Robust Online Adaptive Sensor-Driven Survey Planning for Single and Multiple Autonomous Underwater Vehicles

by

Liam Paull

Previous Degrees (BEng. Computer Engineering, McGill University, 2004)

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

In the Graduate Academic Unit of Electrical and Computer Engineering

Supervisor(s):	Howard Li, PhD Electrical and Computer Engineering
	Mae Seto, PhD Mechanical Engineering
	Liuchen Chang, PhD Electrical and Computer Engineering
Examining Board:	Kevin Englehart, PhD., Electrical and Computer Engineering,
	Philip Parker, PhD., Electrical and Computer Engineering,
	Zengtao Chen, PhD., Mechanical Engineering
External Examiner:	Christopher Williams, PhD.,
	Institute for Ocean Technology,
	National Research Council Canada

This dissertation is accepted by the

Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

October, 2013

Abstract

The objective of the thesis is to propose new survey planning methods for autonomous underwater vehicles (AUVs). Historical approaches to the problem usually involve preplanning paths before the start of the survey in a structured, either zig-zag or lawn mower, pattern. This has three notable shortcomings: 1) The AUV may not be able to follow the prescribed path exactly, for example if there are unexpected disturbances, 2) Object detection in imagery obtained from the survey is dependent on many factors, some of which may not be known *a priori*, resulting in overly conservative plans, and 3) In the absence of pre-placed beacons, once the vehicle submerges, its location estimate will drift, again resulting in deviation from the pre-specified path.

These issues are overcome here using new online and adaptive approaches to survey planning. First, an algorithm is developed that maintains a model of the area coverage over the workspace in real-time and plans paths within that workspace. This sensor-driven algorithm is also able to account for uncertainty in the factors that affect object detection while underway. Second, the algorithm is augmented to explicitly account for the uncertainty in the AUV location estimate. The link between area coverage and state estimation is made. It is motivated that for coverage, estimating the full trajectory (or smoothing) as opposed to just the present pose (or filtering) is beneficial. In addition, the planning strategies previously developed are augmented to operate within the new probabilistic framework.

The main benefit of the proposed approach is that it is robust to localization sensor noise and can guarantee coverage in a real field coverage sense not just a planning sense.

Finally, the algorithms are extended to coverage with multiple AUVs. If a team of AUVs possess precisely synchronized clocks, then when they communicate underwater they can also calculate their relative ranges from the time-of-flight of the acoustic signals. A cooperative trajectory estimation algorithm is developed that is particularly well-suited to operate within the challenging underwater communications channel. Data to be transmitted in each broadcast communication scales linearly with the number of vehicles in the AUV team and does not backlog in the case of communication failures, which are common underwater.

The algorithms are tested in simulation and in the field on the Iver2 AUV. These tests show the effectiveness of the proposed methods.

Acknowledgements

First and foremost I would like to express my sincerest gratitude to my primary supervisor Dr. Howard Li. Without his inspiration, guidance, attention to detail, and not least of all funding, this thesis would simply not have been possible. Dr. Mae Seto's contribution to my work has also been invaluable. From guiding my field work to providing feedback on manuscripts and even being kind enough to let me into her home. Her level of encouragement and support have been over and above the normal call of duty for a graduate supervisor. Additionally, Dr. Liuchen Chang played the central role in enticing me to come to UNB and pursue graduate studies, and has been a beacon of integrity and a source of wisdom throughout.

I would also like to thank the review committee for taking the time to review the thesis and provide insightful feedback.

The University of New Brunswick, and specifically the department of Electrical and Computer Engineering, has been a wonderful place to work. The smiling faces of Karen, Shelley, and Denise in the office, and the additional guidance from Drs. Julian Meng, Yevgen Biletskiy, Maryhelen Stevenson, Chris Diduch, and Kevin Englehart among others has been amazing.

The mine and harbour defense group at DRDC-A, specifically Vince, Warren, Owen, Shawn, and Jonathan and Ricky and Rich at the acoustic barge for all the help with the field trials.

I would like to acknowledge the National Science and Engineering Research Council (NSERC) who have funded a large portion of my graduate studies, as have Defense Research & Development Canada.

To all of my colleagues/friends in the COllaboration Based Robotics and Automation (COBRA) lab over the years: Carl, Amr, Tyler, Alex, Denise, Arnaldo, Mostafa, and Shang and Yao, it has been an absolute pleasure working with you. A special mention should go to Sajad Saeedi, who has been an absolute joy to spend the last four years with. I will be lucky if I ever find someone else as capable and enjoyable to work with in my professional career as Sajad.

Finally, my family. My parents Jacqui and Richard, who always challenged me to do my best. My sister, Georgina, and brother, Brendan, brother-in-law and great friend Travis, and in-laws Anne and Bryan. And, lastly but most certainly not leastly, my wife Kirsten who has supported me through the ups and the downs and all the associated challenges of being married to a graduate student.

Table of Contents

List of Figures

Notation Conventions

Notation	Description
X	All capital letter variables are random variables (RVs)
x	All lower case variables are values
X^i	Superscripts are used to index either robot index or cell index
X_t	Subscripts used to index time t
$X_{1:t}$	Shorthand notation for $X_1, X_2, ..., X_t$
$p(x)$	Shorthand for $p(X = x)$, which is the value of the density function of X at x , often written as $f_X(x)$.
$\boldsymbol{X}$	All bold font variables are arrays or matrices: $\boldsymbol{X} = [X^1, X^2, ..., X^N]$
$X^{[i]}$	The ith element in the array X
$X^{[i:j]}$	The block submatrix $[X^{[i]}, ..., X^{[j]}]$ (similar for 2D arrays: $X^{[i_1:j_1, i_2:j_2]}$
$\mathcal{X}$	Italics used to denote a set.
$\mathcal{X}[i]$	The ith member of the set $\mathcal{X}$

Table 1: Conventions and shorthand notations.

List of Acronyms

AUV	Autonomous underwater vehicle	CNA	Communications and navigation aid
MCM	Mine countermeasures	SIFT	Scale-invariant feature transform
GPS	Global positioning system	SURF	Speeded up robust features
GVD	Generalized Voronoi diagram	VAN	Visually augmented navigation
PRM	Probabilistic roadmap	FLS	Forward-looking sonar
TDMA	Time-division multiple access	CPP	Coverage path planning
LBL	Long baseline	SLAM	Simultaneous localization and mapping
USBL	Ultra-short baseline	RV	Random variable
TOF	Time of flight	UAV	Unmanned aerial vehicle
IMU	Inertial measurement unit	SAS	Synthetic aperture sonar
DVL	Doppler velocity log	SSS	Sidescan sonar sensor
INS	inertial navigation system	ESPRESSO	Extensible performance and evaluation suite for sonar
ADCP	Acoustic Doppler current profiler	EER	Expected entropy reduction
EKF	Extended Kalman filter	IG	Information gain
UKF	Unscented Kalman filter	BE	Branch entropy
PF	Particle filter	DAG	Directed acyclic graph
MSIS	Mechanically scanning imaging sonar	MOOS-IvP	Mission-oriented operative suite with interval programming
FH-FSK	Frequency shift keying with frequency hopping	CGM	Coverage grid map
PSK	Phase shift keying	BN	Bayes' network
MAP	Maximum *a posteriori*	MAC	Medium access control
ASC	Autonomous surface craft		

Table 2: Acronyms and abbreviations.

