19. It follows that this equality holds when the filter $h(t)$ is matched to the signal $s(t)$ such that $H(j\omega) = k\bar{S}(j\omega)e^{-j\omega t}$. That is, the transfer function of the matched filter is the complex conjugate of the Fourier Transform of the signal, with an amplitude k and delay factor $e^{-j\omega\Delta}$. This result is intuitively appealing. The amplitude of the matched filter at any frequency is the amplitude of the signal. This implies the filter minimizes the amplitude of the noise at frequencies where there is little signal. The phase of the matched filter is the conjugate of the signal, or reversed in time. Thereby, when the matched filter is convolved with the measurement, this can also be seen as a cross-correlation of the measurement $x(t)$ and the known signal $s(t)$ that produces an impulse response (not an exact impulse

[1]The Cauchy Schwarz Inequality states that the sum of the lengths of two vectors must be greater than or equal to the length of the sum of both vectors. This can be stated more generally in inner product space as: $|\langle u, v\rangle|^2 \leq |\langle u, u\rangle|^2 + |\langle v, v\rangle|^2$

response as it has non-zero width) at the Δ time in the measurement where the known signal occurs.

For the purposes of the algorithm, the signal and measurement are represented by discrete samples. The Fourier transform is computed as a Discrete Fourier Transform (DFT), and the linear matched filter is represented not as a transfer function but as a vector (and in this specific implementation, a matrix of vectors as wide (repeated as many times) as the number of receivers in the synthetic array). Additionally, we incorporate the PHAT as a pre-whitening weighting that discards all amplitude information from the spectrum since the transmitted chirp does not have amplitude modulation and the acoustic noise is decidedly non-white [166]. The notation $()_i$ denotes elementwise operations, i.e. $(*)_i(*)_i$ is an elementwise multiplication.

$$\hat{X(i\omega)} = \frac{X(i\omega)}{|X(i\omega)|} \tag{4.22}$$

Here it is implemented by dividing the frequency components of the DFT of the measurement by their magnitudes, with a very small offset to account for floating-point numbers very close to zero.

We evaluate the performance of the matched-filter by generating empirical SNR estimates from experimental data. For the following results, the length of the chirp signal was $m = 1024$ samples and the length of the measurement was 8000 samples, sampled at a rate of 37500Hz. The length of the zero-padded DFT was $n = 8192$. The range of the chirp signal was 200-2050Hz. More details on the experimental conditions are provided in Section 5.1.

Fig. 4-1 presents Power Spectral Density (PSD) plots computed using Welch's method of overlapping averaged segments on experimental data. Each segment of length 227 samples is windowed with a Hamming window. The top plot is a representative sample of background noise without any transmitted signal, color-coded by element of the pyramidal array. The sources of low frequency noise are expected to be wind and wave action, as well as some propeller noise while the ASV is in transit. High frequency noise (>10kHz) in clear repeating bands was observed and attributed

Algorithm 1: Matched Filter

Data: $s(t)$, the chirp signal, $x_i(t)$, the measurement signal for $i \in 1...m$ array elements, n, the number of samples, m, the number of array elements

Result: $y_i(t)$, the match filtered signal for $i \in 1...m$ array elements, $\mathbf{H}(\omega)$, a matched filter operating on m measurements of n length

Function `prepMF(`s, n, m`):`

begin

$S = DFT(s)$ `// zero-padded FFT where` $n \geq$ `len(`s`)`

$\mathbf{H}[1 : \text{len}(s), 1...m] = \bar{S}$

return H

end

Function `MatchFilter(`x_i, **H**`):`

begin

$\mathbf{Y}[:, 1...m] = DFT(x_i) \quad \forall i \in 1...m$

$\mathbf{Y} = \dfrac{\mathbf{Y}}{|\mathbf{Y}| + \text{1e-9}}$ `// PHAT transform`

$y = DIFT((Y)_{ij}(H)_{ij})$ **return** $y[:, 1...m]$

end

return

to fluctuations on the power electronics while the motor was on and the ASV was in transit. Here we note that while the Micro-Modem was isolated from the motor's power circuit inside the main electronics box, the separate payload computing running the pyramidal array was unfortunately not.

A hundred samples of background noise (with the ASV in and out of transit) were averaged and the resulting PSD is plotted in the bottom image. The regularly spaced peaks of electronic noise become pronounced, and can also be viewed as bright bands in the spectrograms presented in Fig.5-5. The PSD for a single experimental waveform is shown in blue. The matched filter's output is shown in red. Since the chirp is limited to low frequency bands, the matched filter effectively eliminates the electronic noise. The component of low frequency noise specifically in the <1kHz range is the most significant. One possible way to address this issue is to reduce electronic noise on the power circuits in hardware, and then transmit a chirp in a higher frequency band.

We compute empirical SNR values for each trial presented in our experimental

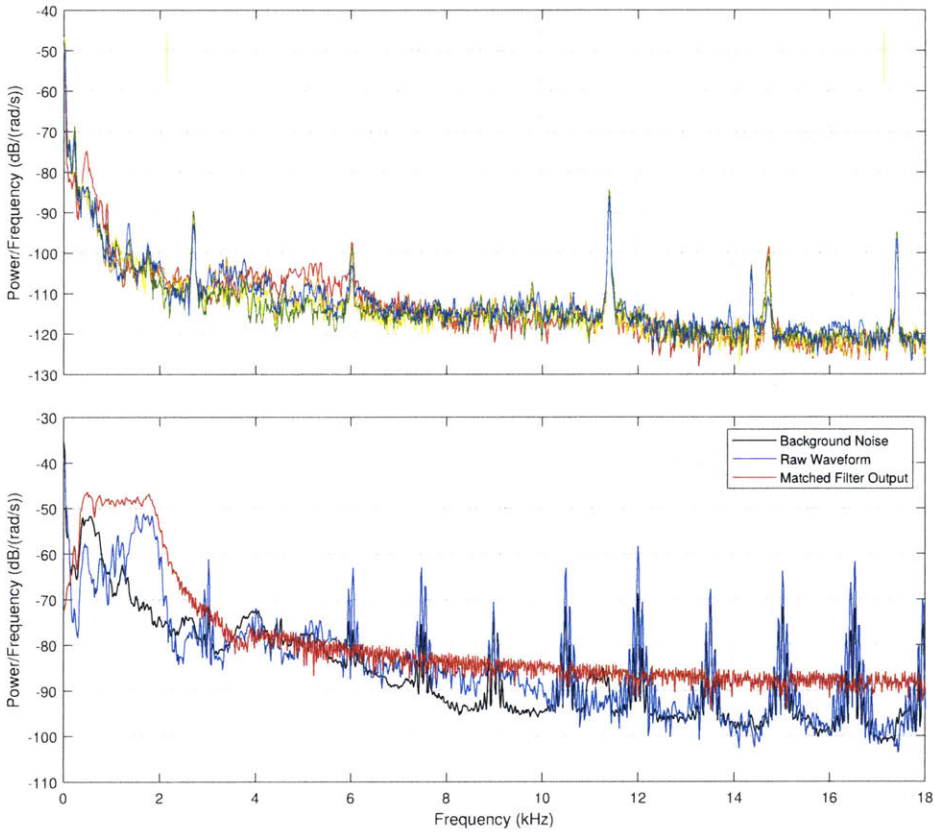

Figure 4-1: Power spectral density (PSD) plots for acoustic waveforms received at pyramidal array. (*Top*:) Single sample of background noise color-coded by array element. (*Bottom*:) PSD of raw waveform from a single array element shown in blue and its matched filter output shown in red. Background noise averaged over 100 samples is shown in black. Beacon chirp is in the range 200-2050Hz. Computed SNR for this datapoint is 10dB.

results. Section‘5.1 provides the details of each trial. Fig. 4-2 shows histogram distributions for a stationary beacon and a drifting beacon. The total number of samples per mission is approximately 300-500. This varies based on the length of the mission, while the beacon transmits a chirp every 1 second. The average SNR of the signal is approximately 10dB for these trials. Fig. 4-3 shows histogram distributions for a Beacon towed by a motorboat. These experimental sets report a wider spread of SNR and on average lower SNR, most likely due to the presence of noise from the motorboat's engine.

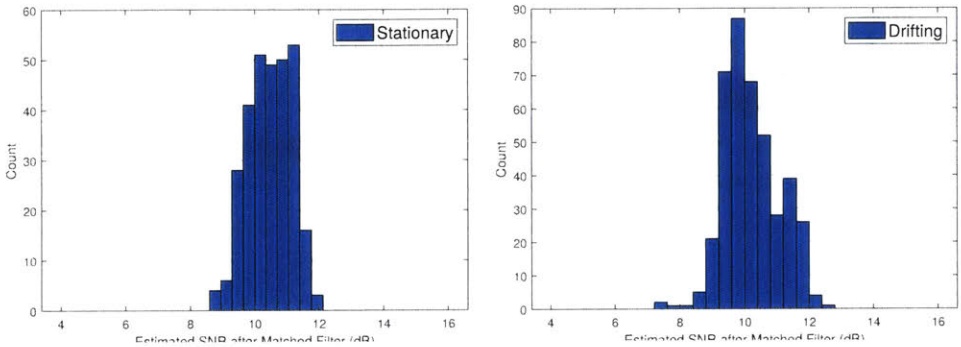

Figure 4-2: Histograms of empirically estimated SNR after matched filtering, with uniform bin widths. (*Left:*) Stationary Beacon, 300 waveforms at a chirp-transmit rate of 1Hz. (*Right:*) Drifting Beacon, 400 samples over mission length.

The output of the matched filter can now be used to obtain an empirical discrete distribution over the Time-of-Flight (TOF) of the waveform and thus, with a sound speed profile, the range from the receiver to the beacon. For the purpose of our experiments with a fixed array, the speed of sound in water is assumed to be $c = 1481\text{m/s}$.

$$R_i = \frac{cy_i}{F_s} \tag{4.23}$$

The theoretical output of the matched filter on a signal corrupted by stationary white noise is an impulse response centered at time delay Δ at which the signal was detected. In practice, this response is a distribution over time delays. A Gaussian fit can be used to implement a range factor or constraint in a typical SLAM framework. The non-Gaussian SLAM framework works with samples drawn from arbitrary prob-

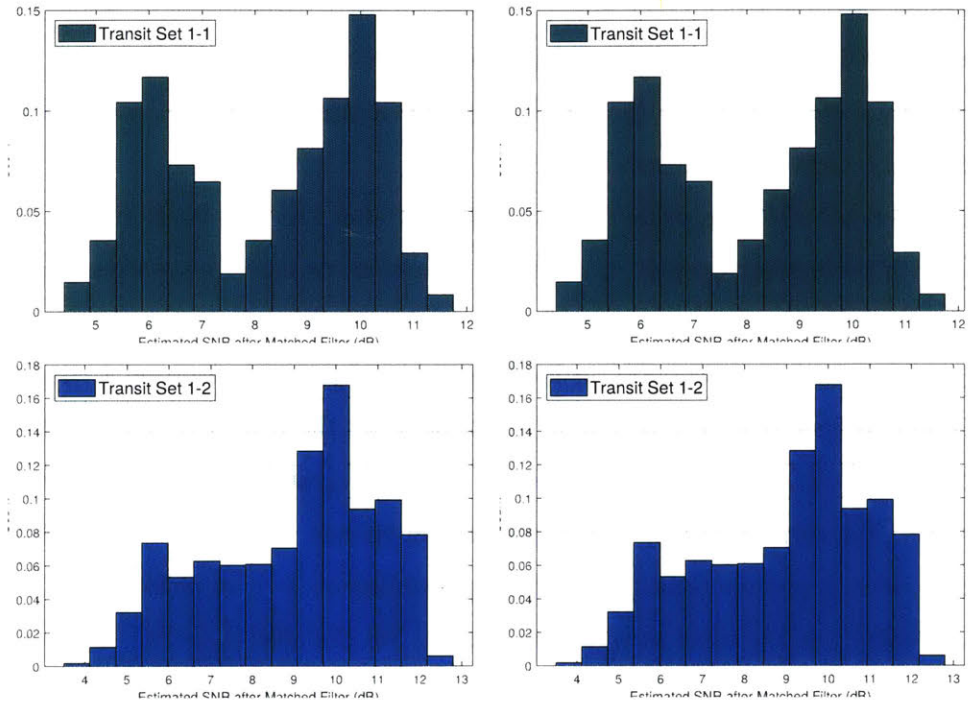

Figure 4-3: Histograms of empirically estimated SNR after matched filtering, with uniform bin widths and normalized by total number of samples for cross-set comparison. (*Left:*) Transit Beacon Set 1, 2100 samples in two trials. (*Right:*) Transit Beacon Set 2, 700 samples in two trials. These trials are characterized by a lower average and wider spread of SNR

ability distributions, thus we need a method to both represent and sample arbitrary discrete probability distributions.

4.2.2 Alias Scalar Sampler

The general alias method is an algorithm that, provided with a set of discrete random values and their respective probabilities, returns samples from that arbitrary probability distribution in $O(1)$ time [186]. An intuitive explanation of the method is, more probable random values are generated aliases with less probable values until the uniform distribution over the set of original samples and aliases corresponds to the desired arbitrary distribution.

Consider an arbitrary probability distribution over the set M of discrete outcomes with probability weights p_i, $i \in 1...m$. The alias method constructs a uniform distribution over the same number of events m, i.e each event is equally probable with likelihood $1/m$, represented in implementation by a table indexed i. Store in each entry of the table a probability of selecting either the corresponding value from the original set M_i of discrete outcomes, or an alias A_i, also from the same set but $A_i \neq M_i$. The alias is chosen with the following ruleset: if the probability of that random value is greater than the uniform probability $p_i > 1/m$, arbitrarily choose another entry j of the table such that $p_j < 1/m$. Set the alias $A_j = M_i$. Update the stored value of $p_i = p_i + k_i - 1/m$. Repeat until all stored probabilities are nominally equal to 1.

Alias Sampler:

1. Generate a probability table $\mathbf{P}$ and an alias table $\mathbf{A}$ of size m, instantiating $P_i = np_i$.
2. Store a list $\mathbf{L}$ of unaliased indices $i = 1 \cdots n$
3. While the length of L is greater than 1
   (a) Find $i \in \mathbf{L}$ such that $P_i > 1$, arbitrarily choose entry $j \neq i$ such that $P_j < 1$.

(b) Set the alias $A_j = i$.

(c) Set the probability $P_i = P_i + P_j - 1$

(d) Remove the aliased index j from the list

4. Set the probability of the last remaining index to 1

Samples can then be drawn using set of aliases as follows: Roll a fair m-sided die to determine the index i. Then, flip a biased coin, selecting M_i with probability P_i and the aliased value A_i otherwise. A new sampler is pre-computed for every distinct probability distribution in $O(n \log n)$ time, in this case, every recorded waveform processed with the matched filter. The construction of the probability weights is presented below, and an aliasing sampler algorithm as described above is provided by the Julia package *Statsbase* under the function *alias sample*.

In order to build an empirical distribution from an arbitrary signal, we first remove elements in the lowest specified quantile (typically 60%-80%), so as to mitigate the effects of noise in the estimate. The remaining elements are then adjusted so that the resulting vector is non-negative and sums to unity. Algorithm 2 details the process of generating discrete probability weights.

Algorithm 2: Compute Alias Sampler Discrete Probabilities

Data: $\bar{y}(t)$, an input signal to be cast as a sampled distribution, p_o, probability lower bound

Result: $\bar{p}$, probability weights for each discrete sample

Function `compProbabilityWeights`(y_i, c, F_s, p_o)**:**

begin

$\bar{y}\,[\bar{y} < 0] = 0$

$\bar{p} = \frac{\bar{y}}{\sum_{i=1}^{n} y_i}$

$\bar{p} \leftarrow \bar{p} - \bar{p}\,[j \quad \forall\, p_j \leq p_o]$

$\bar{p}\,[\bar{p} < 0] = 0$

$\bar{p} \leftarrow \frac{\bar{p}}{\sum_{i=1}^{n} p_i}$

return $\bar{p}$

end

return

Alias sampling is constructed from the output of the matched filter (Section 4.2.1)

to represent the probability distribution over range, and from the output of the beamformer (Section 4.2.4) to represent the distribution over azimuth. For each discrete sample we need to compute the probability that the signal $s(t)$ was detected with some time delay δt in the measurement $x(t)$. Following from [195], the probability density of a sample $n(t)$ of white, Gaussian noise can be written:

$$p(n) = k \exp\left[-\frac{1}{N_o}\int_I n^2(t)dt\right] \tag{4.24}$$

where $N_o/2$ is the double-ended spectral density of the noise, i.e. $\frac{\bar{n}^2(t)}{2F_s} = \frac{N_o}{2}$, k is a constant and I is the observation integral.

Then, the likelihood that the observed waveform sample $x(t)$ represents the signal is the likelihood that the noise assumes the form $n(t) = x(t) - s(t-\Delta)$ [175].

$$p(\Delta) = k \exp\left[-\frac{1}{N_o}\int_I \left(x(t) - s(t-\Delta)\right)^2 dt\right] \tag{4.25}$$

$$= k \exp\left[-\frac{1}{N_o}\int_I x^2(t)dt + \right. \tag{4.26}$$

$$\left. \frac{2}{N_o}\int_I s(t-\Delta)x(t)dt - \frac{1}{N_o}\int_I s^2(t-\Delta)\right]$$

The first term is the energy of the measurement, the second term we recognize as the output of the matched filter, and the last term is the energy of the chirp. For discrete samples, the integral becomes a sum over the observation length.

Figs. 4-4 and 4-6 present the output of the matched filter by range to the beacon in meters (Eqn. 4.23. Chirps have been vertically grouped as a sequence of 8, where 8 elements is the default length of our synthetic array. Each subgroup is sequential in time at a rate of 1Hz, and vertical axes have been scaled within each chirp sub-group to reflect the maximum number of samples. We observe in some of the returns poor performance in the form of a relatively wide spread of ranges, ex. Chirp 310 and 311. Additionally, the first chirp group 70-78 in the Stationary Beacon experiment display a consistently strong second peak, which is most likely an echo from the hard seawall several meters behind the dock. Although the experimental data is characterized

with reasonable SNRs, we find that a procedure of consistent, robust estimation is still necessary to locate the source accurately.

Figs. 4-5 and 4-7 present the set of histograms generated by alias sampling 1000 samples from each chirp's matched-filtered output as a distribution, at a lower bound quantile of 0.6 (i.e. all samples below cumulative probability density of 40% are discarded). These can be interpreted as a visual representation of the discrete probability distributions over range sampled by the non-Gaussian SLAM solver, which we argue the alias sampler is an effective method of expressing.

4.2.3 Chirp Z-Transform

Before beamforming, a Chirp Z-Transform (CZT) is applied to convert the recorded waveforms to the frequency domain. We choose to use the CZT instead of a conventional DFT as this procedure samples the recorded waveform within a given set of bandwidth frequencies, i.e. that of the known beacon signal, and at a smaller bin size. This step reduces computation required for the beamforming step as it disregards irrelevant frequencies, and improves the resolution and accuracy around the desired output signal. The algorithm for the CZT leverages the speed of an Fast Fourier Transform (FFT) operation to efficiently evaluate the z-transforms of a discrete sequence of samples at arbitrary complex intervals [136]. Note that this 'chirp', referring to the waveform with linearly increasing frequency in time is a discrete sequence used by the algorithm to efficiently evaluate convolution with arbitrary complex numbers and is not directly related to the chirp sent by the acoustic source, although it is informed by the frequency range in which the beacon chirp varies.

The purpose of the CZT algorithm is to evaluate the z-transform of a finite number N of discrete samples $\bar{x} = [x_1 \cdots x_n \cdots x_N]$ at the sequence of complex points $\bar{z} = [z_1 \cdots z_k \cdots z_M]$, where $\bar{x}$ in this application is the output of the matched filter.

$$\bar{X} = \sum_{n=0}^{N-1} x_n z_k^{-n} \qquad k = 0 \cdots M-1 \tag{4.27}$$

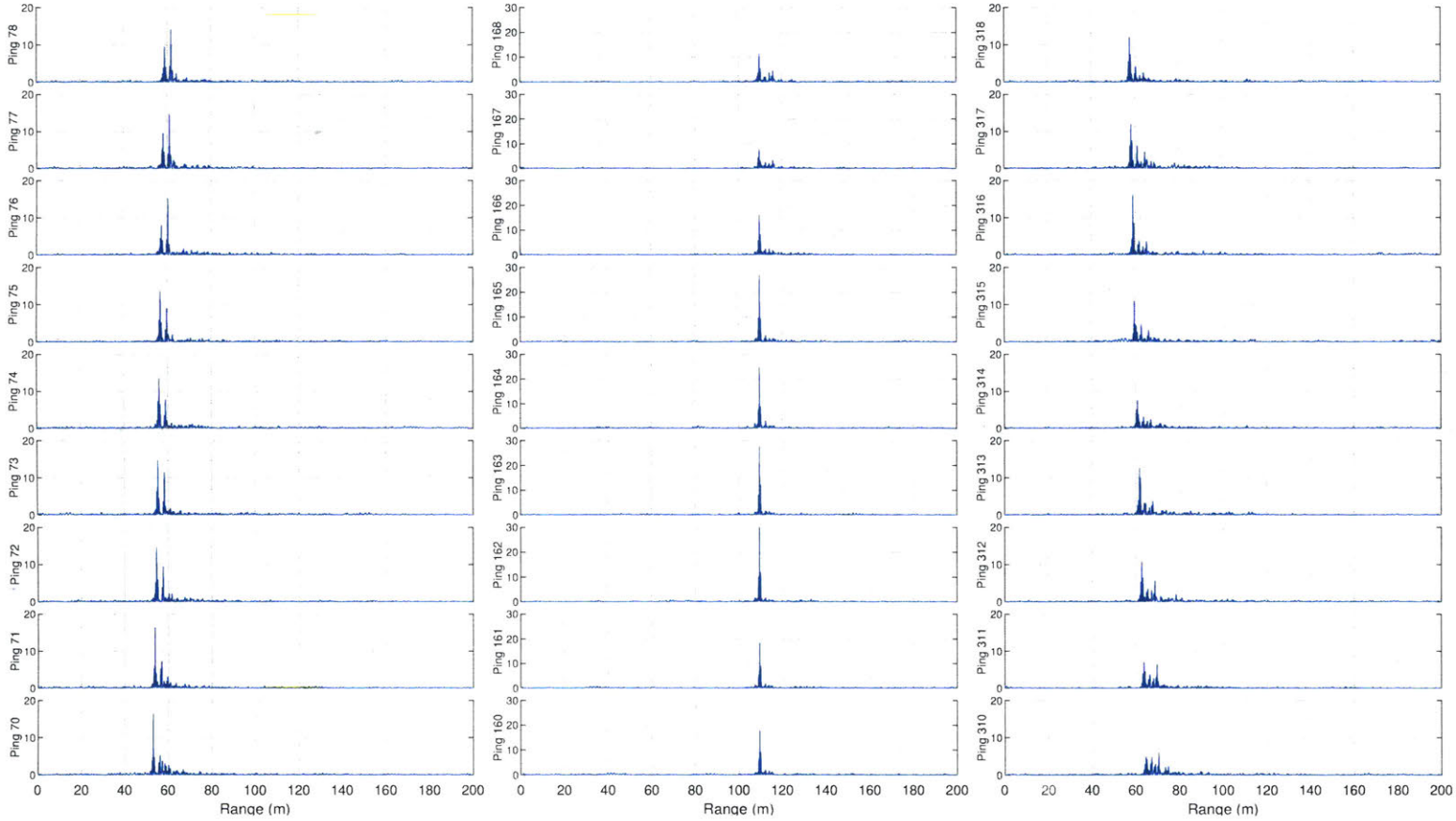

Figure 4-4: Stationary Beacon, Matched-filtered output by chirp group. Power distribution over range can be modeled as a discrete probability distribution by the alias sampler. Vertical axes scaled per chirp group.

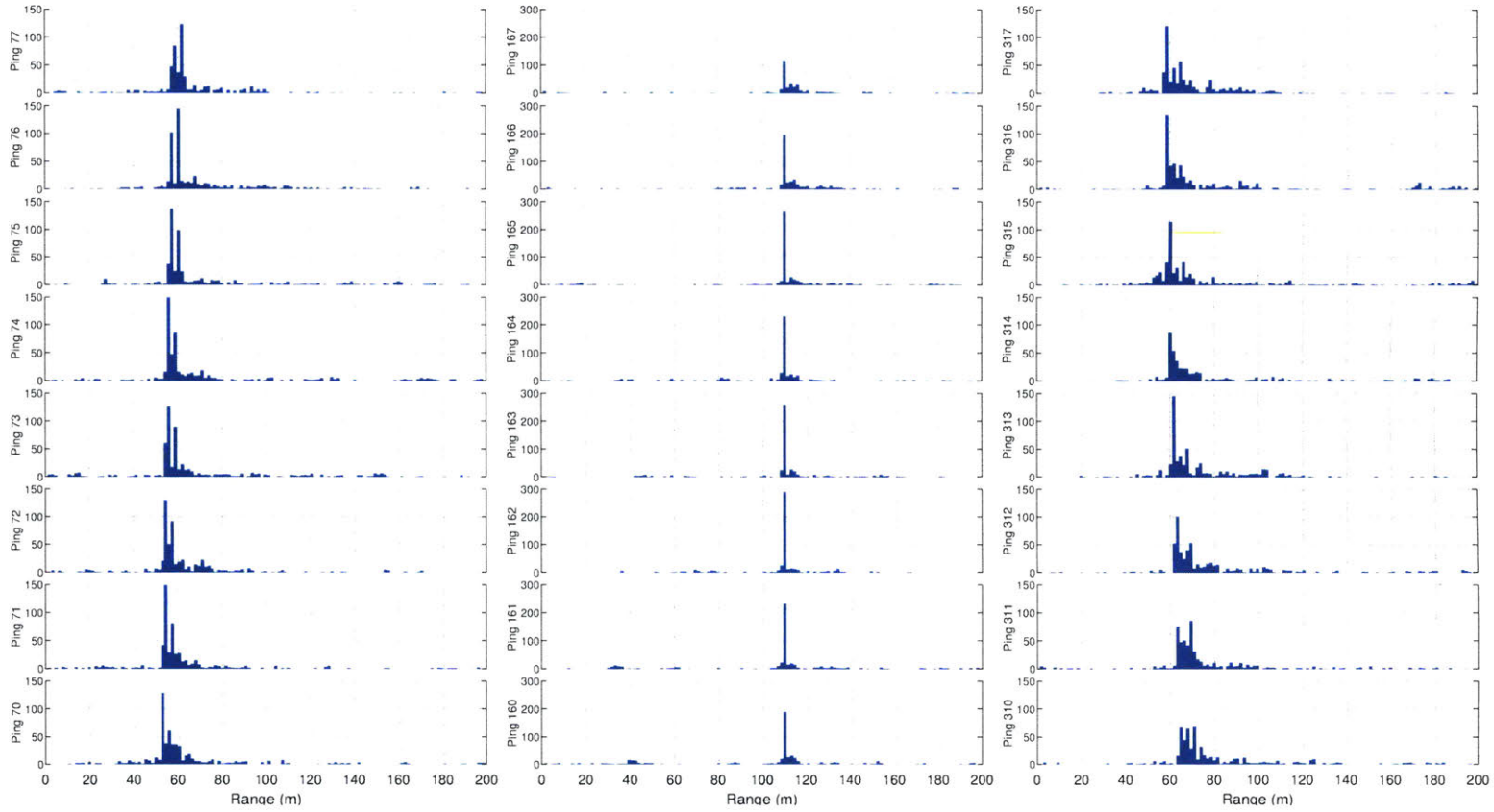

Figure 4-5: Stationary Beacon, Histograms of alias sampler output by chirp group, totaling 1000 samples. Vertical axes scaled per chirp group.

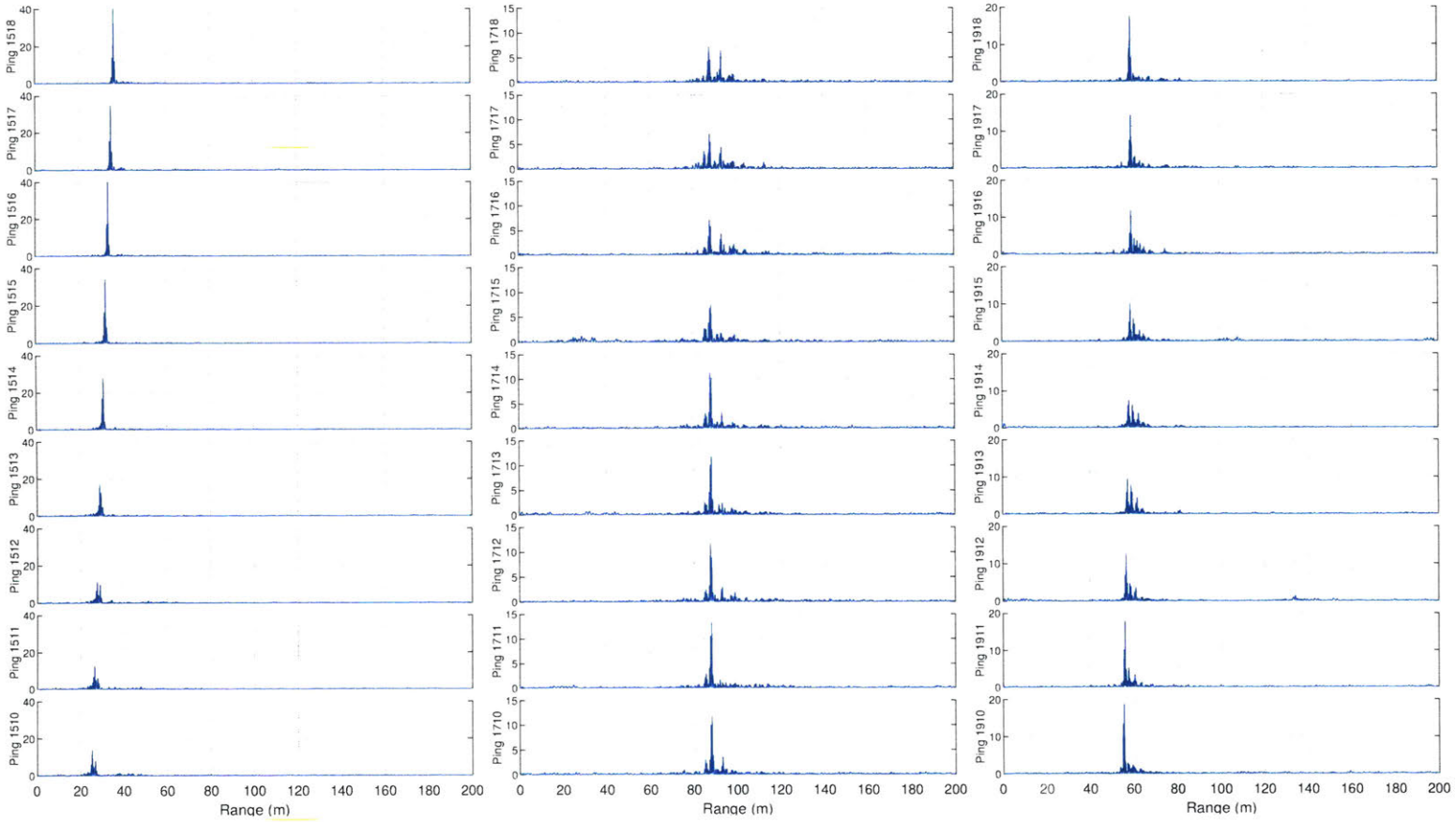

Figure 4-6: Drifting Beacon, Matched-filtered output by chirp group. Power distribution over range can be modeled as a discrete probability distribution by the alias sampler. Vertical axes scaled per chirp group.

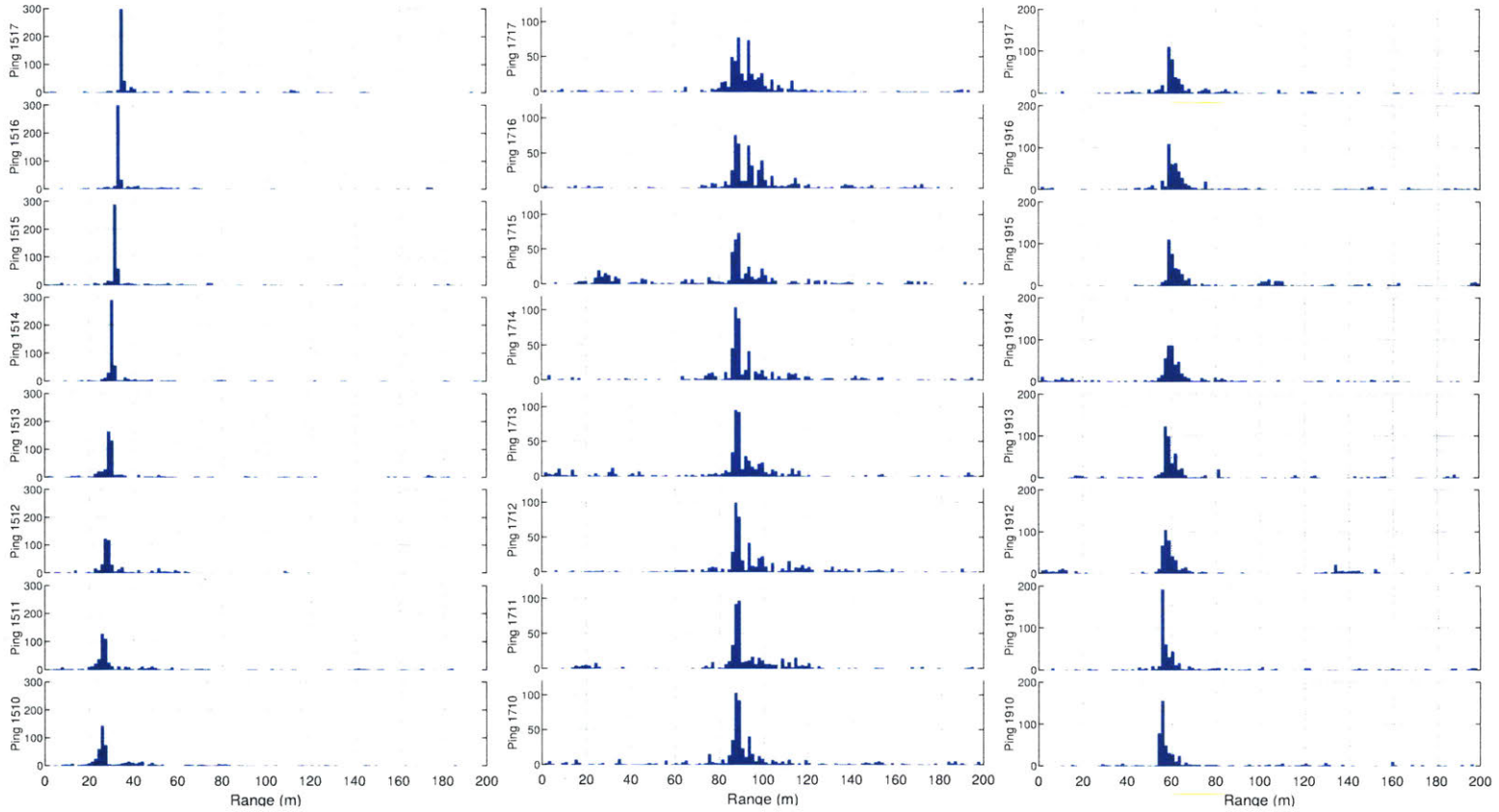

Figure 4-7: Drifting Beacon, Histograms of alias sampler output by chirp group, totaling 1000 samples. Vertical axes scaled per chirp group.

If points $z_k = e^{j2\pi k/N} \quad k = 0, 1, ...N-1$ are equally spaced around the unit circle, then this transform is the DFT. Instead let z_k be a general contour described by points $z_k = a\bar{w}^{-k}$, where a is the origin of the contour and $\bar{w}_k$. If the origin and the contour are arbitrary complex numbers:

$$a = a_o e^{j2\pi\theta_o} \tag{4.28}$$

$$\bar{w}_k = w_o e^{j2\pi\phi_o} \tag{4.29}$$

Then the contour in the s-plane is described by a straight line-segment whose length and sampling density can be arbitrarily defined.

$$s_k = s_o + k(\Delta\sigma + j\Delta\omega) = \frac{1}{T}\left(\ln(a) - k\ln(\bar{w}_k)\right) \tag{4.30}$$

$$k = 0 \cdots M-1 \tag{4.31}$$

The CZT is the z-transform of the signal at these points is:

$$\bar{X} = \sum_{n=0}^{N-1} x_n a^{-n} \bar{w}_k^{nk} \qquad k = 0 \cdots M-1 \tag{4.32}$$

Applying the substitution from Bluestein's algorithm [18]:

$$nk = \frac{1}{2}(n^2 + k^2 - (k-n)^2) \tag{4.33}$$

$$\bar{X} = \sum_{n=0}^{N-1} x_n a^{-n} W^{n^2/2} W^{k^2/2} W^{-(k-n)^2/2} \tag{4.34}$$

Since the summation is taken over the length N of the original signal, for each index n, the term dependent only on n, namely $x_n a^{-n} W^{n^2/2}$ can be precomputed as a weighting of the original signal. The term dependent only on k, namely $W^{k^2/2}$, can be computed after the sum as an elementwise multiplication. Thus, from the substitution we find that the remaining step of multiplication by the term $W^{-(k-n)^2/2}$ can be efficiently performed as a convolution. The Chirp Z-transform algorithm following this logic is given below.

Chirp Z-Transform:

1. Weight x_n by $a^{-n}W^{n^2/2}$

$$x_n^1 = x_n a^{-n} W^{n^2/2} \qquad n = 0 \cdots N-1 \tag{4.35}$$

2. Convolve x_n^1 with the chirp sequence $v_n = W^{-n^2/2}$.

$$x_k^1 = \sum_{n=0}^{N-1} x_n^1 v_{k-n} \qquad k = 0 \cdots M-1 \tag{4.36}$$

3. Elementwise multiply x_k^1 by $W^{k^2/2}$

$$\bar{X} = x_k^1 W^{k^2/2} \qquad k = 0 \cdots M-1 \tag{4.37}$$

Applying circular convolution over a vector of length $L \geq N + M - 1$ where L is a power of 2 means preconstructing the transform vectors $\bar{A} = a^{-n}W^{n^2/2}$, $\bar{F} = DFT(W^{-n^2/2})$ and $\bar{w}_k = W^{k^2/2}$ takes one DFT operation on a vector of length L, and computing the transform takes one DFT and one Inverse Discrete Fourier Transform (IDFT) on a vector of length L, returning a vector of length k. The algorithm is presented below, and we refer the reader to [136] for more implementation details. We evaluate the transform at evenly spaced points between lower and upper chirp frequencies f_1 and f_2.

Algorithm 3: Chirp Z-Transform

Data: n, the length of the input signal, m, the length of the transformed signal, $\bar{x}$, the input signal, f_1, f_2 chirp frequencies, F_s, sampling frequency, a, origin of the sequence

Result: $\bar{y}$, the transformed signal, $\bar{A}, \bar{F}, \bar{w}_k$, vectors comprising a czt operation on a signal of n length

Function `prepCZT(`n, m, f_1, f_2, F_s, a`)`**:**

begin

$w = e^{-j2\pi(f_2-f_1)/(nF_s)}$

$\bar{k} = [0 \cdots \max(n,m) - 1]$ `// linear range vector of length` n

$\bar{w}_k = w^{0.5(\bar{k})_i^2}$

$\bar{A} = \left(a^{-\bar{k}}\right)_i (\bar{w}_k)_i$

$\bar{F}[1:n] = (\bar{w}_k[n:-1:1])_i^{-1}$

$\bar{F}[n:n+m-1] = (\bar{w}_k[1:m])_i^{-1}$

$\bar{F} \leftarrow DFT(\bar{F})$

return $\bar{A}, \bar{F}, \bar{w}_k[1:m]$

end

Function `czt(`$\bar{x}$, $\bar{A}, \bar{F}, \bar{w}_k$`)`**:**

begin

$\bar{y} = DFT(\bar{x}. * \bar{A})$

$\bar{y} \leftarrow DIFT(\bar{y}. * \bar{F})$

$\bar{y} \leftarrow \bar{y}. * \bar{w}_k$

return $\bar{y}$

end

return

4.2.4 Conventional Beamformer

Where the matched filter is a temporal filter that allowed us to estimate the signal's TOF and thereby the range from array to beacon, beamforming is the process by which the signal's Direction-of-Arrival (DOA) is estimated, i.e. the look-angle from array to beacon. In general, a beamformer is defined as a spatial filter that linearly weights and combines spatially sampled waveforms from an array of sensors [180]. This in turn can be applied to many purposes in addition to DOA estimation, such as signal detection or noise filtering. The process of selecting weights steers the beamformer such that the gain of the output is maximized along a specified look-angle. If the DOA of the source is known, this can be used to maximize the SNR of the signal. If the DOA of the source is not known, as in our application, iterating through all possible look-angles and computing the maximum power allows us to estimate the most probable DOA of the source.