Chapter 1

Introduction

Autonomous underwater systems technology is lagging behind its ground and aerial robotics counterparts. Many tasks are more challenging underwater due to the harsh nature of the underwater environment. For example, some sensing modalities that are the cornerstones of terrestrial robotics systems, such as global positioning system (GPS) receivers and cameras, are rendered essentially useless underwater. Additionally, developing autonomous underwater systems tends to be expensive and time-consuming if operation of a research vessel is required. Nevertheless, these obstacles must be overcome as there are many applications where underwater technologies could aid particularly for tasks that are inherently very dangerous or costly for humans.

One particular such application is mine countermeasures (MCM). The U.S. Navy has referred to underwater mine removal as the most problematic mission facing unmanned undersea vehicles and the Navy at large [157]. MCM involves the gathering of sonar imagery, detection and classification of potential threats, followed by reacquisition and disposal by a remotely operated vehicle or human diver. The MCM task pipeline is shown in Fig. 1.1. This thesis focusses on the first two phases of the MCM problem - “survey mission planning” and “data collection survey”.

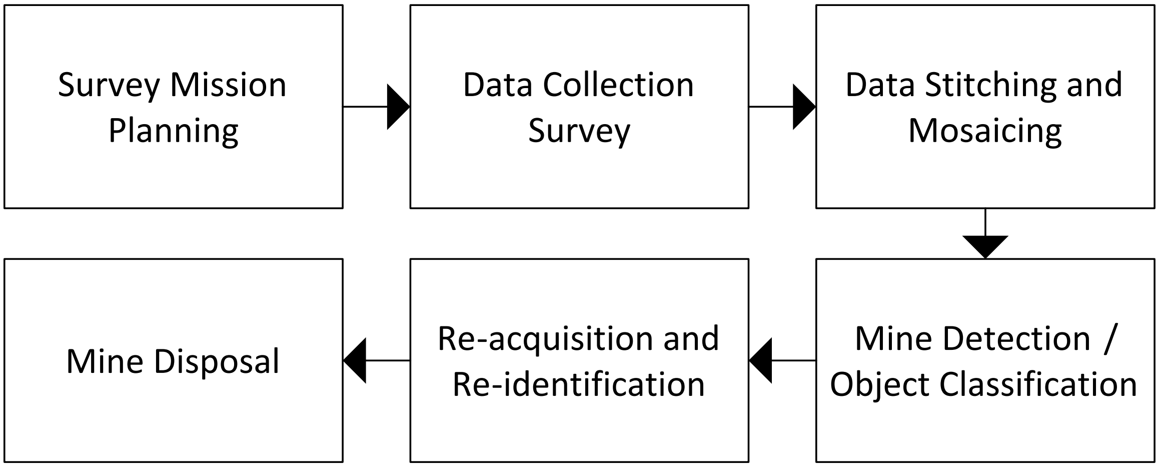

Survey Mission Planning
Data Collection Survey
Data Stitching and Mosaicing
Mine Detection / Object Classification
Re-acquisition and Re-identification
Mine Disposal

Figure 1.1: Autonomous underwater vehicle MCM task pipeline.

Using towed arrays for a variety of tasks such as mine hunting has been done for almost a hundred years [136]. In the 1970’s, engineers began experimenting with using autonomous underwater vehicles (AUVs) with on-board sonar sensors. This had significant advantages in terms of safety to operators and covertness. However, one challenge is that the AUV must navigate autonomously as opposed to a towed body whose position can be more accurately measured relative to the ship that is towing it. Additionally, the gathered sonar data from an AUV cannot be sent to the surface for analysis during the mission. Instead an entire mission is planned beforehand and then imagery is analyzed by an expert or automatic target recognition system to classify objects of interest after the completion of data collection. This puts greater importance on careful survey mission planning for the AUV.

The objective of a seabed coverage survey MCM mission is usually to clear a path or working area of mines before entering with a Navy vessel or some other high valued asset. Usually clearance or confidence is requested up to a specified threshold. If mine-like objects are identified during the survey, another task-specific AUV or a human diver will take a closer look and possibly dispose of the objects. It is critically important to localize the mine-like objects as accurately as possible on the first pass so that they can be easily and quickly reacquired.

This approach to MCM is only capable of detecting mines that are sitting on the seabed. Mines are detected from the gathered sonar imagery which is historically done with a template matching approach on the shadows generated by objects sitting proud of the seabed. As a result, MCM is treated as a 2D planning problem. The approach extends to 3D but detecting other types of mines, such as floating or tethered, would require some other mine sensing technology.

1.1 Problem Statement

Historically, AUV seabed surveys have been designed using primitive heuristic planning methods, such zig-zag or lawn mower (up and back) paths [164]. Such an approach has several limitations:

- Sonar performance is dependent on many factors, such as propagation conditions, seabed type and others, only some of which are known beforehand at the mission planning stage,
- Plans are not adaptive to the situation where the vehicle does not exactly follow the prescribed path either because it is infeasible or due to unexpected external disturbances,
- Plans do not account for the uncertainty in the platform location that is induced when the AUV traverses underwater without access to a globally bounded position estimate other than possibly using a cross-track error threshold to decide when to surface.

The current trend in robotics is towards multi-agent [140] and multi-robot [196] systems. Such systems come with significant potential benefits in terms of efficiency and robustness. These advantages can be exploited for AUV seabed surveying. However, with the benefits come the additional costs in terms of complexity of algorithms required to control and coordinate multiple vehicles that have limited on-board computation capabilities. Communicating among vehicles in water is limited due to lack of available bandwidth, slow throughput rate, and unreliability of the acoustic channel.

1.2 Thesis Overview and Significance

This thesis focuses on AUV survey planning that is: 1) adaptive in the sense that plans can be updated based on sensor data that is gathered during the mission and 2) accurate in the sense that the uncertainty of the vehicle is explicitly accounted for to produce a more descriptive representation of the coverage. In addition, the methods are extended to the multi-AUV case where the ability for vehicles to sense each other using acoustics is exploited for cooperation within the challenging underwater environment.

These proposed methods have significant advantages for AUV seabed surveying operations:

- The mission can be completed more efficiently.
- A more accurate representation of the risk associated with moving an asset through a waterway allows decision makers to be better informed. If the uncertainty of the AUV is not accounted for this can result in a an overestimation of the confidence that all threats are cleared, which can put people and assets in jeopardy.
- Localizing potential targets more accurately means that they can be re-acquired and disposed of more quickly. The disposal of underwater mines has been done by human divers and is inherently very dangerous.

1.3 Summary of Contributions

Detailed theoretical contributions will be listed in each individual chapter, but in overview the contributions of this thesis are:

1. An adaptive approach to the robotics area coverage problem that is particularly well-suited to the case of seabed surveying where gathered data is not available until survey completion. This problem is unique in that the coverage sensor (sidescan sonar) performance is variable and affected by factors such as seabed type or propagation characteristics that can be unknown beforehand.
2. An explicit link between area coverage and state estimation through the definition of a probability of coverage. The adaptive planning strategies are extended to operate within this probabilistic framework.
3. The development of a cooperative trajectory estimation algorithm within a least-squares optimization framework to extend the algorithms to multiple survey AUVs.
4. Field trials and simulations that demonstrate that the proposed methods are effective at achieving adaptive seabed coverage.

1.4 Outline

The rest of the thesis is organized as follows: Chapter 2 provides background and literature review. Specifically, sections are devoted to path planning (Section 2.1), information theory (Section 2.2), the sidescan sonar sensor (Section 2.3) and AUV localization (Section 2.4). Chapter 3 presents the algorithm for sensor-driven online coverage planning with field results. Chapter 4 extends the approach of Chapter 3 by explicitly accounting for the platform uncertainty in the coverage map and drawing the links to state estimation and planning. Chapter 5 extends Chapter 4 to the multi-AUV case where localization error is reduced (and consequently coverage efficiency is increased) by exploiting inter-vehicle range measurements. Finally future research directions are highlighted and conclusions are presented in Chapter 6.

Chapter 2

Background

This thesis is at the intersection of many different research domains. A rigorous treatment of all the related background and results in the literature is not possible. Nevertheless, this chapter presents some of the most important areas, mainly relating to:

- Robotics path planning, with a special focus on AUV path planning, coverage path planning, and path planning with robot uncertainty,
- Information theory, the flexible theoretical framework that is used to achieve various objectives throughout the thesis,
- Sidescan sonar sensor, the imaging sonar which is used for in this work for seabed surveying,
- AUV localization.

2.1 Robot Path Planning

In its most general form, robotics path planning consists of determining the sequence of robot configurations that will result in achieving the stated mission[1]. Often, particularly in AUV applications, paths are represented by a set of waypoints to be visited in succession. These waypoints are used as an outer loop reference that assumes that some inner loop controller will be able to stabilize the vehicle and track the reference. Given that this outer loop reference is independent of the platform control, very generic algorithms can be developed that are agnostic to the AUV system hardware.

Waypoints can be generated one at a time based on current up-to-date information, or can be fully pre-planned *a priori*. For example, traditional seabed surveys use a scripted series of waypoints that define the survey, usually a ladder or zig-zag path [164]. An alternative to waypoint-defined paths is a more reactive paradigm. For example, an AUV trajectory could be specified by maintaining altitude over the seafloor, avoiding obstacles, or following a structure such as a dam. In this style of approach, no model of the environment, or world, is required to aid in the decision making process so global goals are difficult to achieve.

The ability of an autonomous agent to plan a path depends on many factors, such as the level of information it has about the environment, the desired quality of the planned path, and the mission objectives. If detailed environmental information is available beforehand, then path generation can be slow without consequence. However, if the AUV is required to plan its mission *on-the-fly*, then an algorithm that can run in near real-time will be required.

2.1.1 Problem Formulation

In order to formally define the path planning problem, we require a few fundamental terms:
Configuration space (Q): The set of all configurations, q, that the robot can achieve. Q can be subdivided into free configuration space, Q_{free} and occupied configuration space, Q_{obs}. Any configuration in Q_{obs} will result in contact with an obstacle, and any configuration in Q_{free} will not.
Workspace ($\mathcal{W}$): The physical world in which the robot exists.
Path ($\tau(s)$): Curve through Q_{free} parameterized by s.
Degrees of Freedom: The minimum number of independent variables required to represent the robot configuration. To fully describe an AUV requires six degrees of freedom: $q = \{x, y, z, \phi, \theta, \psi\}$, where x, y, and z represent the position in 3D space, and ϕ, θ, and ψ are the Euler angles. Sometimes planning for an AUV is done in 2D by assuming that $\theta = \theta_{ref}$, $\phi = \phi_{ref}$ and $z = z_{ref}$, in which case there are three degrees of freedom: $q = \{x, y, \psi\}$.

We can then define the task of path planning as finding a curve in the free configuration space, Q_{free}, that connects a start location, $q = q_s \triangleq \{x_s, y_s, z_s, \phi_s, \theta_s, \psi_s\}$, to a goal location, $q = q_g \triangleq \{x_g, y_g, z_g, \phi_g, \theta_g, \psi_g\}$. More specifically, as described in [133], the path can be expressed as a curve $\tau : [0, 1] \to Q_{free}$ with $\tau(0) = q_s$ and $\tau(1) = q_g$.

The following terms define properties of any path planning algorithm, and can be used as a basis for comparison:
Optimality: An algorithm that optimizes (maximizes or minimizes) some objective.
Completeness: A plan is complete if it always finds a solution if one exists or determines that no solution exists in finite time. A path can also be considered **resolution complete**

[1] The bulk of the review presented here is published in our book chapter (Chapter 4 in “Marine Robot Autonomy” [173]).

if it is complete subject to discretization. Alternately, an algorithm is said to be **probabilistically complete** if it is guaranteed to converge towards completeness.
Offline planning: The plan is completed before execution begins.
Online planning: The plan is incrementally constructed during the mission.
Sensor-based planning: Sensor information is processed online and used for path planning.
Deliberative: Sense → Plan → Act cycle. The sensor data are usually used to construct a model of the world [132].
Reactive: Use sensory information to accomplish the mission without representation of the entire environment [132].

Path planning approaches come in many flavours. Categories that will be described include: heuristic, potential functions, roadmaps, and cell decomposition. In order to compare and contrast the various algorithms, the simple 2D environment shown in Fig. 2.1 will be used. Red randomly placed blocks represent obstacles and the start and goal are as indicated.

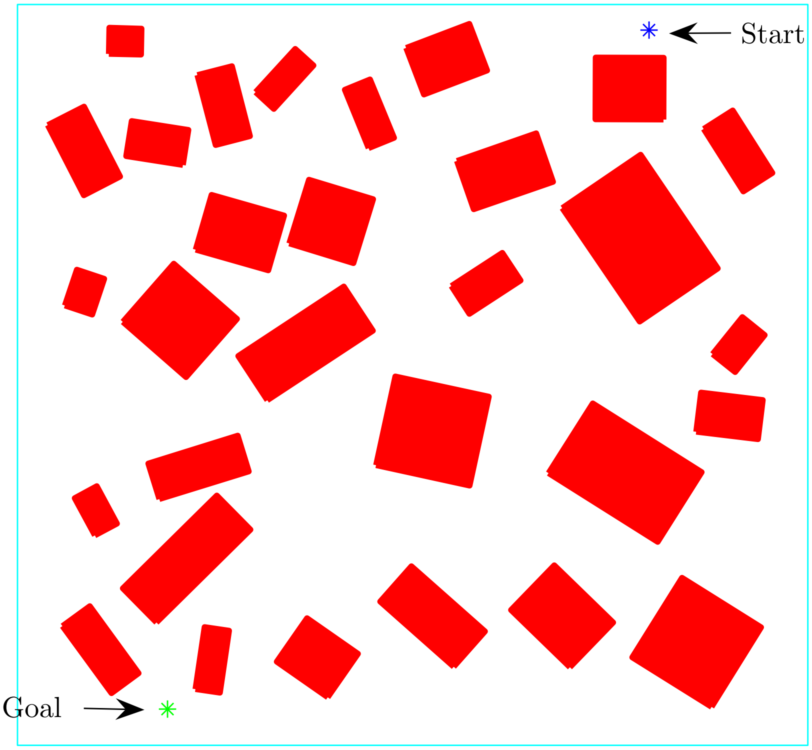


Figure 2.1: Simple environment with polygonal obstacles that will be used to compare path planning algorithms [173].

2.1.2 Heuristic-Based Algorithms

Heuristic-based methods use a set of rules in order to achieve an objective. An example are the simple “bug” algorithms [47].