Beamformers can be classified into data-independent and adaptive methods. Data-independent methods such as the conventional delay-and-sum beamformer used in this thesis do not assume any specific noise or input signal structure, these methods are typically more robust to errors such as sensor calibrations and uncertainty in array element locations. When the weighting is chosen based on minimizing the expected noise or maximizing SNR, this is known as adaptive beamforming, for which there are many well known methods such as Maximum Likelihood (ML) methods [162], subspace methods [153]. Adaptive methods can result in higher resolutions and better interference rejection, although as the name suggests they require model assumptions that are subsequently error-sensitive.

Ultimately the SAS-factor applies the principles behind CBF to produce both an approximate discrete probability distribution over the look-angle from a synthetic array to the acoustic source, and vice-versa, a distribution over the positions of the synthetic array given a source location. This construction of both 'forward' and 'backward' paths then allows the process of simultaneous inference over the factor graph with SAS-factors.

An array of acoustic receivers can be used to estimate the angle between the array and a sound source through CBF [174], as illustrated in Fig. 4-9. Assuming the wave of sound incident onto the array is planar (a valid assumption when the source is in the far-field), we use trigonometric principles and the known array geometry to apply time-delays (or phase-shifts in the frequency domain) to the received signals for a user-specified look-angle. The far-field assumption holds when the distance from the source to the array is large relative to the length of the array, and our field trials are prepared so as to preserve this assumption, as detailed in Section 5.1. The aim of beamforming is to estimate the angle to the sound source by varying this look-angle over a set of candidate directions. The look-angle which results in maximum power (constructive interference on all received signals) then corresponds to the true bearing of the source. Define the geometry of the two-dimensional CBF:

Cartesian:	Spherical:	
$x = r\cos(\varphi)\cos(\vartheta)$	$r = \sqrt{x^2 + y^2}$	(4.38)
$y = r\cos(\varphi)\sin(\vartheta)$	$\theta = \arctan\left(\frac{y}{x}\right)$	(4.39)
$z = \sin(\varphi)$		(4.40)

where r is the range, φ is the elevation angle and ϑ is the azimuth angle to be estimated. The reference directions are ϑ the positive x-axis and positive rotation defined by the right hand rule counter-clockwise about the origin, φ measured counter-clockwise from the x-y plane. Therefore the unit vector pointing towards the beacon ϑ, φ is:

$$\vec{u}(\vartheta, \varphi) = \begin{bmatrix} \cos(\varphi)\cos(\vartheta) \\ \cos(\varphi)\sin(\vartheta) \\ \sin(\varphi) \end{bmatrix} \tag{4.41}$$

Let the velocity of propagation be c, the speed of sound in water, λ the wavelength of the incident wave and F_c, the frequency of the incident wave, then the wavenumber vector is:

$$\vec{k}(\vartheta,\varphi) = \frac{2\pi F_c}{c}\vec{u} = \frac{2\pi}{\lambda}\vec{u} \tag{4.42}$$

The array we consider for this application is a 1-dimensional linear array for the purposes of synthetic aperture sonar. This simplification is mainly for experimental concerns, where the ASV is used as a stand-in for an underwater vehicle at a fixed depth and the ASV's positions in time function as array elements. However we note that extending this approach to three dimensions can be done by extending the beamformer over a two or three dimensional array, which while not trivial, has been widely studied. Conventionally the line of the array is defined along the x-axis, although it is here algorithmically convenient to define the array element positions in space as $\bar{p}_i = [x_i, y_i]^\mathsf{T}$, with the origin of the array arbitrarily defined at the first (and for a synthetic array oldest) element. The field of view of this array is $\Omega = \{(\vartheta,\varphi) : \vartheta \in [0°, 180°), \varphi = 0°\}$.

Notably, the geometry of a linear array does not distinguish directions which are symmetric with respect to the line of the array, nor can elevation be distinguished. This becomes important as Non-Gaussian SLAM is able to effectively retain multi-modal solutions, which are the natural result of our acoustic data. Consider a linear array of N elements with locations $\vec{p}$. In order to check the far-field assumption, define the array aperture:

$$l_a = \max_{\forall i,j} \|\bar{p}_i - \bar{p}_j\| \tag{4.43}$$

If the range $r > 2l_a^2/\lambda$ then the source is in the array far-field (the *Fraunhofer* zone) and assuming an incident plane wave is valid [106]. This distance is approximately 20-200m for our synthetic array lengths. Let the waveform recorded at the origin array element at time t be $f(t)$. Let τ_i represent the propagation delay between a plane wave incident onto the origin element and then arriving at array element i. Then the waveform arriving at element i is a time-delayed version, $f(t-\tau_i)$.

For array geometry as in Fig. 4-8, the propagation delay is:

$$\tau_i = -\frac{\vec{p_i}^\mathsf{T}\vec{u}_i}{c} \tag{4.44}$$

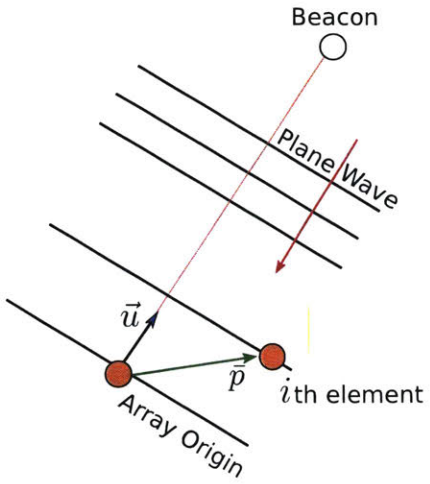

Figure 4-8: Geometry of plane wave incident on element i of linear array

where c is speed-of-sound in water. This time-delay applied to the signal f_i from array element i is identical to a phase-shift in frequency:

$$F_i[\omega] = \int_{-\infty}^{\infty} f(t-\tau_i)e^{-j\omega t}dt \tag{4.45}$$

$$= \int_{-\infty}^{\infty} f(t)e^{-j\omega\tau_i}e^{-j\omega t}dt \tag{4.46}$$

$$= F_i[\omega]e^{-j\omega\tau_i} \tag{4.47}$$

$$= F_i[\omega]e^{-j\left(\vec{p}_i^{\mathsf{T}}\vec{k}_i(\vartheta)\right)} \tag{4.48}$$

$$\rightarrow f_i[n-\tau_i] \overset{DFT/IDFT}{\rightleftharpoons} F_i[\omega]e^{-j\vec{p}_i^{\mathsf{T}}v\bar{e}c_i(\vartheta)} \tag{4.49}$$

where $\vec{\omega}$ is the vector of frequencies where we evaluate the n-point DFT. The array manifold vector $\bar{a}$, defined as the complex array response to a unit amplitude plane wave from look-angle ϑ can then be written as:

$$a_i(\vartheta)* = e^{-j\left(\vec{p}_i^{\mathsf{T}}\vec{k}_i(\vartheta)\right)} \tag{4.50}$$

$$\bar{a} = [a_1 \cdots a_N] \tag{4.51}$$

These time delays are induced by the array geometry, and beamforming works to negate these by using an element-wise spatial filter $e^{j\vec{\omega}\tau_i}$ that applies opposing

phase-shifts, followed by a summation over all signals for a single look-angle ϑ, as in Fig. 4-9:

$$Y[\omega, \vartheta] = \sum_{i=1}^{m} H_i[\omega, \vartheta] \cdot F_i[\omega] \tag{4.52}$$

$$\text{where } H_i[\omega, \vartheta] = e^{j\vec{\omega}\tau_i}$$

The output power of the beamformer averaged over all frequencies $\vec{\omega}$ is then:

$$|\widetilde{Y}[\vartheta]|^2 = \frac{1}{n} \sum_{n} |Y[\omega_n; \vartheta]|^2 \tag{4.53}$$

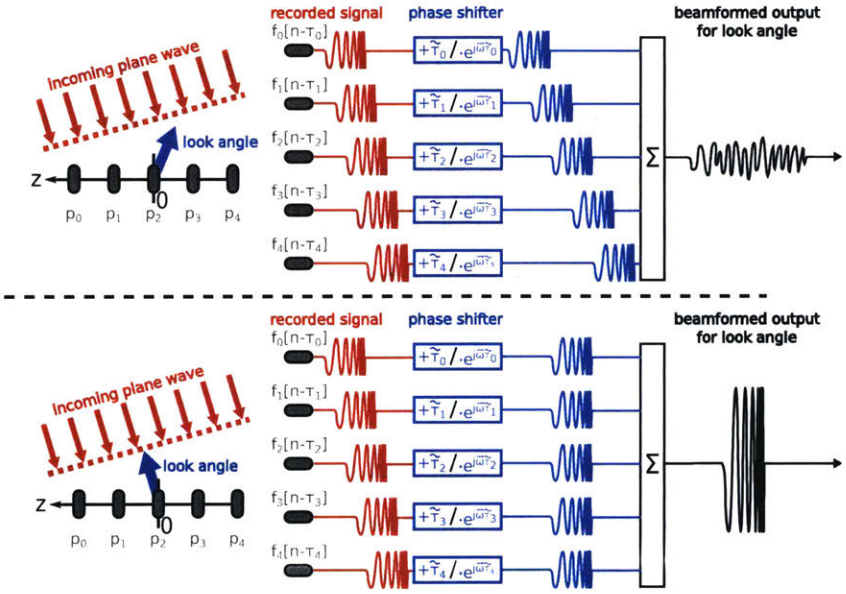

Figure 4-9: Conceptual Illustration of Conventional Beamforming. (*Top*:) The output power of the beamformer is low when the look-angle is not directed toward the incoming acoustic plane wave. (*Bottom*:) Signals received by each element constructively combine when the look-angle is directed toward the acoustic source, which is reflected by a high beamformer output power.

Conventional beamforming is typically performed using a geometrically fixed array of acoustic receivers. SAS extends this concept: if a single element coherently receives acoustic energy over time (either by actively ensonifying an object in the environment at every timestep or, as in this application, by receiving chirps from an acoustic

beacon in sync with the receiver recording), then the aperture can be increased with distance. In other words, after the receiving element has traveled a distance l_a, the effective length of the synthetic array, the received signals resemble those received by the elements of a physical linear array over that distance, and processing those signals by the same operations as a physical array produces the effect of a much larger aperture. One downside of SAS is the memory overhead associated with storing these signals over time and possibly the bandwidth required to communicate them with collaborating vehicles. A change to Eq. 4.44 that reflects this dependency over the most recent n timesteps:

$$\tau_i = -\frac{\vec{u}^T \vec{p}_{t,i}}{c} \qquad \text{where } i = 0, 1, 2, ..., n \tag{4.54}$$

Accurate estimation of the positions of the receiver(s), $\vec{x}_{t,i}$ the *micronavigation* problem is critical to the performance of the SAS system, and must be performed using the few measurements available: received signals, odometry estimates, and occasional absolute positioning measurements. The half-power beamwidth, in radians, of a linear array with identical elements can be expressed as:

$$\beta = \frac{\lambda}{l_a} \tag{4.55}$$

We do not include a factor of 2 as our onboard transducer only receives signals from a beacon assumed to be distant. The linear resolution in azimuth can be written in terms of the range to the source R, and the beamwidth β [40]:

$$\delta_\vartheta = \beta R \tag{4.56}$$

$$= \frac{\lambda R}{l_a} \tag{4.57}$$

Thus, increasing the resolution of the beamformer usually incurs the cost of increasing the frequency of transmission (and the power required to overcome increased attenuation), or increasing the size of the aperture. With SAS, the latter does not have to correspond to building or towing a larger physical array. The sampling rate of the

synthetic aperture should be finer than $\lambda/2$ [72]. For our single-receiver vehicle, the along-track speed is constrained by:

$$v_a \leq \frac{\lambda}{2T} \approx \frac{7.4}{2} \tag{4.58}$$

In order to resolve the most probable azimuth, the output of the beamformer is evaluated over the space of azimuths $[0, 2\pi]$ at the beamformer's maximum resolution of $\lambda/l_a \approx 2°$. As always, the algorithm is defined in such a way that the linear filter can be pre-computed and stored in memory.

Algorithm 4: Conventional Beamformer

Data: $\bar{p}$ array element locations, $\bar{\vartheta}$ vector of azimuths in $[0, 2\pi]$ of length k, c the speed of sound in water, $\bar{f}$ the vector of FFT frequencies of length m, $\bar{f}_i$ waveforms for $i \in 1 \cdots N$ array elements, $\bar{t}$ a temporary array of length m

Result: $\bar{y}$, beamformer power output, $\mathbf{H}$ a CBF filter of size (k, m, n)

Function `prepBF(`$\bar{p}$, $\bar{\vartheta}$, c, $\bar{f}$`)`**:**

begin

foreach $\vartheta \in \bar{\vartheta}$ **do**

$\vec{u} = [-\cos(\vartheta), -\sin(\vartheta)]$

$\tau = \dfrac{(\bar{p}^{\mathsf{T}}\vec{u})}{c}$

for $i \leftarrow 1$ **to** N **do**

$\mathbf{H}[\vartheta, :, i] = \exp[-j2\pi\tau\bar{f}_{ij}]_{ij}$ `// element-wise exponent`

$\mathbf{H}[\vartheta, :, :] \leftarrow \left(\mathbf{H}^H[\vartheta, :, :]\right)_{ij}$ `// matrix adjoint`

end

end

return H

end

Function `delaysum(`$\bar{f}_i$, **H**`)`**:**

begin

foreach $\vartheta \in \bar{\vartheta}$ **do**

$\bar{t} = \sum_i (f_i)_{ij}(\mathbf{H}[\vartheta, :, i])_{ij}$

$\bar{t} \leftarrow \|\bar{t}\|_{ij}$

$\bar{y}[\vartheta] = \sum \bar{t}$

end

return $\bar{y}$

end

return

Figs. 4-10 and 4-11 show the output power of the conventional beamformer by azimuth (or beampattern) for a single beacon chirp, and by length of the synthetic array. Each array element corresponds to approximately 1m of additional array length. This is not always the case depending on the vehicle's trajectory for that experiment, which are presented in detail in Fig. 5-1. The output power of a longer array is always greater as the power has not been normalized by number of elements.

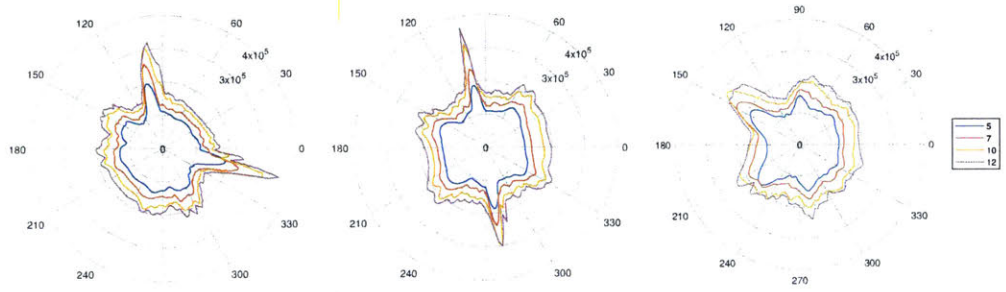

Figure 4-10: Stationary Beacon, CBF beam patterns for Chirps 70, 160 and 310 respectively, color-coded by the number of elements in the synthetic array. Chirps are representative of the start, middle and end of the elliptical mission trajectory. The ground truth azimuths of the stationary beacon are $98°, 108°, 149°$ respectively.

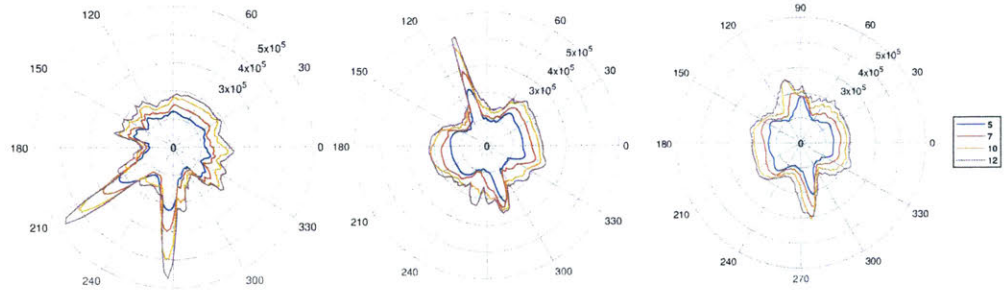

Figure 4-11: Drifting Beacon, CBF beam patterns for Chirps 1520, 1720 and 1920 respectively, color-coded by the number of elements in the synthetic array. Chirps are representative of the start, middle and end of the circular mission trajectory with the beacon at the center of the circle. The ground truth azimuths of the drifting beacon are $131°, 117°, 130°$ respectively.

We observe two results of note to the SLAM implementation. Firstly, the linear array does not resolve ambiguity symmetric about the axis of the array, usually resulting in two equally favored modes, see Fig. 4-10 left and middle, and Fig. 4-11 left. In some cases, due to unmodeled errors in the synthetic array's geometry, one

symmetric peak is of greater magnitude, see Fig. 4-11 middle and right. However, this peak is not necessarily the true azimuth. While this kind of error may adversely affect a ML result, the non-parametric SLAM solver does not discard information regarding the smaller symmetric peak.

A second and related effect is that for some trajectories (i.e. synthetic arrays) the beampattern becomes spread over a wide number of possible azimuths, see Fig. 4-10 and 4-11 right. Additionally, computing over a larger synthetic aperture is not always possible or optimal, as errors in position and linearity of the array accumulate over time. As we discuss in Section 4.2 and Chapter 5, the SLAM formulation does not optimize over the length of each synthetic array (i.e. number of elements) to minimize positioning error across the entire array, however the length of the array can vary by factor and data pre-processing can be applied to this purpose during construction of the factor graph. One solution could be to pre-preemptively check for and cull results from highly non-linear trajectories, however, we also show below that after sampling, useful azimuthal information is still returned. Another possible solution is to reformulate the beamformer to allow for additional dimensions of variation for the array.

Figs. 4-12 and 4-13 plot representations of the beam pattern that is modeled by the aliasing sampler of Section 4.2.2 as a probability distribution over azimuth. Beam patterns are color-coded by the length of the array, ranging from 5 to 12 elements. For figure display purposes, the output power is floored at the minimum power. Discrete samples above the 80% quantile of power are shown as points on the beam pattern, these are the samples passed as probability weights to the aliasing sampler as described in Algorithm 2.

The aliasing sampler is an algorithmic representation of the non-parametric probability density over azimuth computed by the beamformer and sampled directly by the Non-Gaussian SLAM algorithm. At the cost of increased computation and space requirements, the inherently ambiguous, multi-modal aspect of the response is preserved without the need to fit a parametric distribution. Poorer resolution resulting from unmodeled errors in the synthetic array still contribute information to the solu-

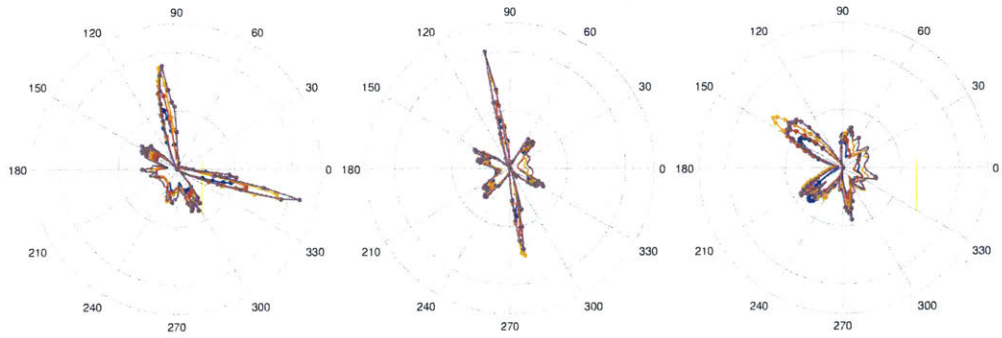

Figure 4-12: Stationary Beacon, CBF beam patterns for Chirps 70, 160 and 310 respectively. Chirps are transmitted at 1Hz and time-stamped from the start of the mission. The discrete 80% quantile samples are shown as points on the beam pattern, which is floored by minimum power for this plot only. This representation of the beam pattern is modeled by the SLAM solver as a probability distribution using the alias sampling algorithm discussed in Section 4.2.2.

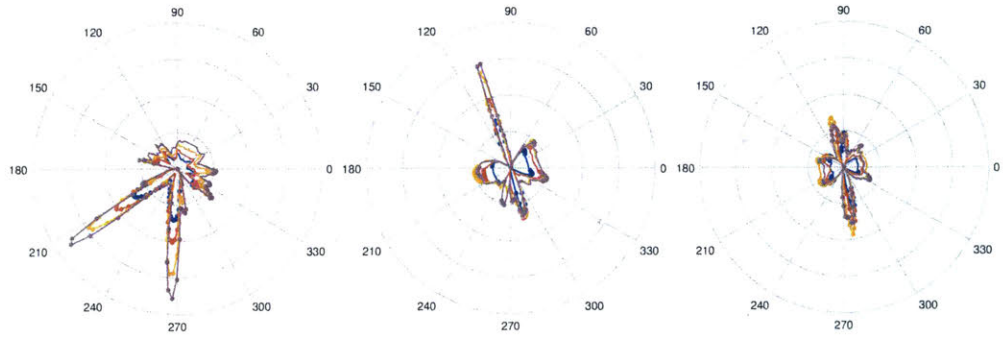

Figure 4-13: Drifting Beacon, 80% quantile CBF beam patterns and discretized samples for Chirps 1520, 1720 and 1920 respectively.

tion, as we discuss later, the greater spread over likely azimuths is corrected by the sum-product method.

Figs. 4-14 and 4-15 plot each element of the synthetic array, arranged by length, and the beam patterns from Figs. 4-12 and 4-13 over the origin element. Both axes are plotted in meters. This is a visual representation of the measurement from the SAS-factor associated with that series of poses. In general we find there is a trade-off to be made when increasing the length of the synthetic array, as trajectory errors accumulate that violate the linear array assumption (see Fig. 4-14 Chirp 310, the 12-element array at top).

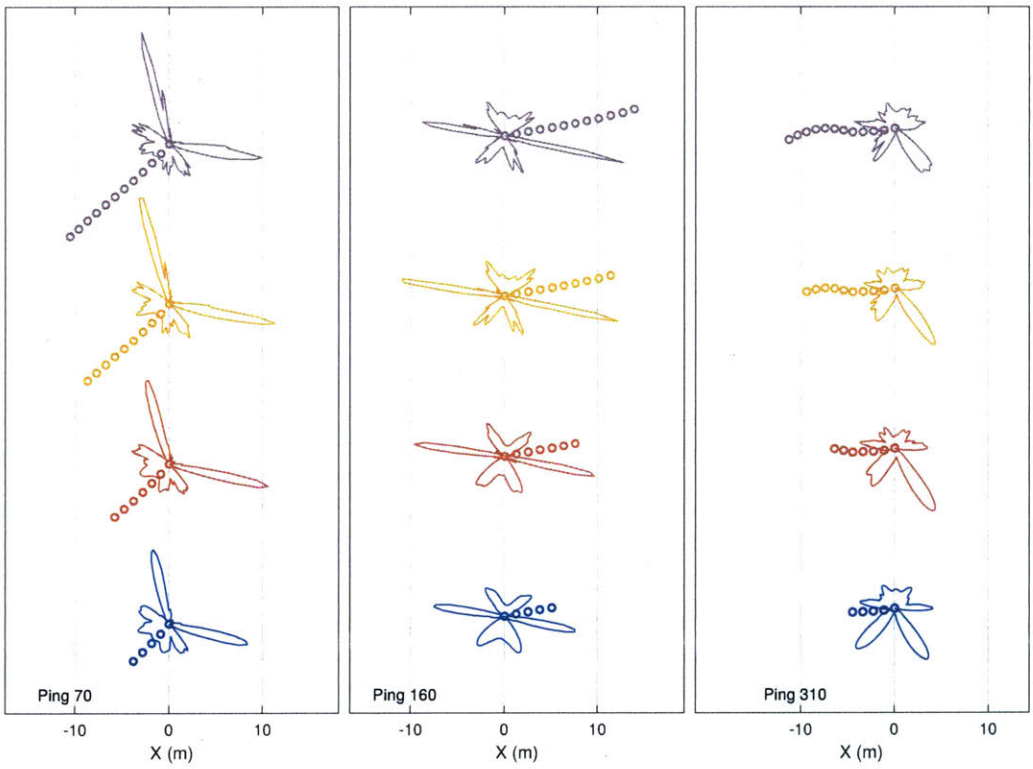

Figure 4-14: Stationary Beacon, CBF beam patterns for Chirps 70, 160 and 310 respectively, represented at the origin of element their synthetic arrays. Since the CBF assumes a linear array and beacon in the far-field, small deviations in the trajectory of the ASV (as in Ping 310) reduce the total power at the peak, and the accuracy of the azimuthal estimate.

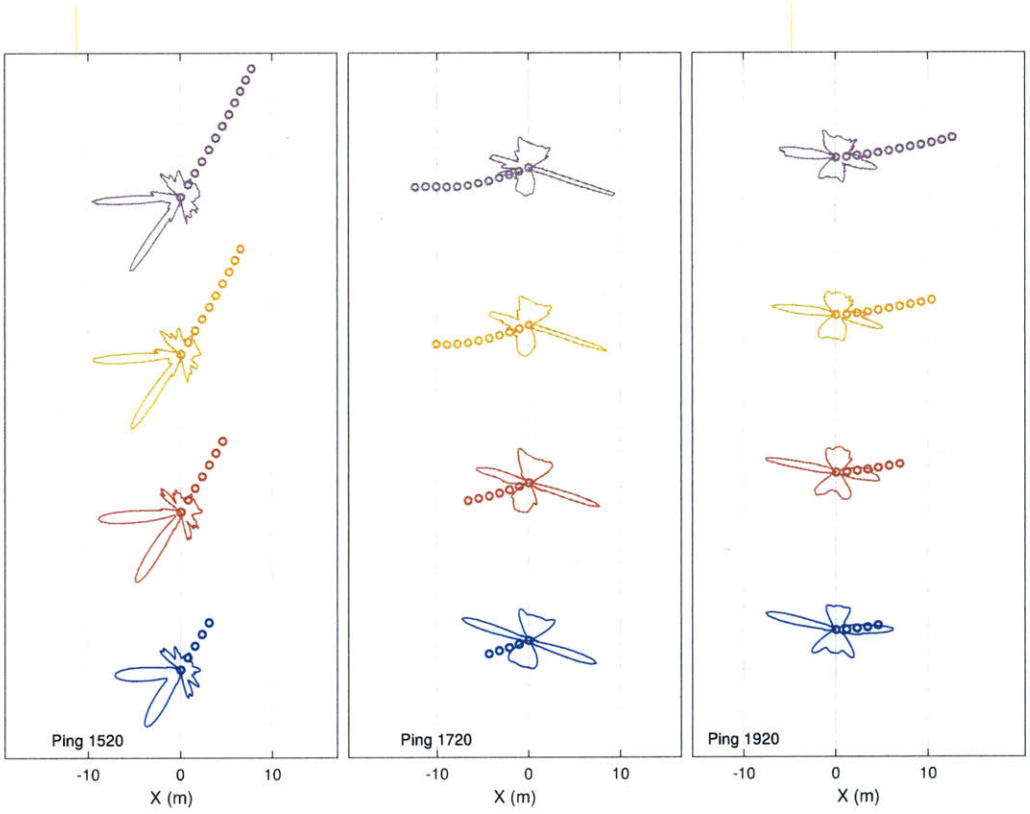

Figure 4-15: Drifting Beacon, CBF beam patterns for Pings 1520, 1720 and 1920 respectively, represented at the origin element of their synthetic arrays. The ground truth azimuths are $131°, 117°, 130°$ respectively.

4.3 Caesar: Multi-Modal Inference

This section covers in brief the multi-modal incremental smoothing and mapping inference algorithm (mm-iSAM) [60] for performing inference over Non-Gaussian factor graphs, and the design of a SAS-factor that consistently describes synthetic aperture sonar with acoustic measurements within this framework. The reader is referred to Fourie ([58],[59]) for a detailed exposition of this algorithm, which will be used as the non-Gaussian SLAM back-end. We posit that non-Gaussian (and multimodal) factor graph formulations (with inference) provide greater flexibility when representing challenging real-world environments, events and measurement paradigms. More complicated measurement processes such as the present SAS likelihood factor require a much richer non-parametric (multi-modal and non-Gaussian) structure.

The underlying inference problem that estimates state variables Θ from non-ideal sensor measurements $\mathbf{Z}$ is described by a bi-partite factor graph language [92]. The true measurements $\mathbf{z}_i^*$ are assumed to originate from the true hidden variable states θ_i^* and the associated measurement process $\mathbf{z}_i^* \sim [\,\mathbf{Z}_i \,|\, \Theta_i = \theta_i^*\,]$. Factors (measurement models) are modeled by means of algebraic structure along with meaningful stochastic models. While much of the existing state estimation, localization, and mapping work in autonomy relies on point estimates and/or unimodal Gaussian approximations these yield less computational intensive algorithms, but they can over simplify the representation of uncertainty, miss important mission-relevant information, and yield brittle performance.

In this work, the measurement likelihood model shown in Fig. 4-29 is completely non-Gaussian and used along with multimodal incremental smoothing and mapping (mm-iSAM) to perform belief space inference over all sensor data without having to resort to unimodal Gaussian assumptions. Modeling and computing non-Gaussian posterior densities is potentially exponential in computational complexity and mm-iSAM uses nine principles in a combination of exact and asymptotically correct approximation to reduce this load. One major feature of mm-iSAM is that the full marginal posterior belief of any or all variables in the factor graph can be estimated.

We outline the two methods unique to this formulation of the non-Gaussian SLAM problem: The algorithmic representation of non-parametric probability distributions and their products in a computational tractable way (i.e. computing marginals), and the sum-product approach to inference on the Bayes tree (i.e. propagating marginals).

4.3.1 Belief Propagation on the Bayes Tree

A set of random variables Θ, and independent measurements $\mathbf{Z}$ is fully described by the joint probability belief $[\,\Theta \,|\, \mathbf{Z}\,] \in \mathcal{P}$, where $\mathcal{P}$ is the space of functions that describe their probability densities. By the Chain Rule, the theoretical posterior joint probability density is proportional to the product of independent measurements in the form of likelihoods $[\,\mathbf{Z}_i \,|\, \Theta_i\,]$ and variable prior potentials $[\,\Theta_j\,]$:

$$[\,\Theta \,|\, \mathbf{Z}\,] \propto \prod_i [\,\mathbf{Z}_i \,|\, \Theta_i\,] \prod_j [\,\Theta_j\,] \tag{4.59}$$

The state of each variable can be correctly estimated given their joint probability density. Unfortunately, computing the full joint is typically intractable, especially for high-dimensionality SLAM systems with many robot states to be inferred over, and furthermore, for the purposes of robot localization and navigation, we are usually only interested in solving for actual state estimates, as opposed to a detailed representation of the full posterior. This suggests the form of a solution that is able to bypass or approximate computing the full joint.

iSAM2, the current state-of-the-art SLAM solver [86], solves the problem by computing a max-product optimization for θ^*, assuming Gaussian likelihoods:

$$z_i = h_i(\theta_i, l_j) + v_k \tag{4.60}$$

$$\rightarrow [\,z_i \,|\, \theta_i, l_j\,] \propto \exp\left[-\frac{1}{2}\|h_i(\theta_i, l_j) - z_i\|^2_{\Sigma_i}\right] \tag{4.61}$$

where l_j indicate landmarks, h_i is a measurement equation and w_i, v_k are zero-mean

Gaussian noise with respective covariance matrices Γ_i, Σ_k. $\|x\|^2_\Sigma$ denotes the Mahalanobis distance $x^\mathsf{T}\Sigma^{-1}x$. Then, Eqn. 4.59 can be written as a nonlinear least squares problem [43]:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \left[\Theta \mid \mathbf{Z} \right] \tag{4.62}$$

$$= \underset{\theta}{\operatorname{argmin}} - \log \left(\left[\Theta \mid \mathbf{Z} \right] \right) \tag{4.63}$$

$$= \underset{\theta}{\operatorname{argmin}} - \log \left(\prod_i \exp \left[-\frac{1}{2} \| h_i(\theta_i, l_j) - z_i \|^2_{\Sigma_k} \right] \right) \tag{4.64}$$

$$= \underset{\theta}{\operatorname{argmin}} \Sigma_i \| h_i(\theta_i, l_j) - z_i \|^2_{\Sigma_k} \tag{4.65}$$

Eqn. 4.65 can subsequently be linearized about a current estimate, producing a linear least-squares problem that can be solved by very efficient methods, i.e. Cholesky factorization, QR factorization etc. The resulting joint posterior distribution is a multivariate Gaussian distribution and the max-product approach resolves the mean (first moment). This approach has been extended to resolve multi-modal or multi-hypothesis likelihoods as well, by tracking and maintaining the various modes in a bank [79], and to Non-Gaussian distributions by an algebraic transformation of the distribution [145].

For Gaussian likelihoods only, the max-product and sum-product algorithms are identical [192]. This is not the case for non-parametric distributions. mm-iSAM approaches the problem of computing Eqn. 4.59 from the sum-product algorithm for factor graphs. Computational concerns are addressed in two broad categories, exact (operations on the special structure of the Bayes Tree), and approximate (stochastic sampling-based approximate density products and convolutions).

Consider a general set of variables $x_i \forall i \in 1 \cdots n$, taking values in the domain $\mathcal{D}_i$. Let $f(x_1, \cdots, x_n)$ be a function with domain $\mathcal{S} = \mathcal{D}_1 \times \mathcal{D}_2 \times \cdots \times \mathcal{D}_n$, and codomain $\mathcal{R}$. $\mathcal{S}$ is known as the configuration space for the variables x_i, and each element of $\mathcal{S}$ is a configuration of x_i that assigns each variable a value.

Associate each function $f(x_1, \cdots, x_n)$ with n marginal functions $f_i(x_i)$. For each

value $d \in \mathcal{D}_i$, $f_i(d)$ can be obtained by summing the value of $f(x_1, \cdots, x_n)$ over all configurations where $x_i = d$. The following example introduces the notation for summing over all variables except x_i [92] :

$$\sum_{\setminus\{x_2\}} f(x_1, x_2, x_3) : \sum_{x_1 \in \mathcal{D}_1} \sum_{x_3 \in \mathcal{D}_3} f(x_1, x_2, x_3) \tag{4.66}$$

Then the marginal function for x_i of f is:

$$f_i(x_i) = \sum_{\setminus\{x_i\}} f(x_1 \cdots x_n) \tag{4.67}$$

If the functions f_i represent probability distributions and the variables x_i, random variables X_i, this representation recovers the standard marginal distribution for a subset of a collection of random variables. If $[X_1 \cdots X_N]$ is the joint probability distribution for $X_1 \cdots X_N$ random variables, then the marginalization rules are:

$$[X_1] = \int_{\setminus\{X_1\}} [X_1 \cdots X_N] \, dX_i \tag{4.68}$$

$$[X_1] = \int_{\setminus\{X_1\}} [X_1 \,|\, X_2 \cdots X_N] \, [X_2 \cdots X_N] \, dX_i \tag{4.69}$$

With the discrete case:

$$[X_1] = \sum_{\setminus\{X_1\}} [X_1 \cdots X_N] \tag{4.70}$$

$$[X_1] = \sum_{\setminus\{X_1\}} [X_1 \,|\, X_2 \cdots X_N] \tag{4.71}$$

The sum-product algorithm terminating on an arbitrary factor graph computes and stores the marginal posterior probability distributions for every variable node in the graph, and the conditional probabilities in every factor node. An assumption is made that summation and products are well-defined over the functions f_i. Of those, the product of two probability densities is non-trivial and the implementation

is presented in more detail in Section 4.3.2.

A factor graph is a bipartite graph with nodes representing variables x_i and functions f_i. Variable nodes are connected only to function nodes and vice-versa. A variable node is connected to a function node only if the variable is an argument of that function. In other words, the factor graph is a graphical model of how the joint probability distribution can be factored. Fig. 4-16 is a small example that represents the following relationship:

$$f(x_1, x_2, x_3, l_1, l_2) = f_1(x_1, x_2) f_2(x_1, l_1) f_3(x_2, l_1) f_4(x_2) f_5(l_1, l_2) f_6(x_3, l_2) \tag{4.72}$$

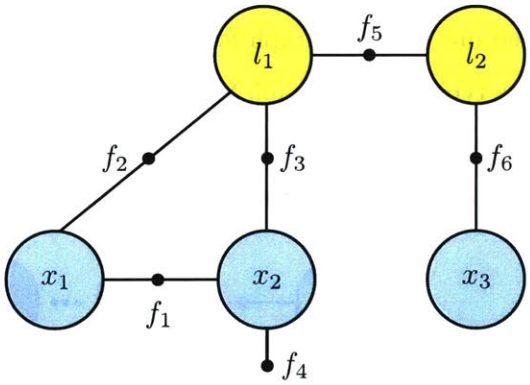

Figure 4-16: Example factor graph with five variable nodes and six function nodes.

The marginal function for each pose can be written:

$$f_1(x_1) = \sum_{x_2} f_1(x_1, x_2) f_4(x_2) \left(\sum_{l_1} f_2(x_1, l_1) f_3(x_2, l_1) \right. \tag{4.73}$$

$$\left. \left(\sum_{l_2} f_5(l_1, l_2) \left(\sum_{x_3} f_6(x_3, l_2) \right) \right) \right) \tag{4.74}$$

$$f_2(x_2) = f_4(x_2)\left(\sum_{x_1} f_1(x_1, x_2)\left(\sum_{l_1} f(x_1, l_1) f(x_2, l_1)\right.\right. \tag{4.75}$$

$$\left.\left.\left(\sum_{l_2} f_5(l_1, l_2)\left(\sum_{x_3} f_6(x_3, l_2)\right)\right)\right)\right) \tag{4.76}$$

$$f_3(x_3) = \sum_{x_2} f_(x_2)\left(\sum_{x_1} f_1(x_1, x_2)\right. \tag{4.77}$$

$$\left.\left(\sum_{l_1} f(x_1, l_1) f(x_2, l_1)\left(\sum_{l_2} f_5(l_1, l_2) f_6(x_3, l_2)\right)\right)\right) \tag{4.78}$$

This general framework is a good description of the SLAM inference problem, as function nodes encode measurements and variable nodes encode poses or states those measurements are made from or of. Functions represent the constraints of the optimization carried out over the graph for the value (or expected value) at the nodes. Fig. 4-17 shows a typical factor graph formulation of a general SLAM problem. Landmarks variables denoted l_1, l_2 are typically expected to be associated with ground truth information. State variables x_i represent robot poses.

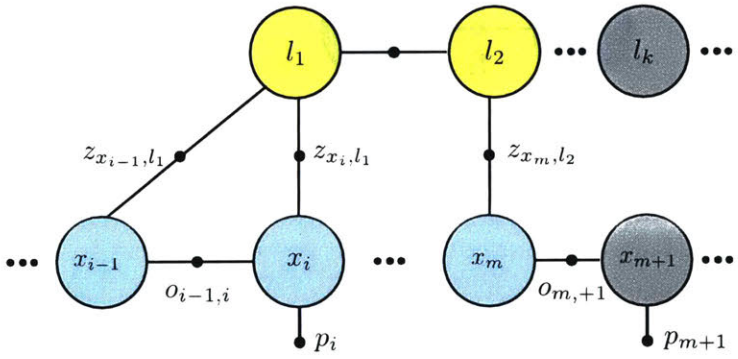

Figure 4-17: Factor graph formulation of a general SLAM problem, where x_i may represent robot poses, l_i represent landmarks in the world. z_i are observations made of each landmark at instance i, and p_i are priors.

Given a factor graph, the sum-product algorithm computes all marginal functions by, as expected, a series of sums and products. Messages are passed along the edges of the factor graph, terminating when two have passed on each edge, one in each di-

rection. The appropriate variable marginals can then be computed at each respective node as the product of all incoming messages.

A message $m_{x_i \to f_i}$ passed from a variable node x_1 along an edge e to node f_i is the product of all messages received at that node other than along e, then x_1 is marginalized out:

$$m_{x_i \to f_i}(x_i) = \prod_{f_j \in N(x_i) \backslash \{f_i\}} m_{f_j \to x_i}(x_i) \tag{4.79}$$

where $N(x_i)$ is the set of all neighbors of x_i. This can also be written as a conditional belief over the subset of measurements other than the one along e.

$$m_{\Theta_i \to Z_k} = \prod_{Z_j \in N(\Theta_i) \backslash \{Z_k\}} [\, Z_k \,|\, Z_j \,] \tag{4.80}$$

As the factor graph is bipartite, the set of all neighbors for a variable node contains only functions, and vice-versa for function nodes, thus only two messages need to be defined. Fig. 4-18 illustrates this set of messages. A message passed from a function node f_i along an edge e to node x_i is the product of that function with all messages received at that node other than along e, and marginalized for x_i:

$$m_{f_i \to x_i}(x_i) = \sum_{\backslash \{x_i\}} \left(f_i(N(f_i)) \prod_{x_j \in N(f_i) \backslash \{x_i\}} m_{x_j \to f_i} \right) \tag{4.81}$$

This can be written as:

$$m_{z_k \to \Theta_i} = \sum_{\backslash \{\Theta_i\}} \left([\, Z_k \,|\, \Theta_j \,] \prod_{N(x_i) \backslash \{Z_k\}} [\, Z_j \,|\, \Theta_j \,] \right) \tag{4.82}$$

Message-passing using this set of update rules will terminate in a non-cyclic factor graph after two messages have passed on each edge. The marginal belief of each variable then contains all the information introduced by the factors in the graph:

$$\left[\hat{\Theta}_i \,|\, \mathbf{Z} \right] = \int [\, \Theta \,|\, \mathbf{Z} \,]\, d \backslash \theta_i. \tag{4.83}$$

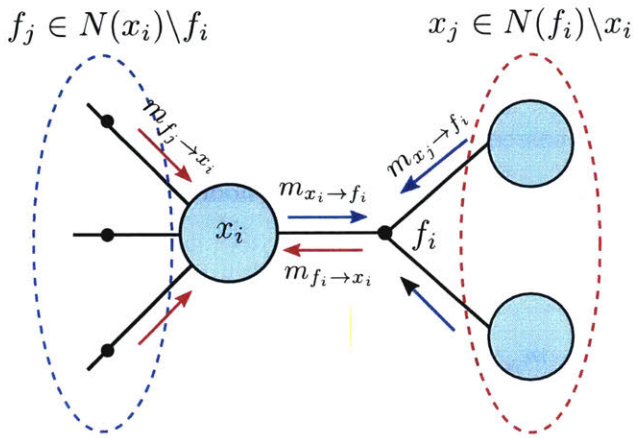

Figure 4-18: Sum-Product Message-Passing on the Factor Graph

If the factor graph is cyclic, which occurs often in SLAM formulations, the resulting factor graph has implicit algebraic loops. The consequence of which is the sum-product algorithm does not naturally terminate as messages can arrive and be passed multiple times on a given edge. This increases the complexity of computing over the graph and affects the accuracy of the final result.

So far a general factor graph framework has been described, with few assumptions made over the variables and nodes. A factor graph representation is in one sense simple to construct as the relationships between edges and nodes are straight-forward to construct from a given set of functions. mm-iSAM builds the SLAM inference problem starting from a factor graph representation. The problems associated with the cyclic factor graph can be addressed by converting the graph into a Bayes network, which is a directed acylic graph, and from there, into a Bayes tree, which allows for efficient computation and termination of the sum-product algorithm [86]. Message-passing on the Bayes net or Bayes tree is also known as belief-propagation [129]. We now assume that each variable is a random variable and each factor is associated with a residual prediction model $\delta(\theta_\cdot, z_\cdot)$ representing a probability density over with measurements $\mathbf{z}_k^*$.

A variable elimination algorithm is used to convert a factor graph into a directed Bayes net that is algebraically equivalent to the original graph [86]. One variable is converted to the Bayes net at a time, eliminating it from the factor graph, proceeding

until all variables have been eliminated from the factor graph. We modify our previous notation $\backslash\{x_i\}$ to define in a probabilistic sense, x_i as the frontal variable, and the set of all variables not including the frontal as the separator S_i. This notation, in the probabilistic context of the factor graph, indicates all variables x_i is conditioned on.

Variable Elimination:

1. Let Θ_i be the variable to be eliminated and add it to the Bayes net. Remove from the factor graph all neighboring factors $[\mathbf{Z}_j \,|\, \Theta_j] \in N(\Theta_i)$, that is all variables Θ_j in the separator S_i.

2. Form the local joint density $[\Theta_i, S_i \,|\, \mathbf{Z_i}] \propto \prod [\Theta_j \,|\, \mathbf{Z_j}] [\Theta_j]$

3. Using the Chain Rule, factor out the separator variables:
   $[\Theta_i, S_i \,|\, \mathbf{Z_i}] = [\Theta_i \,|\, S_i, Z_i] [S_i \,|\, Z_j]$.

4. Add the factor $[S_i \,|\, Z_j]$ back to the factor graph. Add the conditional $[\Theta_i \,|\, S_i, Z_i]$ to the Bayes net.

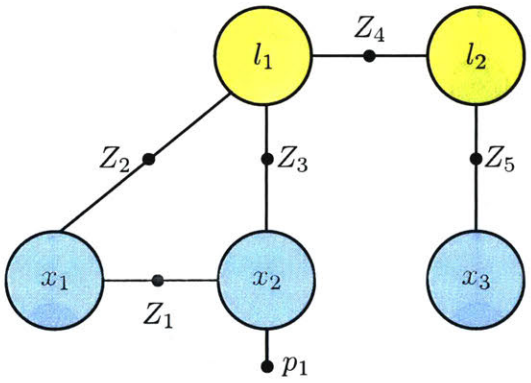

Figure 4-19: Example factor graph with five variable nodes, five relative measurements and one prior.

Consider the factor graph in Fig. 4-19, representing the following joint probability:

$$[\Theta \,|\, \mathbf{Z}] \propto [X_2] [Z_1 \,|\, X_1, X_2] [Z_2 \,|\, X_1, L_1] \times \\ [Z_3 \,|\, X_2, L_1] [Z_5 \,|\, L_1, L_2] [Z_6 \,|\, X_3, L_2] \tag{4.84}$$

Fig. 4-20 shows the steps of the variable elimination algorithm carried out in the order x_3, l_2, l_1, x_1, x_2. In Step (2), eliminating the variable x_3 removes the factor $[\, Z_5 \,|\, l_2, x_3 \,]$ from the factor graph. The separator S_{x_3} contains only l_2. The local joint density is factored as $[\, x_3, l_2 \,|\, Z_5 \,] = [\, x_3 \,|\, l_2, Z_5 \,] [\, l_2 \,|\, Z_5 \,]$ and the new marginal $[\, l_2 \,|\, Z_5 \,]$ is added back to the factor graph as a prior.

In step (4, eliminating variable l_1 removes the factors $[\, Z_2 \,|\, l_1, x_1 \,], [\, Z_3 \,|\, l_1, x_2 \,]$. The separator contains $S_{l_1} = x_1, x_2$. The local joint density is factored as $[\, l_1, x_1, x_2 \,|\, Z_2, Z_3 \,] = [\, l_1 \,|\, x_1, x_2, Z_2, Z_3 \,] [\, x_1, x_2 \,|\, Z_2, Z_3 \,]$ and the new marginal $[\, x_1, x_2 \,|\, Z_2, Z_3 \,]$ is added back to the factor graph as a factor between nodes x_1, x_2.

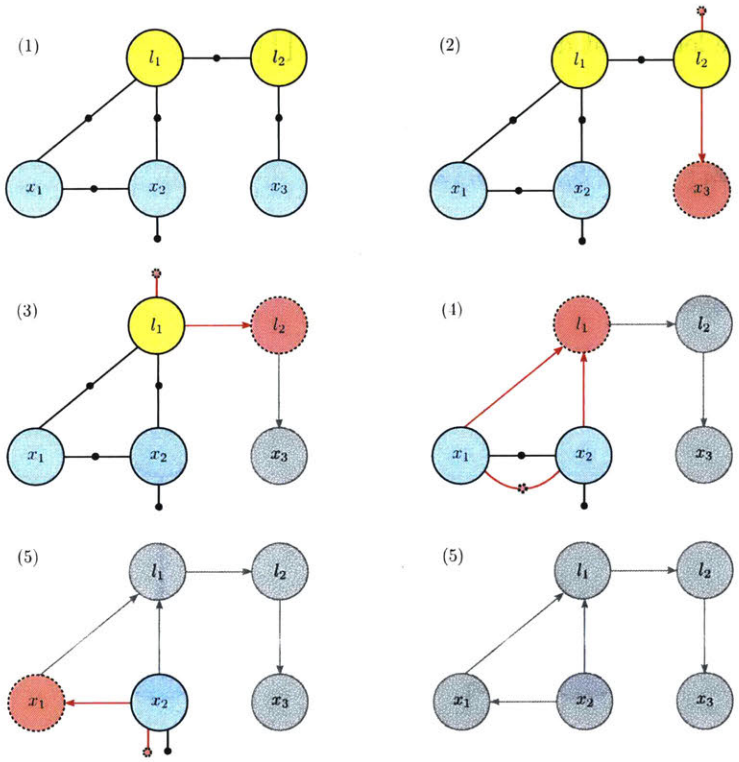

Figure 4-20: Variable elimination to convert the factor graph in Fig. 4-16 to an equivalent Bayes net, using variable ordering x_3, l_2, l_1, x_1, x_2.

The variable ordering carried out by the elimination algorithm is arbitrary and different orderings may produce a different network. A poor ordering produces a factorization that has a negative impact on the performance of the algorithm. An ideal ordering minimizes the size of the cliques, that is, it attempts to enforce sparsity

in the resulting conditionals on the Bayes net. As shown in [87], in the parametric case a poor ordering results in the equivalent of *fill-in*, wherein variable reordering during factorization destroys the sparsity of the information matrix. mm-iSAM, following from iSAM1 and iSAM2, uses the Constrained-COLAMD heuristic to obtain good variable orderings [42].

The resulting Bayes net is chordal (or triangulated), i.e. any cycle of length greater than 3 must have a chord, a node connecting two non-consecutive vertices in the graph. This is guaranteed since the variable elimination algorithm means the variable order is a perfect elimination ordering. Each frontal variable and its separator form a clique. We now construct a Bayes tree as in [86], by discovering the cliques in the Bayes net [169].

Algorithm 5: Bayes Tree Construction from Bayes Net after Variable Elimination

foreach $[\Theta_j \mid S_j, Z_j]$ *in reverse elimination order* **do**
  if $S_j = \{\}$ **then**
    Start a new root clique C_R containing Θ_j
  else
    Identify parent clique C_P that contains the first eliminated variable of S_j as a frontal variable
    if *the nodes of the parent are equal to the separator:* $\Theta_{F,P} \cup \Theta_{S,P} = S_j$ **then**
      Insert $[\Theta_j \mid S_j, Z_j]$ into clique C_P
    else
      Start a new clique C' as a child of C_P containing Θ_j
    end
  end
end

For the example given in 4-20, the following conditional probabilities, given in variable elimination order, are added to the Bayes tree:

$$[\Theta \mid \mathbf{Z}] = [X_3 \mid L_2, Z_5]\,[L_2 \mid Z_5]\,[L_2 \mid L_1, Z_5]\,[L_1 \mid Z_5]\,[L_1 \mid X_1, X_2, Z_2, Z_3] \quad (4.85)$$
$$[X_1 \mid Z_2]\,[X_1 \mid X_2, Z_1]\,[X_2 \mid Z_1]\,[X_2]$$

Using reverse variable ordering x_2, x_1, l_1, l_2, x_3, the Bayes Tree shown in Fig. 4-21

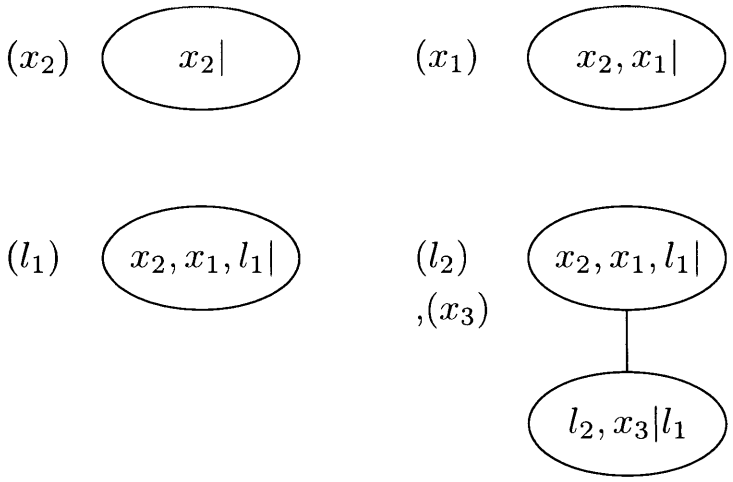

Figure 4-21: Example of Clique Finding Algorithm (Alg. 5) on the Bayes Network in Fig. 4-20, using variable ordering x_2, x_1, l_1, l_2, x_3. The variables inserted at each step is shown on the left. An equivalent Bayes Tree with two cliques is produced.

is generated. Two cliques were found by Algorithm 5, one of dimension 3 and one of dimension 2. The message passed up from the leaf contains the conditional marginal probability of l_1 only.

We can show that this Bayes Tree is not unique, nor is the variable ordering the most efficient. Intuitively, a tree representing the most factored form of the joint probability is preferred, reducing the computational complexity within each clique. Consider the variable elimination order x_2, x_3, x_1, l_2, l_1 instead. This adds the following conditionals to the Bayes tree given in Fig. 4-22:

$$\begin{aligned}[\Theta \mid \mathbf{Z}] = &[X_2 \mid X_1, L_1, Z_1, Z_3]\,[X_3 \mid L_2, Z_5] \times \\ &[X_1 \mid L_1, Z_1, Z_3]\,[X_1 \mid L_1, Z_2]\,[L_2 \mid Z_5]\,[L_2 \mid L_1, Z_4] \\ &[L_1 \mid Z_2]\,[L_1 \mid Z_1, Z_3]\,[L_1 \mid Z_4]\end{aligned} \tag{4.86}$$

where Clique 1, the root, and its two children 2 and 3 contain respectively:

$$\text{Clique 1:} \quad [L_1, L_2 \mid \mathbf{Z}] = [L_2 \mid Z_5][L_2 \mid L_1, Z_4] \times [L_1 \mid Z_2][L_1 \mid Z_1, Z_3][L_1 \mid Z_4] \tag{4.87}$$

$$\text{Clique 2:} \quad [X_1, X_2 \mid L_1, Z_1, Z_2, Z_3] = [X_2 \mid X_1, L_1, Z_1, Z_3] \times [X_1 \mid L_1, Z_2][X_1 \mid L_1, Z_1, Z_3] \tag{4.88}$$

$$\text{Clique 3:} \quad [X_3 \mid L_2, Z_5] \tag{4.89}$$

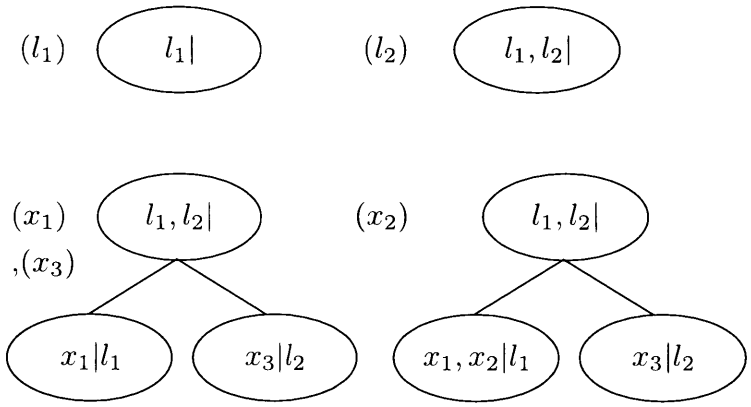

Figure 4-22: Example of Clique Finding algorithm to convert the Bayes Network in Fig. 4-20 to an equivalent Bayes Tree, using variable ordering l_1, l_2, x_1, x_3, x_2. The resulting Bayes Tree has one more clique than the previous ordering and each clique has at maximum 2 frontal variables. This tree structure is preferred as more computation has been factored into individual cliques.

The Bayes tree representation has special structure that can be exploited for efficiently computing the marginal likelihoods of each frontal variable [85]. Frontal and separator variables within a clique are always disjoint, and frontal variables between cliques are also always disjoint. Intuitively, a variable will not be conditioned on itself, and its marginal probability can only be added to the tree once, after which it has been removed from the factor graph. It also follows that frontal variables do not travel up the tree, i.e. parent cliques never contain frontal variables of their children. Therefore, cliques are only related through their separator variables, which travel up the tree until they appear as a frontal variable.

The special structure of the Bayes tree stores in each clique the partial posterior belief calculated over that clique's frontal variables, which are subsequently marginalized out of the up-message. That is, at each level of an ideal Bayes Tree, the maximum number of sibling cliques with the minimum number of frontal variables in each are present. Let the variables of a clique $\mathcal{C}_i$ where $i \in 1 \cdots r$ be the set $\Theta_{(\mathcal{C},i)} = \{\Theta_{(\mathcal{F}_k,i)}, \Theta_{(\mathcal{S}_k,i)}\}$ where $\Theta_{(\mathcal{F}_k,i)}$ is the k-th frontal variable of the clique and $\Theta_{(\mathcal{S}_k,i)}$ is the k-th separator variable. The full marginal posterior over the frontal variables $\Theta_{(\mathcal{F},i)}$ of any clique can be written as:

$$\begin{aligned}\left[\,\Theta_{(\mathcal{F},i)} \,|\, \mathbf{Z}\,\right] &= \int_{\mathcal{D}} \left[\,\Theta \,|\, \mathbf{Z}\,\right] d\Theta_{\setminus(\mathcal{F},i)} \\ &\propto \int_{\mathcal{D}} \prod_{j=1}^{r} \left[\,\mathbf{Z}_j \,|\, \Theta_{(\mathcal{C},j)}\,\right] d\Theta_{\setminus(\mathcal{F},i)}\end{aligned} \tag{4.90}$$

The frontal variables can be marginalized out, but the partial belief is still conditioned on the clique's separator. In particular, marginalizing the frontal variables of the root, which by definition has no separator variables, factors the joint product over the domains of each clique:

$$\begin{aligned}\left[\,\Theta_{(\mathcal{F},1)} \,|\, \mathbf{Z}\,\right] \propto \int_{\mathcal{D}_2} \left[\,\mathbf{Z}_2 \,|\, \Theta_{(\mathcal{C},2)}\,\right] \int_{\mathcal{D}_3} \left[\,\mathbf{Z}_3 \,|\, \Theta_{(\mathcal{C},3)}\,\right] \cdots d\Theta_{(\mathcal{F},3)} \times \cdots \\ \times \int_{\mathcal{D}_r} \left[\,\mathbf{Z}_r \,|\, \Theta_{(\mathcal{C},r)}\,\right] \cdots d\Theta_{(\mathcal{F},r)} d\Theta_{(\mathcal{F},2)}\end{aligned} \tag{4.91}$$

where each expansion denoted by $(\cdots)$ contains the children of that clique. From our example in Fig. 4-22, this is:

$$\begin{aligned}\left[\,L_1, L_2 \,|\, \mathbf{Z}\,\right] \propto \left[\,Z_4 \,|\, L_1, L_2\,\right] \times \int \left[\,Z_5 \,|\, L_2, X_3\,\right] dx_3 \times \\ \int \left[\,Z_1 \,|\, X_1, X_2\,\right] \left[\,Z_2 \,|\, X_1, L_1\,\right] \left[\,Z_3 \,|\, L_1, X_2\,\right] \left[\,X_2\,\right] dx_2 dx_1\end{aligned} \tag{4.92}$$

By traversing the tree from the leaves to the root and passing up each computed partial belief, reaching the root which by definition contains no separator variables

is equivalent to having enough information to compute the full joint posterior. This distinction in wording is important as in practice the algorithm never specifically computes the full joint. Rather, for a SLAM formulation, we are interested in the estimate at each variable. Information can then be propagated back down the tree to compute full posterior marginals for each frontal variable.

Through the factorization represented by the Bayes Tree, the problem of computing the joint posterior distribution is divided by clique into problems of lower dimension, and the sum-product algorithm sketched out earlier defines the belief messages that are passed on the Bayes Tree. If Gaussian likelihoods are assumed, per **iSAM2**, the max-product and sum-product approaches are identical. Belief can be propagated as max-products and computed using linear least-squares to obtain a very fast solution. Furthermore, as we note later, the structure of the Bayes Tree can also be exploited for incremental inference each time the factor graph is updated.

In order to compute inference over non-Gaussian distributions, we now define how belief messages can be constructed based on the general sum-product approach. First we derive the messages to be passed along the tree and the general belief propagation algorithm that generates the marginal probabilities of every variable. Then in Section 4.3.2 we describe how these messages are stochastically approximated by the mm-iSAM implementation of approximate density sums and products.

With a slight abuse of notation, let Z_i' be the collection of measurements of clique i and all of its children, and denote any priors contained within as P_i. Therefore, the root $i = 1$ has $Z_1' = \mathbf{Z}$, the collection of all measurements, and each leaf j has $Z_j' = Z_j$, the collection of measurements the leaf is conditioned on. The message that is passed up from each clique i is:

$$m_i := \left[\Theta_{(\mathcal{S},i)} \,|\, Z_i' \right] \tag{4.93}$$

At any leaf clique L, this is equivalent to:

$$m_L := \left[\Theta_{(\mathcal{S},i)} \mid Z_L \right] = \int_{\mathcal{D}_L} \left[\Theta_{(\mathcal{C},L)} \mid Z_L \right] d\Theta_{(\mathcal{F},L)}$$
$$\propto \int_{\mathcal{D}_L} \prod \left[Z_L \mid \Theta_{(\mathcal{C},L)} \right] \prod P_L \, d\Theta_{(\mathcal{F},L)} \tag{4.94}$$

Subsequent messages passed from any clique that is not a leaf also include m_c, all messages that are passed up from their children. The Bayes tree structure guarantees that parent nodes are not conditioned on the frontal variables of their children, i.e. frontal variables never travel up the tree. Marginalizing local frontal variables out of messages passed up reduces the necessary complexity of the density product at each clique.

$$m_i := \left[\Theta_{(\mathcal{S},i)} \mid Z_i' \right] = \int_{\mathcal{D}_i} \left[\Theta_{(\mathcal{C},L)} \mid Z_i' \right] \, d\Theta_{(\mathcal{F},L)}$$
$$\propto \int_{\mathcal{D}_i} \prod \left[Z_i \mid \Theta_{(\mathcal{C},i)} \right] \prod P_i \prod m_c \, d\Theta_{(\mathcal{F},L)} \tag{4.95}$$

Eqns. 4.94 and 4.95 are equivalent to the Chapman-Kolmogorov equation. Similarly, to recover full marginal densities over every frontal variable, belief messages are then passed from parent to each child back down the tree, starting from the root. The down-message from every parent is the belief over the separator variables of the child. As in the sum-product approach on the general factor graph, the mm-iSAM algorithm concludes when two messages have been passed on each edge. If full marginals on variables lower in the tree are not required by the implementation, down-message passing can be terminated early.

The factor graph for a small marine range-only example as shown in Fig. 4-23, and its corresponding Bayes Tree in Fig. 4-24. Nine variables $x_1 \cdots x_9$ encode a vehicle's 2D position in time. l_1 is a beacon variable whose 2D location is considered unknown. Relative odometry factors connect each consecutive pose. The first and last pose in the chain have GPS priors, representing a vehicle periodically surfacing. Finally, the vehicle ranges the beacon three times, providing relative range measurements on

x_1, x_4 and x_7. These factors are provided in Section 4.4.3.

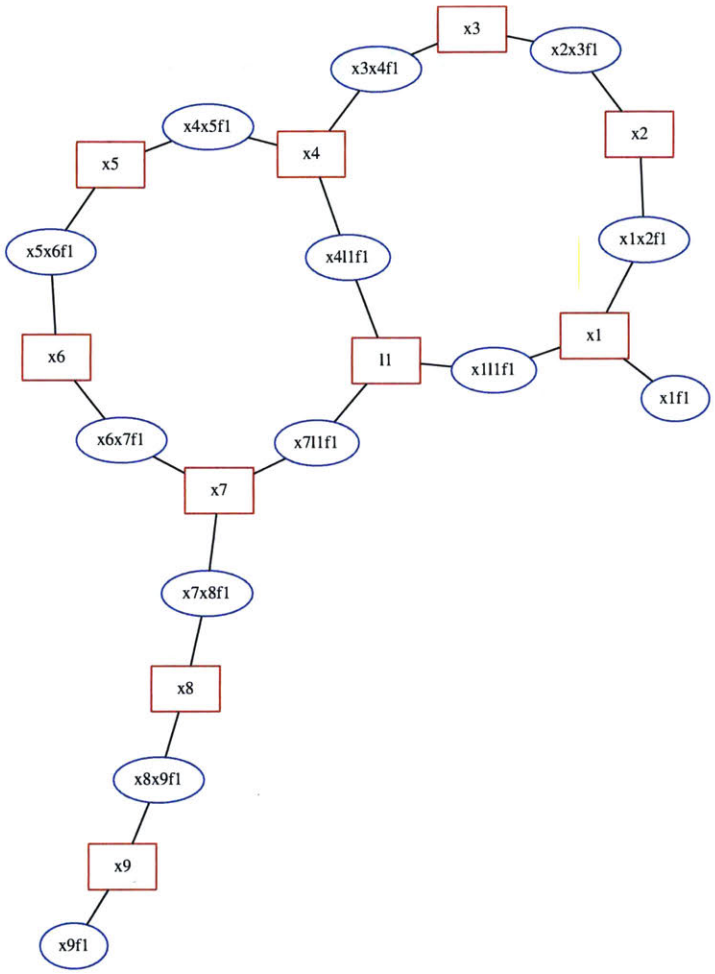

Figure 4-23: Example factor graph for a marine Range-only SLAM formulation with ten variables shown in red. Factors are shown in blue. Each subsequent pose is linked by an odometry factor. Priors are provided at the start and end of the chain (typical of a periodically surfacing vehicle). Three range factors on x_1, x_4 and x_7 respectively link the vehicle poses to the beacon, whose position is considered unknown.

Fig. 4-25 shows the association matrices for up-message generation from each clique, representing the sources of information available to each clique. For example, the clique with frontal variable x_1 processes a relative odometry factor ($[Z \mid x_1, x_2]$) (row 1), a prior on itself (row 2), and an up-message from its child node over x_4 and x_7.

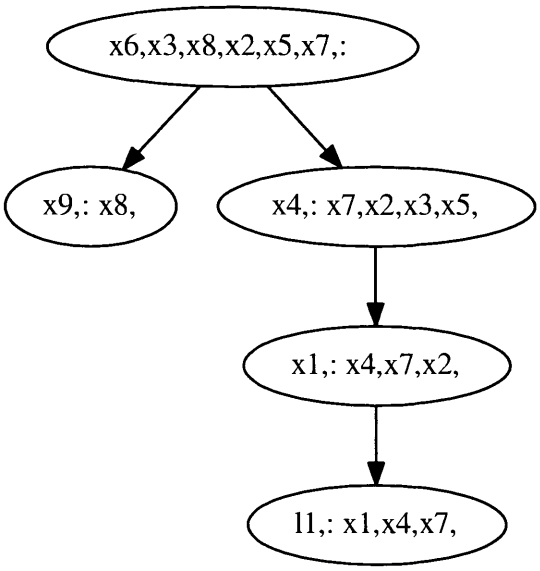

Figure 4-24: Bayes Tree generated for the factor graph shown in Fig. 4-23. Variables present only in the odometry chain appear at the root of the tree. The left branch forms due to the presence of a prior on x_9. The right branch encodes the information provided by the range factors and the prior on x_1. Since the variable ordering has pushed the unknown beacon to the leaf, its up-message is expected to be uninformative.

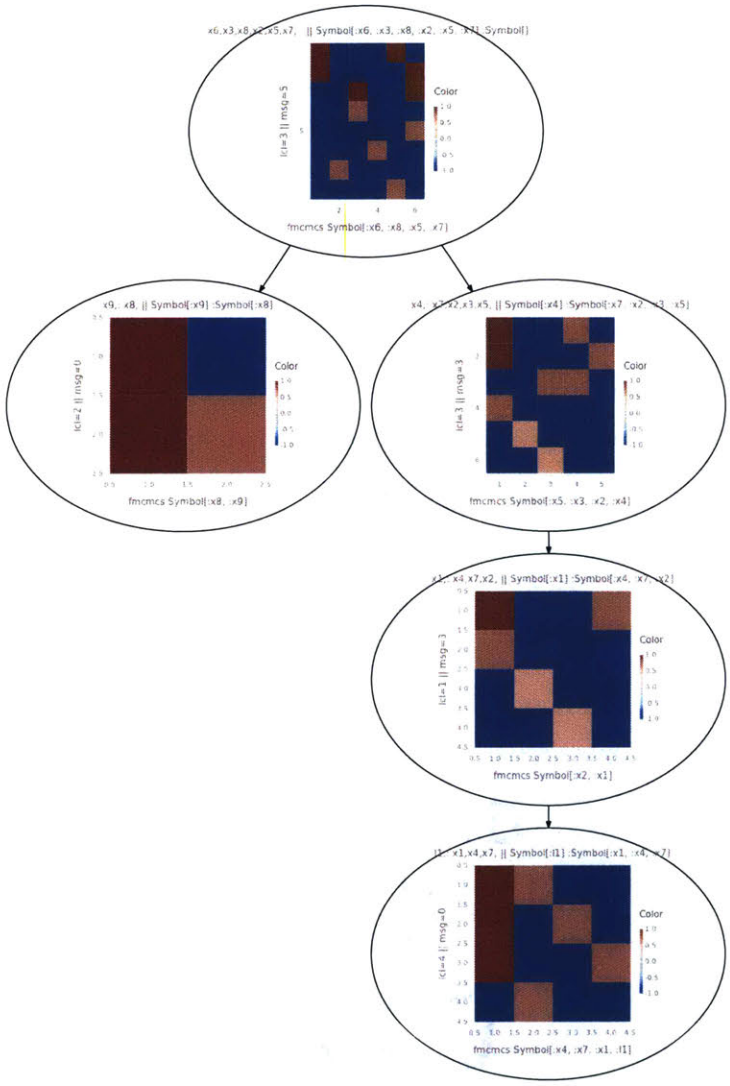

Figure 4-25: Bayes Tree for factor graph in Fig. 4-23 showing association matrices for up-message generation. Columns are labeled by variable, rows are labeled by factor. No association is shown in blue. Frontal variables, separator variables and children up-messages are shown in different intensities of red. The association matrix in the leaf node indicates the are three range factors between the beacon and its separator variables, as well as a prior on x_1 shown in the last row. In its parent clique, the first row indicates the odometry factor between x_1, x_2, the second is the prior on x_1 and the last two are up-messages propagating belief on x_4, x_7 up the tree.

Since the position of beacon l_1 is considered unknown, its up message to its parent is a wide distribution over position, as shown in Fig. 4-26. While this message contributes very little useful information to the state of x_1 in its parent, via the sum-product approach, it does not negatively affect the estimate as probability density is concentrated by products against other factors. The final distribution over the vehicle's poses are shown far right in Fig. 4-26.

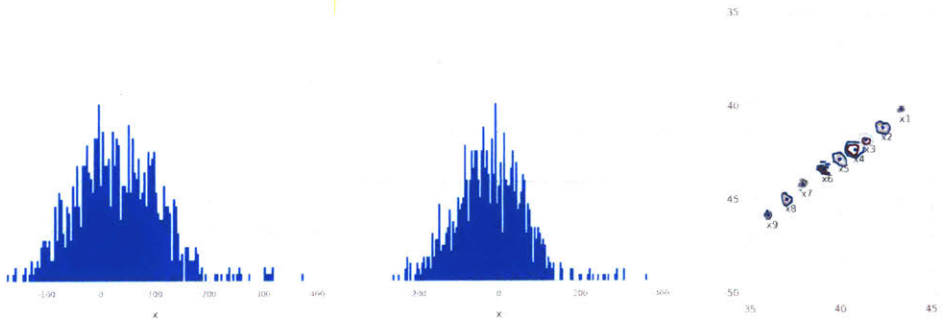

Figure 4-26: (*Left, Middle:*) Up-message from leaf clique in Fig. 4-24 expressing belief over the position of variable x_4, expressed as histograms in dimensions x, y (m). The uninformative distribution is a consequence of the beacon's unknown location. (*Right:*) In a sum-product approach, the tight accuracy over the final estimate over each vehicle pose is unaffected.

A consequence of the structure of the Bayes Tree, in which information only travels up the tree, comprising the full joint posterior in the root, is the computational savings to be gained from incremental inference. That is, if the factor graph is updated with new variables and factors incrementally to the old structure, cliques that remain unchanged can be re-used. Since frontal variables do not travel up the tree, pre-computed up-messages from cliques that are siblings to any cliques being added or changed, can be reused in the updated Bayes Tree. Fig. 4-28 illustrates this, where the clique shaded in grey remains unchanged if variable x_4 and factors z_5, z_6 are added to the example in Fig. 4-19.

In practice, mm-iSAM uses a functional state machine to step through operations that happen at an inter-clique level. Additional steps of implementation are required to perform initialization on the Bayes Tree. The initialization algorithm chooses frontal variables to initialize heuristically, Variables with high inferred dimensionality

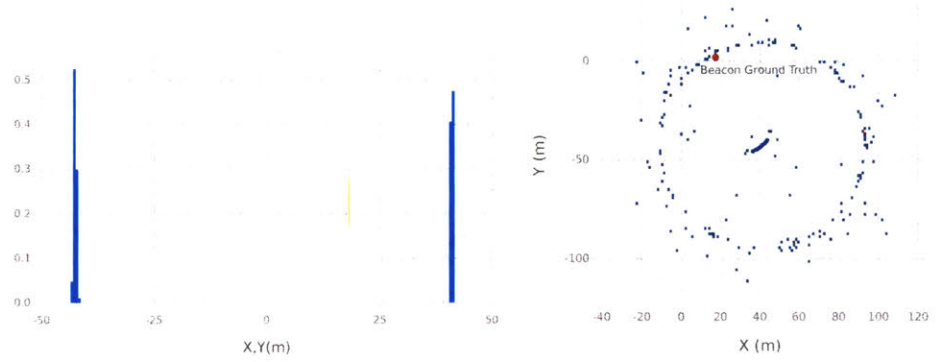

Figure 4-27: (*Left:*) Down-message from clique with frontal variable x_1, from separator variable x_4 towards beacon l_1. (*Left:*) Marginal posterior distribution over the location of x_4 after down-solve. On down-solve step, the tightly estimated location of x_4 now informs an accurate marginal posterior over l_1. (*Right:*) X-Y 2D-Histogram showing distribution generated by approximate convolution over the range factor $[\, Z \,|\, x_4, l_1 \,]$ towards beacon l_1. This fully non-Gaussian distribution is captured by the mm-iSAM solver as a KDE.

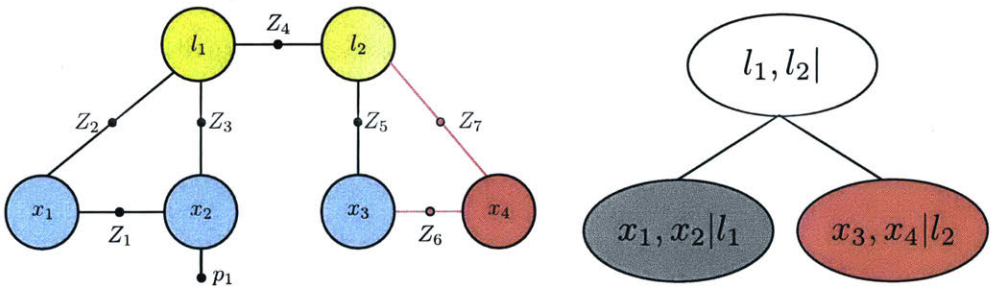

Figure 4-28: (*Left:*) Incrementally updated factor graph adding variable x_4 and factors Z_6, Z_7, shown in red. (*Right:*) Bayes Tree generated using variable ordering $l_1, l_2, x_1, x_3, x_4, x_2$. The clique to be updated is shown in red, while its sibling clique shown in gray can be reused.

of information over their posteriors are preferred, and variables that are deeper in the tree are preferred. Once a variable is initialized, partial up or down messages are passed to allow neighboring cliques to initialize as well. In summary, this section detailed a belief propagation algorithm on the Bayes Tree, an acylic graphical model with special structural properties. Subsequently, we complete the summary of the mm-iSAM algorithm with a discussion of the process by which the density sums and products that are necessary to construct the belief messages passed on the tree can be approximated.

4.3.2 Approximate Density Products

This section summarizes the current mm-iSAM strategy for representing and operating on non-Gaussian likelihoods. Generating clique up-messages (Eqns. 4.94, 4.95), i.e. computing the Chapman-Kolmogorov integral, requires evaluating the product between two non-Gaussian densities in the same function space.

Density estimation falls into three primary categories, parametric, semi-parametric and non-parametric [68]. Parametric estimation is applicable if the underlying distribution is known or simple to model. Semi-parametric methods such as mixture models are more flexible than the former, and non-parametric models assume the least about the structure of the modeled distribution. While non-parametric models stand to provide the most accurate representation of an arbitrary distribution, there is a significant computational cost to take into account.

mm-iSAM uses Kernel Density Estimation (KDE) to approximate non-Gaussian beliefs. Consider the problem of modeling the density of a set of sample points Θ, for example, the output of the matched filter which represents the probability of the beacon being at a certain range. KDE is a widely used non-parametric density estimation method that represents the density of a particular point as a weighted sum of kernels centered on each sample representing that distribution.

$$\left[\hat{\Theta} \mid \cdot\right] = \frac{1}{N}\sum_{j=1}^{N} w_{i,j} K\left(\Lambda_{i,j}\|\hat{\theta} - \theta_i\|\right) \tag{4.96}$$

where $w_{i,j}$ is a weighting parameter, K is a kernel function centered on the sample point Θ_j and $\Lambda_{i,j}$ is a bandwidth parameter that scales the kernel. Specifically, mm-iSAM uses N evenly weighted Gaussian kernels, with leave-one-out likelihood cross validation for bandwidth selection Λ_i [154].

$$\left[\hat{\Theta} \,|\, \cdot\right] = \frac{1}{N} \sum_{j=1}^{N} w_{i,j} \mathcal{N}\left(\hat{\theta}\,;\, \mu = \Theta_i \sigma = \Lambda_{i,j}\right) \tag{4.97}$$

There are many possible strategies to selecting appropriate kernels, weights and bandwidths. In the interest of reducing computational costs, the approximation is limited by the following simplifications. The number of kernels is constant for each distribution. Each sample is evenly weighted ($w_{i,j} = w_j$), and share common bandwidth parameters ($\Lambda_{i,j} = \Lambda_j$). Bandwidths are instead tracked across individual dimensions, i.e. a two-dimensional sample (x, y) is represented by bandwidths $\Lambda_{j,x}, \Lambda_{j,y}$.

Each hypothesis in a multi-modal distribution is tracked by a Gaussian distribution. When taking the product of two multi-modal distributions, the resulting number of modes may exponentially increase, but the number of kernels used to represent the product remains constant. This trade-off tracks the N most dominant modes of the distribution, with approximation accuracy increasing with N. In Chapters 4 and 5, distributions represented by KDEs are plotted as histograms of samples drawn from that distribution and contour plots enclosing the probability quantiles computed based on the number of contours.

The remaining step is to compute the Chapman-Kolmogorov integral on a distribution represented by the KDE of a given set of sample particles, which consists of two general iterative steps. The first is sampling the conditional distribution over every variable in the clique (Eqn. 4.95) through approximate convolution of the measurement likelihood residuals, any priors, and any incoming up-messages. The second is propagating those samples in such a way that their distribution reflects the true conditional posterior. When done stochastically with Markov chain transitions, this is an MCMC procedure.