2.1.2.1 Bug1

In Bug1 [144], it is assumed that the robot has only a contact sensor and knowledge of where the goal is. The robot begins by moving towards the goal as if the start and end points were joined by a straight line. If an obstacle is encountered, it is circumnavigated to find the point on the obstacle that is closest to the goal. The robot then travels to that point and continues until the goal is reached. If the heading line intersects with a previously visited obstacle, it is inferred that there is no path to the goal. It has been proven that this method is complete, meaning that the robot is guaranteed to find the goal or report that there is no possible path to the goal in finite time. The details of the approach are presented in Algorithm 1.

Algorithm 1 Bug1 algorithm

```
Input: Map
Output: Path
while Not arrived at the goal do
    Take a step in the direction of the goal
    if Obstacle encountered then
        if Obstacle is previously unvisited then
            Circumnavigate the obstacle until encountering the initial contact point
            Move to the point on the obstacle that is closest to the goal
            Mark obstacle visited
        else
            Path does not exist
        end if
    end if
end while
```

Algorithm 2 Bug2 algorithm

```
Input: Map
Output: Path
Generate line $l$ that intersects start and goal
while Not arrived at the goal do
    Take a step along $l$
    if Obstacle encountered then
        Move around the obstacle until hitting $l$
        if Robot revisits an already visited point then
            Search fails
        end if
    end if
end while
```

The sample path generated with the Bug1 algorithm is shown in Fig. 2.2.

2.1.2.2 Bug2

In Bug2 [144], an imaginary line is drawn between the start point and the goal point. The robot begins by following this line. If an obstacle is encountered, it begins to circumnavigate the obstacle until it intersects the line and then it continues. Bug2 is generally faster than Bug1, but has the disadvantage that it is not guaranteed to find the goal in finite time, as it can be trapped by certain oddly shaped obstacles. If the robot ever encounters an already visited point in the map, then the algorithm fails. The details of the approach are presented in Algorithm 2. The path generated is shown in Fig. 2.3.

There is also a variation of Bug2 called TangentBug. In TangentBug [103], it is assumed that the robot is equipped with a range sensor. The robot begins by traveling towards the target point. If an obstacle is detected by the finite range sensor then the robot alters its trajectory to the point on the range sensor that will avoid the obstacle and bring it towards the goal. The resulting path is tangential to the obstacles. This method vastly outperforms

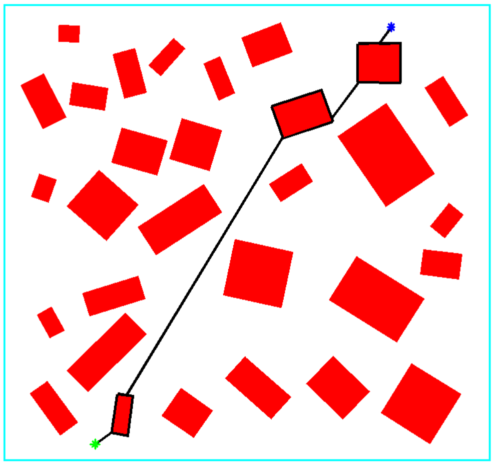


Figure 2.2: Path planned using Bug1 algorithm (appears in joint publication [173]). In Bug1, obstacles that are encountered are circumnavigated to find the closest point to the goal.

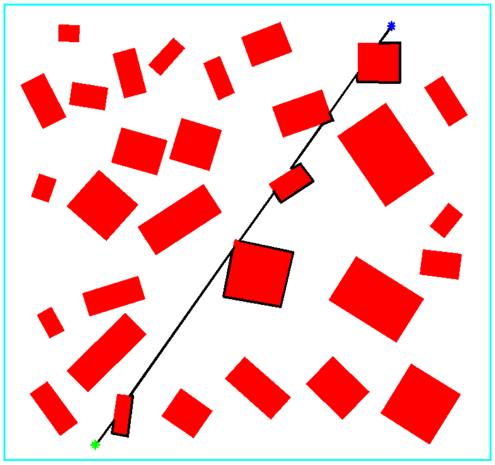


Figure 2.3: Path planned using Bug2 algorithm (appears in joint publication [173]). In Bug2, obstacles are avoided until the robot returns to the line that intersects the start and goal locations.

Bug1 and Bug2, but requires the use of an additional sensor. In all Bug algorithms, the direction to circumnavigate the obstacles must be chosen prior to the search. Usually this choice is made randomly, but can have a significant effect on the length of the resultant path.

2.1.3 Potential Functions

Available knowledge of the environment such as goal and obstacle locations can lead to the development of a potential function over the free configuration space. Paths are generated by following the negative gradient of the potential function. The approach tends to be computationally simple because no searching is required to determine the path, instead the direction of the negative gradient of the potential field is automatically chosen.

2.1.3.1 Potential Fields

In the potential fields method, the potential function is generated by considering the goal as an attractive force and the obstacles as repulsive forces. The potential function at any point in the space is the sum of the positive and negative force fields [47]:

$$\begin{aligned} U(q) &= U_{att}(q) + U_{rep}(q) \\ \nabla U(q) &= \nabla U_{att}(q) + \nabla U_{rep}(q) \end{aligned} \tag{2.1}$$

where the positive force could be defined as:

$$\begin{aligned} U_{att}(q) &= \frac{1}{2}\xi d^2(q, q_{goal}) \\ \nabla U_{att}(q) &= \xi(q - q_{goal}) \end{aligned} \tag{2.2}$$

where $d(q_i, q_j) = (q_i - q_j)$ is a function that defines some measure of distance between any two locations, q_i and q_j,80 in the configuration space, and ξ is a scalar multiplier. The negative force could be defined as:

$$U_{rep_i}(q) = \begin{cases} \frac{1}{2}\eta(\frac{1}{d_i(q)} - \frac{1}{Q_i^*})^2 & , d_i(q) \leq Q_i^* \\ 0 & , d_i(q) > Q_i^* \end{cases} \tag{2.3}$$

$$\nabla U_{rep_i} = \begin{cases} \eta(\frac{1}{Q_i^*} - \frac{1}{d_i(q)})\frac{q-c_i}{d_i^3(q)} & , d_i(q) \leq Q_i^* \\ 0 & , d_i(q) > Q_i^* \end{cases} \tag{2.4}$$

$$\nabla U_{rep} = \sum_{i=1}^{n} \nabla U_{rep_i} \tag{2.5}$$

where $d_i(q)$ is the minimum distance to obstacle i, and c_i is the closest point to q on obstacle i. Q_i^* is the domain of influence for obstacle i, and η is a scalar multiplier. The path is generated by following the direction of the negative gradient $(-\nabla(U(q))$ which corresponds to the direction that is progressing towards the goal the fastest.

In this form, complete knowledge of the location of obstacles in the environment is required assuming Q_i is large enough. However, online versions of the potential fields method have been suggested [114]. For example, if the robot is equipped with a range sensor, then the

Q_i values could be adjusted to the range of the sensor. The main drawback of this method is that it is possible for the robot to enter locations that are not the goal where the sum of the vectors is zero. These are referred to as local minima, for an example refer to Fig. 2.4.