Gibbs sampling is an Markov Chain Monte Carlo (MCMC) technique that draws

samples from the approximate conditional posterior distribution through a stochastic iterative process [64]. If the desired posterior distribution can be represented by the stationary distribution of a Markov chain, then iterative sampling on that Markov chain will converge to samples from that distribution. Consider the following first-order Markov chain on a sequence of random samples $\boldsymbol{\Theta}^1, \boldsymbol{\Theta}^2, \cdots \boldsymbol{\Theta}^t$, represented by the transition probability:

$$\left[\boldsymbol{\Theta}^t = \theta^t \,|\, \Theta^{t-1} = \theta^{t-1}, \cdots, \Theta^1 = \theta^1 \right] = \left[\boldsymbol{\Theta}^t \,|\, \Theta^{t-1} = \theta^{t-1} \right] \tag{4.98}$$

Let the state transition probabilities be the conditional distributions of each variable within a clique. Let $\Theta_{j,k}$ be the jth sample of the kth variable of a clique, where $k \in 1 \cdots K$, $j \in 1 \cdots N$. $\mathbf{Z}'$ indicates all measurements and up-messages passed to that clique. Samples are drawn and propagated in batches according to a Gibbs sequence. At each iteration step i, N samples are drawn for each variable according to the distributions:

$$\begin{aligned}
\{\Theta_{1,1} \cdots \Theta_{N,1}\}^i &\sim \left[\hat{\Theta}_1 \,|\, \hat{\Theta}_2^{i-1}, \cdots, \hat{\Theta}_K^{i-1} \mathbf{Z}' \right] \\
\{\Theta_{1,2} \cdots \Theta_{N,2}\}^i &\sim \left[\hat{\Theta}_2 \,|\, \hat{\Theta}_1^{i}, \hat{\Theta}_3^{i-1}, \cdots, \hat{\Theta}_K^{i-1} \mathbf{Z}' \right] \\
\{\Theta_{1,3} \cdots \Theta_{N,3}\}^i &\sim \left[\hat{\Theta}_3 \,|\, \hat{\Theta}_1^{i}, \hat{\Theta}_2^{i}, \hat{\Theta}_4^{i-1}, \cdots, \hat{\Theta}_K^{i-1} \mathbf{Z}' \right] \\
&\vdots \\
\{\Theta_{1,K} \cdots \Theta_{N,K}\}^i &\sim \left[\hat{\Theta}_K \,|\, \hat{\Theta}_1^{i}, \cdots, \hat{\Theta}_{K-1}^{i}, \mathbf{Z}' \right]
\end{aligned} \tag{4.99}$$

As each variable is iterated over, its conditional belief estimate is updated and used for the next sample. After sufficient iteration, the MCMC procedure eventually propagates each vector of samples $\Theta_{j,k}$ towards a set of samples drawn from the Markov chain's stationary distribution. Samples are drawn by solving for the roots of each measurement residual. Consider a measurement function $Z_{1,2}$ of which $\Theta_{j,1}, \Theta_{j,2}$ are arguments. The conditional prior belief estimate over Θ_2^{i-1} computed during a previous iteration of Gibbs sampling is:

$$\left[\Theta_2^{i-1}\right] = \text{KDE}(\{\Theta_{1,2}^{i-1}, \cdots, \Theta_{N,2}^{i-1}\}) \approx \left[\hat{\Theta}_2 \,|\, \hat{\Theta}_1^{i-1}, Z_{1,2}\right] \tag{4.100}$$

$$= \int_{\mathcal{D}} \left[\Theta_2 \,|\, \Theta_1, Z_{1,2}\right] \left[\hat{\Theta}_1^{i-1}\right] d\Theta_1 \tag{4.101}$$

Samples can then be drawn using a nonlinear trust-region solver to find $\Theta_{j,1}^i$ for which the measurement residual is satisfied.

$$\Theta_{j,1}^i = \underset{\Theta_{j,1}}{\text{solve}} \left[\delta_{1,2}\left(\Theta_{j,1}, \Theta_{j,2}^{i-1}, n_{1,2}\right) = 0\right] \tag{4.102}$$

where $\delta_{1,2}$ is the measurement residual and $n_{1,2}$ is its associated noise model. The roots of the equation exist if the set of equations defined by Eqn. 4.103 is full-dimensioned, for example, an odometry factor with additive white Gaussian noise on every Cartesian dimension. If the measurement is under-constrained, for example a range factor fixing one dimension less than a full Cartesian pose, the root-finding algorithm is replaced with a nonlinear minimization that performs gradient descent iterating the sampling procedure over one dimension at a time. The optimizer then draws samples according to:

$$\Theta_{j,1}^i = \underset{\Theta_{j,1}}{\text{argmin}} \| \left[\delta_{1,2}\left(\Theta_{j,1}, \Theta_{j,2}^{i-1}, n_{1,2}\right)\right] \| \tag{4.103}$$

The necessary conditions for the MCMC method to converge to the true conditional posterior distribution are that the Markov chain represented by Gibbs sampling the measurements residuals given to the factor graph is ergodic, has a stationary distribution and is reversible. These conditions are satisfied if the Markov chain satisfies the detailed balance criterion (Kolmogorov's Criterion):

$$\left[\hat{\Theta}_i\right] \left[\Theta_j \,|\, \Theta_i, Z_{i,j}\right] = \left[\hat{\Theta}_j\right] \left[\Theta_i \,|\, \Theta_j, Z_{i,j}\right] \tag{4.104}$$

Factors designed for the non-Gaussian mm-iSAM algorithm must define measurement residuals that satisfy this criterion for the estimated marginals to converge to the true distributions. Additional proof of convergence is discussed in [58]. Finally,

the product between two kernel density estimates is computed through a multi-scale Gibbs sampling method described in [81]. Assuming each likelihood is represented by a kernel density estimate with N Gaussian kernels, the algorithm draws samples from the product of $k = 1 \cdots K$ kernel density estimates with mean and variance μ_k, Λ_k by computing the weights and parameters of the resulting product:

$$w_i = \frac{\prod_{k=1}^{K} w_l \mathcal{N}(\theta_j;\ \mu_k, \Lambda_k)}{\mathcal{N}(\theta_j;\ \mu_i, \Lambda_i)} \tag{4.105}$$

$$\Lambda_i = \sum_{k=1}^{K} \Lambda_k \tag{4.106}$$

$$\Lambda_i \mu_i = \sum_{k=1}^{K} \Lambda_k m u_k \tag{4.107}$$

If the exact product of K mixtures is computed, then the number of components scales exponentially as N^d. Instead, the product can be efficiently computed using a Gibbs sampling strategy to select the individual components to be sampled from. This individual component of the product mixture is then used to sample from the posterior product, and following earlier discussion on Gibbs sampling, converges asymptotically towards samples from the true product distribution after iteration. Let each component be labeled l_k for $k \in 1 \cdots K$. As an initialization step, the Gibbs sampling algorithm randomly selects $K-1$ labels, then subsequent labels are selected by iterating:

$$l_1^i \sim \left[l_1 \,|\, l_2^{i-1}, \cdots, l_K^{i-1} \right] = \text{Cat}\left[\rho_1, \rho_2, \cdots, \rho_N \right]^{i-1} \tag{4.108}$$

$$l_2^i \sim \left[l_2 \,|\, l_1^i, l_3^{i-1}, \cdots, l_K^{i-1} \right] = \text{Cat}\left[\rho_1, \rho_2, \cdots, \rho_N \right] \tag{4.109}$$

$$\vdots$$

where the sampling weights are computed by normalized weight the mean of each component N against the quasi-product of $k-1$ components, as in Eqn. 4.105.

While the strategy of approximate Gibbs sampling in mm-iSAM allows for highly

non-Gaussian measurement models in the factor graph, incorrect modeling of the measurement process would still yield an inconsistent system and result in unpredictable behavior - therefore providing an accurate measurement model remains critical.

4.4 SAS-Factor Formulation

In this section, we arrive at the 'payoff' that the non-Gaussian SLAM mm-iSAM solver offers to an underwater multi-vehicle target-tracking mission, the formulation of a factor over non-Gaussian, multi-modal measurement likelihoods. We incorporate a full synthetic aperture sonar procedure from raw acoustic waveform recordings into a SAS-factor that can then be inferred over as part of a larger factor graph containing a variety of other sensor data. Fig. 4-29 shows the multi-variable SAS-factor connecting every element of the synthetic array (that is, poses that may be generated by a vehicle in time, or multiple vehicles simultaneously), with the acoustic beacon l_1 and with other odometry factors, range factors, GPS priors, etc.

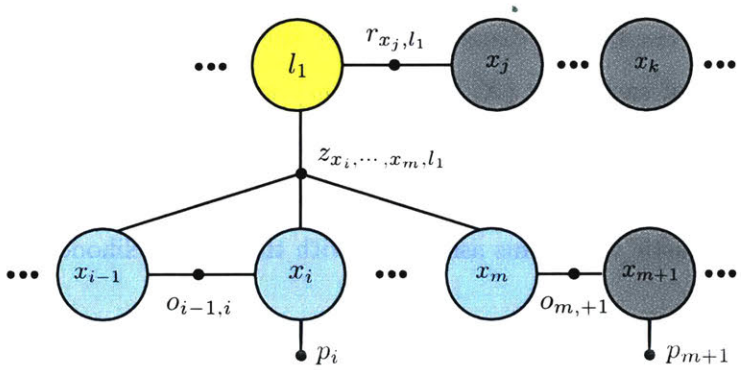

Figure 4-29: Illustration of a SAS-factor $z_{x_1,\cdots,x_m,l_1}$ between m vehicle poses and the landmark l_1 used in combination with other variables and factors in a factor graph representation. Each consecutive vehicle pose x_i corresponds to a single element of the synthetic array and contains the associated acoustic waveform measurement. The residual defined by $z_{x_1,\cdots,x_m,l_1}$ is fully non-Gaussian.

The SAS-factor describes a SAS-measurement residual function for the probabilistic convolution of factors with current belief estimates of the dependent variables. In order to satisfy the detailed balance criterion (Eqn. 4.104), the likelihoods must be reversible, that is, providing a sample to any argument of the factor should inform

the other arguments. We define the 'forward' projection (akin to conventional SAS) as using receiver positions and recorded waveforms to resolve a likelihood over the position of the acoustic source. Similarly, we define the 'backward' projection to estimate a single receiver position from the beacon's position and all other neighboring poses. Note that this is not conventional inverse beamforming but a leave-one-out cross-correlation procedure over every sample pose. The naming scheme reflects the reversible operation taken by the solver during the Gibbs sampling step, depending on which argument variable of the factor was selected to be sampled, and does not imply sequential evaluation.

4.4.1 Forward Convolution

'Forward' probabilistic convolution is computed 'towards the beacon L', i.e. to compute a marginal conditional distribution over the position of L from the current marginal beliefs of each pose $\{X_i \ \forall i \in 1 \cdots m\}$ where m is the length of the synthetic array. In order to compute the approximate convolution via Gibbs sampling (Eqn. 4.103), a residual function $\delta\left(\mathbf{L}_1, \mathbf{X}_1, \cdots, \mathbf{X}_m; \mathbf{Z}_1\right)$ is defined where $\mathbf{Z}_1$ represents the recorded acoustic waveforms associated with the SAS likelihood factor.

$$\left[\hat{\mathbf{L}} \,|\, \mathbf{X}_1, \cdots, \mathbf{X}_m, \mathbf{Z}_1, \cdot\right] \propto \left[\mathbf{Z}_1 \,|\, \mathbf{L}, \mathbf{X}_1, \cdots, \mathbf{X}_m\right] \prod_{j=1}^{m} \left[\hat{\mathbf{X}}_j \,|\, \cdot\right] \tag{4.110}$$

Conventional SAS beamforming quantifies the distribution in received acoustic energy over azimuths for a series of vehicle poses and acoustic measurements. For the forward case, the beam pattern produced by the CBF over azimuth angle $[0, 2\pi)$ is converted into a representative probability density $\left[\tilde{\Theta} \,|\, \mathbf{X} = \mathbf{x}_1, \cdots, \mathbf{X}_m; \mathbf{Z} = \mathbf{z}_1\right]$ after a process of noise floor truncation and re-normalization. This is illustrated in Fig. 4-34, and corresponds to the distribution represented by an alias sampling procedure described in Algorithm 2. The output power of the beamformer averaged

over all frequencies $\vec{\omega}$, and the resulting distribution computed by:

$$Y[\omega_n, \vartheta] = \sum_{i=1}^{m} H_i[\omega_n, \vartheta] \cdot F_i[\omega_n] \tag{4.111}$$

$$\text{where } H_i[\omega_n, \vartheta] = e^{j\vec{\omega_n}\tau_i} \tag{4.112}$$

$$\tau_i = -\frac{\vec{u}^T \vec{p}_{t,i}}{c} \qquad i \in [0, m] \tag{4.113}$$

$$\|\widetilde{Y}[\vartheta]\| = \frac{1}{N} \sum_{n=1}^{N} \|Y[\omega_n; \vartheta]\| \tag{4.114}$$

$$\left[\tilde{\Theta} \,|\, \cdot \right] = \frac{1}{Q} \int_{\mathcal{D}} \|\widetilde{Y}[\vartheta]\| \, d\vartheta \tag{4.115}$$

such that Q is a hypothetical partition function that normalizes the beamformer output so that it sums to 1. This density is re-calculated each time a unique set of poses is sampled, i.e. each Gibbs sampling step. Strictly speaking the beam pattern also includes reproducible structure dependent on the environment and physical setup which can be deconvolved from the calculated beam pattern [170]; that process of channel-modeling and calibration is not in the scope of this work. The model in Eq. (4.115) can then be used to sample azimuth angles $\phi^{\{n\}}$ for numerical approximation of the convolution:

$$\theta^{\{n\}} \sim \left[\Theta \,|\, \mathbf{X} = \mathbf{x}_1, \cdots, \mathbf{X}_m, \mathbf{Z} = \mathbf{z}_1 \right]. \tag{4.116}$$

For both time-synchronized one-way, or active SAS two-way time-of-flight operation, the range between beacon and transducer $r^{\{n\}}$ can be calculated via matched filtering, and the correlator output of the matched filter is used to construct an additional pseudo-probability density for the SAS-factor:

$$r^{\{n\}} \sim [\,\mathbf{P} \,|\, \mathbf{X} = \mathbf{X}_1, \cdots, \mathbf{X}_m, \mathbf{Z} = \mathbf{z}_1\,] \tag{4.117}$$

The combination of range and bearing samples then informs a bearing-range constraint between the SAS reference frame and beacon location:

$$\delta_F \left(\mathbf{x}_1^{\{n\}}, \cdots, \mathbf{X}_m^{\{n\}}, l_1^{\{n\}}; \theta^{\{n\}}, r^{\{n\}} \right) \tag{4.118}$$

Using numerical root-finding, and given the independence of the kernels in each of the pose marginal estimates, each of the pose values $\mu^{\{n\}}$ are used in conjunction with fresh samples $\theta^{\{n\}}, r^{\{n\}}$ to find the best value for each $l_1^{\{n\}}$, $\forall n \in [1, N]$

$$^{*}L_1^{\{n\}} = \underset{L_1}{\operatorname{argmin}} \| \left[\delta_F \left(\mathbf{x}_{1..m}^{\{n\}}, l_1^{\{n\}}; \theta^{\{n\}}, r^{\{n\}} \right) \right] \| \tag{4.119}$$

A synthetic example over one SAS-factor with ten elements is presented in Figs. 4-30 and 4-31. In this example the array elements are located between $[-4.5, 4.5\ (\text{m})]$ on the x-axis. Received waveforms are simulated from a beacon traveling in arc between azimuths of (2,5) radians at an SNR of 10dB. To demonstrate the marginal distribution produced over the unknown beacon location l_1, for this example each pose variable $x_1 \cdots x_{10}$ is given a tight prior over its position. The multi-variate construction of the SAS residual results in a Bayes Tree (not shown) with a single clique containing all variables as frontals.

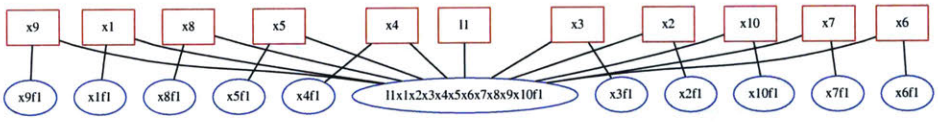

Figure 4-30: Example factor graph consisting of a single SAS-factor with ten receiver variables $x_1 \cdots x_m$ and a single beacon l_1 to be localized. Approximate convolution is computed over this factor graph for Figs. 4-31 to 4-39

Beam patterns and matched-filter range estimates are shown in Fig. 4-31 for the simulated data. The resulting beacon location and the direction of its travel is ambiguous about the central axis of the array. In a target-tracking application, the trajectory of the vehicle should be motivated towards resolving this ambiguity.

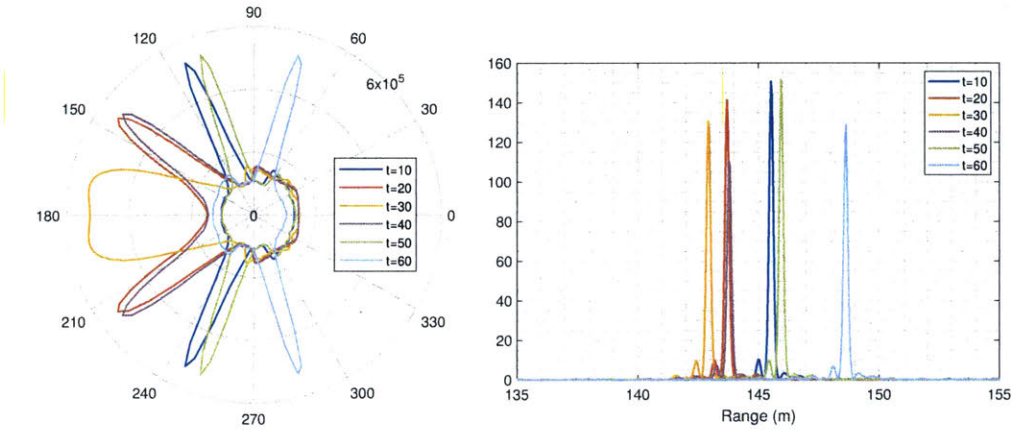

Figure 4-31: Azimuth and Range measurement distributions for Synthetic dataset consisting of a Moving Beacon with chirp received at 10-element array located on the x-axis. (*Left:*) Beam patterns over time as the source translates from a look-angle of 2 to 5 radians. The location and velocity of the source is ambiguous over the central axis of the array. (*Right:*) Matched Filter range estimates over time.

Fig. 4-32 illustrates the marginal posterior distribution generated by a single instance of forward convolution from the array elements (shown in blue) towards an unknown beacon at time steps $t = \{1, 3, 6\}$. A histogram of the samples $L_1^{\{n\}}$ from the distribution is plotted over a probability contour plot of the kernel density estimate, with each contour corresponding to 20% probability mass. The beam pattern from Fig. 4-31 is overlaid at the origin of the array for reference. The location samples are concentrated at the approximate convolution of the range estimate and the beamformer azimuthal estimate, resulting in an arc of samples around the multi-modal ambiguous peaks of the beam pattern. At time $t = 3$, endfire effects result in a much wider distribution over azimuth.

A single factor example using an arbitrary sample set of acoustic waveforms recorded during field experiments is shown in Fig. 4-33. The output beam pattern of the CBF for a single array of five elements is drawn in magenta at the array's first

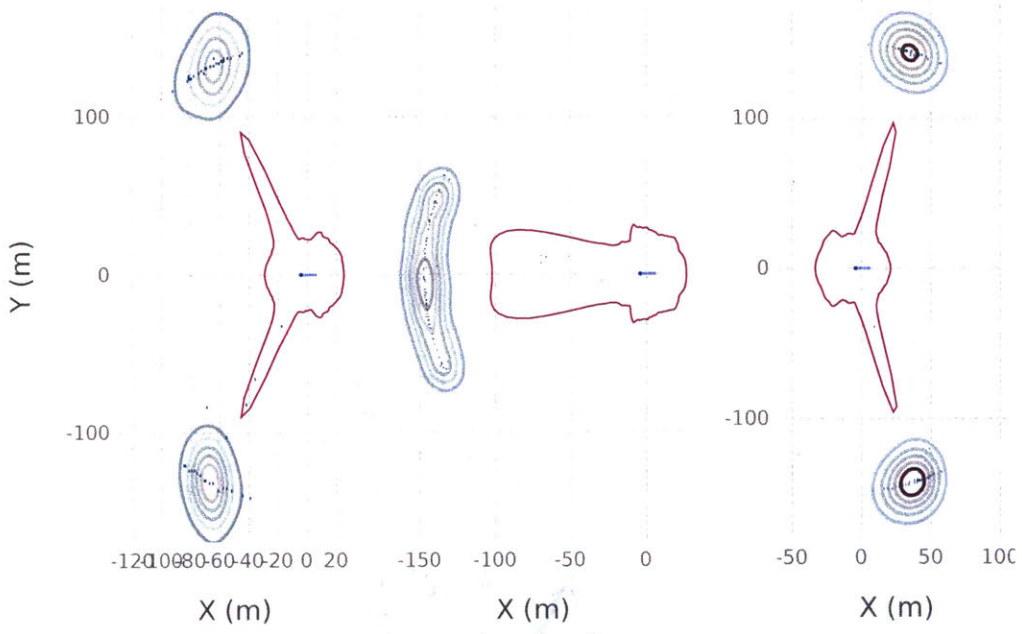

Figure 4-32: Marginal posterior probability densities over the beacon l_1 for Synthetic dataset, at times $t = \{1, 3, 6\}$. The look-angle translates from 2 to 5 radians over 6 time steps. Array elements shown in blue are located on the x-axis between (-4.5, 4.5)(m). The marginal posterior estimate for the beacon's x-y position is plotted as a histogram of samples and a KDE probability contour. Each contour contains 20% probability mass. A scaled CBF beam pattern is shown in magenta for reference.

pose arbitrarily chosen as its origin. The range and azimuthal probability distribution is sampled to form a proposed posterior distribution for the beacon's location, here shown as a histogram of sampled locations and a contour plot of the resulting kernel density estimate representation. As typical for a CBF, the likelihood is symmetric about the linear array. Here, the beacon is located at (17.0, 1.78). In this example, with an array of shorter length and noisier data, the distribution over range and azimuth is larger than in the synthetic dataset.

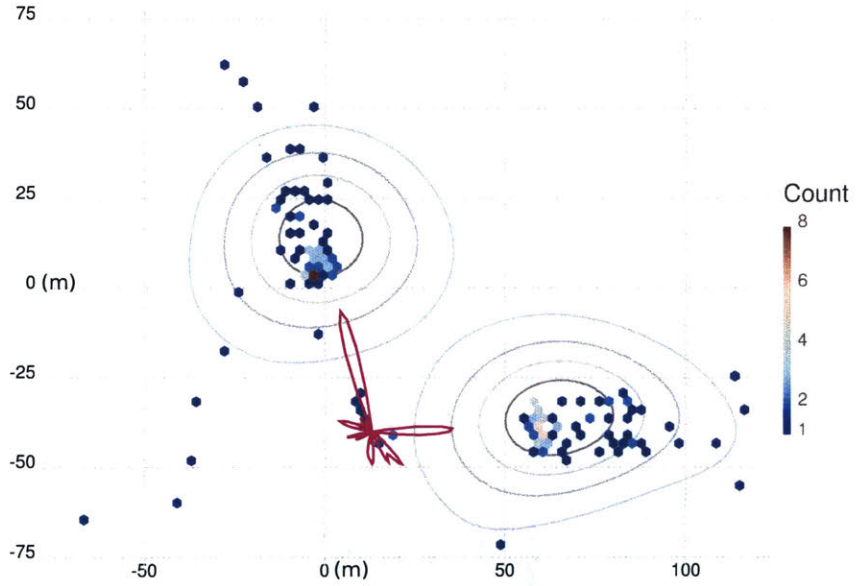

Figure 4-33: Marginal posterior probability density over the beacon l_1 for Field dataset of Stationary Beacon located at ground truth (17.0, 1.78) (m), after a single step of forward convolution over one SAS-factor, represented by a histogram of propagated samples and a KDE probability density contour plot. Each contour contains 20% probability mass. Without any prior information, the CBF resolves two equally likely modes symmetric about the axis of the linear array, resulting in a non-Gaussian distribution with two main modes.

4.4.2 'Backward' Convolution

'Backward' approximate convolution is the process by which the position of the beacon L_1 and the remaining $m-1$ elements of the array inform the position of the element currently being sampled. This is a result of the reversibility criterion required for the measurement residual likelihood to converge to the true marginal posterior. As the

process of Gibbs sampling iterates through each variable and subset of dimensions, the SAS residual iterates through a set of conditional probabilities consistent with the acoustic measurements. We note that this is not a forward-backward solve in the typical sense. The backward convolution is approximated by the mm-iSAM infrastructure using the factor residual function for all kernels $n \in [1, N]$ as in Eqn. 4.119, however, in the backwards case we directly optimize the correlation of one element at a time in a leave-one-out strategy:

$$ {}^{*}x_i^{\{n\}} = \underset{p_i}{\operatorname{argmin}} \| \left[\delta_F \left(\mathbf{x}_{j \in 1..m, j \neq i}^{\{n\}}, L_1^{\{n\}}; \mathbf{Z}_1 \right) \right] \| \tag{4.120} $$

A leave-one-out strategy is used to ensure pose estimates are calculated independently of their previous estimates, such that $\vec{x}_j, j \in 1...m \backslash i$. When solving for the leave-out element $\mathbf{x}_i$, the leave-in array element measurement waveforms are time shifted according to their positions $\left[\mathbf{X}_j = \hat{\mathbf{x}}_j^{\{n\}} \mid \cdot \right]$.

The origin of the synthetic array is arbitrarily chosen as one of the leave-in elements $\mathbf{x}_O$, $O \in j$, and the estimated position of the beacon is $[\mathbf{L}_1 \mid \cdot]$. The relative azimuth between pose and beacon is computed from the current belief over the beacon's location and the origin. The conventional beamformer filter is applied to all of the leave-in elements, and combined with the proposed phase-shifted waveform. This produces a residual function as shown in Figs. 4-34 to 4-39, the largest peak of which is the position that most accurately resolves the current look-angle.

$$ \vartheta = \tan^{-1} \left(\frac{y_B - y_O}{x_L - x_O} \right) \tag{4.121} $$

$$ \hat{\tau}_i = -\frac{\vec{u}^T \hat{x}_i}{c} \tag{4.122} $$

$$ \tau_j = -\frac{\vec{u}^T \vec{x}_j}{c} \qquad \text{where: } \vec{u} = [\cos(\vartheta) \ \ \sin(\vartheta)]^T \tag{4.123} $$

$$ Y[\omega; \vartheta] = \frac{1}{m-1} \sum_{j \in (1,m) \backslash i} e^{j\vec{\omega}\hat{\tau}_j} \cdot F_j[\omega] + e^{j\vec{\omega}\hat{\tau}_i} \cdot F_i[\omega] \tag{4.124} $$

Algorithm 6: Leave-one-out CBF Measurement Residual

Data: N, the length of the recorded waveforms, m, the number of receiver elements, $\bar{p}$, proposed receiver poses, $\bar{p}_L$, proposed beacon's location, i the label of the leave-out element, O the label of the array origin, $\mathbf{Y}_{ij}$, associated waveforms of size $(m \times N)$, $\mathbf{H}$, the CBF filter pre-constructed from $\bar{p}$, $\bar{f}$ the vector of FFT frequencies of length N,

Result: δ, the measurement residual

Function `LIECorr(`$\mathbf{Y}_{ij}$`,`$\mathbf{H}$`):`

begin

$\mathbf{Y} \leftarrow (\mathbf{Y})_{ij}(\mathbf{H})_{ij}$

$\mathbf{Y} \leftarrow \sum_{n=1}^{N} \mathbf{Y}$

$\mathbf{Y} \leftarrow \dfrac{(\mathbf{Y})_{ij}}{(\|\mathbf{Y}\|)_{ij}}$

end

Function `LOOPhaseShift(`$\bar{p}_L, \bar{p}_O, \bar{p}_i, \bar{f}$`):`

begin

$\Delta\bar{p} = \bar{p}_L - \bar{p}_O$

$\theta = \tan^{-1}\left(\dfrac{\Delta y}{\Delta x}\right)$

$\vec{u} = [-\cos(\theta), -\sin(\theta)]$

$\tau = \dfrac{\left(\bar{p}_i^{\mathsf{T}}\vec{u}\right)}{c}$

$\mathbf{F} = \exp\left[(-j2\pi\tau)\,\bar{f}\right]_n$

$\mathbf{Y}_i \leftarrow \left(\mathbf{F}^H\right)_n (\mathbf{Y}_i)_n$

end

Function `compResidual(`$\bar{x},\ \bar{A}, \bar{F}, \bar{w}_k$`):`

begin

$\delta =$ `LIECorr(`$\mathbf{Y}_{ij}$`,`$\mathbf{H}$`)`

$\delta \leftarrow \delta +$ `LOOPhaseShift(`$\bar{p}_L, \bar{p}_O, \bar{p}_i, \bar{f}$`)`

$\delta \leftarrow -\|\delta\|$

end

return

The measurement residual computed by a leave-one-out procedure for a single synthetic array is illustrated in Figs. 4-34 to 4-39, on both synthetic and field data as in the previous section. The CBF beam patterns for each respective dataset are shown in Fig. 4-34. Although the measurement residuals are plotted against both deviation from ground truth position and azimuth to illustrate their overall structure, at each sampling step the mm-iSAM back-end solver batch-samples from the residual, holding all other variables constant (i.e. at a single azimuthal angle only). That is, a set of N samples describing the KDE of the beacon's position will each be propagated through the computation at one sampled look-angle. The resulting measurement residual is nonlinear and retains the ambiguities inherent to a CBF process with a linear array as shown below.

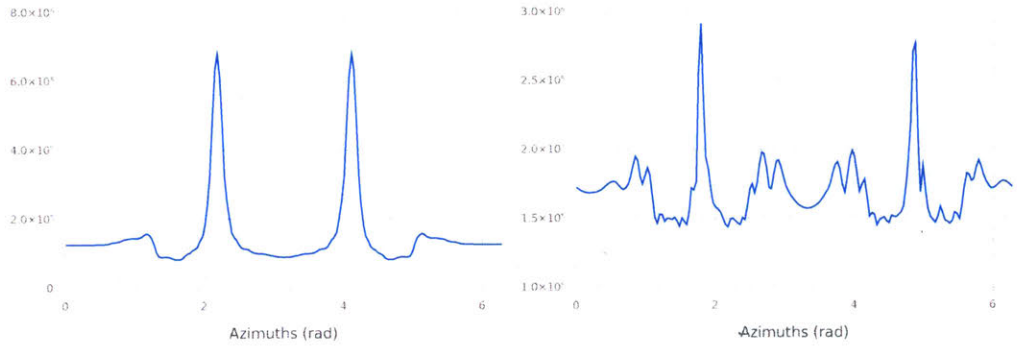

Figure 4-34: Beam patterns against azimuth angle (radians) for Synthetic and field datasets presented in Figs. 4-35 to 4-39. The distribution of SAS measurement residuals by azimuth is the leave-one-out CBF beam pattern.

Figs. Fig. 4-35 and Fig. 4-36 show the measurement residuals generated as the leave-out array element is perturbed along and perpendicular to the true azimuthal direction respectively. In Fig. 4-35, two sharp peaks are resolved at zero deviation from the ground truth position of the leave-out element. The phase-shifted signal correlates strongly with the CBF output when the array element is in its true location. We observe from the left view that the distribution along azimuth resolves the CBF beam pattern. In Fig. 4-36, the position of the leave-out element is perturbed along the direction perpendicular to the true azimuth. In this case, due to the far-field assumption, the phase-shift experienced by this element of the linear array remains

constant despite the deviation in position, and the residual does not distinguish a clear peak.

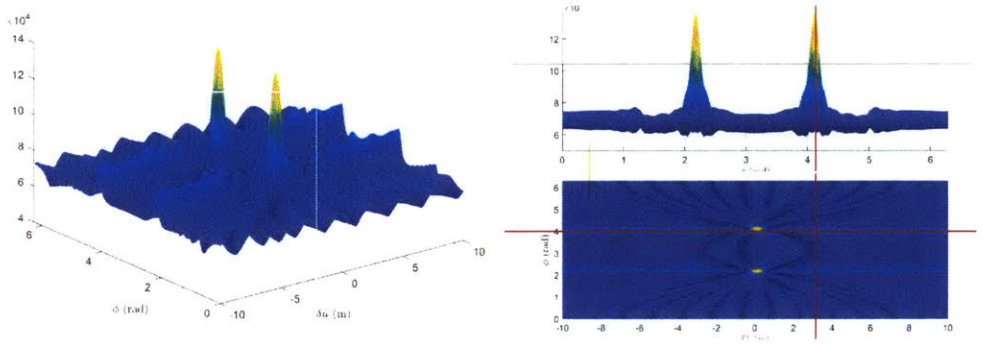

Figure 4-35: Backward Convolution correlation SAS measurement residual towards array element x_2, for a Synthetic dataset, where $\hat{u}$ is the unit vector pointing along the true look-angle. (*Top-Right*:) Distribution along azimuth angles returns the beam pattern of Fig. 4-34. (*Bottom-Right*:) Intensity map shows that the residual has a maximum centered around zero variation in position, which is the ground truth location of that array element. In practice, the residual function is implemented with a sign-change for the minimization algorithm.

Fig. 4-37 shows the sample histograms and probability density contours for the second element of the ten-element array along the x-axis after one step of backward convolution, at time $t = 1, 3, 5$. The average error is around 0.2m along the look-angle and up to 2m in the perpendicular direction. Notably, the case of endfire ($\theta = 180°$, middle plot) computes a tighter distribution around the receiver's position despite the forward convolution computing a wider distribution around the beacon's position.

The residual distribution for sample experimental data with a stationary beacon and an array of five elements is shown in Figs. 4-38 and 4-39. The residual is significantly noisier, but the same double peak pattern is observed for the along-azimuthal direction at the two modes of the CBF beam pattern, and wide envelope for the perpendicular direction.

Sampled solution locations for a single array element x_2 are shown in Fig. 4-40. GPS ground-truth for five synthetic array elements are shown as red crosses. The clear line of solutions perpendicular to the ray from array to beacon represents the far-field assumption, as moving linear array elements perpendicular to the correct

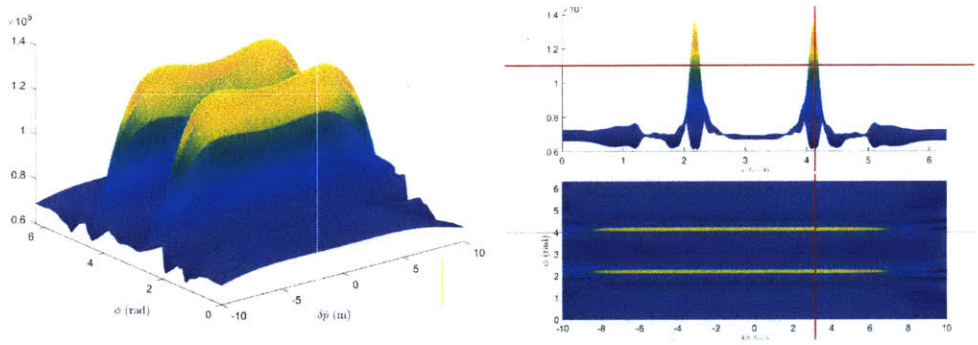

Figure 4-36: Backward Convolution measurement residual towards array element x_2, for a Synthetic dataset, where $\hat{p}$ is the unit vector pointing perpendicular to the true look-angle. (*Top-Right*:) Distribution along look angles returns the beam pattern of Fig. 4-34. (*Bottom-Right*:) Intensity map shows maximum of residuals does not distinguish variations in position along the angle perpendicular to the true look angle as the phase shifts applied to the waveform are unchanged.

look-angle does not change the phase-shift applied to the measurement waveform and its corresponding correlation.

We have shown here that the marginal distributions for the SAS-factor along range and azimuth are typically non-Gaussian and multi-modal. Traditional approaches to fusing TDOA and DOA measurements typically estimate over the full non-Gaussian distribution to track a maximum likelihood point measurement. The mm-iSAM factor graph formulation maintains each mode using a kernel density estimate, and the sum-product approach to belief propagation on the Bayes Tree ensure unlikely modes are stochastically eliminated during the approximate convolution.

4.4.3 Other Factors

This section presents the other variables and factors used in the full formulation of experimental factor graphs that will be described in Section 5. These are standard measurement residual functions implemented in Julia as part of the Robot Motion Estimate (RoME) package [61]. The measurement noise distributions are experiment-dependent and may vary.

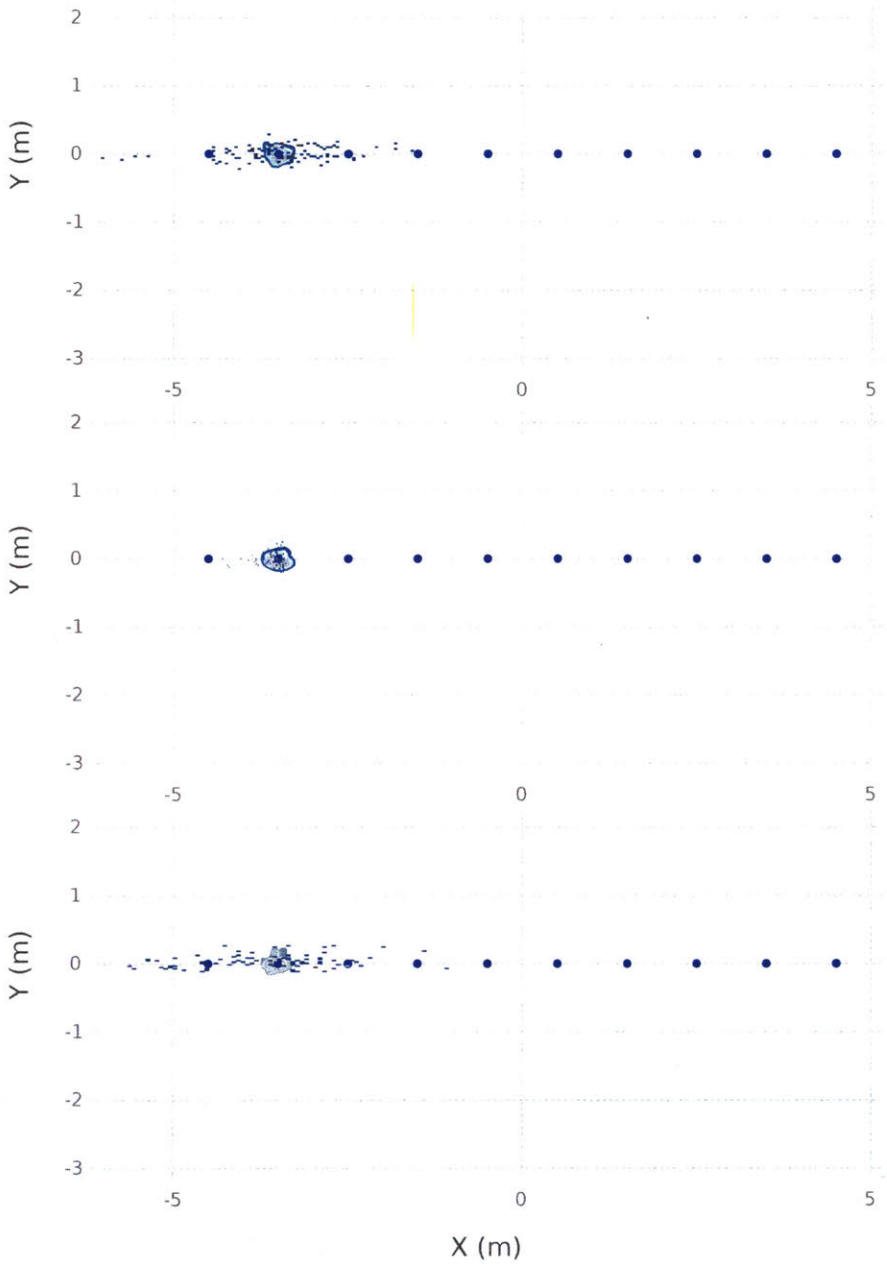

Figure 4-37: Marginal posterior distributions after a single step of backward convolution towards array element x_2, for a Synthetic dataset. The 10-element array is located on the x-axis, as shown in blue. The distribution is represented by a histogram of samples and a kernel density estimate probability contour for the beacon's x-y position. Most of the probability mass is concentrated at the right location, however the likelihood residual constructed from the linear array CBF does not resolve ambiguity along the direction perpendicular to the look-angle.

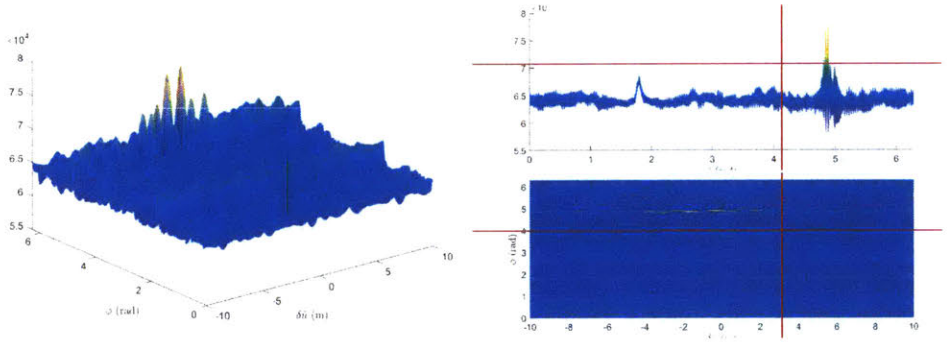

Figure 4-38: Backward Convolution measurement residual towards array element x_2, for a single sample from a field dataset with a stationary beacon located at ground truth (17.0, 1.78)(m). (*Top-Right*:) Distribution along look angles returns the beam pattern of Fig. 4-34. (*Bottom-Right*:) Intensity map does not resolve the peak as strongly in the presence of noise.

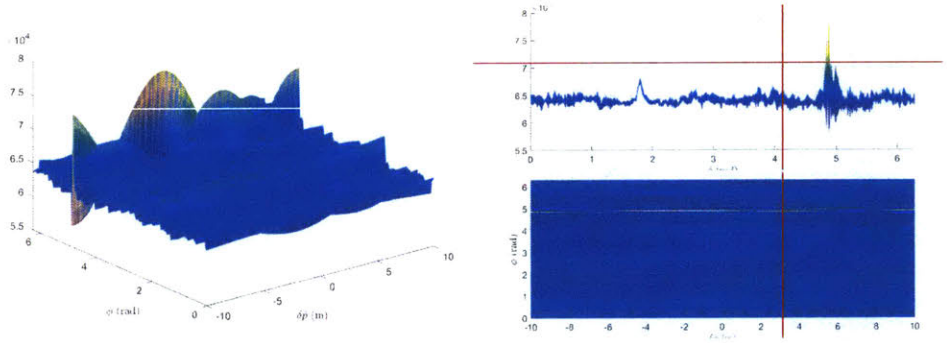

Figure 4-39: Backward Convolution measurement residual towards array element x_2, along the direction perpendicular to the look-angle. (*Top-Right*:) Distribution along look angles returns the beam pattern of Fig. 4-34. (*Bottom-Right*:) Intensity map.

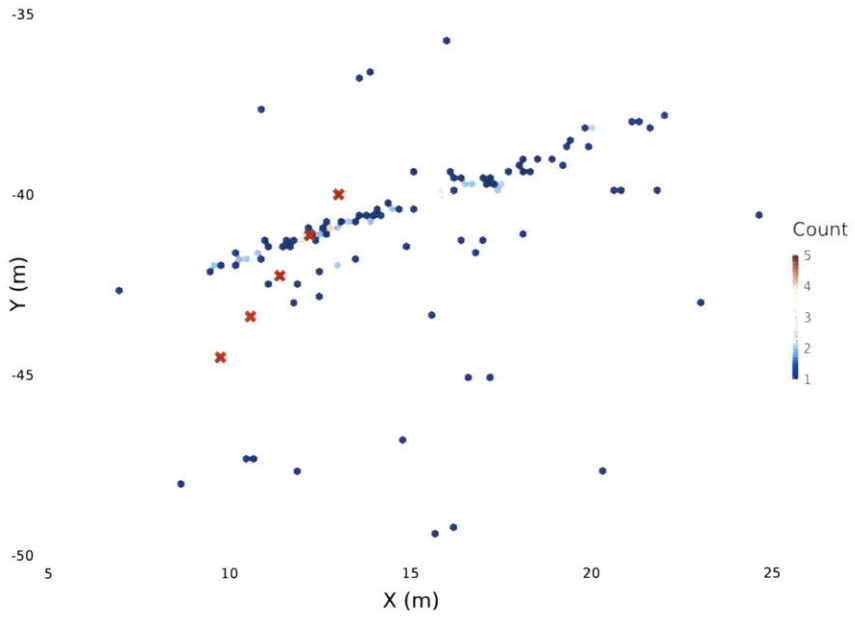

Figure 4-40: Example of coherence during backward convolution, on a field dataset. GPS ground-truth for five array elements are shown as red crosses. The stationary beacon is located at (17.0, 1.78)(m). Samples are drawn from the residual distribution and plotted as a 2D histogram. The estimated marginal location of the array element being sampled is ambiguous along the line perpendicular to the true look-angle.

Variables

Variables are represented by N samples of dimension m, where m is the dimension of the variable's domain $\mathcal{D}$.

(Point2) 2D Cartesian Point with 2 sampled dimensions $\boldsymbol{\Theta} = \{x, y\}$.

(Pose2) 2D Cartesian Pose with 3 sampled dimensions $\boldsymbol{\Theta} = \{x, y, \theta\}$.

(DynPoint2) 2D Dynamic Cartesian Point with 4 sampled dimensions $\boldsymbol{\Theta} = \{x, y, \dot{x} = v, \dot{y} = y\}$ and user-defined time index t, which is not sampled.

Factors

Factors are represented by the argument variables, the distribution of noise over the measurement and the measurement likelihood residual.

(PriorPoint2) Direct observation of a **Point2** variable. Samples are drawn directly from a distribution given over $\boldsymbol{\Theta} = \{x, y\}$. For example, a prior given by Θ_o is defined:

$$\begin{aligned} \mathbf{z} &\sim [\, \mathbf{Z}_i \,|\, \Theta_i = \theta_i \,] = \mathcal{N}\left(\mu = \Theta_o, \Sigma\right) \\ \Sigma &= \begin{bmatrix} \sigma_x & 0 \\ 0 & \sigma_y \end{bmatrix} \end{aligned} \tag{4.125}$$

(Point2Point2) Odometry measurement likelihood observing $\mathbf{Z} = \Delta\theta$ in **Point2** dimensions $\boldsymbol{\Theta}_i = \{x, y\}$. For example, with Gaussian noise:

$$\begin{aligned} \mathbf{z} &\sim [\, \mathbf{Z} \,|\, \Theta_i = \theta \,] = \mathcal{N}\left(\mu = \delta\left(\theta, \hat{\theta}_j, \mathbf{Z}^*\right), \Sigma\right) \\ \delta &= \Delta\theta - (\theta_i - \theta_j) \\ \Sigma &= \begin{bmatrix} \sigma_x & 0 \\ 0 & \sigma_y \end{bmatrix} \end{aligned} \tag{4.126}$$

(Point2Point2Range) Range measurement likelihood observing $\mathbf{Z} = r$, one inferred dimension over **Point2** dimensions $\mathbf{\Theta}_i = \{x, y\}$. For example, with Gaussian noise $\mathcal{N}$ over range r and uniform distribution $\mathcal{U}$ over bearing $\phi \in [0, 2\pi)$:

$$\begin{aligned} \mathbf{z} \sim [\,\mathbf{Z} \,|\, \Theta_i = \theta\,] = \mathcal{N}\left(\mu = \delta\left(\theta, \hat{\theta}_j, \mathbf{Z}^*\right), \Sigma\right) \\ \delta = \|\Theta_i - \left(r[\cos(\phi), \sin(\phi)]^\mathsf{T} + \theta\right)\| \\ \phi \sim \mathcal{U}\,(0, 2\pi) \end{aligned} \tag{4.127}$$

(PriorDynPoint2) Direct observation of a **DynPoint2** variable. Samples are drawn directly from the distribution. For example, a prior given by Θ_o is defined:

$$\mathbf{z} \sim [\,\mathbf{Z}_i \,|\, \Theta_i = \theta_i\,] = \mathcal{N}\left(\mu = \bar{X}_o, \Sigma\right) \tag{4.128}$$

$$\Sigma = \begin{bmatrix} \sigma_x & 0 & 0 & 0 \\ 0 & \sigma_y & 0 & 0 \\ 0 & 0 & \sigma_v & 0 \\ 0 & 0 & 0 & \sigma_w \end{bmatrix}$$

(DynPoint2DynPoint2) Odometry measurement likelihood observing $\mathbf{Z} = \Delta\theta$ in **DynPoint2** dimensions $\mathbf{\Theta}_i = \{x, y, v, w\}$, propagating velocity as a first-order system. For example, with Gaussian noise:

$$\mathbf{z} \sim [\,\mathbf{Z}_i \,|\, \Theta_i = \theta_i\,] = \mathcal{N}\left(\mu = \delta\left(\theta, \hat{\theta}_j, \mathbf{Z}^*\right), \Sigma\right) \tag{4.129}$$

$$\delta = (\mathbf{Z} - (\Theta_i - \Theta_j))^2 + \left(\frac{\Delta\bar{p}}{\Delta t} - \frac{1}{2}\left([v, w]_i + [v, w]_j\right)^2\right) \tag{4.130}$$

$$\Sigma = \begin{bmatrix} \sigma_x & 0 & 0 & 0 \\ 0 & \sigma_y & 0 & 0 \\ 0 & 0 & \sigma_v & 0 \\ 0 & 0 & 0 & \sigma_w \end{bmatrix}$$

4.5 Conclusion

This chapter has presented a non-Gaussian Synthetic Aperture Sonar factor, the general multi-modal factor graph SLAM framework in which it is used, and the specific mm-iSAM non-Gaussian SLAM algorithm for solving the inference problem. Results from field trials in the Charles River and evaluation of the overall method are presented in the following chapter.

Raw acoustic waveforms from a single receiver are incorporated directly into the SLAM inference problem through a novel procedure of generating reversible SAS measurement likelihoods. A process of sequential match-filtering and conventional beamforming computes non-Gaussian likelihoods over the measurements of range and look-angle that can be efficiently and discretely sampled by an alias sampler. Fig. 4-41 provides a schematic of this process, outlining the steps that take place inside the iteratively computed SAS-factor. Pose samples are computed by first sampling from the range distribution. Those samples are propagated in a Gibbs sampling step that approximates samples from the conditional posterior distribution defined by the SAS measurement residuals. We construct residuals for forward and backward convolution such that the received TDOA and DOA measurements can inform either the location of the acoustic beacon, or the pose of the receiver element, whichever is the current variable being sampled. The forward likelihood function is derived from the beam pattern of a conventional beamformer. The backward likelihood function is computed by a leave-one-out correlation approach. The fundamental ambiguities arising from the TDOA and DOA beamformer approach result in multi-modal residual distributions that are enabled by the non-Gaussian framework.

The factor graph and its respective Bayes Tree were presented as a framework for describing the inference problem. Inference on the factor graph to compute the joint posterior of a large set of variables was posed as a message-passing algorithm on the Bayes Tree. In particular, the sum-product algorithm for belief propagation takes advantage of the tree's special structure to reduce overall complexity. The cliques of the Bayes Tree were used to distinguish between frontal variables, which can be safely

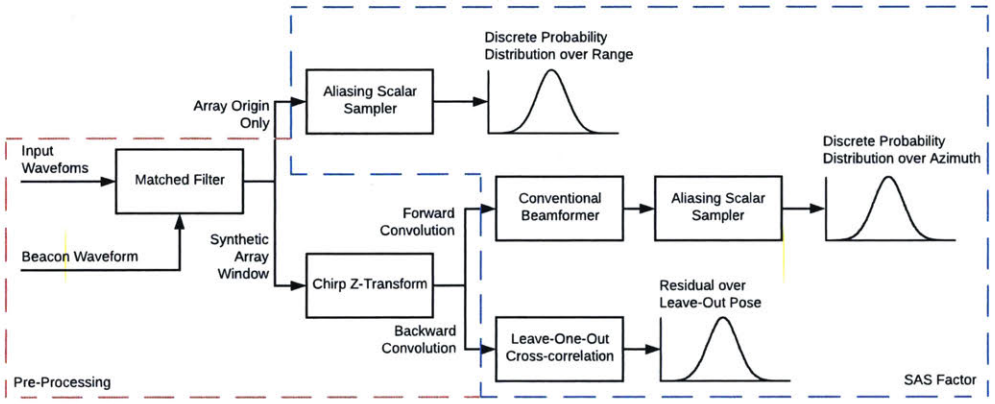

Figure 4-41: Schematic representation of acoustic waveform pre-processing and computation taking place within a SAS-factor. Pre-processing steps are computed once for each factor, while sampling within the factor happens iteratively during inference.

marginalized out of the inference at that level in the tree, and separator variables, whose conditional distributions are propagated up the tree. The up-solve step passes messages from every child defining the conditional distributions over their separator variables. The down-solve step passes messages from every parent defining marginal distributions over the frontal variables of the parent. The sum-product approach on the Bayes Tree concludes after a full up-solve and down-solve with marginal densities computed for every variable in the factor graph.

We apply the mm-iSAM algorithm to solve the non-Gaussian inference problem [58]. The belief messages passed on the Bayes Tree are marginal conditional distributions for each variable in a clique, and can be computed by evaluating the Chapman-Kolmogorov transit integral over every other variable in the clique. mm-iSAM computes this approximate convolution by batch sampling each variable in a Gibbs sequence, as illustrated in Fig. 4-42. The samples are updated by non-linear minimization over a measurement residual.

The sum-product approach to passing non-Gaussian likelihoods represented by kernel density estimates captures multiple hypotheses without the need for explicit

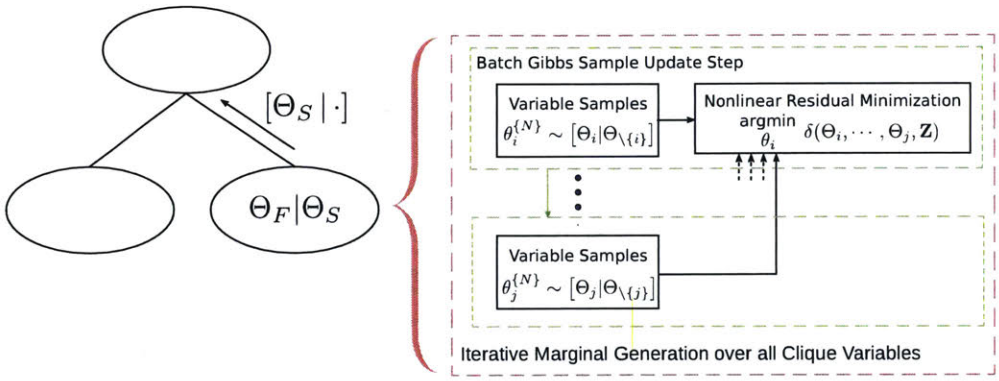

Figure 4-42: Schematic representation of mm-iSAM Batch Gibbs Sampling and Belief Propagation on the Bayes Tree

hypothesis tracking. Each Gaussian kernel potentially represents an individual hypothesis. The product of arbitrary distributions during belief propagation naturally eliminates unlikely modes from the posterior density, provided that consistent measurement models are used. Perhaps most importantly, unlikely modes are still captured in the conditional probabilities and are not permanently eliminated. If new information is added to the factor graph, modifying the Bayes Tree, those modes can re-emerge from the updated inference operation.

Chapter 5

Field Experiments and Results

In this chapter we present results from performing non-Gaussian SAS-SLAM inference on four experimental datasets collected in the Charles River. We demonstrate that a factor graph SAS formulation with just a few SAS-factors accurately localizes an unknown acoustic beacon and simultaneously informs the trajectory of a moving receiver to correct for inertial drift. mm-iSAM captures the multi-modal aspect of marginal distributions produced by ambiguous range measurements or linear array CBF range and bearing measurements, and the multi-modality collapses to the correct mode given enough measurements to resolve ambiguity. Kernel density estimate mean and max values are presented as convenient representations of the distribution, keeping in mind that the inference is fully non-parametric. The performance of the inference algorithm is analyzed by several metrics, namely the accuracy of localization compared to RTK-GPS ground truth, the multi-modality of the conditional and posterior distributions, along with the number of factors, computation time and size of the graph. Additionally, we present for comparison a conventional range-only SLAM formulation on the same datasets. Navigational and acoustic data in this chapter were recorded onboard our testbed ASV, and a discussion of experimental conditions is provided.

5.1 Field Experiments

Let the experimental datasets be designated by labels *Stationary*, *Drifting* and *Transit* in reference to the trajectory of the acoustic beacon for each type of experiment. Within each experiment we designate as separate trials those windows in which the ASV travels a full planned trajectory once. Fig. 5-1 shows full ground-truth RTK-GPS trajectories for the *Stationary* and *Drifting* experiments. In both experiments types, the ASV autonomously travels in an elliptical or circular loiter pattern relative to the beacon at a constant forward speed of approximately 1m/s. In the *Stationary* experiment, the beacon was mounted off the dock at the MIT Sailing Pavilion. Its ground truth location was independently verified by differential GPS to be $[x = 17.05, y = 1.78]$ (m). The trajectory of the drifting beacon is shown in red.

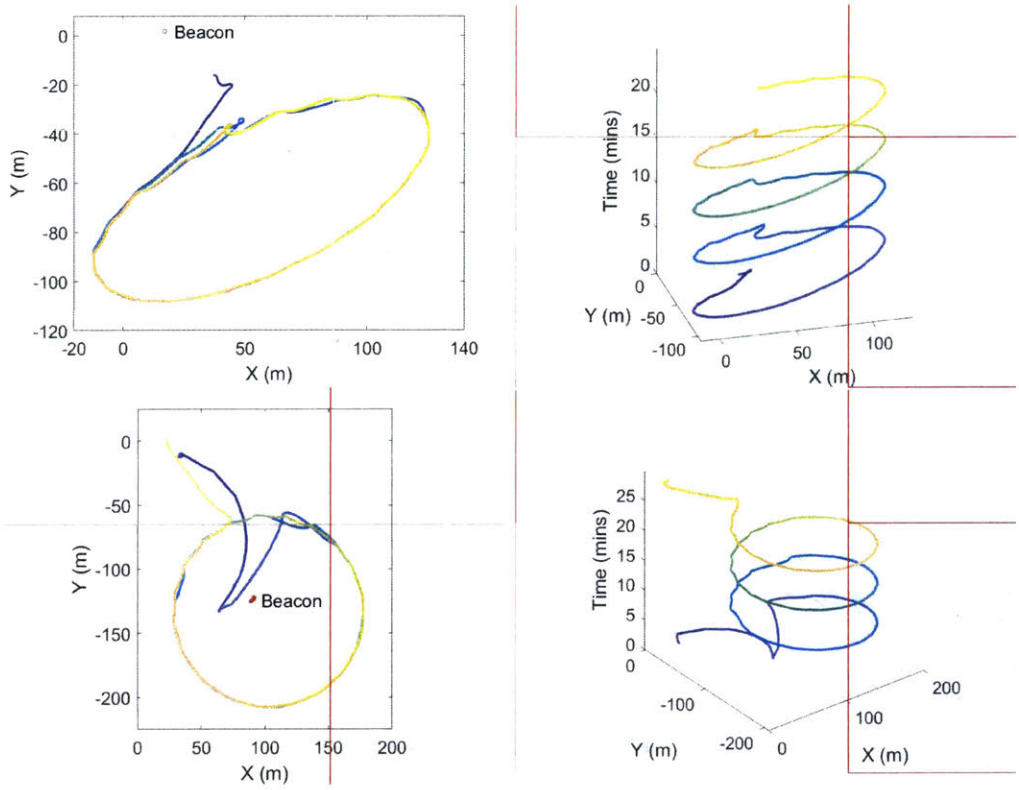

Figure 5-1: Full ASV and Beacon ground truth trajectories by experiment, color-coded by time. In 3D, the vertical axis is mission time in minutes. (*Top:*) *Stationary* Beacon at $[17.05, 1.78]$ (*Bottom:*) *Drifting* Beacon with trajectory shown in red

In the *Drifting* experiment, the beacon was hung off an anchored motorboat in an open section of the Charles River. The motorboat is free to drift with surface waves up to the limit of the anchor cable, estimated to be a maximum radius of 10m, and with its engine fully off. In this dataset, ground truth positions for the beacon are reported as a trajectory, with covariances up to the accuracy of RTK-GPS.

The ASV has three pre-planned behaviors relative to the moving beacon for the *Transit* experiment. *Simple-Parallel* and *Simple-Perpendicular* describe trajectories that run parallel to the beacon at a constant stand-off. These two trajectories are shown in Fig. 5-2 and Fig. 5-3. In the final behavior, *Loiter*, the vehicle executes a circular loiter pattern of constant radius around the moving beacon.

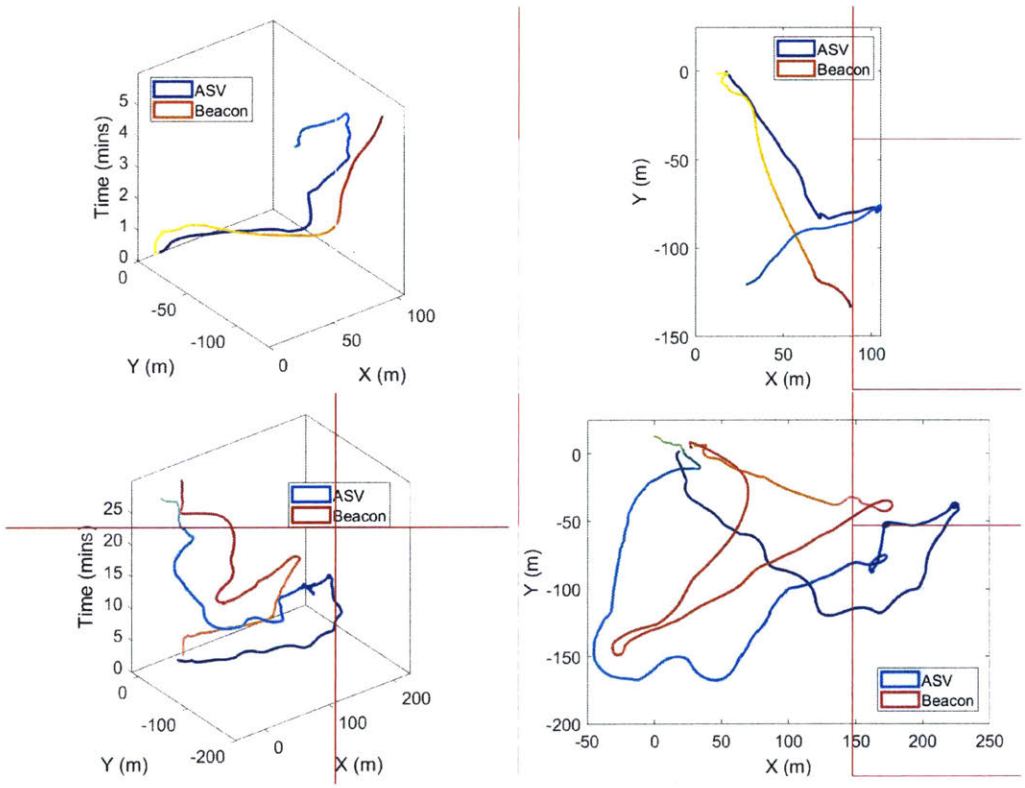

Figure 5-2: Full ASV and Beacon ground truth trajectories by experiment, color-coded by mission time in minutes. (*Top:*) *Simple-Parallel* and *Simple-Perpendicular* behaviors, where the trajectory of the ASV is parallel or perpendicular to the beacon (*Bottom:*) *Constant-Follow* behavior where the ASV follows the beacon at a constant stand-off distance.

For the purposes of trajectory planning and control, the ASV is provided with the ground-truth location of the beacon (motorboat). Trajectories are planned in advance, and the SLAM solution is not computed online. We note that although the autonomy behavior is intended to emulate an AUV, the experimental ASV is able to travel much faster (up to 4m/s) and has a smaller turning radius than a typical AUV, as mentioned in Section 2.2. Therefore, we assume the ASV is always able to execute the planned trajectory. The overall mission autonomy comprises of the system of tuned PIDs for low-level thrust and heading control onboard the vehicle, the mission-level MOOS-IVP track-line, waypoint and loiter behaviors, and the WiFi-based communication infrastructure of MOOS.

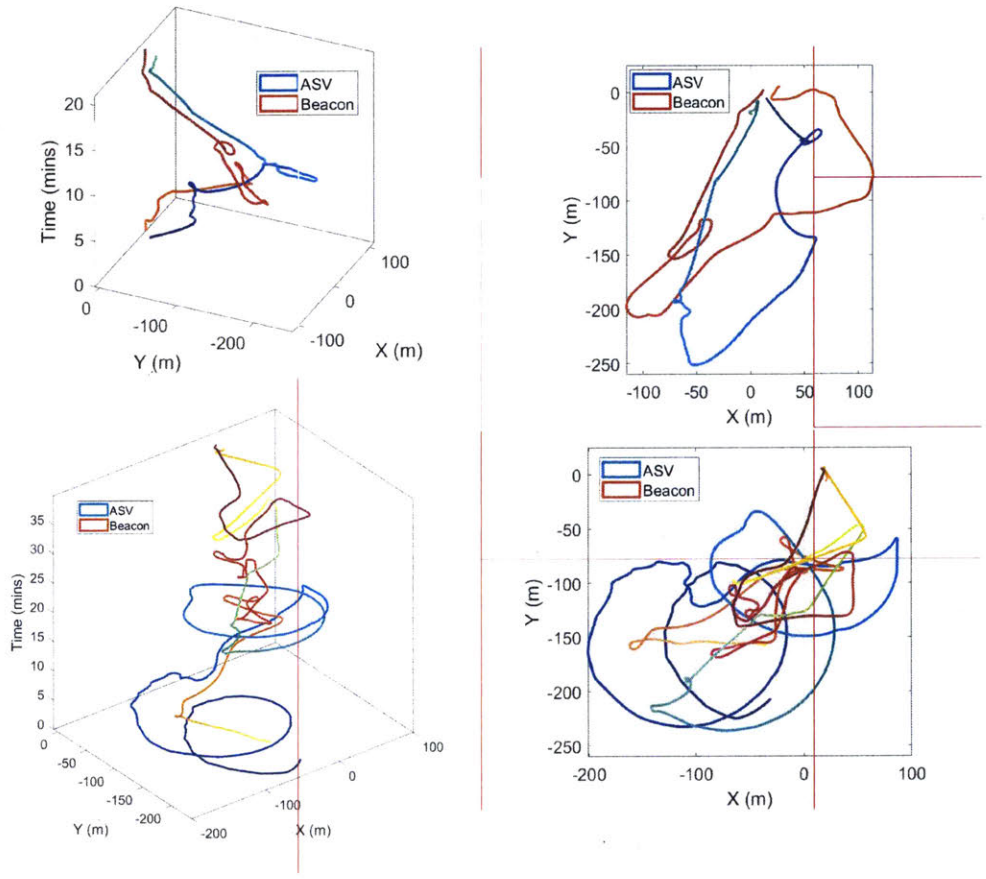

Figure 5-3: Full ASV and Beacon ground truth trajectories by experiment, color-coded by mission time in minutes. (*Top:*) Stationary Beacon at Dock (*Middle:*) Drifting Beacon (*Bottom:*) Beacon in Transit

There are several additional considerations to keep in mind for field experiments conducted in the Charles River. The presence of significant stochastic spikes in ambient acoustic noise may be observed from passing ships such as ferries, other motorboats, and changing weather conditions over the course of the mission. These may be noted in the relevant discussion where applicable. Additionally, while the ASV is configured to autonomously cut power to its motor during receive windows in order to reduce noise, the motorboat does not have the same capability and actively affects SNR during experiments where it is towing the beacon. During experiments where either the ASV or the motorboat are a significant distance (>200m) away from the shore station, intermittent dropouts may affect WiFi connectivity. WiFi dropouts do not affect the transmit or receive schedules of the acoustic beacon and transducer array, the trajectory of the human-driven motorboat or the onboard autonomy of the ASV. However, they will alter the planned trajectory of the ASV due to late or missed updates for the ground-truth location of the beacon. Similarly, RTK-GPS requires WiFi connectivity to the base station server, and will default to a local fix when it drops out.

For reference of scale, Fig. 5-4 shows each ground-truth mission trajectory on a satellite image of the operational area around the MIT Sailing Pavilion. The *Stationary* and *Drifting* trials are plotted in blue and orange respectively, while the trajectory of the ASV during two *Transit* experiments is plotted in purple and yellow. Each dataset takes place on a separate experimental day. Since the ASV does not experience inertial drift, its trajectory is collectively perturbed with simulated cumulative drift between any provided GPS priors, and its poses in the factor graph are informed by the drifted measurements. Both ground truth and drifted trajectories are provided in the results. Localization error is compared directly to ground truth.

Acoustic waveforms are recorded with a stern-mounted pyramidal hydrophone array at a fixed depth of 1.5m (Section 2.2.3). Waveforms are recorded at all five elements of the pyramidal array. Since the ASV is a stand-in for an AUV equipped with a single receiver, only one element is used during synthetic aperture beamforming, although in the future, LBL measurements from the array could potentially also

Figure 5-4: ASV Mission Trajectories relative to satellite image. *Stationary* Beacon and *Drifting* Beacon trials are shown for reference in blue and orange respectively. Simple and Loiter transit trajectories are shown in purple and yellow.

be included in the factor graph representation as additional factors [148]. Acoustic waveform measurements are recorded onboard the vehicle at the rate of 1Hz and sampled at a frequency of 37.5 kHz for a total of 8000 samples per waveform. Vehicle poses are generated at a rate of 10Hz, and recorded synchronously with the waveform measurement.

A Lubell acoustic beacon is used to transmit a known, time-synced waveform. The waveform is a Linear Frequency Modulation chirp in the range 250 to 2050Hz, with a bandwidth $\Delta f = 2$kHz. The theoretical distance resolution achievable with this chirp is:

$$\delta_r = \frac{c}{2\Delta f} = \frac{1481}{4000} \approx 0.37\text{m} \tag{5.1}$$

Fig. 5-5 presents the FFT of the chirp waveform in the frequency domain, and a representative spectrogram of the received signal before and after the CZT filter. Banding observed in the spectrogram is identified as electronic noise on the power distribution circuit, which we have shown in Section 4.2.1 to be effectively filtered out by the matched filter. In all field trials, the beacon is mounted or towed at an approximately constant depth of 1.5m.

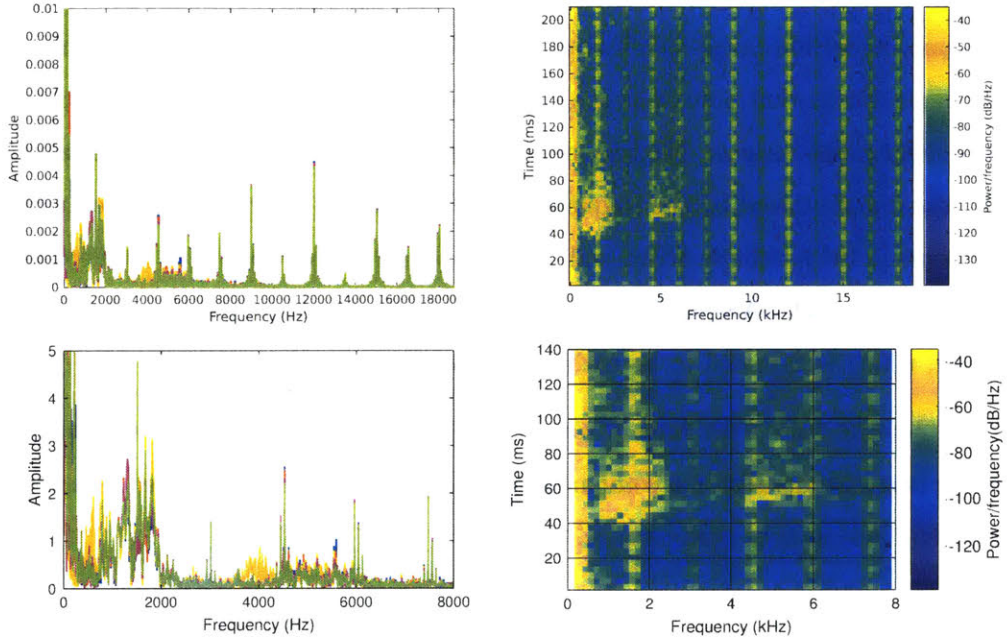

Figure 5-5: (*Left:*) Frequency content of pre-generated beacon chirp waveform after a Fourier Transform (Top) unaltered chirp and (Bottom) after taking a chirp z-transform (*Right:*) Spectrogram of sample waveform received at single element of pyramidal array (Top) unaltered (Bottom) after taking a chirp z-transform

5.2 SAS-SLAM Results

Experimental results are presented categorized by dataset and by construction of the factor graph. Two variable-types are used to describe the vehicle and beacon, Point2 and DynPoint2 (see Section 4.4.3). To investigate the performance of the SLAM solver in detail, we vary the availability of ground-truth navigation information in the form of GPS priors. First we show results from a range-only SLAM formulation in which matched-filtered acoustic data is used to generate range measurement likelihoods for alias-sampling within range-only factors. This formulation differs from a conventional implementation in that non-Gaussian inference is performed over the fundamentally multi-modal range returns. Next we describe several factor graph SAS formulations. These are differentiated by the frequency of SAS factors (observations), the number of elements in the synthetic array, and the type of inertial pose assigned to the vehicle and beacon. In each SAS-SLAM implementation, no prior is assigned to the beacon's position, which is assumed to be entirely unknown.

The 2-norm error from both a Mean-Fit (Mean) estimate and a MAP estimate over the posterior distribution compared against ground-truth is presented as a simple parametric indicator of performance.

$$e_b = \sqrt{(\hat{x} - x_o)^2 + (\hat{y} - y_o)^2} \tag{5.2}$$

Where relevant, the average 2-norm error e_v for vehicle poses compared to ground truth is also tabulated. The multi-modality of the posterior is evaluated by computing the Kullback-Leibler divergence (KLD) between the KDE and a fitted multivariate normal distribution.

$$\mathcal{D}_{KL}\left(p(\Theta)|q(\Theta)\right) = \int_{\mathcal{D}} p(s) \log\left(\frac{p(s)}{q(s)}\right) ds \tag{5.3}$$

Since the KLD is not symmetric, and does not obey the triangle inequality, we take the maximum divergence as representative.

5.2.1 Range-Only Results

As a basis for comparison, a non-Gaussian range-only factor graph (Fig. 5-6) is constructed from the same experimental datasets. Range measurement likelihoods are generated from the output of the matched-filter, as in Figs. 4-4 and 4-6, and formulated as range-only factors. During the Gibbs sampling step, the measurement likelihood is alias-sampled, and residual root-finding is performed to find the optimal sample value for that step.

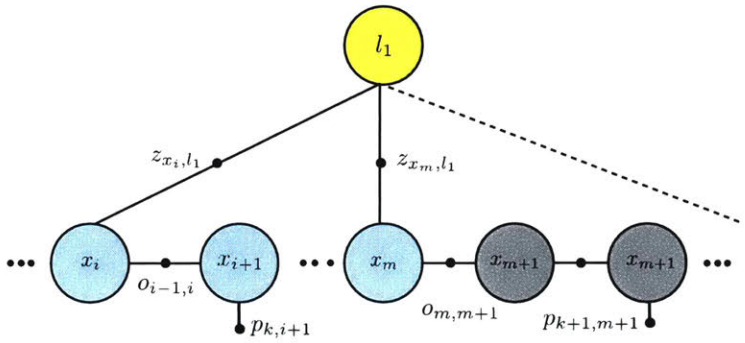

Figure 5-6: Range-only Factor Graph representation with measurements every m poses and priors every k poses.

Fig. 5-7 shows the results of a formulation in which every vehicle pose is provided with a tight GPS prior, essentially posing the equivalent of a mapping-only or acoustic localization problem. The beacon's approximate marginal posterior distribution is presented as a kernel density estimate contour plot, with each contour corresponding to 20% of the probability mass.

As the number of range measurements provided along the vehicle's trajectory decreases (from top to bottom, left to right), the posterior distribution over the beacon position is increasingly multi-modal. With only 6 factors, as in the bottom-right plot, the stochastic sampling method draws samples from two equally likely hypotheses of the multi-modal distribution. In this specific case, the MAP estimate would land on the wrong mode. With at least 10 poses and range factors, the true position of the beacon becomes strongly localized, and as expected the error decreases as the number of measurements increase.

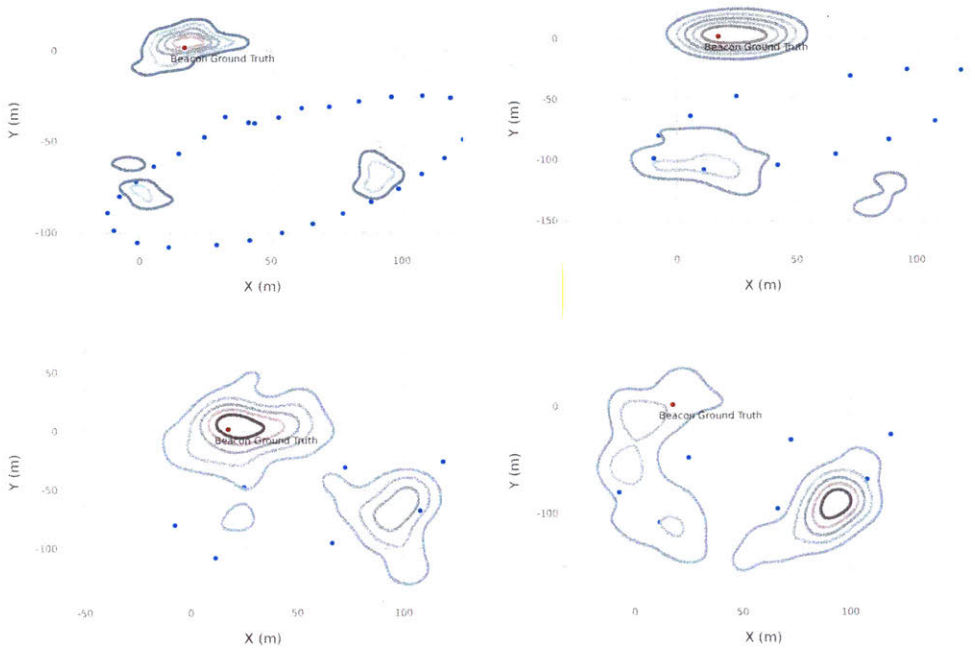

Figure 5-7: Marginal Posterior probability distribution over the unknown beacon's location, shown as probability contours each containing 20% probablity mass (20/40/60/80% confidence). All vehicle locations shown as blue points are pose priors with RTK-GPS covariances. (*Top-Left:*) 30 variables and range factors. (*Top-Right:*) 13 variables and range factors. (*Bottom:*) Two equally likely posterior distributions given only 6 variables and factors.

This proceeds into a full SLAM formulation by introducing inertial odometry factors between each vehicle pose. GPS priors are provided only at the very start and end of the trajectory, simulating the fix received by an underwater vehicle at the surface between dives. Fig. 5-8 and Fig. 5-9 are an example of the factor graph and equivalent Bayes Tree from a 290 pose formulation for the *Stationary* dataset. The factor graph is characterized by long odometry chains, forming a loop closure with each consecutive range observation of the beacon. This results in a Bayes Tree with a deep branch containing the landmark as a frontal variable.