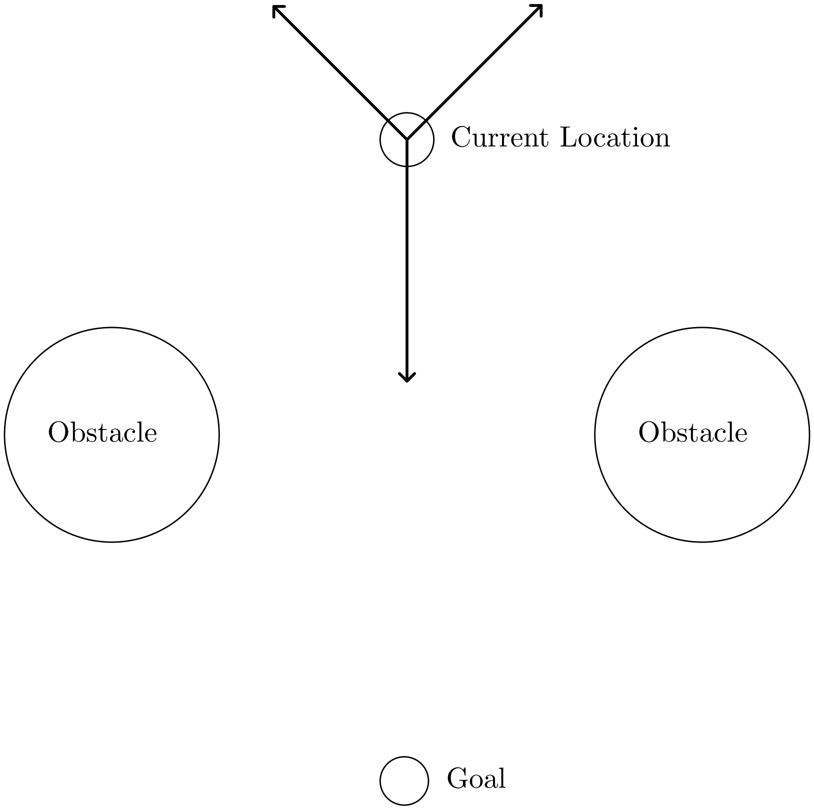


Figure 2.4: A local minimum. The repulsive forces from the obstacles and the attractive force of the goal all sum to zero so the robot will not move.

The path generated by the potential fields algorithm is shown in Fig. 2.5.

2.1.3.2 Wavefront Planner

The wavefront planner [22] is similar to the potential fields method, but differs in several important ways. The potentials at each point in the free space are generated by simulating a wave propagating from the goal location to the start. The details are given in Algorithm 3. The free space is represented by a grid, all are initially marked as unvisited. To initialize, the goal cell is marked as visited and given a value of 2. The algorithm then iteratively finds unvisited neighbors of visited cells and assign them values of one greater than the visited cell. The result is a ‘wavefront’ radiating outwards from the goal cell as new cells are assigned increasing values.

Once the wavefront reaches the start cell, the potential function is complete and a path can be derived based on gradient descent. This is accomplished by beginning at the start location and iteratively choosing the neighbor with the minimum value.

The wavefront planner has the advantage over the potential fields method in that no local minima are generated. However, it is computationally more expensive and requires prior knowledge of the locations of all of the obstacles (offline algorithm).

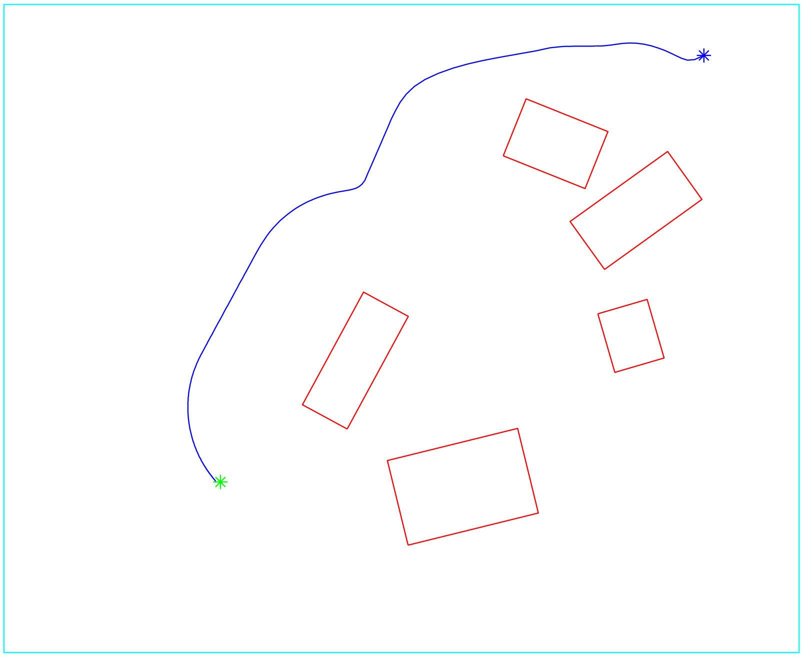


Figure 2.5: Path generated from the potential field algorithm (appears in joint publication [173]).

Fig. 2.6 shows an example implementation of the algorithm. In the figure, the lighter areas of free space are at a higher potential. The obstacles shown in black are considered infinite potential. In this case, the free space is discretized into a grid and only four neighbors are considered in the gradient descent step.

2.1.4 Roadmaps

A roadmap [132] is generated as a subset of the configuration space that satisfies three main rules:

1. It is possible to get from any start point in the configuration space to the roadmap.
2. It is possible to get from any point on the roadmap to any other point on the roadmap.
3. It is possible to get from the roadmap to any goal point in the configuration space.

Different spaces that have been proven to be roadmaps include the visibility graph [143], which is generated by connecting every vertex of obstacles that can be connected with a straight line, and the Voronoi diagram [11], which is the set of points that are equidistant from at least two obstacles.

Once a roadmap is generated, it can be represented as a graph of edges and nodes, which can be searched with classical search methods, such as A*, D*, breadth-first search and many others. Generating the roadmap requires prior knowledge of the positions of all of the obstacles. However, online methods where the roadmap is built as the robot traverses the environment do exist and are referred to as 'sampling-based methods' [47].

Algorithm 3 Wavefront algorithm

```
Input: Map $Q$, start cell $q_{start}$, goal cell $q_{goal}$
Output: Path $P$
mark all free space as unvisited
$index \leftarrow 2$
$q_{goal} \leftarrow index$
mark $q_{goal}$ visited
while $q_{start}$ not visited do
    set all unvisited cells neighboring a cell with value $index$ to $index + 1$
    mark all cells with value $index + 1$ as visited
    $index \leftarrow index + 1$
end while
$q \leftarrow q_{start}$
add $q$ to $P$
while $q_{goal}$ not in $P$ do
    $q \leftarrow$ neighbor of $q$ with minimum value
    add $q$ to $P$
end while
```

2.1.4.1 Generalized Voronoi Diagram

The generalized Voronoi diagram (GVD) is the set of all points in the free space of a map that are equidistant from at least the two closest obstacles. The GVD can be interpreted as a topological representation of the map structure that contains the key information intrinsic to the map but in a much more compact, one-dimensional form, and is a roadmap [47].

An easy way to generate the GVD is to assign a circle with a small radius to each grid of the map, q_i, then increase the radius of the circle until it contacts at least one obstacle. If there is only one contact point, then q_i does not belong to the GVD otherwise it does. Algorithm 4 gives the detail of the approach.

An example of a GVD is shown in Fig. 2.7.

2.1.4.2 Visibility Graph

The visibility graph is an alternate roadmap representation of the free space.