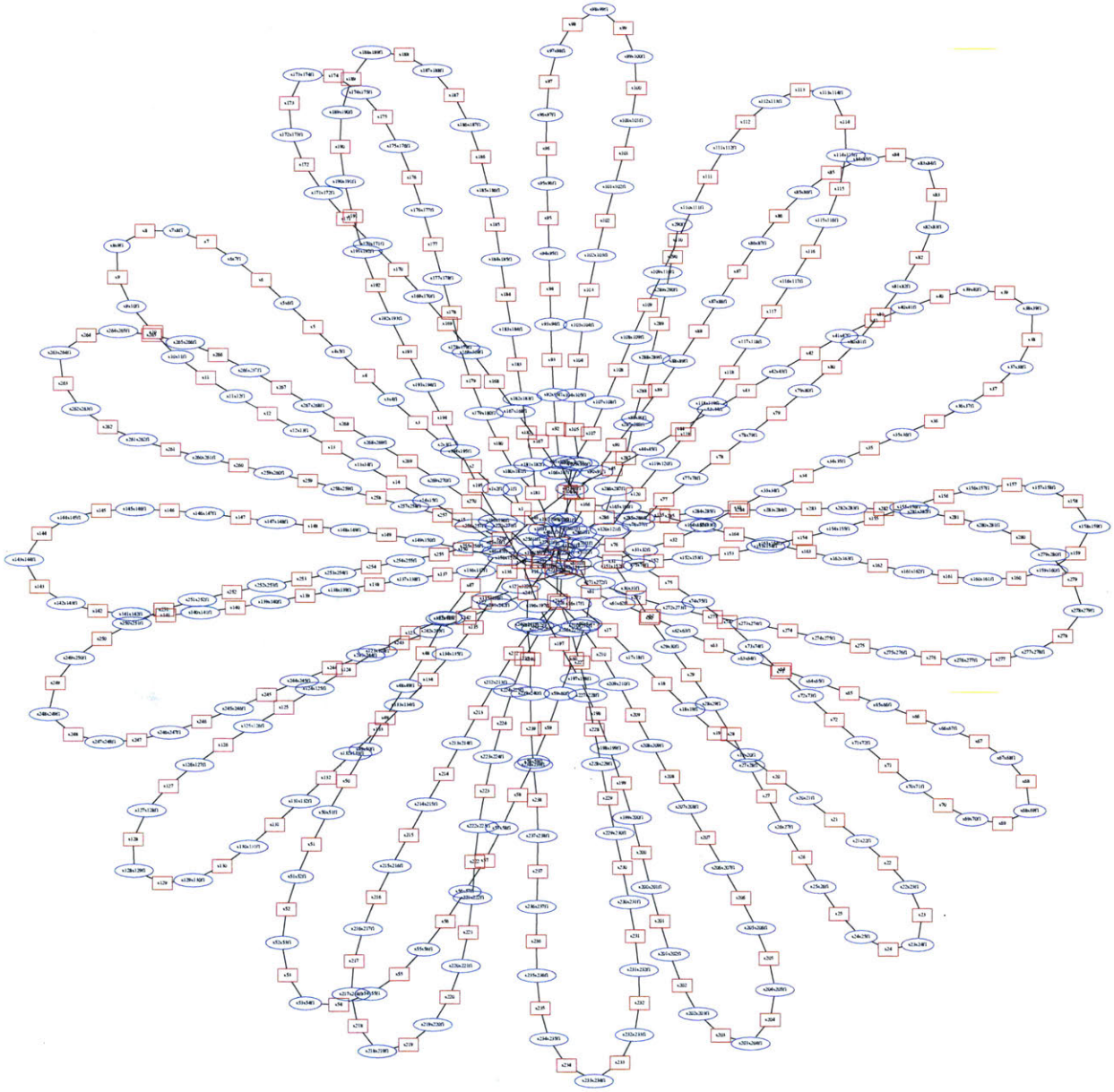

Figure 5-8: Example Factor Graph generated for 290 poses of the Stationary dataset, with 21 range factors in total. Factor nodes are shown as ovals and variable nodes are shown as rectangles. Long odometry chains form loop closures within each consecutive pair of range factors. The beacon is plotted at the center of the graph.

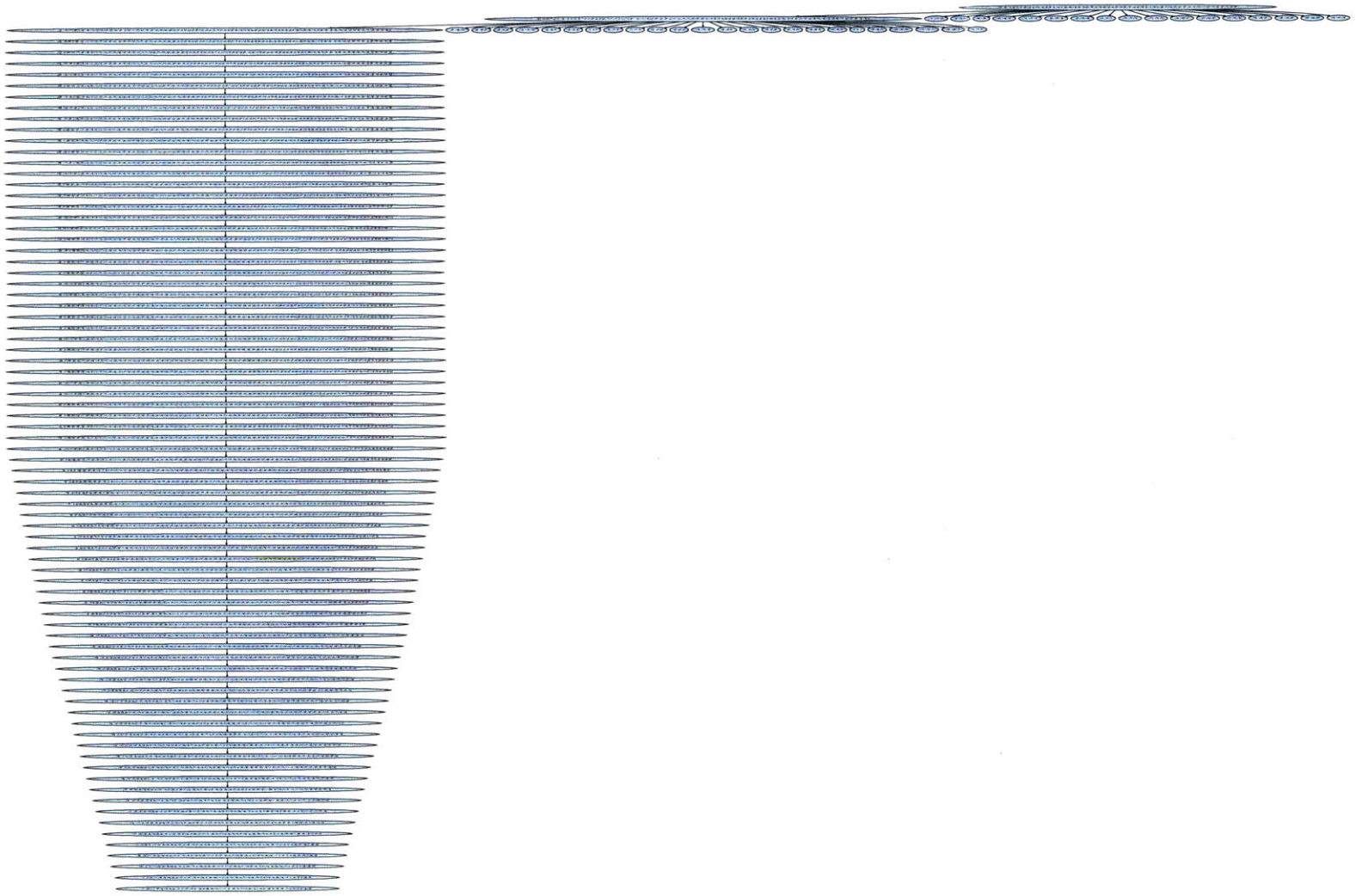

Figure 5-9: Example Bayes Tree generated for truncated trajectory shown in Fig. 5-12. This range-only acoustic beacon localization solution is characterized by a deep branch containing the range factors, and a series of shallow sibling branches for unaffected odometry factors.

From Table. 5.1, we observe that the range-only localization can be as accurate as sub-5m to the beacon's true position, and any accuracy beyond this point is beyond the resolution of the method. We emphasize that the mm-iSAM solver represents the final result over each variable as a set of samples, and the mean or max error tabulated here is a simplifying parametric result that might not capture the true behavior of the posterior distribution. The *Drifting* dataset is characterized by lower errors in general as the circular loiter trajectory taken by the vehicle clearly resolves ambiguity after half the trajectory. This can also be seen from Fig. 5-12.

Table 5.1: Range-Only SLAM

Experiment (.Trial)	Poses	#-Factors Range	e_b (m) Mean	Max	KLD
Station.1	290	12	39.27	28.13	0.003
Station.2	295	11	38.99	28.43	0.011
Station.3	290	12	33.42	64.08	0.034
Station.1	290	20	40.40	21.87	0.04
Station.2	295	20	42.34	46.18	0.29
Station.3	290	20	51.84	50.16	0.059
Station.1	290	58	8.49	9.56	0.04
Station.2	295	59	8.45	7.41	0.03
Station.3	290	58	7.07	7.42	0.056
Drift.1	400	16	24.02	28.45	0.059
Drift.2	398	16	21.52	17.89	0.25
Drift.3	398	16	21.70	29.36	0.087
Drift.1	400	25	12.52	11.13	0.02
Drift.2	398	25	12.62	10.07	0.042
Drift.3	398	25	14.16	16.54	0.060
Drift.1	400	65	2.67	4.77	0.012
Drift.2	398	65	7.92	9.46	0.042
Drift.3	398	65	2.75	1.63	0.011

Fig. 5-11 shows the marginal posterior distribution over the beacon's position for *Stationary* and *Drifting* datasets as a contour plot and a histogram of samples. The maximum and mean estimates computed over the KDE for the beacon position are

plotted in orange and labeled. For simplicity, final vehicle poses are shown in blue as their MAP estimates over their marginal posterior distributions, and the ground-truth RTK-GPS trajectory is shown as a green trace. By varying the frequency of range measurements used during factor graph construction, we can observe that the estimate converges to its best achievable accuracy at greater than 20 observations distributed along the entire trajectory. Additionally, With trajectories that are designed to resolve ambiguity, as in the *Drifting* experiment, this convergence can be much faster. Similarly, Fig. 5-12 and Fig. 5-13 plot the posterior distribution by mission time along trajectory and by intra-mission trial.

5.2.2 All Priors

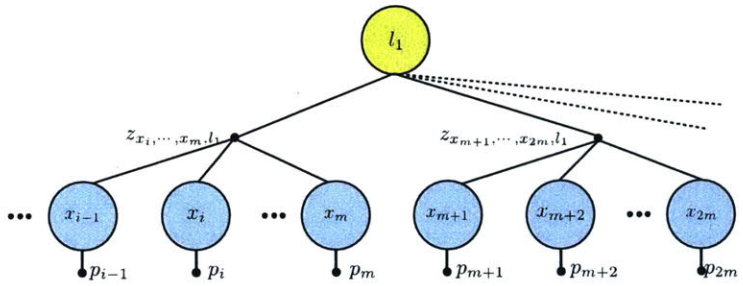

Figure 5-10: All Priors Factor Graph representation with one SAS-Factor every m poses and GPS priors on every pose.

First, we show results with the assumption that every vehicle pose is supplied with an accurate GPS ground-truth prior, i.e. a surface vehicle SLAM formulation, represented by the factor graph in Fig. 5-10. Since the vehicle's globally-referenced poses inform the position of the acoustic beacon, this is acoustic localization via synthetic aperture sonar in a non-Gaussian SLAM framework. We specifically analyze the performance of the forward convolution optimization step via the synthetic aperture sonar factor. Fig. 5-14 shows an instance of the *Stationary* dataset with six SAS factors, each comprising of eight poses (synthetic array length of approximately 8m) selected at equal intervals around the mission's elliptical trajectory.

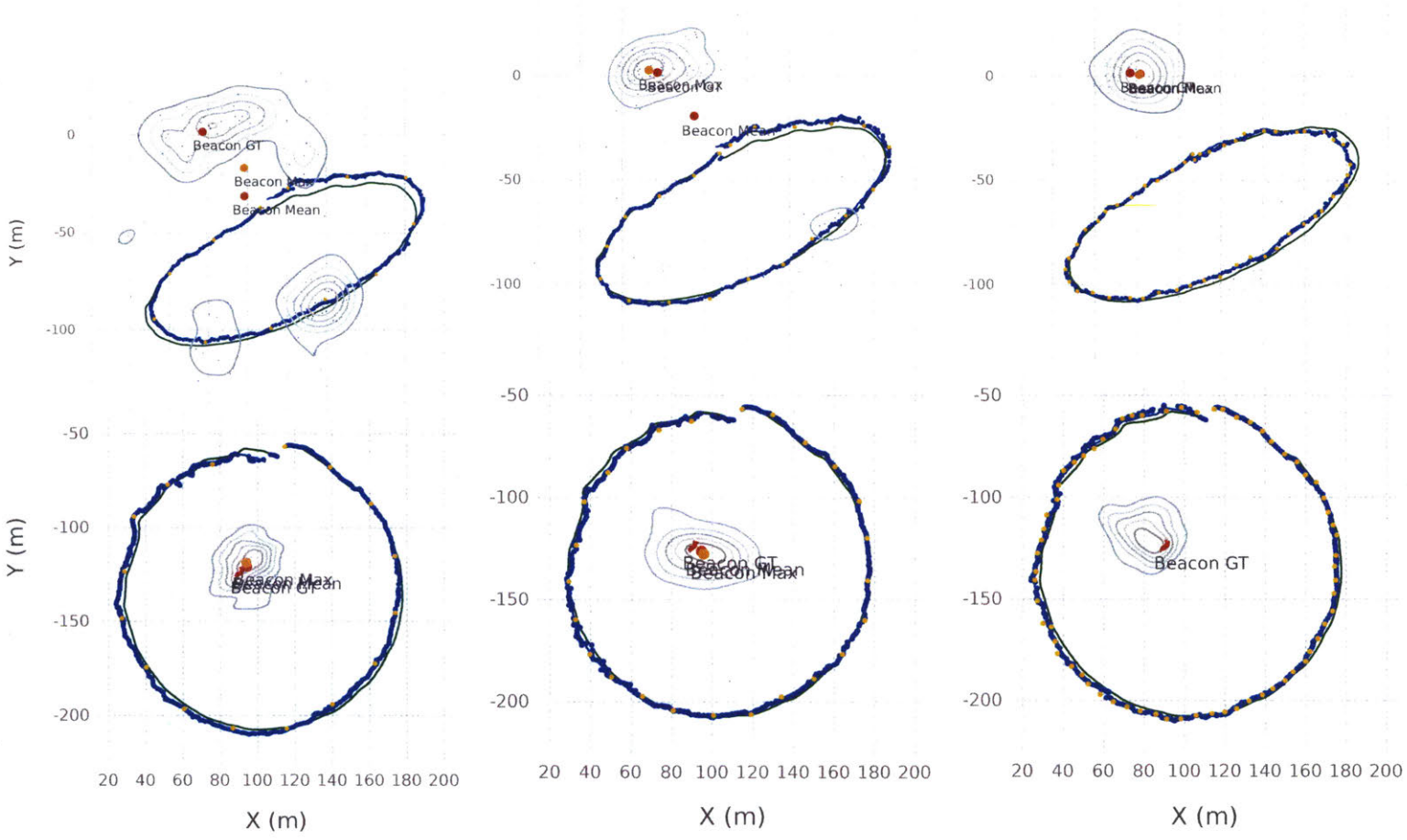

Figure 5-11: Marginal Posterior beacon distribution by number of range measurements. (*Top:*) *Stationary* Beacon $\{12, 20, 58\}$ range factors total. (*Bottom:*) *Drifting* Beacon $\{16, 25, 65\}$ poses total.

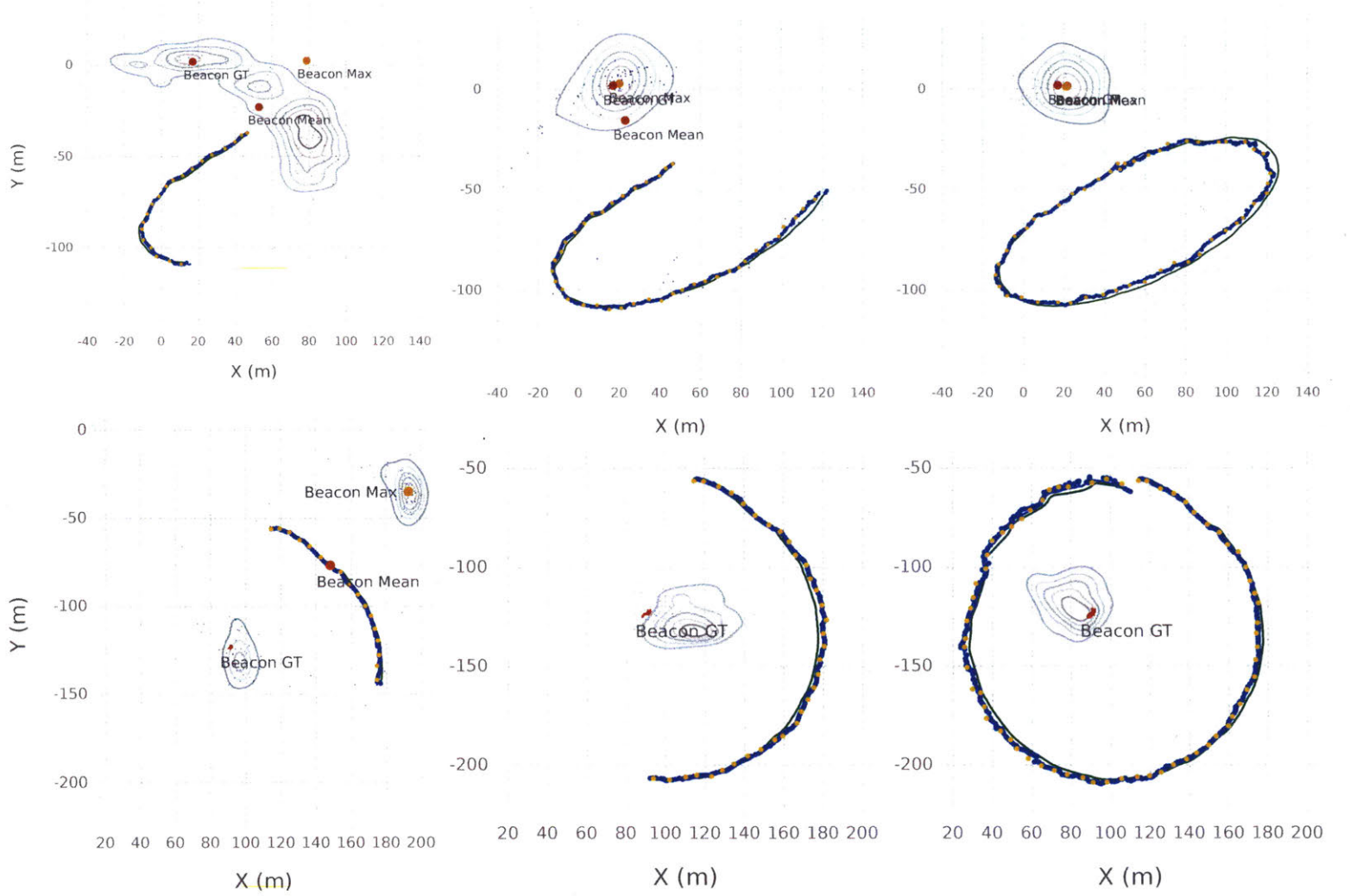

Figure 5-12: Marginal Posterior beacon distribution by Trajectory. (*Top:*) *Stationary* Beacon $\{100, 200, 300\}$ poses total. (*Bottom:*) *Drifting* Beacon $\{100, 300, 400\}$ poses total. Bimodal distributions collapse to the right mode given enough measurements along the trajectory.

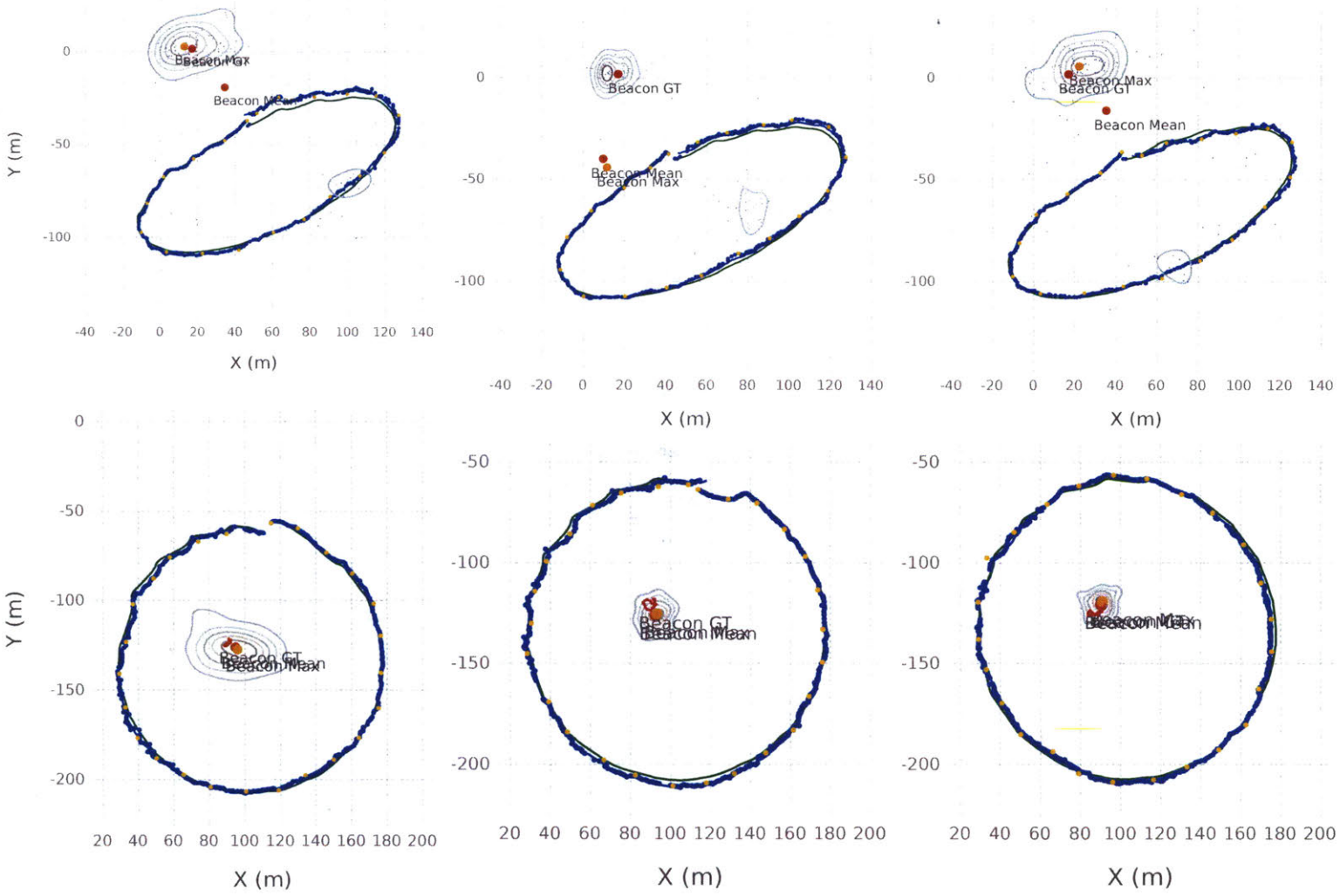

Figure 5-13: Marginal Posterior beacon distribution by Intra-experiment Trial. (*Top:*) *Stationary* Beacon at mission start time $\{410, 720, 1050\}$ seconds respectively. (*Bottom:*) *Drifting* Beacon at mission start time $\{1550, 1952, 2301\}$ seconds respectively.

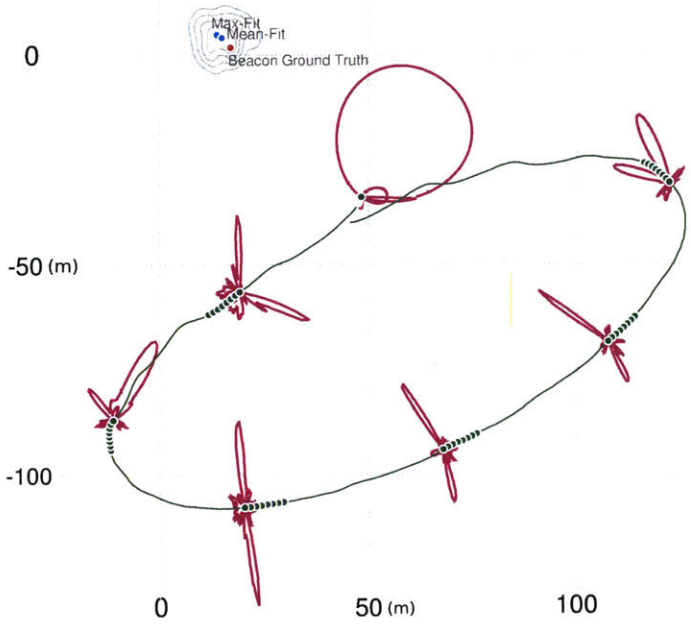

Figure 5-14: SAS-SLAM result with GPS priors on every pose for the *Stationary* dataset. Ground truth GPS trajectory of the vehicle is traced in green, with poses used for SAS shown as points. Estimated beam patterns are shown as magenta traces at the origin of each array (arbitrarily selected as the earliest pose). The true, MAP, and mean-fit beacon locations are shown, underlaid by the full posterior distribution represented as a density contour plot, with each contour containing 20% of the probability mass.

With accurate positioning, the beamformer strongly identifies the correct azimuth (and its equally likely symmetric second mode) for every synthetic array, with the exception of the very first one (closest to the beacon). The first circular pattern represents the case where no resolution in azimuth is obtained due to the very short effective array length (as the vehicle was relatively stationary at the start of the mission). This type of inconsistent data is accounted for by testing the effective length of the array before contributing the beamformer measurement. The ability to resolve the correct look-angle is also affected by end-fire conditions, as shown by the third and seventh factor (counting counter-clockwise).

We conclude that given even just three SAS-factors, inference over the joint distribution produced from the CBF resolves the posterior probability of the beacon's

location very accurately. The observed increase in prediction error from 7 factors to 16 can be attributed to the increasing number of SAS-factors with uninformative beamformer distributions (as discussed in Fig. 5-14).

Table 5.2: SAS-SLAM All Priors

Experiment (.Trial)	#-Factors SAS	e_b (m) Mean	Max	KLD
Station.1	3	5.50	6.22	0.018
Station.1	7	3.12	4.52	0.025
Station.1	16	7.42	7.05	0.033
Drift.1	3	5.51	8.05	0.359
Drift.1	8	4.10	6.70	0.027
Drift.1	14	4.53	4.40	0.048

5.2.3 Static Point Odometry

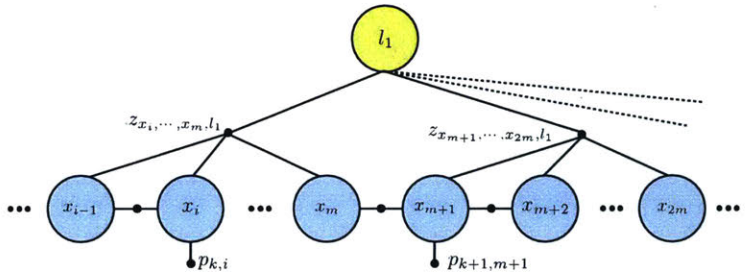

Figure 5-15: Static and Dynamic Odometry Factor Graph representation with one SAS-Factor every m poses and GPS priors every k poses.

Subsequently, GPS priors are provided on poses at different rates simulating dives of varying length. Navigational data is corrupted with a specified amount (0.2, 0.4 m/s) of cumulative navigation drift as a proxy for the accumulation of inertial error via odometry on an underwater vehicle. Drift is only accumulated between GPS priors, reflecting fixes obtained at the surface. Vehicle poses are initialized with corrupted position estimates, and consecutive vehicle poses are related with odometry factors ($\sigma = 0.5$m in x, y). Fig. 5-16 presents the evolution of the beacon's density estimate as the factor graph is incrementally supplied with SAS observations.

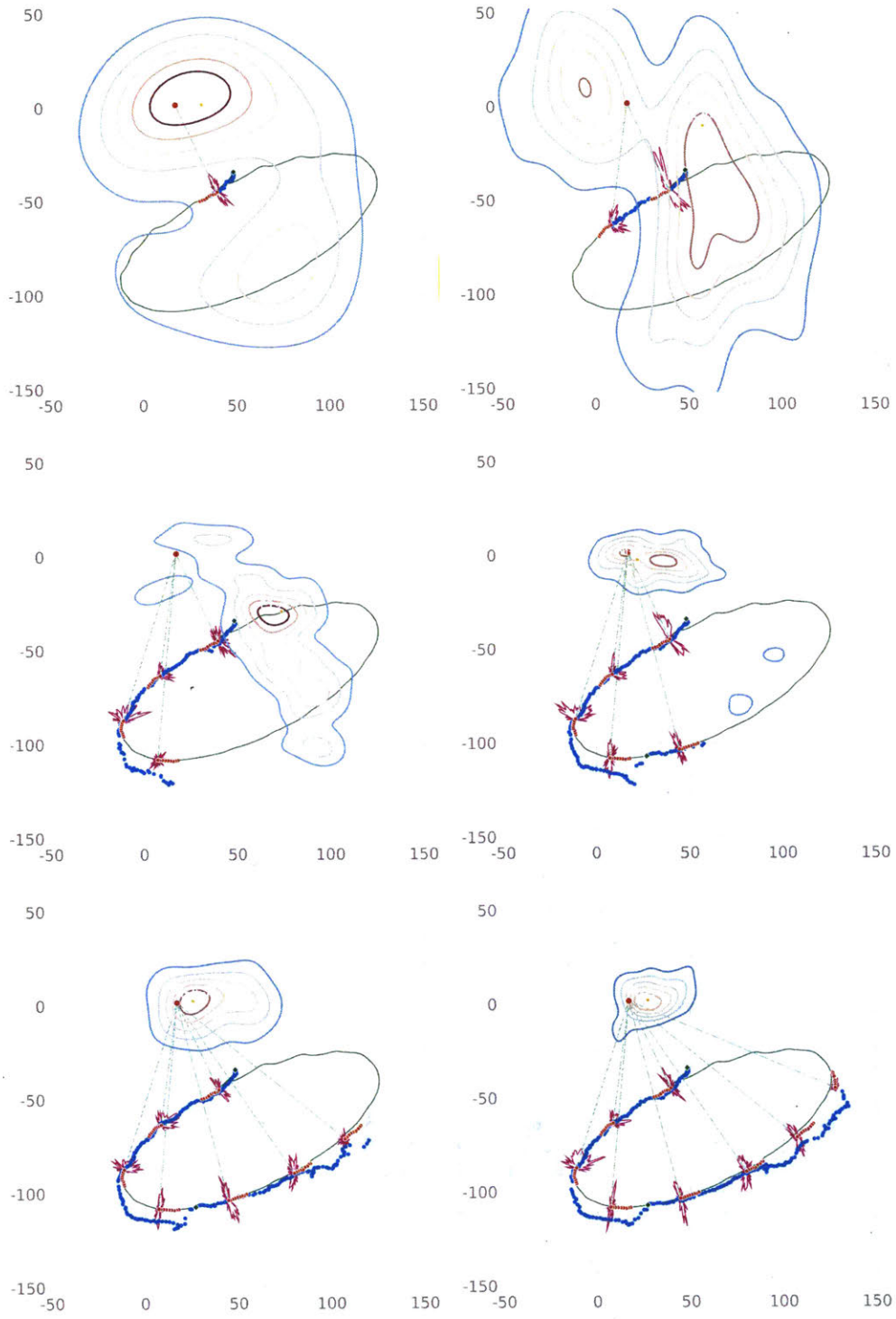

Figure 5-16: Evolution of posterior distribution over beacon represented as a density contour plot with the addition of each synthetic array to the factor graph

Here, GPS priors are shown as green diamonds and ground truth GPS trajectory is in green. Drifted position estimates (proxy inertial measurements) are shown as red points. The final MAP values of poses after inference is shown as blue circles, and the raw CBF beampatterns is shown as a magenta trace. The beacon locations are: ground truth (solid red line), MAP estimate and Mean-Fit estimate (blue points), and its posterior distribution is represented as a density contour plot, with each contour containing 20% of the probability mass. Five SAS-factors are created, each with eight array elements. X-Y scale is shown in (m). Array poses are shown in red. The CBF output of each synthetic array is shown in cyan. Each grey contour contains 50% of the probability mass.

Fig. 5-17 (Left) shows an instance of the *Drifting* dataset with five SAS-factors, instantiated at equal intervals around the trajectory encircling the beacon from approximately 75m. As a comparison, using the same dataset, Fig. 5-17 (Right) shows the density estimates from each of the five synthetic arrays and their beam patterns where each synthetic array is computed only from drifted position data (no SLAM). While ultimately the most probable location for the beacon lies on the interior of the trajectory, the fusion of the non-Gaussian acoustic measurements is key to identifying the posterior distribution of that probability given relevant symmetric modes.

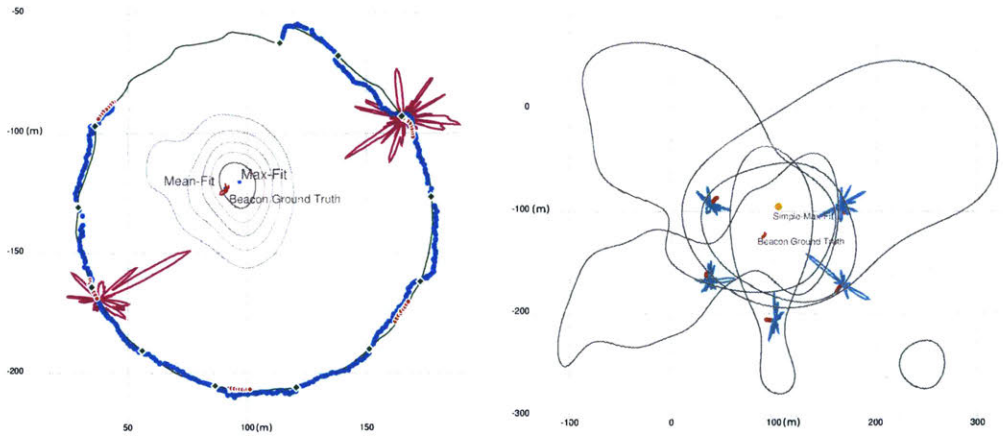

Figure 5-17: (*Left*:) Full SAS-SLAM result with inference over GPS, odometry and SAS factors. (*Right*:) Comparison without SAS-SLAM: The overall beacon location predicted with a Max-Fit (orange point) over all five distributions after 'forward' convolution only.

In addition to beacon localization errors, we compute the initial (drift) and final errors in vehicle positions e_p, and evaluate estimation performance under different conditions: number of GPS priors, number of SAS factors, and variance of the velocities drift in the position data ($\sigma_{\dot{x},\dot{y}}$). The MAP estimate was used to compute both e_b and e_p. The results, tabulated in Table 5.4, show that the algorithm succeeds in estimating the position of both beacon and vehicle to comparable levels of accuracy, bringing e_b and e_p to the order of a few meters. Increasing the drift in odometry measurements appears to have a more noticeable effect on the accuracy of the beacon estimate rather than the vehicle's.

Table 5.3: SAS-SLAM (Static Point)

Experiment	#-Factors		$\sigma_{\dot{x},\dot{y}}$	e_b (m)		KLD	e_p (m)	
(.Trial)	GPS	SAS		Mean	Max		Initial	Final
Station.1	3	5	0.2	15.3	26.3	0.258	7.6	2.6
	3	11	0.2	4.48	5.16	0.173	16.9	4.8
	3	16	0.2	2.92	3.12	0.035	13.7	2.6
Station.1	13	10	0.2	2.28	4.65	0.456	48.3	4.4
	13	10	0.4	3.38	4.30	0.037	72.9	6.4
	13	10	0.6	6.66	14.5	0.024	65.1	5.87
Drift.1	2	5	0.2	35.5	36.5	0.030	57.2	8.01
	2	12	0.2	6.77	11.0	0.044	22.4	5.7
	2	21	0.2	22.2	21.2	0.015	25.7	3.5
Drift.1	5	5	0.2	6.07	8.24	0.006	18.3	3.81
	18	5	0.2	6.27	5.56	0.006	3.08	2.29
	35	5	0.2	2.33	3.67	0.051	2.75	1.91

For the *Stationary* dataset, the increase in the number of observations quickly resolves the initially unknown beacon position estimate to within 10m of its true value. Platform position error, despite the low number of absolute measurements, remains low (<5m). For Trial B, even with only five SAS-factors, the localization performance was within 10m. The maximum initial squared error as a result of drift and the maximum final error in each pose e_p are reported. We observe that with very few GPS priors, increasing the number of SAS observations successfully decreases the

localization error and resolves the unknown position of the beacon as well as the case with all GPS priors.

5.2.4 Dynamic Point Odometry

We extend the factor graph formulation from static poses to dynamic poses by adding to the measurement residual a dynamic model that takes into account the first-order correlation between the vehicle's observed position and x-y velocities (see Section 4.4.3). This simple model observes the dynamic relationship between subsequent poses in time, inferring over four dimensions, but does not model the vehicle's non-holonomic maneuvering characteristics. For now, the landmark is still modeled as a static point. Representative intermediate results showing the change in posterior density over two iterations of the *Drifting* dataset is shown in Fig. 5-18, with respective factor graph (Fig. 5-19) and Bayes Tree (Fig. 5-20). Each SAS-factor of array length 8 is separated by an odometry chain of length 16 (in mission time 16 seconds).

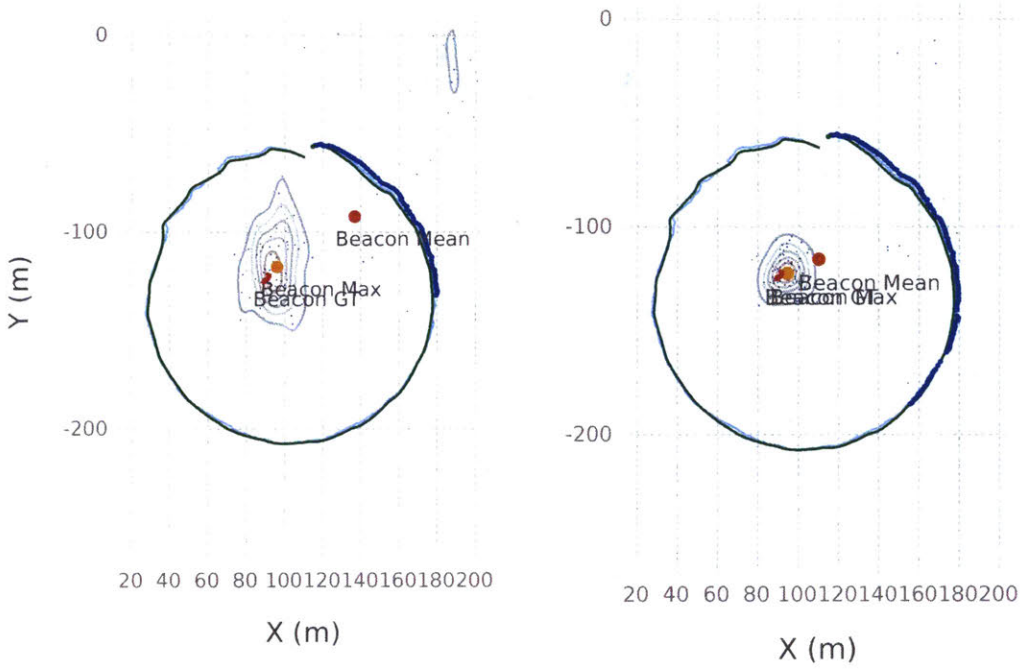

Figure 5-18: Sample intermediate SAS-SLAM solves with 4 and 6 SAS-factors and 1 and 2 GPS Priors respectively.

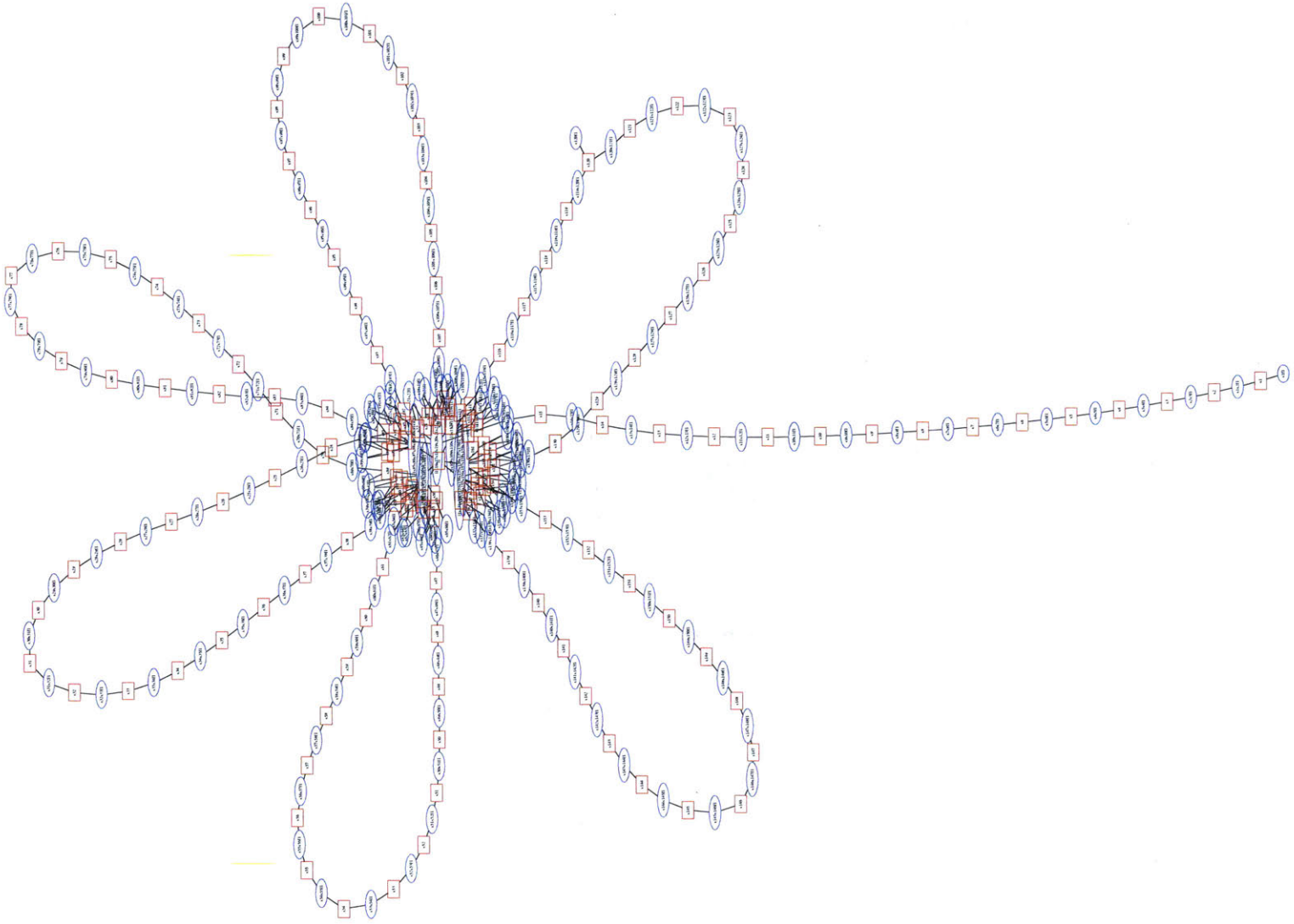

Figure 5-19: Intermediate factor graph generated for Fig. 5-18, with 6 SAS-factors along half of the full trajectory. Factor nodes are shown as ovals and variable nodes are shown as rectangles. SAS-factors (drawn at center) are loop-closures that connect every element of the array to the landmark.

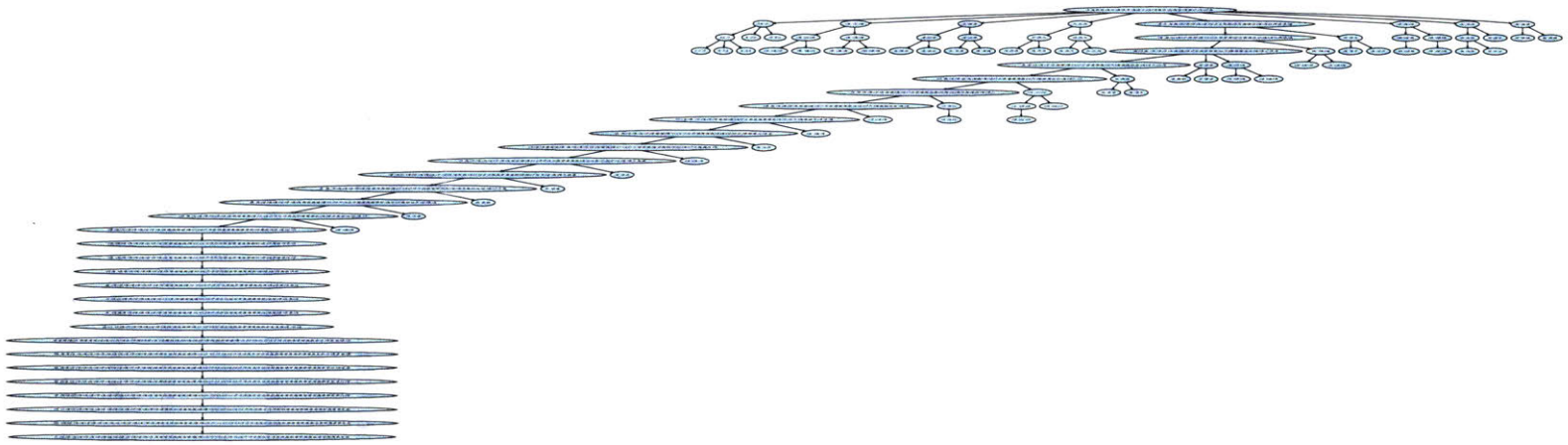

Figure 5-20: Respective Bayes Tree for Fig. 5-18 and Fig. 5-19. Deep branching structure is observed with frontal variables over the elements of each synthetic array.

The beacon's location is unambiguously resolved after 5 SAS-factors have been observed along the vehicle's trajectory. While keeping in mind that the max and mean estimates may not be good representations of the true multi-modal distributions, we show the change in localization error for the acoustic beacon in Figs. 5-21 and 5-22, for different datasets and specified number of GPS priors. Here, at each solve iteration the odometry chain is extended along the ASV's trajectory until another SAS-factor of the specified length can be added to the factor graph, and the factor graph is incrementally re-solved. Iteration does not refer to specific minimization iterations within the mm-iSAM back-end solver, and errors are computed after the final inference down-message passing concludes.

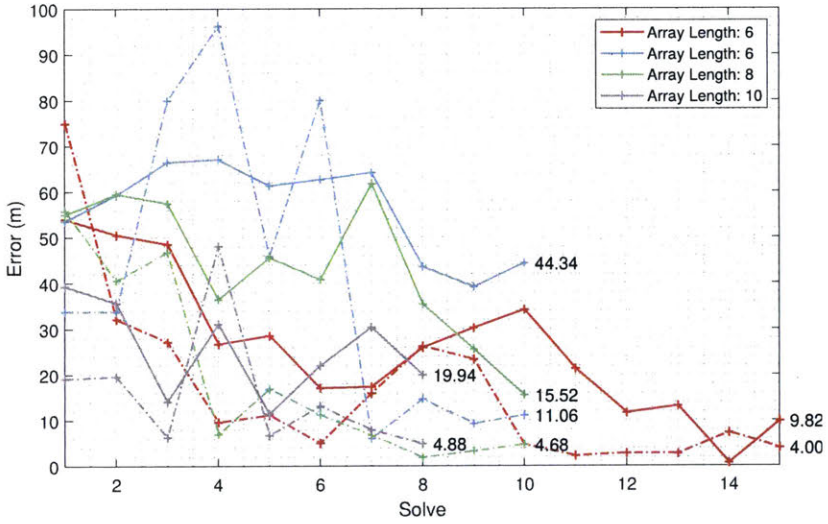

Figure 5-21: Mean (Solid) and Max (Dotted) KDE Estimates Error over beacon location by Incremental solve iteration for the *Stationary* dataset, color-coded by trial and array length. 5 GPS Priors total. Final errors are indicated in text.

Since the linear array beamformer is ambiguous about the axis of the array and the measurement distribution typically results in two modes, we observe that the Max KDE estimate that represents the mode with more likelihood density is more accurate than the mean that lies between both modes. Large spikes in max error such as at the 9th iteration in Fig. 5-22 occur when the Gibbs sampling procedure generates more samples around the incorrect mode.

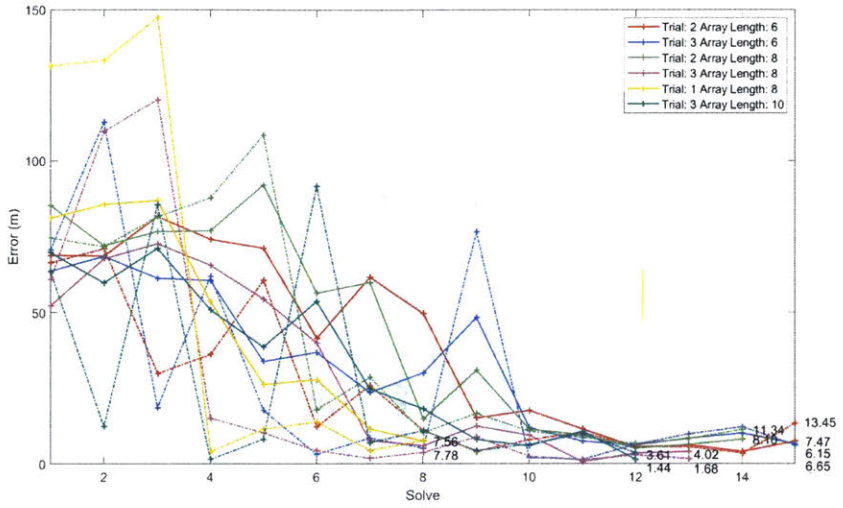

Figure 5-22: Mean (Solid) and Max (Dotted) KDE Estimates Error over Beacon location by Incremental solve iteration for the *Drifting* dataset, 2 GPS Priors total. Final errors are indicated in text.

Non-Gaussian inference over the factor graph SAS-SLAM formulation accurately observes the beacon's position even when the number of GPS priors is reduced to 2 (one at the start and one at the end) of an approximately 7 minute mission. Four to five SAS-factors distributed along half or all of the mission trajectory is sufficient for resolving ambiguity around the beacon's position, much less than for the range-only formulation and preserving the ability to sample less likely modes without explicit hypothesis tracking. The numerical performance of this formulation is summarized in Table 5.4, with the key parameters being, the number of GPS priors provided, the number of SAS-factors provided, and the length (number of elements) of the synthetic array. Several representative plots along different metrics are shown in Figs. 5-24, 5-25, 5-26 and 5-27.

The length of the synthetic array in number of elements is an interesting trade-off. Longer synthetic arrays provide better resolution but require longer trajectories, increasing the likelihood of navigation errors and decreasing the rate at which observations are generated. Less total SAS observations can be made over the trajectory, but faster updates can be generated. In our results computed offline, we observe that

increasing the array length reduces total navigation error. However, one advantage of the factor graph formulation is that factors with differing array lengths can be used in the same graph.

Results were computed on (4) Intel Xeon E5-263 CPUs (8 cores, 2.40GHz), without multi-threading. The computation times and allocated memory usage of the factor graph construction and mm-iSAM back-end optimization are presented in Fig. 5-23. Incremental construction of the SAS-SLAM factor graph benefits from the additional computational strategies of quasi-fixed-lag and incremental solving on the Bayes Tree. The former reduces iteration over variables older in time and the latter reuses computation on the Bayes Tree for incremental update. For further detail, the reader is referred to the mm-iSAM codebase maintained at `https://github.com/JuliaRobotics/Caesar.jl`.

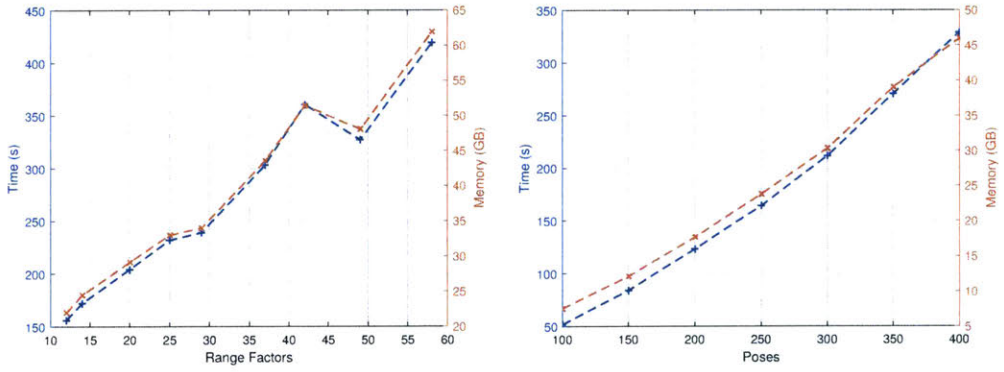

Figure 5-23: Computational performance of Range-only SLAM. (*Left:*) Varying the number of range factors in a trajectory of fixed length. (*Right:*) Varying the number of poses with range-factors at a fixed frequency.

In summary, this chapter has presented an effective Synthetic Aperture Sonar factor that fuses non-Gaussian range and bearing measurement likelihoods in a general factor graph SLAM inference framework, accurately localizing an unknown beacon and the vehicle even with very sparse global GPS reference. Inference was performed directly over non-Gaussian and multi-modal measurement likelihood distributions representing the experimental acoustic and navigation data collected onboard our testbed ASV in the Charles River. In the next chapter we conclude with a discussion and analysis of the overall method.

Table 5.4: SAS-SLAM (Dynamic Point)

Experiment (.Trial)	Poses	Array Length	Observations GPS	Observations SAS	e_b (m) Mean	e_b (m) Max	KLD	e_p (m) Initial	e_p (m) Final
Station.3	262	6	2	10	12.0	7.98	0.66	1.95	1.28
Station.1	290	6	2	15	2.14	2.91	0.007		2.62
Station.1	290	6	3	10	3.87	4.69	0.14	3.51	2.45
Drift.1	400	6	3	14	3.57	4.46	0.005		3.04
Drift.2	395	6	3	15	5.89	6.01	0.025	3.97	2.35
Drift.3	395	6	3	15	6.65	6.14	0.028	2.91	2.68
Drift.1	400	6	6	14	1.20	2.12	0.001	2.84	3.07
Station.3	276	8	2	9	13.52	5.52	0.019	2.30	1.28
Station.1	280	8	2	10	12.38	3.39	0.96	2.56	1.97
Drift.1	400	8	2	10	7.94	9.43	0.055		2.68
Station.1	290	8	2	12	6.61	5.95	0.016		2.21
Drift.1	384	8	3	8	7.55	7.78	0.17	2.07	2.79
Drift.1	400	8	3	10	4.81	5.22	0.027	2.28	3.84
Drift.2	398	8	3	19	4.25	8.58	0.04		2.27
Drift.3	371	8	3	13	4.01	1.68	0.001	3.71	1.74
Station.1	290	8	3	22	3.41	5.31	0.034	2.54	2.59
Station.1	280	8	4	10	5.52	4.65	0.77	2.51	1.26
Drift.3	371	8	6	13	1.82	3.22	0.010	3.60	2.18
Drift.1	400	8	6	9	7.80	7.93	0.025		3.36
Drift.1	400	8	6	22	1.96	1.84	0.031		1.88
Drift.1	397	8	6	22	1.91	2.16	0.001	2.38	2.08
Station.1	256	10	2	8	2.64	1.64	0.23	2.66	1.78
Station.3	278	10	2	9	10.25	9.09	0.12	2.70	3.23
Station.1	290	10	2	10	2.89	1.19	0.015		2.76
Station.2	295	10	2	11	11.4	6.12	0.82	2.90	1.52
Drift.1	360	10	2	12	2.79	3.81	0.03	2.46	2.39
Drift.3	373	10	3	12	1.44	3.60	0.03	2.39	1.65
Drift.1	398	10	4	19	1.01	0.15	0.006	2.77	2.93

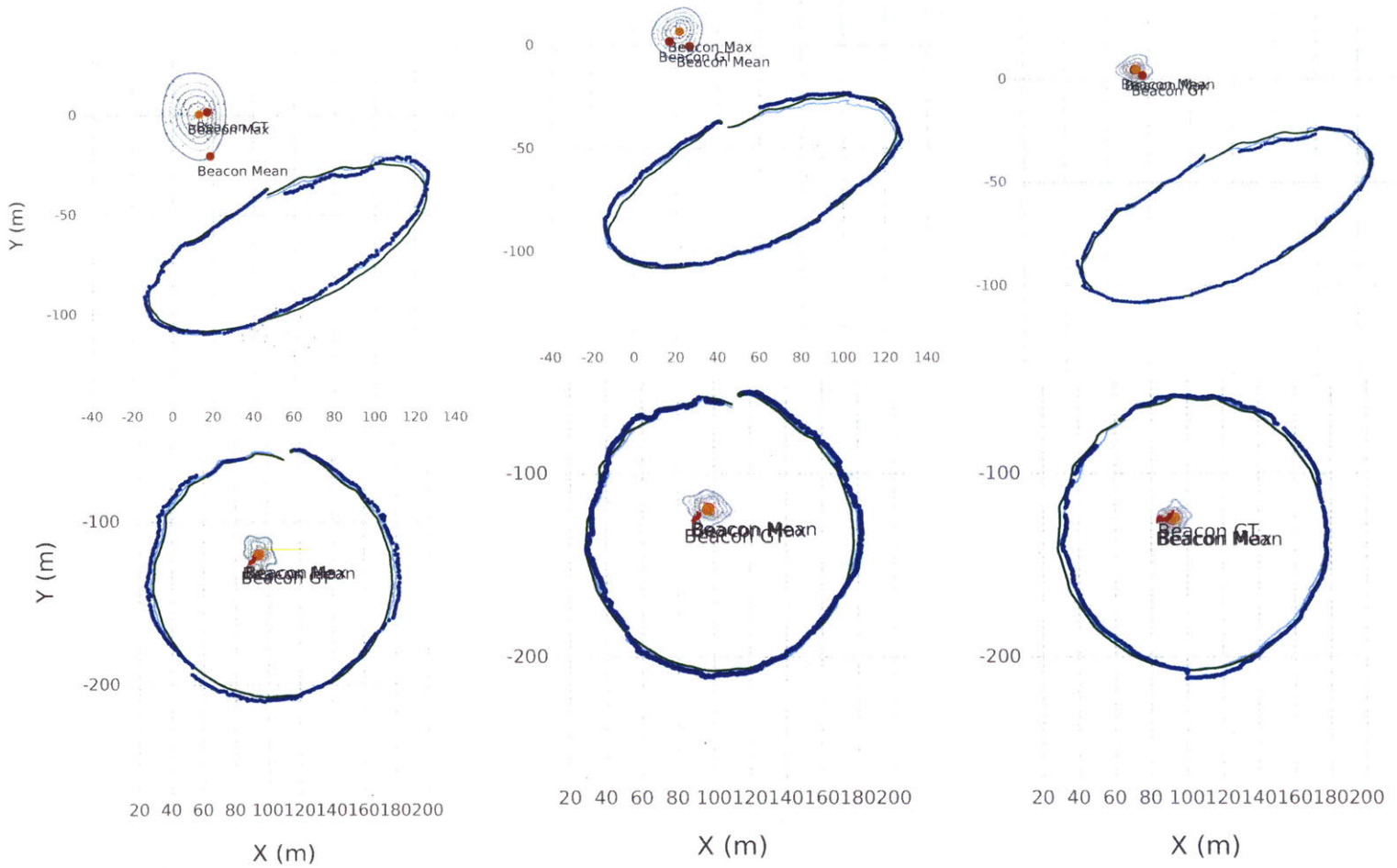

Figure 5-24: Marginal Posterior beacon distribution by number of SAS-factors. (*Top:*) *Stationary* Beacon $\{7, 14, 21\}$ factors total, 2 GPS Priors. (*Bottom:*) *Drifting* Beacon $\{9, 13, 21\}$ poses total, 3 GPS Priors. Each array has 8 elements. Ground truth trajectory shown in green and drifted trajectory shown in light blue.

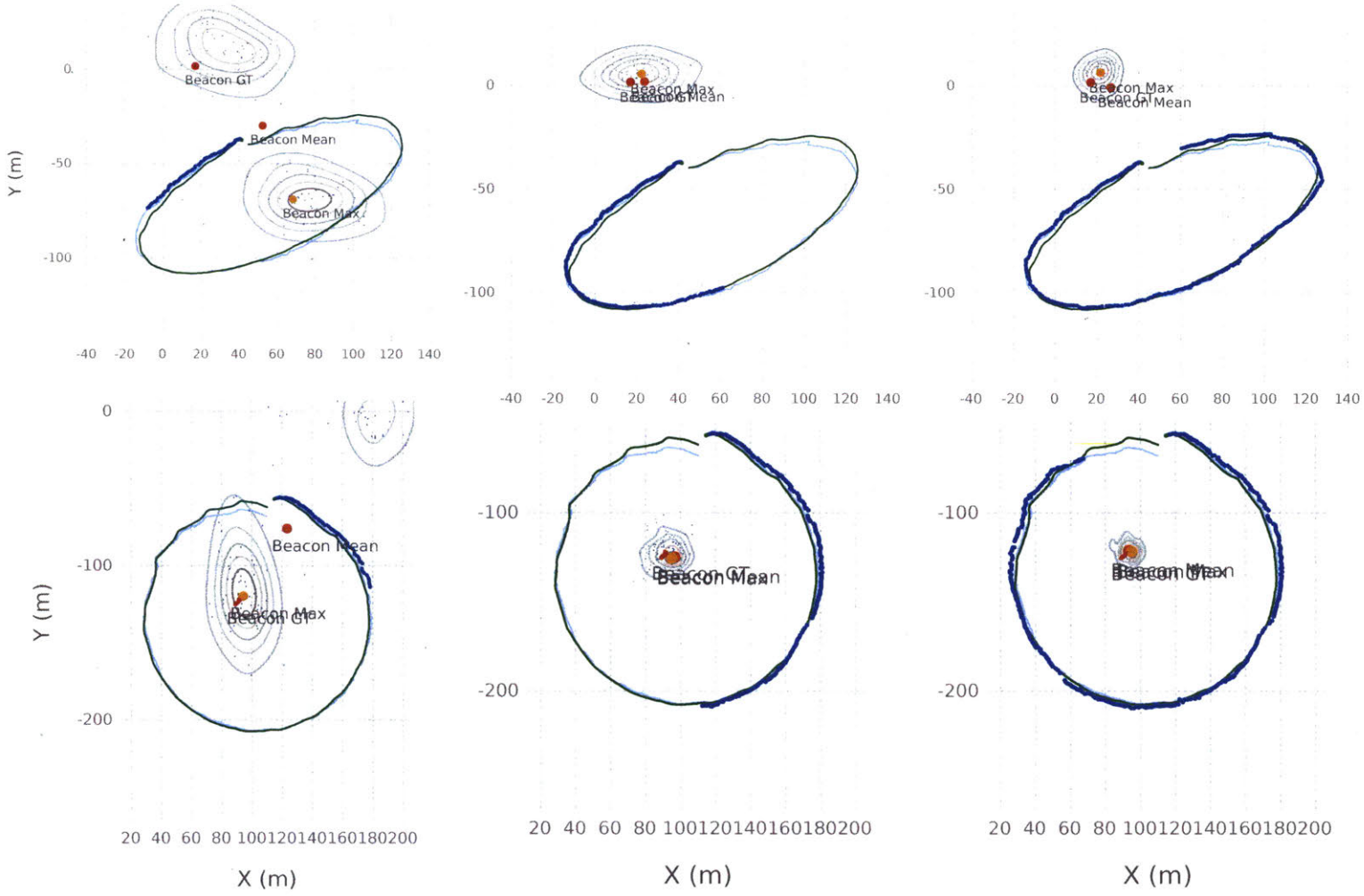

Figure 5-25: Marginal Posterior beacon distribution by Trajectory, $\{3, 7, 14\}$ SAS-factors total. (*Top:*) *Stationary* Beacon (*Bottom:*) *Drifting* Beacon. A few SAS-Factors are sufficient to resolve the ambiguous mode.

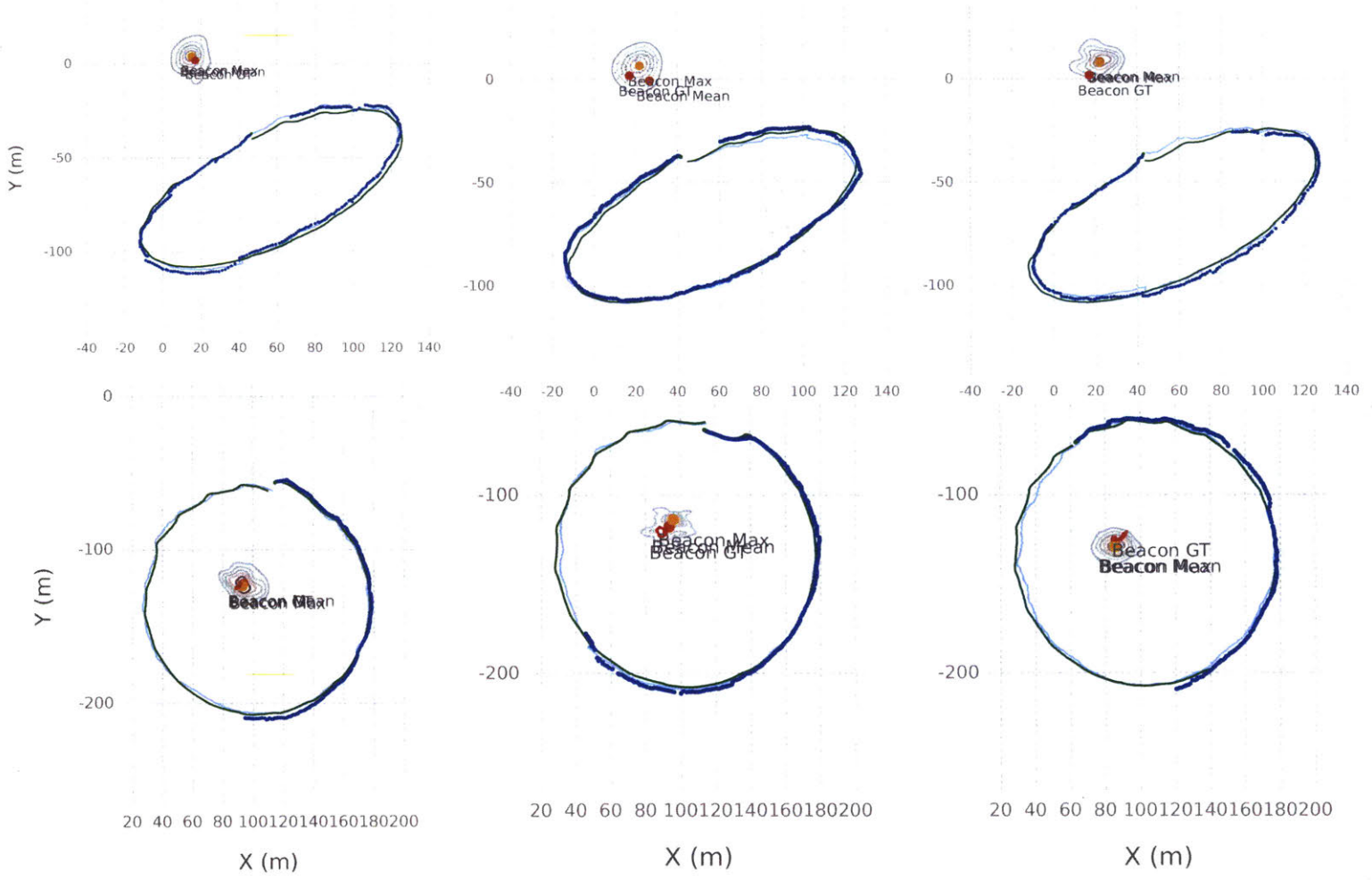

Figure 5-26: Marginal Posterior beacon distribution by Intra-Esxperiment Trial. (*Top:*) *Stationary* Beacon at mission start time $\{410, 720, 1050\}$ seconds respectively. (*Bottom:*) *Drifting* Beacon at mission start time $\{1550, 1952, 2301\}$ seconds respectively.

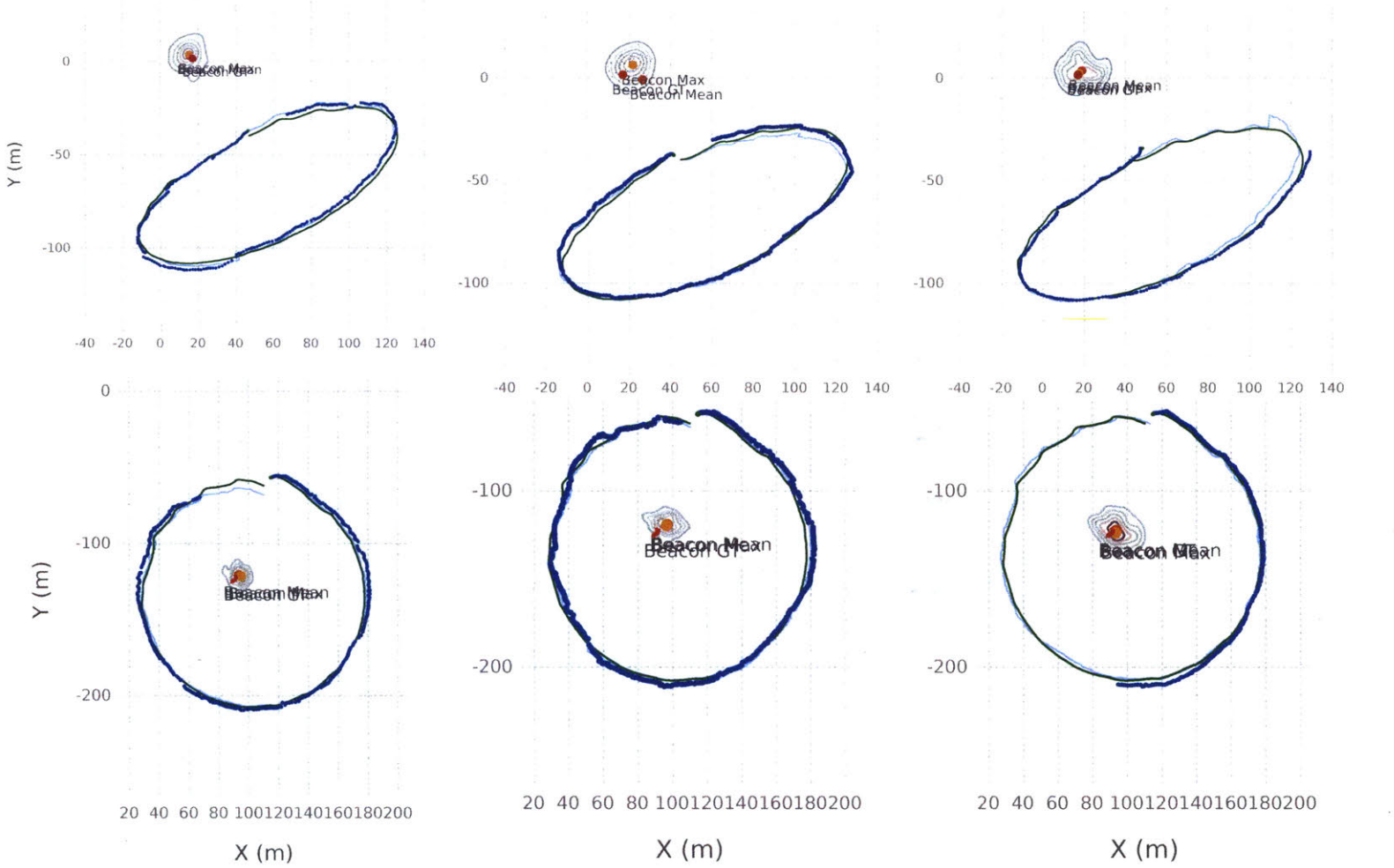

Figure 5-27: Marginal Posterior beacon distribution by Synthetic Array Length, $\{6, 8, 10\}$ seconds (elements) respectively. (*Top:*) *Stationary* Beacon. (*Bottom:*) *Drifting* Beacon. Longer arrays tradeoff accuracy with the vehicle's transit time and incurred navigational errors.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 6

Conclusion

There are a wealth of important reasons for studying, monitoring and working in our oceans, however operating in an underwater environment has been and remains a challenging, expensive and dangerous task, particularly for humans. Rapid advances in autonomous robotic systems over the last few decades provide an effective low-risk, low-cost solution, as navigation, autonomy and computation technology becomes increasingly sophisticated, and individual robotic platforms are now commercially available and exploited in many underwater applications. In the fields of ocean science, mapping, archaeology, oil and gas industry, fishing industry, naval operations, etc. these systems have already made a huge impact.

Now we look forward to fielding fleets of co-operating underwater robots able to carry out missions of increased complexity and capability, however the marine domain is fundamentally challenging for collaborative multi-robot operations. An unknown underwater environment is time-consuming to map and accurately model, especially with robots that have limited access to a high-quality localization solution such as GPS. Communications over distances greater than one hundred meters necessitates the use of acoustics (acomms), resulting in bandwidth and networking challenges. To successfully perform a highly dynamic co-operative task such as target-tracking in this environment, we must optimize the process of information-gathering, probabilistic inference over disparate, noisy sensor data, and communication of local information over a costly communication link.

The central topic of this thesis has been the behavior and control of co-operating robotic vehicles in order to enhance their collaborative capabilities of sensing, maneuvering and task-completion. We have shown two formulations of a dynamic target-tracking underwater mission, and validated both with experimental data collected in the field on our custom-built testbed ASVs.

6.1 Critical Analysis

Chapter 3 presented a de-centralized joint estimation and control framework for multiple underwater agents collaborating to track a maneuvering target. Range-only (TDOA) pursuit was experimentally demonstrated in a shallow-water acoustic environment with acoustics-in-the-loop, in which the acoustic modem was used as the measurement as well as the main communications channel. An experimental frequency analysis of the dynamic system showed that the TDMA cycle time was the main factor limiting performance when packet losses were low and vehicle speeds were relatively fast, i.e. the vehicles tracked the target up the maximum Nyquist rate of the information link. This articulates the distributed sensing challenge of increasing time delays when moving to larger fleets tracking over longer distances, which cannot be easily addressed from a filtering or control design perspective.

There are many possible solutions. Introducing a system of packet buffering could help to smooth trajectories and predict control inputs over a horizon, i.e. model predictive control and trajectory planning. Improving the modeling fidelity of the local dynamics places less emphasis on the capability of the communication network. Improving the communication link itself, through multiplexing, advanced coding/decoding techniques, is another rich field of research. The direction of this thesis has been to address and formulate the more general inference problem for the marine domain, of which tracking and pursuit is one possible mission, that is SLAM, or the simultaneous construction of a map of a robot's environment and the estimation of its state. Specifically, we have presented a novel non-Gaussian, multi-modal factor graph SLAM formulation for the underwater domain.

Much work has been done with point estimate representations and noisy sensor data represented by Gaussian distributions, and the state-of-the-art is an extremely powerful and capable tool. However, the marine domain presents two significant challenges to the usual formulation. Firstly, measurements are usually obtained from diverse sensors such as ranging and imaging sonars, optical modems, oceanographic instruments (pressure, temperature, bathymetry), and often require extensive filtering and parametrization to fuse in a Gaussian syntax. Secondly, acomms-in-the-loop between multiple robots significantly limits the amount of local information that can be shared over the network. Chapters 4 and 5 presented a non-Gaussian, non-parametric, multi-modal factor graph formulation of SLAM for marine robots undertaking a cooperative maneuvering target-tracking task underwater, utilizing the method of synthetic aperture sonar (SAS) to relate disparate acoustic measurements together in a consistent probabilistic framework. Although the method may be relatively slow and memory-intensive at present, there are many functional improvements being implemented on the codebase for significant speed-up, including multi-threading, in-place memory operations and fixed-lag solving.

Thus, the two main challenges are addressed by this work as follows. A matched-filter and conventional beamforming pipeline expresses raw acoustic data as measurement likelihood densities over bearing and range to the source. These are used directly to construct a reversible synthetic aperture sonar factor, i.e. informing the location of the beacon from the receiver and vice versa by computing a cross-correlation residual that can be used in the general minimization framework. The factor is a novel formulation of direct data fusion between raw acoustic waveforms and navigational measurements, used in a completely general SLAM framework. mm-iSAM [58] was the backend algorithm used to solve the joint inference problem. We showed results on experimental datasets collected in the Charles River that demonstrated the effectiveness of the SAS-factor, and the multi-modal nature of the resulting approximate marginal posterior distributions.

In terms of computation speed and resource-usage, the non-Gaussian multi-modal solver cannot be compared with existing state-of-the-art Gaussian solvers, with solve

times on the order of hours for factor graphs of approximately 1000 nodes. Performance with the approximate MCMC sampling method, however, only stand to improve with greater access to parallel-processing capabilities. Additionally, while the front-end is able to represent highly non-Gaussian measurement likelihoods, the measurements must remain consistent for the inference to be accurate, an ongoing challenge with acoustic data corrupted by non-white ambient noise.

6.2 Future Work

There are a number of logical extensions to the method that are the subject of ongoing future work. Distributing the information captured by each agent's factor graph to multiple collaborating agents such that they are able to share their local measurements in an efficient way is an important step to multi-vehicle operations. One solution is a local, incrementally updated fixed-lag factor graph onboard each vehicle, that may be iterated over in a short time, while agents intermittently communicate with a server-based factor graph that iterates over the full solve to produce more accurate estimates. Collaborating agents should be able to share full measurement likelihoods over the acoustic channel, as opposed to large amounts of raw data, or parametric point estimates.

Another interesting possibility is the addition of belief-space planning for future trajectories. We have seen that the performance of acoustics can vary spatially, most dramatically observed as acoustic dead zones, and that path-planning for multiple agents in the presence of ocean currents, mission objectives and accumulated inertial drift is a challenging problem. New factors would have to be formulated to represent potential information obtained in the future, tied to navigational, such that the inference solver can be used to simultaneously optimize future trajectories for maximum information gain.

Appendix A

ASV Specifications

Figure A-1: MRG-ASV *Nostromo* on the Charles River, Boston.

Table A.1: Hardware Components and Specifications

Part	Specification
Power	
LiFePo4 Batteries	100Ah, 4 cells, 12.8V nominal. Energy Density: 75.26 Wh/kg
Motor and Steering	
Minn-Kota Riptide 55 Trolling Motor	55lbs Maximum Thrust, 12V Nominal
Robotec LDC1430 Motor Driver	35V Maximum Battery Voltage, 30V Maximum Motor Voltage 120A Maximum Current RS232, 0-5V Analog or Pulse (RC) Command Modes Forward and Reverse Control
Servocity MEGA MS530-1 Servo	2000 oz-in Maximum Torque 0.9s Transit Time (60°) 7.4V nominal
Sensors	
OceanServer OS5000 3-Axis Compass	0.5° RMS Heading Roll & Pitch Rotation to 1° Tilt-compensated (Electronically Gimbaled) 1"x1"x0.3" footprint Hard and soft-iron compensation
uBlox NEO-6 GPS receiver	GPS and PPP 2.7V - 3.6V Nominal 2.5m Horizontal Position Accuracy 0.1m/s Velocity accuracy 0.5° Heading accuracy
Novatel GPS-701-GG Antenna	4.6V - 18V, 35mA current L1 Frequency
Tritech PA500 Altimeter	0.1-10m range 12V Nominal 500kHz 6°

Table A.2: Hardware Components and Specifications

Part	Specification
Electronics and Communication	
Arduino Mega 2560 Microcontroller	54 I/O pins (15 PWM) 16 Analog Inputs, 4 UART Serial Ports 7V-12V nominal, 20mA draw per I/O Pin, 50mA draw per 3.3V Pin 256KB flash memory
Ubiquiti Bullet-M BM2HP Wireless Radio	24V Maximum, POE 7W Maximum Power Consumption 28dBm Output Power, up to 54 Mbps
Freewave FGR2-PE 900 MHz	97km Range (with clear LOS) 1W Maximum Output Power 902-928 MHz Frequency Range 115.2 kbps Standard RF Data rate 6V-30V Nominal 550mA Transmit, 150mA Receive, 71mA Idle Draw
Gumstix Duovero Zephyr COM	Texas Instruments OMAP4430 Processor ARM Cortex-A9 Architecture 1GHz base clock speed, dual core 1GB LPDDR RAM Wireless and bluetooth modules 3.3V - 4.2V nominal

THIS PAGE INTENTIONALLY LEFT BLANK

Bibliography

[1] CANON: Controlled, Agile and Novel Observing Network. `https://www.mbari.org/science/upper-ocean-systems/canon/`. Accessed: 2019-0-01.

[2] MIT Marine Autonomy Bay. `http://oceanai.mit.edu/pavlab/pmwiki/pmwiki.php`. Accessed: 2019-06-01.

[3] MIT Sailing Pavilion Weather Station. `http://sailing.mit.edu/weather/`. Accessed: 2019-06-01.

[4] D. Ribas M. Carreras A. Mallios, P. Ridao and R. Camilli. Toward autonomous exploration in confined underwater environments. *Journal of Field Robotics*, 33(7):994 1012, 2016.

[5] Ian F Akyildiz, Dario Pompili, and Tommaso Melodia. Underwater acoustic sensor networks: research challenges. *Ad hoc networks*, 3(3):257 279, 2005.

[6] S Alexander, R Bishop, and Robert Ghrist. Capture pursuit games on unbounded domains. *Enseign. Math.(2)*, 55(1-2):103 125, 2009.

[7] A. Bahr. *Cooperative Localization for Autonomous Underwater Vehicles.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, February 2009.