In a visibility map, all obstacle vertices are nodes in the graph. Edges connect nodes that are in the line of sight of others in the free space. The full algorithm for generating the visibility graph is given in Algorithm 5. This algorithm requires that obstacles be polygonal. The set $\{v_i\}_{i=1}^{V}$ represents all obstacle vertices where V is the total number of vertices in all obstacles. Each possible combination of vertices is connected with and a straight line, denoted by Q_{line}. If all configurations on this line do not intersect with any obstacles then an edge between these two nodes is added to the graph.

This algorithm produces a graph with a high level of connectivity. The graph can be pruned by considering the following two definitions:

- *Supporting lines* are tangent to two obstacles with both of them on the same side of the line.
- *Separating lines* are tangent to two obstacles with each on opposite sides of the line.

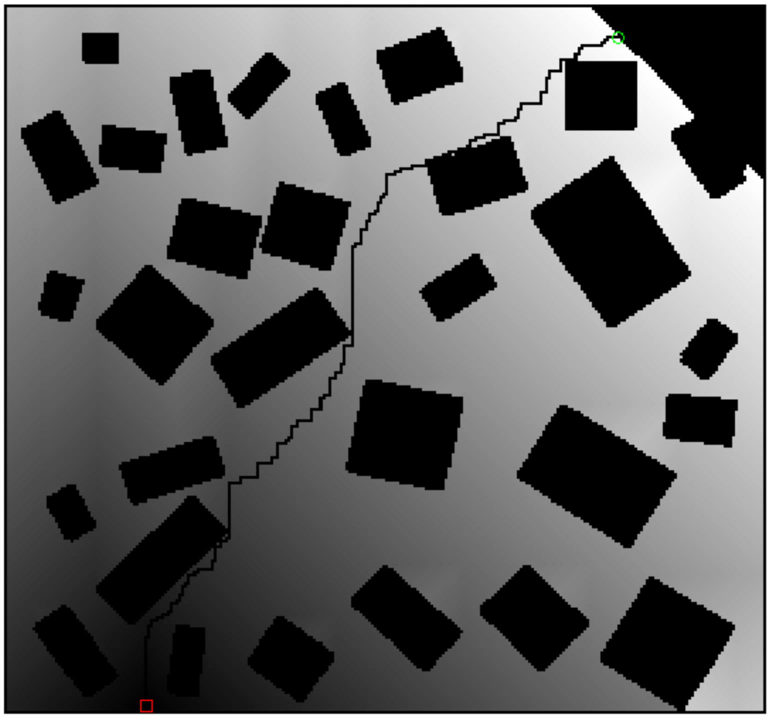


Figure 2.6: Path planned with wavefront algorithm.(This figure appears in joint publication [173])

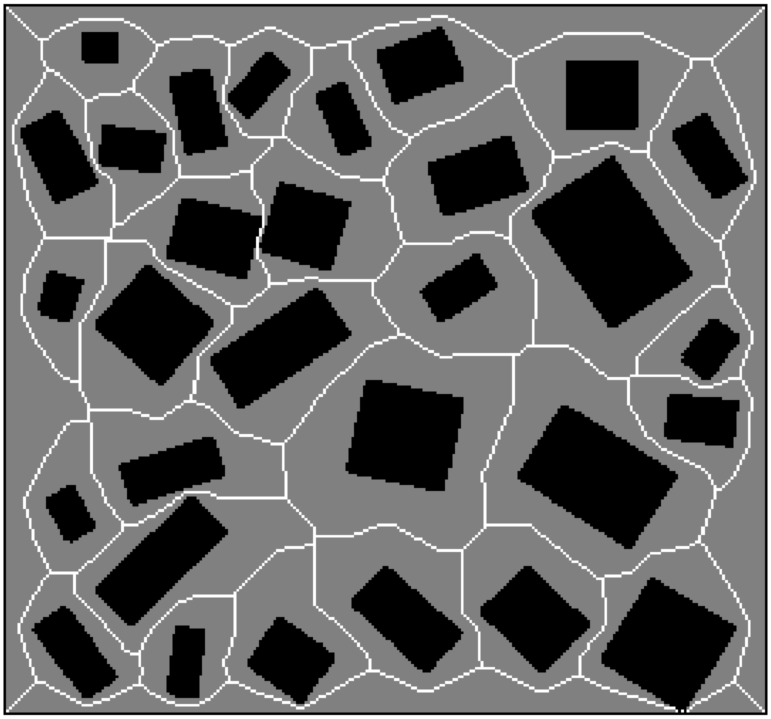


Figure 2.7: A generalized Voronoi diagram (GVD) (This figure appears in joint publication [173]). Each point in the GVD is equidistant from at least the two nearest obstacles.

Algorithm 4 Generate GVD

```
Input: Map $Q$
Output: GVD
GVD $\leftarrow \emptyset$
for all $q_i \in Q_{free}$ do
    $r \leftarrow 0$
    $C \leftarrow$ circle with center $q_i$ and radius $r$
    $cp \leftarrow \emptyset$
    while $cp = \emptyset$ do
        $r \leftarrow r + \Delta r$
        $cp \leftarrow C \bigcap Q_{obs}$
    end while
    if number of elements in $cp$ is greater than 1 then
        Add $q_i$ to GVD
    end if
end for
```

Algorithm 5 Visibility Graph

```
Input: Map, $Q$
Output: Visibility graph $VG$
Add each obstacle vertex $v_i$ to $VG$ as node $n_i$
for $i = 1 : V$ do
    for $j = 1 : V$ and $j \neq i$ do
        $q_i \leftarrow$ closest valid configuration to vertex $v_i$
        $q_j \leftarrow$ closest valid configuration to vertex $v_j$
        $Q_{line} \leftarrow$ set of configurations connecting $q_i$ to $q_j$ by a straight line in configuration space
        if $Q_{line} \bigcap Q_{obs} = \emptyset$ then
            Add edge between $n_i$ and $n_j$ in $VG$
        end if
    end for
end for
```

It can be proven [47] that the best solution will be made of supporting and separating lines. Therefore all other edges can be removed.

A visibility graph is shown in Fig. 2.8. In reality it can be useful to implement a safety margin (buffer) around all obstacles so that the paths do not closely approach obstacles.

The visibility graph has the advantage over the GVD roadmap in that it is very simple to generate. However, the resulting graph in general has more edges and nodes and therefore will be harder to search. In addition, the path generated from following the roadmap is guaranteed to come very close to obstacles, and the algorithm requires that obstacles be represented as polygons.

2.1.4.3 Probability Roadmaps

In some cases, generating the roadmap using either the GVD or the visibility graph is either too difficult, or results in a graph with so many nodes and edges that it cannot be

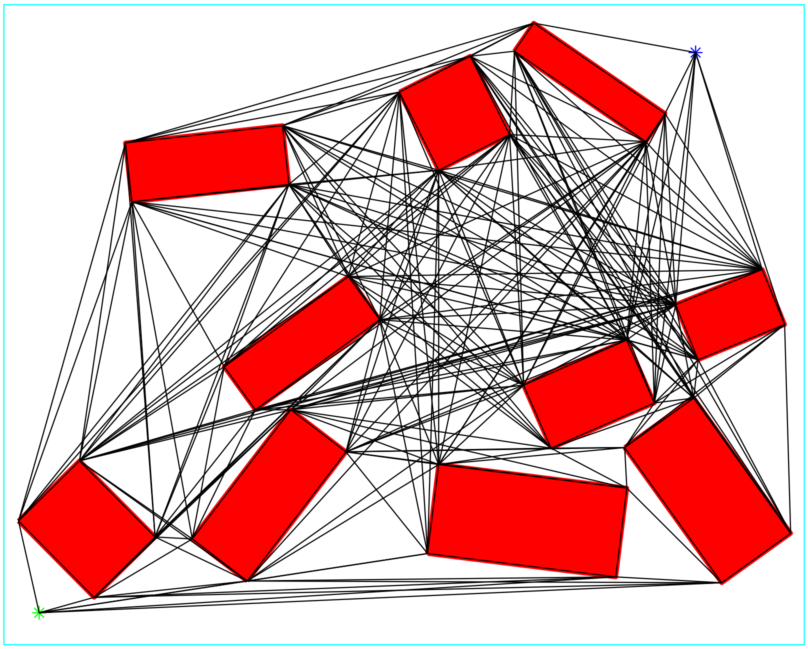


Figure 2.8: A visibility graph (This figure appears in joint publication [173]). The roadmap is built by connecting all vertices of all obstacles together.

searched efficiently. This can be the case if there are many obstacles, or particularly if the dimensionality of the configuration space is high. For example, an underwater robotic arm with multiple links that is used for docking or precise tasks will have a configuration space dimensionality equal to the number of links in the manipulator. To account for these problems, sampling-based methods have been proposed. While these algorithms do not possess desirable qualities such as completeness or optimality, they are effective at solving real life problems efficiently [129].

There are different types of sampling-based algorithms, but essentially they all aim to build a roadmap by randomly generating sample locations in the free configuration space. The first sampling-based algorithm was the probabilistic roadmap (PRM) planner [129]. According to this planner, the entire free configuration space is covered with random samples which are then connected at a higher level to generate a roadmap. The PRM planner is designed such that the roadmap building and the query phases of operation are separate; the entire roadmap is built before a path is found from start to goal. The challenging aspects of this type of algorithm are: 1) determining whether a randomly generated sample lies in the free configuration space at all, and 2) determining whether the edge between two nodes remains in the free configuration space. An implementation of the PRM planner is given in Algorithm 6.

A path planned with the PRM algorithm is shown in Fig. 2.9. Note that this algorithm is probabilistically complete, meaning that it is guaranteed to converge towards an optimal solution if enough nodes are generated.

Other sampling-based methods find a path to the goal as samples are generated. One popular algorithm is the Rapidly-Exploring Random Trees planner [125].

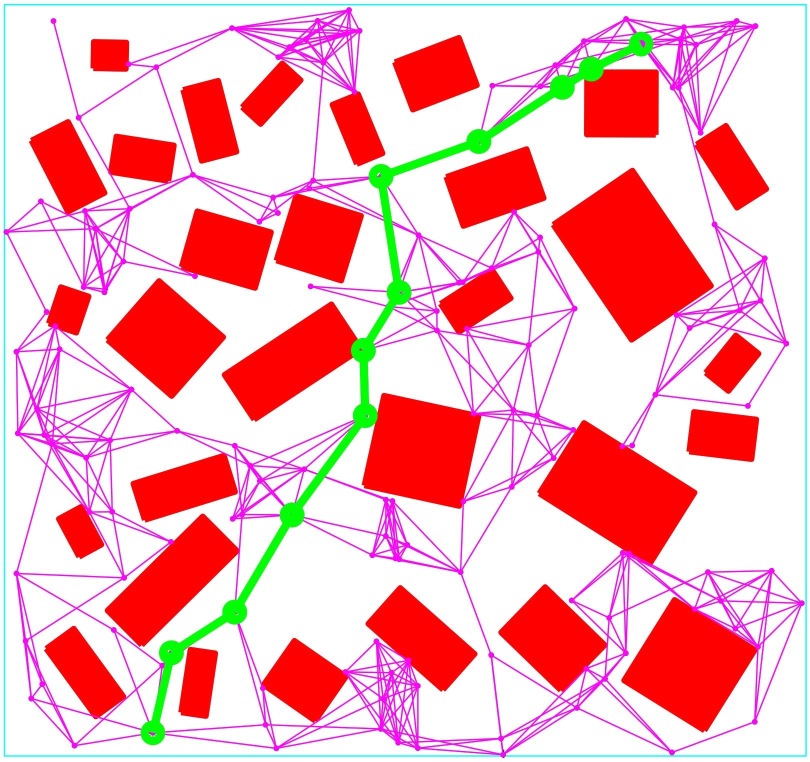


Figure 2.9: Path planned using the probability roadmap algorithm (figure appears in joint publication [173]). The free space is randomly sampled and then connected. This roadmap is used to find a path from the start to the goal.

Algorithm 6 Probabilistic Roadmap

```
Input: Map, $Q$
Output: Probabilistic Roadmap $PRM$
Add $q_{start}$ and $q_{goal}$ to $PRM$
while No path exists from $q_{start}$ to $q_{goal}$ in $PRM$ do
  Randomly generate $q$ from $Q$
  if $q \in Q_{free}$ then
    Add $q$ to $PRM$
    for all node $n \in PRM$ with $n \neq q$ do
      if Good path exists from $n$ to $q$ then
        Add edge in $PRM$ between $n$ and $q$
      end if
    end for
  end if
end while
```

2.1.5 Cell Decomposition

Cell decomposition refers to any method that partitions the free configuration space into a set of smaller cells [49]. Once the decompostion has been done, a connectivity graph can be built, where the cells are the nodes, and cells that share a common edge are connected in the graph.

Cell decompositions are classified as either exact, semi-approximate, or approximate [49]. Within the cell decomposition methods, the major differences are generally based on how to generate the cells and what shape they should be. Pioneering work in this field was done by Acar and Choset. The Boustophedeon cell decomposition is first introduced in [48] and is based on previous work by Canny and Lin [43]. The cell decomposition is generated by determining critical points in the workspace where the connectivity of a slice, or 1-D line, of the free space changes as it is swept across the workspace. This is expanded in [1, 2] to include a method in which critical points can be detected online and the decomposition is referred to as a Morse decomposition. The connectivity of the cells is represented by the Reeb graph. In [3], the approach is applied to a terrestrial demining task where the robot is able to detect and exploit the pattern of the mines to search more efficiently. In [4], the algorithm is extended to apply to a robot whose sensor swath is larger than the platform footprint. In this approach, two heuristics are combined: one for coverage of wide open spaces based on Boustrophedon search, and one for constricted areas which is based on the GVD.

Once the cell decomposition is made, a graph is generated where each cell represents a node in the tree and adjacent cells are connected with an edge. As mentioned, this type of decomposition is most useful for achieving coverage, but start-to-goal planning can be done by reducing each cell to one or a few points. In Fig. 2.10 each cell has been reduced to its centroid. The centroids of cells in the figure are denote by black dots. The blue lines represent the critical slices. Note that every critical slice is tangent to an obstacle but cannot penetrate an obstacle. As per the earlier figures, the obstacles are shown in red. The path found by connecting centroids is shown as the yellow line. Note that in this simplistic case, a suboptimal path is generated because of the reduction of the cells to only their centroids.