[8] A. Bahr, J.J. Leonard, and M.F. Fallon. Cooperative localization for autonomous underwater vehicles. *Intl. J. of Robotics Research*, 28(6):714 728, 2009.

[9] T. Bailey and H.F. Durrant-Whyte. Simultaneous localisation and mapping (SLAM): Part I. *Robotics & Automation Magazine*, 13(2):99 110, June 2006.

[10] T. Bailey and H.F. Durrant-Whyte. Simultaneous localisation and mapping (SLAM): Part II. *Robotics & Automation Magazine*, 13(3):108 117, September 2006.

[11] Stephen Barkby, Stefan B Williams, Oscar Pizarro, and Michael V Jakuba. Bathymetric particle filter slam using trajectory maps. *The International Journal of Robotics Research*, 31(12):1409 1430, 2012.

[12] T. Bean, J. Canning, G. Beidler, M. O'Rourke, and D. B. Edwards. Designing and implementing collaborative behaviors for autonomous underwater vehicles. *Proceedings of UUST 2007*, pages 19 22, 2007.

[13] D. Bechler, M. S. Schlosser, and K. Kroschel. System for robust 3d speaker tracking using microphone array measurements. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 3, pages 2117 2122 vol.3, Sep. 2004.

[14] A. Bellettini and M. A. Pinto. Theoretical accuracy of synthetic aperture sonar micronavigation using a displaced phase-center antenna. *IEEE Journal of Oceanic Engineering*, 27(4):780 789, 2002.

[15] M. R. Benjamin, J. J. Leonard, H. Schmidt, and P. M. Newman. An Overview of MOOS-IvP and a brief users guide to the IvP Helm autonomy software. *CSAIL Technical Reports*, 2009.

[16] Wallace M Bessa, Max S Dutra, and Edwin Kreuzer. Depth control of remotely operated underwater vehicles using an adaptive fuzzy sliding mode controller. *Robotics and Autonomous Systems*, 56(8):670 677, 2008.

[17] Brian Bingham, Brendan Foley, Hanumant Singh, Richard Camilli, Katerina Delaporta, Ryan Eustice, Angelos Mallios, David Mindell, Christopher Roman, and Dimitris Sakellariou. Robotic tools for deep water archaeology: Surveying an ancient shipwreck with an autonomous underwater vehicle. *Journal of Field Robotics*, 27(6):702 717, 2010.

[18] Leo Bluestein. A linear filtering approach to the computation of discrete fourier transform. *IEEE Transactions on Audio and Electroacoustics*, 18(4):451 455, 1970.

[19] Andrew Branch, Mar M. Flexas, Brian Claus, Andrew F. Thompson, Yanwu Zhang, Evan B. Clark, Steve Chien, David M. Fratantoni, James C. Kinsey, Brett Hobson, et al. Front delineation and tracking with multiple underwater vehicles. *Journal of Field Robotics*, 36(3):568 586, 2019.

[20] L. Brignone, J. Alves, and J. Opderbecke. GREX sea trials: First experiences in multiple underwater vehicle coordination based on acoustic communication. In *OCEANS*, pages 1 6. IEEE, 2009.

[21] A. Caiti, V. Calabro, G. Dini, A. L. Duca, and A. Munafo. AUVs as mobile nodes in acoustic communication networks: Field experience at the UAN10 experiment. In *OCEANS, 2011 IEEE-Spain*, pages 1 6. IEEE, 2011.

[22] Andrea Caiti, P. Felisberto, T. Husoy, S. M. Jesus, I. Karasalo, R. Massimelli, Tor Arne Reinen, and A. Silva. UAN underwater acoustic network. In *OCEANS 2011 IEEE-Spain*, pages 1 7. IEEE, 2011.

[23] Romano Capocci, Gerard Dooly, Edin Omerdić, Joseph Coleman, Thomas Newe, and Daniel Toal. Inspection-class remotely operated vehicles a review. *Journal of Marine Science and Engineering*, 5(1):13, 2017.

[24] Y.T. Chan and K.C. Ho. A simple and efficient estimator for hyperbolic location. *Signal Processing, IEEE Transactions on*, 42(8):1905 1915, 1994.

[25] Xianhui Che, Ian Wells, Gordon Dickers, and Paul Kear. Re-evaluation of rf electromagnetic communication in underwater sensor networks. *IEEE Communications Magazine*, 2010.

[26] M. Cheung, J. Leighton, and F. Hover. Multi-armed Bandit Formulation for Autonomous Mobile Acoustic Relay Adaptive Positioning. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[27] M. Chitre. *Underwater acoustic communications in warm shallow water channels*. PhD thesis, National University of Singapore, 2006.

[28] M. Chitre. Path planning for cooperative underwater range-only navigation using a single beacon. In *2010 International Conference on Autonomous and Intelligent Systems, AIS 2010*, pages 1 6, June 2010.

[29] Timothy H Chung, Joel W Burdick, and Richard M Murray. A decentralized motion coordination strategy for dynamic target tracking. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 2416 2422. IEEE, 2006.

[30] Timothy H Chung, Geoffrey A Hollinger, and Volkan Isler. Search and pursuit-evasion in mobile robotics. *Autonomous robots*, 31(4):299, 2011.

[31] C. M. Clark, C. Forney, E. Manii, D. Shinzaki, C. Gage, M. Farris, C.G. Lowe, and M. Moline. Tracking and Following a Tagged leopard shark with an Autonomous Underwater Vehicle. *Journal of Field Robotics*, 30(3):309 322, 2013.

[32] Brian Claus, James H. Kepper IV, Stefano Suman, and James C. Kinsey. Closed-loop one-way-travel-time navigation using low-grade odometry for autonomous underwater vehicles. *Journal of Field Robotics*, 35(4):421 434, 2018.

[33] P. Corke, C. Detweiler, M. Dunbabin, M. Hamilton, D. Rus, and I. Vasilescu. Experiments with underwater robot localization and tracking. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 4556 4561. IEEE, 2007.

[34] N. A. Cruz, B.M. Ferreira, A.C. Matos, C. Petrioli, R. Petroccia, and D. Spaccini. Implementation of an underwater acoustic network using multiple heterogeneous vehicles. In *OCEANS*, pages 1 10. IEEE, 2012.

[35] Nuno A Cruz and Aníbal C Matos. Reactive auv motion for thermocline tracking. In *OCEANS'10 IEEE SYDNEY*, pages 1 6. IEEE, 2010.

[36] Rongxin Cui, Yang Li, and Weisheng Yan. Mutual information-based multi-auv path planning for scalar field sampling using multidimensional rrt. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 46(7):993 1004, 2015.

[37] J. Curcio, J. Leonard, and A. Patrikalakis. SCOUT - a low cost autonomous surface platform for research in cooperative autonomy. In *Proc. of the IEEE/MTS OCEANS Conf. and Exhibition*, Washington DC, September 2005.

[38] J. Curcio, J. Leonard, J. Vaganay, A. Patrikalakis, A. Bahr, D. Battle, H. Schmidt, and M. Grund. Experiments in Moving Baseline Navigation using autonomous surface craft. In *OCEANS, 2005. Proceedings of MTS/IEEE*, pages 730 735. IEEE, 2005.

[39] J. Curcio, J. J. Leonard, J. Vaganay, A. Patrikalakis, A. Bahr, D. Battle, H. Schmidt, and M. Grund. Experiments in Moving Baseline Navigation using Autonomous Surface Craft. In *Proc. of the IEEE/MTS OCEANS Conf. and Exhibition*, volume 1, pages 730 735, September 2005.

[40] Louis J. Cutrona. Comparison of sonar system performance achievable using synthetic?aperture techniques with the performance achievable by more conventional means. *The Journal of the Acoustical Society of America*, 58(2):336 348, 1975.

[41] J. Das, F. Py, T. Maughan, T. O'Reilly, M. Messie, J. Ryan, K. Rajan, and G.S. Sukhatme. Simultaneous tracking and sampling of dynamic oceanographic features with autonomous underwater vehicles and lagrangian drifters. In *Experimental Robotics*, pages 541 555. Springer, 2014.

[42] T.A. Davis, J.R. Gilbert, S.I. Larimore, and E.G. Ng. A column approximate minimum degree ordering algorithm. *ACM Trans. Math. Softw.*, 30(3):353 376, 2004.

[43] F. Dellaert and M. Kaess. Square Root SAM: Simultaneous localization and mapping via square root information smoothing. *Intl. J. of Robotics Research*, 25(12):1181 1203, December 2006.

[44] Frank Dellaert, Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Monte Carlo Localization for Mobile Robots. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, May 1999.

[45] Joseph H DiBiase, Harvey F Silverman, and Michael S Brandstein. Robust localization in reverberant rooms. In *Microphone Arrays*, pages 157 180. Springer, 2001.

[46] A-R Diercks, VL Asper, M Woolsey, JL Williams, and F Cantelas. Niust auv's study shipwrecks in the northern gulf of mexico. In *2010 IEEE/OES Autonomous Underwater Vehicles*, pages 1 5. IEEE, 2010.

[47] Shenfu Dong, Janet Sprintall, Sarah T Gille, and Lynne Talley. Southern ocean mixed-layer depth from argo float profiles. *Journal of Geophysical Research: Oceans*, 113(C6), 2008.

[48] Marek Doniec, Carrick Detweiler, Iuliu Vasilescu, and Daniela Rus. Using optical communication for remote underwater robot operation. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4017 4022. IEEE, 2010.

[49] R. Eustice, M. Walter, and J. Leonard. Sparse extended information filters: Insights into sparsification. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3281 3288. IEEE, 2005.

[50] Ryan M Eustice, Hanumant Singh, and Louis L Whitcomb. Synchronous-clock, one-way-travel-time acoustic navigation for underwater vehicles. *journal of field robotics*, 28(1):121 136, 2011.

[51] M.F. Fallon, M. Kaess, H. Johannsson, and J.J. Leonard. Efficient AUV navigation fusing acoustic ranging and side-scan sonar. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Shanghai, China, May 2011.

[52] N Farr, A Bowen, J Ware, C Pontbriand, and M Tivey. An integrated, underwater optical/acoustic communications system. In *OCEANS'10 IEEE SYDNEY*, pages 1 6. IEEE, 2010.

[53] Jay A. Farrell, Shou Pang, Wei Li, and Richard Arrieta. Chemical plume tracing experimental results with a remus auv. In *Oceans 2003. Celebrating the Past... Teaming Toward the Future (IEEE Cat. No. 03CH37492)*, volume 2, pages 962 968. IEEE, 2003.

[54] Fausto Ferreira, Gianmarco Veruggio, Massimo Caccia, and Gabriele Bruzzone. Real-time optical slam-based mosaicking for unmanned underwater vehicles. *Intelligent Service Robotics*, 5(1):55 71, 2012.

[55] E. Fiorelli, N.E. Leonard, P. Bhatta, D.A. Paley, R. Bachmayer, and D.M. Fratantoni. Multi-AUV control and adaptive sampling in Monterey Bay. *Oceanic Engineering, IEEE Journal of*, 31(4):935 948, 2006.

[56] EM Fischell, N Rypkema, and H Schmidt. Relative acoustic navigation for sensing with low-cost auvs. In *ICRA 2016 Workshop on Marine Robot Localization and Navigation*, 2016.

[57] T. I. Fossen. *Guidance and control of ocean vehicles.* John Wiley & Sons Inc, 1994.

[58] D. Fourie. *Multi-Modal and Inertial Sensor Solutions to Navigation-type Factor Graphs.* PhD thesis, Massachusetts Institute of Technology and Woods Hole Oceanographic Institution, 2017.

[59] D. Fourie and contributors. Caesar.jl, 2018. `https://github.com/JuliaRobotics/Caesar.jl`.

[60] D. Fourie, J.J. Leonard, and M. Kaess. A nonparametric belief solution to the Bayes tree. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, IROS*, Daejeon, Korea, Oct 2016.

[61] Dehann Fourie. Rome.jl.

[62] D. M. Fratantoni and J. M. Lund. Glider operations in buzzard's bay, ma. Technical report, 2006.

[63] L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski, and K. Ball. The WHOI micro-modem: an acoustic communications and navigation system for multiple platforms. In *OCEANS, 2005. Proceedings of MTS/IEEE*, volume 2, pages 1086 1092, Sept. 2005.

[64] Alan E Gelfand and Adrian FM Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, 85(410):398 409, 1990.

[65] M.D. Gillette and H.F. Silverman. A linear closed-form algorithm for source localization from time-differences of arrival. *Signal Processing Letters, IEEE*, 15:1 4, 2008.

[66] John Gould, Dean Roemmich, Susan Wijffels, Howard Freeland, Mark Ignaszewsky, Xu Jianping, Sylvie Pouliquen, Yves Desaubies, Uwe Send, Kopillil Radhakrishnan, et al. Argo profiling floats bring new era of in situ ocean observations. *Eos, Transactions American Geophysical Union*, 85(19):185 191, 2004.

[67] Mark Grasmueck, Gregor P Eberli, David A Viggiano, Thiago Correa, Glenda Rathwell, and Jiangang Luo. Autonomous underwater vehicle (auv) mapping reveals coral mound distribution, morphology, and oceanography in deep water of the straits of florida. *Geophysical Research Letters*, 33(23), 2006.

[68] Alexander G Gray and Andrew W Moore. Nonparametric density estimation: Toward computational tractability. In *Proceedings of the 2003 SIAM International Conference on Data Mining*, pages 203 211. SIAM, 2003.

[69] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard. A tutorial on graph-based SLAM. *Intelligent Transportation Systems Magazine, IEEE*, 2(4):31 43, 2010.

[70] Matthew Grund, Lee Freitag, James Preisig, and Keenan Ball. The plusnet underwater communications system: acoustic telemetry for undersea surveillance. In *OCEANS 2006*, pages 1 5. IEEE, 2006.

[71] Karol Hausman, Jörg Müller, Abishek Hariharan, Nora Ayanian, and Gaurav S Sukhatme. Cooperative multi-robot control for target tracking with onboard sensing. *The International Journal of Robotics Research*, 34(13):1660 1677, 2015.

[72] M. P. Hayes and P. T. Gough. Synthetic Aperture Sonar: a review of current status. *IEEE Journal of Oceanic Engineering*, 34(3):207 224, July 2009.

[73] Bo He, Yan Liang, Xiao Feng, Rui Nian, Tianhong Yan, Minghui Li, and Shujing Zhang. Auv slam and experiments using a mechanical scanning forward-looking sonar. *Sensors*, 12(7):9386 9410, 2012.

[74] Robert Headrick and Lee Freitag. Growth of underwater communication technology in the us navy. *IEEE Communications Magazine*, 47(1):80 82, 2009.

[75] Anthony J Healey and David Lienard. Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles. *IEEE journal of Oceanic Engineering*, 18(3):327 339, 1993.

[76] J-P Hermand and William I Roderick. Acoustic model-based matched filter processing for fading time-dispersive ocean channels: Theory and experiment. *IEEE Journal of Oceanic Engineering*, 18(4):447 465, 1993.

[77] F. Hidalgo and T. Braunl. Review of underwater SLAM techniques. In *2015 6th International Conference on Automation, Robotics and Applications (ICARA)*, pages 306 311, Feb 2015.

[78] Geoffrey Hollinger, Sanjiv Singh, Joseph Djugash, and Athanasios Kehagias. Efficient multi-robot search for a moving target. *The International Journal of Robotics Research*, 28(2):201 219, 2009.

[79] G. Huang, M. Kaess, J. Leonard, and S. Roumeliotis. Analytically-Selected Multi-Hypothesis Incremental MAP Estimation. In *International Conference on Acoustics, Speech, and Signal Processing*, British Columbia, Canada, May 2013.

[80] T. A. Huang and M. Kaess. Towards acoustic structure from motion for imaging sonar. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 758 765. IEEE, 2015.

[81] Alexander T Ihler, Erik B Sudderth, William T Freeman, and Alan S Willsky. Efficient multiscale sampling from products of Gaussian mixtures. *Advances in Neural Information Processing Systems*, 16:1 8, 2004.

[82] Shojiro Ishibashi, Hiroshi Yoshida, Hiroyuki Osawa, Tomoya Inoue, Junichiro Tahara, Kazuaki Ito, Yohitaka Watanabe, Takao Sawa, Tadahiro Hyakudome, and Taro Aoki. A ROV "ABISMO" for the inspection and sampling in the deepest ocean and its operation support system. In *OCEANS 2008-MTS/IEEE Kobe Techno-Ocean*, pages 1 6. IEEE, 2008.

[83] S.J. Julier and J.K. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *SPIE 12th International Symposium on Aerospace /Defense Sensing, Simulation, and Controls*, volume 3, pages 3 2. Orlando, FL, 1997.

[84] Bong-Huan Jun, Jin-Yeong Park, Fill-Youb Lee, Pan-Mook Lee, Chong-Moo Lee, Kihun Kim, Young-Kon Lim, and Jun-Ho Oh. Development of the auv 'isimi' and a free running test in an ocean engineering basin. *Ocean Engineering*, 36(1):2 14, 2009. Autonomous Underwater Vehicles.

[85] M. Kaess, V. Ila, R. Roberts, and F. Dellaert. The Bayes tree: An algorithmic foundation for probabilistic robot mapping. In *Intl. Workshop on the Algorithmic Foundations of Robotics, WAFR*, Singapore, December 2010.

[86] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research*, 31:217 236, February 2012.

[87] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Trans. Robotics*, 24(6):1365 1378, December 2008.

[88] Peter Kimball, John Bailey, Sarah Das, Rocky Geyer, Trevor Harrison, Clay Kunz, Kevin Manganini, Ken Mankoff, Katie Samuelson, Thomas Sayre-McCord, et al. The whoi jetyak: An autonomous surface vehicle for oceanographic research in shallow or dangerous waters. In *Autonomous Underwater Vehicles (AUV), 2014 IEEE/OES*, pages 1 7. IEEE, 2014.

[89] James C. Kinsey, Ryan M. Eustice, and Louis L. Whitcomb. A survey of underwater vehicle navigation: Recent advances and new challenges. In *IFAC Conference of Manoeuvering and Control of Marine Craft*, Lisbon, Portugal, September 2006. Invited paper.

[90] James C Kinsey, Dana R Yoerger, Michael V Jakuba, Rich Camilli, Charles R Fisher, and Christopher R German. Assessing the deepwater horizon oil spill with the sentry autonomous underwater vehicle. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 261 267. IEEE, 2011.

[91] Charles Knapp and Glifford Carter. The generalized correlation method for estimation of time delay. *IEEE transactions on acoustics, speech, and signal processing*, 24(4):320 327, 1976.

[92] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Inform. Theory*, 47(2), February 2001.

[93] Ryo Kurazume, Shigemi Nagata, and Shigeo Hirose. Cooperative positioning with multiple robots. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 1250 1257. IEEE, 1994.

[94] Lucas C. Laurindo, Arthur J. Mariano, and Rick Lumpkin. An improved near-surface velocity climatology for the global ocean from drifter observations. *Deep Sea Research Part I: Oceanographic Research Papers*, 124:73 92, 2017.

[95] J. J. Leonard and A. Bahr. Autonomous underwater vehicle navigation. In M. R. Dhanak and N. I. Xiros, editors, *Springer Handbook of Ocean Engineering*. Springer, 2016.

[96] J. J. Leonard and H. F. Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *Proc. IEEE Int. Workshop on Intelligent Robots and Systems*, pages 1442 1447, Osaka, Japan, 1991.

[97] Naomi E Leonard, Derek A Paley, Russ E Davis, David M Fratantoni, Francois Lekien, and Fumin Zhang. Coordinated control of an underwater glider fleet in an adaptive ocean sampling field experiment in monterey bay. *Journal of Field Robotics*, 27(6):718 740, 2010.

[98] C. D. Cadena Lerma, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. Leonard. Past, present, and future of simultaneous localization and mapping: Towards the robust-perception age. *IEEE Trans. Robotics*, 2016.

[99] Pierre FJ Lermusiaux, Tapovan Lolla, Patrick J Haley Jr, Konuralp Yigit, Mattheus P Ueckermann, Thomas Sondergaard, and Wayne G Leslie. Science of autonomy: Time-optimal path planning and adaptive sampling for swarms of ocean vehicles. In *Springer Handbook of Ocean Engineering*, pages 481 498. Springer, 2016.

[100] J. Li, M. Kaess, R. M. Eustice, and M. Johnson-Roberson. Pose-graph slam using forward-looking sonar. *IEEE Robotics and Automation Letters*, 3(3):2330 2337, July 2018.

[101] Wei Li, Jay A Farrell, Shuo Pang, and Richard M Arrieta. Moth-inspired chemical plume tracing on an autonomous underwater vehicle. *IEEE Transactions on Robotics*, 22(2):292 307, 2006.

[102] X Rong Li and Vesselin P Jilkov. Survey of maneuvering target tracking. part i. dynamic models. *IEEE Transactions on aerospace and electronic systems*, 39(4):1333 1364, 2003.

[103] E. Liao, G. Hollinger, J. Djugash, and S. Singh. Preliminary results in tracking mobile targets using range sensors from multiple robots. In *Distributed Autonomous Robotic Systems 7*, pages 125 134. Springer, 2006.

[104] Yukun Lin, Jerry Hsiung, Richard Piersall, Connor White, Christopher G Lowe, and Christopher M Clark. A multi-autonomous underwater vehicle system for autonomous tracking of marine life. *Journal of Field Robotics*, 34(4):757 774, 2017.

[105] F. Lu and E. Milios. Globally consistent range scan alignment for environmental mapping. *Autonomous Robots*, 4:333 349, April 1997.

[106] X. Lurton. *An Introduction to Underwater Acoustics: Principles and Applications.* Geophysical Sciences Series. Springer, 2002.

[107] Ivan Markovic and Ivan Petrovic. Speaker localization and tracking with a microphone array on a mobile robot using von mises distribution and particle filtering. *Robotics and Autonomous Systems*, 58(11):1185 1196, 2010.

[108] Agostino Martinelli, Frederic Pont, and Roland Siegwart. Multi-robot localization using relative observations. In *Proceedings of the 2005 IEEE international conference on robotics and automation*, pages 2797 2802. IEEE, 2005.

[109] Ruben Martinez-Cantin and José A Castellanos. Unscented slam for large-scale outdoor environments. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3427 3432. IEEE, 2005.

[110] Pramod Maurya, A Pedro Aguiar, and Antonio Pascoal. Marine vehicle path following using inner-outer loop control. *IFAC Proceedings Volumes*, 42(18):38 43, 2009.

[111] P. H. Milne. *Underwater Acoustic Positioning Systems.* London: E. F. N. Spon, 1983.

[112] Y. Mostofi, T.H. Chung, R.M. Murray, and J.W. Burdick. Communication and sensing trade-offs in decentralized mobile sensor networks: a cross-layer design approach. In *International Symposium on Information Processing in Sensor Networks*, page 16, 2005.

[113] Roozbeh Mottaghi and Richard Vaughan. An integrated particle filter and potential field method applied to cooperative multi-robot target tracking. *Autonomous Robots*, 23(1):19 35, 2007.

[114] A. I. Mourikis and S. I. Roumeliotis. Performance analysis of multirobot cooperative localization. *IEEE Trans. Robotics*, 22(4):666 681, 2006.

[115] Walter H Munk. Sound channel in an exponentially stratified ocean, with application to sofar. *The Journal of the Acoustical Society of America*, 55(2):220 226, 1974.

[116] H. Schmidt N. R. Rypkema, E. M. Fischell. One-way travel-time inverted ultra-short baseline localization for low-cost autonomous underwater vehicles. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages pre print, 2017.

[117] P. Newman and J. Leonard. Pure range-only sub-sea SLAM. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, volume 2, pages 1921 1926, September 2003.

[118] P. M. Newman, J. J. Leonard, and R. J. Rikoski. Towards constant-time SLAM on an autonomous underwater vehicle using synthetic aperture sonar. *Robotics Research*, pages 409 420, 2005.

[119] P.M. Newman. MOOS - a mission oriented operating suite. Technical Report OE2003-07, Department of Ocean Engineering, MIT, 2003.

[120] Richard Nowakowski and Peter Winkler. Vertex-to-vertex pursuit in a graph. *Discrete Mathematics*, 43(2-3):235 239, 1983.

[121] N. R. Rypkema O. A. Viquez, E. M. Fischell and H. Schmidt. Design of a general autonomy payload for low-cost auv r&d. In *Autonomous Underwater Vehicles (AUV), 2016 IEEE/OES*, pages 151 155. IEEE, 2016.

[122] Julie N Oswald, Jay Barlow, and Thomas F Norris. Acoustic identification of nine delphinid species in the eastern tropical pacific ocean. *Marine mammal science*, 19(1):20 037, 2003.

[123] Derek A Paley, Fumin Zhang, and Naomi Ehrich Leonard. Cooperative control for ocean sampling: The glider coordinated control system. *IEEE Transactions on Control Systems Technology*, 16(4):735 744, 2008.

[124] Jim Partan, Jim Kurose, and Brian Neil Levine. A survey of practical issues in underwater networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 11(4):23 33, 2007.

[125] L. Paull, M. Seto, J. J. Leonard, and H. Li. A probabilistic mobile robot area coverage framework and its application to cooperative autonomous seabed surveying. *Intl. J. of Robotics Research*, 2017. To Appear.

[126] Liam Paull, Sajad Saeedi, Mae Seto, and Howard Li. Sensor-driven online coverage planning for autonomous underwater vehicles. *IEEE/ASME Transactions on Mechatronics*, 18(6):1827 1838, 2012.

[127] Liam Paull, Sajad Saeedi, Mae Seto, and Howard Li. AUV navigation and localization: A review. *Oceanic Engineering, IEEE Journal of*, 39(1):131 149, 2014.

[128] Liam Paull, Mae Seto, John J Leonard, and Howard Li. Probabilistic cooperative mobile robot area coverage and its application to autonomous seabed mapping. *The International Journal of Robotics Research*, 37(1):21 45, 2018.

[129] Judea Pearl. Bayesian networks. *Department of Statistics, UCLA*, 2011.

[130] Arvind A Pereira, Jonathan Binney, Geoffrey A Hollinger, and Gaurav S Sukhatme. Risk-aware path planning for autonomous underwater vehicles using predictive ocean models. *Journal of Field Robotics*, 30(5):741 762, 2013.

[131] Stephanie Petillot and Henrik Schmidt. Exploiting adaptive and collaborative auv autonomy for detection and characterization of internal waves. *IEEE Journal of Oceanic Engineering*, 39(1):150 164, 2013.

[132] Alexander B Phillips, Naomi Gold, Nick Linton, Catherine A Harris, Ella Richards, Robert Templeton, Sebastian Thuné, Jeremy Sitbon, Marcus Muller, Iain Vincent, et al. Agile design of low-cost autonomous underwater vehicles. In *OCEANS 2017-Aberdeen*, pages 1 7. IEEE, 2017.

[133] Shokoofeh Pourmehr, Jake Bruce, Jens Wawerla, and Richard Vaughan. A sensor fusion framework for finding an hri partner in crowd. In *IEEE Int. Conf. on Intelligent Robots and Systems, Workshop on Designing and Evaluating Social Robots for Public Settings*, 2015.

[134] Michael Purcell, Dave Gallo, Greg Packard, Mark Dennett, Marcel Rothenbeck, Andy Sherrell, and Sylvain Pascaud. Use of remus 6000 auvs in the search for the air france flight 447. In *OCEANS'11 MTS/IEEE KONA*, pages 1 7. IEEE, 2011.

[135] T. O. Sabo R. E. Hansen, H. J. Callow and S. A. V Synnes. Challenges in seafloor imaging and mapping with synthetic aperture sonar. *IEEE Transactions on Geoscience and Remote Sensing*, 49(10):3677 3687, 2011.

[136] L Rabiner, R W Schafer, and C Rader. The chirp z-transform algorithm. *IEEE transactions on audio and electroacoustics*, 17(2):86 92, 1969.

[137] A. Ranganathan, M. Kaess, and F. Dellaert. Loopy SAM. In *Intl. Joint Conf. on Artificial Intelligence*, pages 2191 2196, Hyderabad, India, 2007.

[138] Caleb Rascon and Ivan Meza. Localization of sound sources in robotics: A review. *Robotics and Autonomous Systems*, 96:184 210, 2017.

[139] B. Reed. Multiple-vehicle resource-constrained navigation in the deep ocean. Master's thesis, Massachusetts Institute of Technology, 2011.

[140] B. Reed, J. Leighton, M. Stojanovic, and F. Hover. Multi-vehicle Dynamic Pursuit using Underwater Acoustics. In *International Symposium on Robotics Research (ISRR)*, 2013.

[141] Brooks Louis-Kiguchi Reed. *Controller design for underwater vehicle systems with communication constraints.* PhD thesis, Massachusetts Institute of Technology, 2015.

[142] D. Ribas, P. Ridao, J.D. Tardós, and J. Neira. Underwater SLAM in man-made structured environments. *Journal of Field Robotics*, 25(11-12):898 921, 2008.

[143] Ian C Robbins, GJ Kirkpatrick, Shelley M Blackwell, J Hillier, Charles A Knight, and Mark A Moline. Improved monitoring of habs using autonomous underwater vehicles (auv). *Harmful Algae*, 5(6):749 761, 2006.

[144] Dean Roemmich, Gregory C. Johnson, Stephen Riser, Russ Davis, John Gilson, W. Brechner Owens, Silvia L. Garzoli, Claudia Schmid, and Mark Ignaszewski. The argo program: Observing the global ocean with profiling floats. *Oceanography*, 22(2):34 43, 2009.

[145] D. M. Rosen, M. Kaess, and J. J. Leonard. RISE: An incremental trust-region method for robust online sparse least-squares estimation. *Robotics and Automation, IEEE Transactions on*, 30(5):1091 1108, October 2014.

[146] S. I. Roumeliotis and G. A. Bekey. Distributed multirobot localization. *Robotics and Automation, IEEE Transactions on*, 18(5):781 795, 2002.

[147] Stergios I Roumeliotis and Ioannis M Rekleitis. Propagation of uncertainty in cooperative multirobot localization: Analysis and experimental results. *Autonomous Robots*, 17(1):41 54, 2004.

[148] N. R. Rypkema. *Underwater & Out of Sight: Towards Ubiquity in Underwater Robotics*. PhD thesis, Massachusetts Institute of Technology, 2019.

[149] N. R. Rypkema, E. M. Fischell, and H. Schmidt. One-way travel-time inverted ultra-short baseline localization for low-cost autonomous underwater vehicles. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4920 4926, May 2017.

[150] Sajad Saeedi, Michael Trentini, Mae Seto, and Howard Li. Multiple-robot simultaneous localization and mapping: A review. *Journal of Field Robotics*, 33(1):3 46, 2016.

[151] Sanem Sariel, Tucker Balch, and Nadia Erdogan. Naval mine countermeasure missions. *IEEE robotics & automation magazine*, 15(1):45 52, 2008.

[152] Peter M Saunders. Practical conversion of pressure to depth. *Journal of Physical Oceanography*, 11(4):573 574, 1981.

[153] Ralph Schmidt. Multiple emitter location and signal parameter estimation. *IEEE transactions on antennas and propagation*, 34(3):276 280, 1986.

[154] B. W. Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

[155] Hanumant Singh, Roy Armstrong, Fernando Gilbes, Ryan Eustice, Chris Roman, Oscar Pizarro, and Juan Torres. Imaging coral i: imaging coral habitats with the seabed auv. *Subsurface Sensing Technologies and Applications*, 5(1):25 42, 2004.

[156] Hanumant Singh, Ali Can, Ryan Eustice, Steve Lerner, Neil McPhee, and Chris Roman. Seabed auv offers new platform for high-resolution imaging. *Eos, Transactions American Geophysical Union*, 85(31):289 296, 2004.

[157] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M.I. Jordan, and S.S. Sastry. Kalman Filtering with intermittent observations. *IEEE Transactions on Automatic Control*, 49(9):1453 1464, 2004.

[158] David A Smallwood and Louis L Whitcomb. Model-based dynamic positioning of underwater robotic vehicles: theory and experiment. *IEEE Journal of Oceanic Engineering*, 29(1):169 186, 2004.

[159] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In *Autonomous Robot Vehicles*, pages 167 193. Springer Verlag, 1990.

[160] Jeff Snyder. Doppler velocity log (dvl) navigation for observation-class rovs. In *OCEANS 2010 MTS/IEEE SEATTLE*, pages 1 9. IEEE, 2010.

[161] J.M. Soares, A.P. Aguiar, A.M. Pascoal, and A. Martinoli. Joint ASV/AUV range-based formation control: Theory and experimental results. In *IEEE Conference on Robotics and Automation (ICRA), 2013*, pages 5579 5585. IEEE, 2013.

[162] Petre Stoica and Arye Nehorai. Performance study of conditional and unconditional direction-of-arrival estimation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(10):1783 1795, 1990.

[163] M Stojanovic and L Freitag. Multiuser undersea acoustic communications in the presence of multipath propagation. In *MTS/IEEE Oceans 2001. An Ocean Odyssey. Conference Proceedings (IEEE Cat. No. 01CH37295)*, volume 4, pages 2165 2169. IEEE, 2001.

[164] Milica Stojanovic. On the relationship between capacity and distance in an underwater acoustic communication channel. *ACM SIGMOBILE Mobile Computing and Communications Review*, 11(4):34 43, 2007.

[165] Milica Stojanovic and James Preisig. Underwater acoustic communication channels: Propagation models and statistical characterization. *IEEE communications magazine*, 47(1):84 89, 2009.

[166] Piergiorgio Svaizer, Marco Matassoni, and Maurizio Omologo. Acoustic source location in a three-dimensional space using crosspower spectrum phase. In *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 231 234. IEEE, 1997.

[167] T. Takasu and A. Yasuda. Development of the low-cost rtk-gps receiver with an open source program package rtklib. In *International Symposium on GPS/GNSS*, 2009.

[168] Tomoji Takasu. Rtklib ver. 2.4. 2 manual. *RTKLIB: An Open Source Program Package for GNSS Positioning*, pages 29 49, 2013.

[169] Robert E Tarjan and Mihalis Yannakakis. Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM Journal on computing*, 13(3):566 579, 1984.

[170] P. V. Teixeira, M. Kaess, F. S. Hover, and J. J. Leonard. Multibeam data processing for underwater mapping. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2018.

[171] I. Tena Ruiz, S. de Raucourt, Y. Petillot, and D. M. Lane. Concurrent mapping and localization using sidescan sonar. *Journal of Oceanic Engineering*, 29(2):442 456, April 2004.

[172] S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot. FastSLAM: An efficient solution to the simultaneous localization and mapping problem with unknown data association. *Journal of Machine Learning Research*, 4(3):380 407, 2004.

[173] Sebastian Thrun, Dieter Fox, Wolfram Burgard, et al. Monte carlo localization with mixture proposal distribution. In *AAAI/IAAI*, pages 859 865, 2000.

[174] H. A. Van Trees. *Optimum array processing: Part IV of detection, estimation and modulation theory*, volume 1. Wiley Online Library, 2002.

[175] George Turin. An introduction to matched filters. *IRE transactions on Information theory*, 6(3):311 329, 1960.

[176] A. Underwood and C. Murphy. Design of a micro-AUV for autonomy development and multi-vehicle systems. In *OCEANS 2017 - Aberdeen*, pages 1 6, June 2017.

[177] Geological Survey (US) and Robert F Breault. *Spatial distribution, temporal variability, and chemistry of the salt wedge in the lower Charles River, Massachusetts, June 1998 to July 1999.* Citeseer, 2000.

[178] Jerome Vaganay, Mike Elkins, Dave Esposito, Will O'Halloran, Franz Hover, and Mike Kokko. Ship hull inspection with the hauv: Us navy and nato demonstrations results. In *OCEANS 2006*, pages 1 6. IEEE, 2006.

[179] Jean-Marc Valin, François Michaud, and Jean Rouat. Robust localization and tracking of simultaneous moving sound sources using beamforming and particle filtering. *Robotics and Autonomous Systems*, 55(3):216 228, 2007.

[180] Barry D Van Veen and Kevin M Buckley. Beamforming: A versatile approach to spatial filtering. *IEEE assp magazine*, 5(2):4 24, 1988.

[181] Jaco Vermaak and Andrew Blake. Nonlinear filtering for speaker tracking in noisy and reverberant environments. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, volume 5, pages 3021 3024. IEEE, 2001.

[182] Keith Vickery. Acoustic positioning systems. a practical overview of current systems. In *Proceedings of the 1998 Workshop on Autonomous Underwater Vehicles (Cat. No. 98CH36290)*, pages 5 17. IEEE, 1998.

[183] R. Vidal, O. Shakernia, H. J. Kim, D. H. Shim, and S. Sastry. Probabilistic pursuit-evasion games: Theory, Implementation, and Experimental Evaluation. *Robotics and Automation, IEEE Transactions on*, 18(5):662 669, 2002.

[184] Emmanuel Vincent, Aghilas Sini, and François Charpillet. Audio source localization by optimal control of a mobile robot. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5630 5634. IEEE, 2015.

[185] Peter Wadhams, Jeremy P Wilkinson, and SD McPhail. A new view of the underside of arctic sea ice. *Geophysical Research Letters*, 33(4), 2006.

[186] Alastair J. Walker. An efficient method for generating discrete random variables with general distributions. *ACM Trans. Math. Softw.*, 3(3):253 256, September 1977.

[187] Jeffrey M. Walls and Ryan M. Eustice. Experimental comparison of synchronous-clock cooperative acoustic navigation algorithms. In *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, Kona, HI, USA, September 2011. Accepted, To Appear.

[188] Ding Wang, Pierre FJ Lermusiaux, Patrick J Haley, Donald Eickstedt, Wayne G Leslie, and Henrik Schmidt. Acoustically focused adaptive sampling and onboard routing for marine rapid environmental assessment. *Journal of Marine Systems*, 78:S393 S407, 2009.

[189] J. Wang, S. Bai, and B. Englot. Underwater localization and 3d mapping of submerged structures with a single-beam scanning sonar. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4898 4905, May 2017.

[190] Wei Wang and Christopher M. Clark. Modeling and simulation of the VideoRay Pro iii underwater vehicle. In *OCEANS 2006-Asia Pacific*, pages 1 7. IEEE, 2007.

[191] Darren B. Ward and Robert C. Williamson. Particle filter beamforming for acoustic source localization in a reverberant environment. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages II 1777. IEEE, 2002.

[192] Yair Weiss and William T. Freeman. On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs. *IEEE Transactions on Information Theory*, 47(2):736 744, 2001.

[193] L. L. Whitcomb, M. V. Jakuba, J. C. Kinsey, S. C. Martin, S. E. Webster, J. C. Howland, C. L. Taylor, D. Gomez-Ibanez, and D. R. Yoerger. Navigation and control of the nereus hybrid underwater vehicle for global ocean science to 10,903 m depth: Preliminary results. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 594 600, May 2010.

[194] Stefan Williams and Ian Mahon. Simultaneous localisation and mapping on the great barrier reef. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 2, pages 1771 1776. IEEE, 2004.

[195] Philip M. Woodward. *Probability and Information Theory, with Applications to Radar*, volume 3. Elsevier, 2014.

[196] Russell B Wynn, Veerle AI Huvenne, Timothy P Le Bas, Bramley J Murton, Douglas P Connelly, Brian J Bett, Henry A Ruhl, Kirsty J Morris, Jeffrey Peakall, Daniel R Parsons, et al. Autonomous underwater vehicles (auvs): Their past, present and future contributions to the advancement of marine geoscience. *Marine Geology*, 352:451 468, 2014.

[197] Kai Y., Jianping A., Xiangyuan B., and Gangcan S. Constrained total least-squares location algorithm using time-difference-of-arrival measurements. *Vehicular Technology, IEEE Transactions on*, 59(3):1558 1562, March 2010.

[198] Kegen Y. and I. Oppermann. Performance of UWB position estimation based on time-of-arrival measurements. In *International Workshop on Ultra Wideband Systems.*, pages 400 404, May 2004.

[199] Namik Kemal Yilmaz, Constantinos Evangelinos, Pierre FJ Lermusiaux, and Nicholas M Patrikalakis. Path planning of autonomous underwater vehicles for adaptive sampling using mixed integer linear programming. *IEEE Journal of Oceanic Engineering*, 33(4):522 537, 2008.

[200] Yanwu Zhang, Michael A Godin, James G Bellingham, and John P Ryan. Using an autonomous underwater vehicle to track a coastal upwelling front. *IEEE Journal of Oceanic Engineering*, 37(3):338 347, 2012.

[201] K. Zhou and S.I. Roumeliotis. Optimal motion strategies for range-only constrained multisensor target tracking. *Robotics, IEEE Transactions on*, 24(5):1168 1185, 2008.