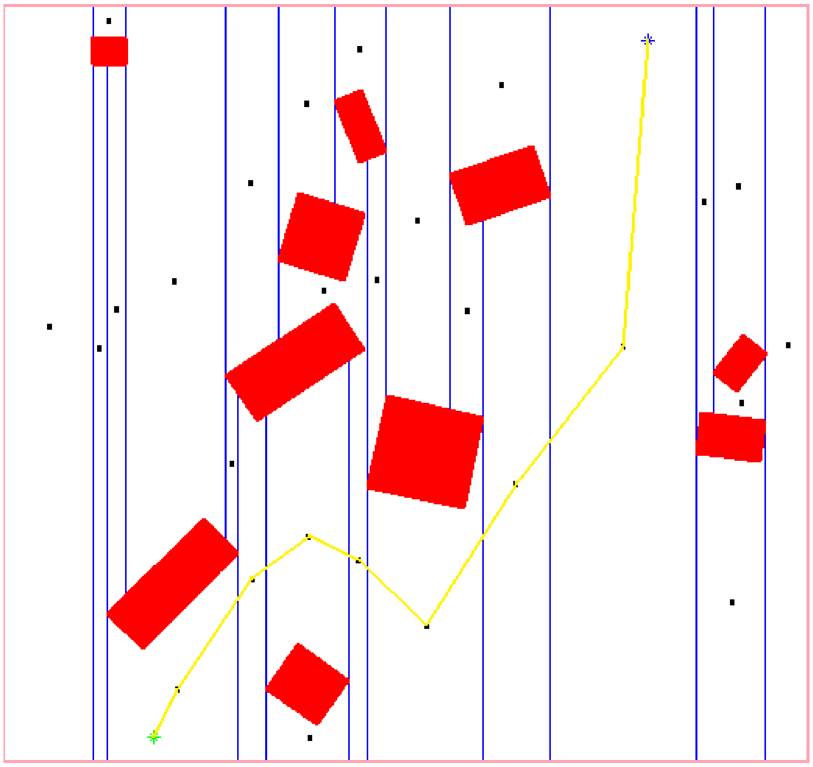


Figure 2.10: A path planned using the Morse cell decomposition (figure appears in joint publication [173]). Once cells are generated, a path is found through the cells that connects the start and the goal.

2.1.6 Coverage Path Planning

Coverage path planning (CPP) is a sub-problem of the more general path planning problem. Instead of generating a path that efficiently navigates from a start point to a goal point, the objective is now to generate a path where every point in the workspace is covered with a sensor. The problem can be formulated as follows: consider a mobile robot with an onboard sensor with some 2D swath $\mathcal{S}$. A trajectory through the workspace $\mathcal{W}$ will result in a set of N swath readings: $\{\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_N\}$. The goal of CPP is generally to generate a path through the workspace such that $\bigcup_{i=1}^{N} \mathcal{S}_i \supseteq \mathcal{W}$, meaning that at some point the sensor swath has passed over every point in the workspace[2].

In the context of CPP, some approaches can be considered as deterministic, which do not consider uncertainty and can provide completeness under good conditions [85]. Alternately, non-deterministic or probabilistic approaches are more robust to sensor and actuator noise, but cannot guarantee completeness. Simple CPP algorithms define structured paths that follow a predetermined geometry like a lawn-mower or a zig-zag to ensure coverage. These approaches tend to work well when the structure of the workspace to be covered is completely known *a priori*. When the area to be covered is unknown, this task is sometimes referred to as 'exploration' and requires more adaptive algorithms [86].

As described in Choset's survey of complete coverage methods, there are heuristic [49], random, and cell decomposition techniques. A heuristic method defines a set of rules to follow that will result in the entire environment being covered. For example, Acar and Choset's complete coverage algorithm based on sensing critical points [1], and Wein's [226] method of building corridors based on maximizing a quality function. A key facet of these approaches is having obstacles to generate the rules. Another recently published method based on a heuristical approach is [135]. Three independent behaviours are defined: spiral path tracking, wall following and virtual wall following resulting in paths that are more energy and time efficient.

Cell decomposition is used to divide up the environment into a manageable number of cells or areas that can be searched like a graph or tree. Once all cells have been covered, then the entire workspace has been covered. Commonly used methods of decomposing the workspace include trapezoidal, or Morse [1] decompositions. Another popular method for solving the CPP problem is the spanning tree coverage algorithm [82]. The workspace is decomposed into cells equal in size to the sensor swath and then a Hamiltonian cycle is determined which visits each cell exactly once. The main drawback of this method is that the path generated contains many sharp turns which violate the dynamic constraints of any non-holonomic platform. Other recently proposed cell decomposition approaches to CPP include: exact cell decomposition for vacuum cleaning [13] and unmanned aerial vehicles [141], 3D terrain coverage for farming [106], and a trapezoidal decomposition for agricultural applications [165].

An important aspect of any coverage planner is whether paths are planned *online* or *offline*. An offline planner assumes that perfect prior knowledge of the workspace is available. Online planners are required if it is desired for the agent to adapt to the dynamic environment. If the planner uses real-time sensor data then the approach is said to be sensor driven [165].

[2] This coverage objective will be explored more fully and expanded upon in Chapter 3 and Chapter 4

2.1.7 Autonomous Underwater Vehicle Path Planning

Much less research has been done on AUV path planning compared to autonomous ground or air vehicle planning. This section will summarize some of the more important and recent results in the field.

One of the first known papers to discuss path planning of AUVs was published by Warren in 1990 [223]. In this approach, potential fields are used to avoid obstacles and local minima are avoided by considering the global path. More recently, an optimal kinematic control scheme is proposed where the cost function to be minimized is the integral of a quadratic function of the velocity components [29]. In [231], a mixed integer linear programming method is used to find a path for adaptive sampling of the ocean using AUVs. This type of algorithm is used as an alternative to static buoys for collecting oceanic data such as temperature, salinity, etc. The metric for benefit in the objective function is a maximum sum of probabilities and paths planned are greedy. In [46], genetic algorithms and dynamic programming are used to find paths for multiple AUVs.

In some cases, such as highly cluttered harbours, obstacle avoidance is an important consideration. In [179], a path is planned using a fast-marching based method. In [178], a sequential-quadratic programming approach is used to avoid obstacles detected with a multi-beam sonar.

A significant consideration in start-to-goal path planning in an underwater environment is accounting for ocean or estuarine currents, for examples refer to [35], [123], [207], [208], [5], [214].

In [209], a coverage algorithm for MCM with a side-looking sonar is proposed that uses cell decomposition and exploits the supposed fact that mines are sometimes laid in lines. In [77], a Boustrophedon decomposition [48] is combined with the GVD to derive paths for coverage of a highly unstructured or non-convex environment. This algorithm presumes that absolute knowledge of the environment is known *a priori* and all planning is done offline.

Algorithms have been presented that optimize the design of structured paths for CPP. For example, in [117], the Cramer-Rao lower bound is used to optimize the track spacing for a ship hull inspection mission. In [228], a coverage algorithm for MCM with a side-looking sonar is proposed that optimizes the spacing between parallel tracks. The metric for optimality is maximizing the mean probability of detection for a given workspace. The dependence of probability of detection on seabed type and range is described.

AUV coverage algorithms for inspection and reconstruction of underwater objects such as ship hulls have also been presented [63, 117, 93, 95].

2.1.8 Path Planning with Uncertainty

Chapter 4 extends coverage planning to explicitly consider the state uncertainty of the robotic platform in a rigorous way. When an AUV submerges to perform a survey, it loses access to a global position reference. As a result, an uncertainty in the location of the platform is induced that grows as the AUV remains submerged, especially in the presence of ocean currents. Without carefully considering this position uncertainty, it is possible to become overly confident that a certain area of seabed has been covered. In a MCM application, this can result in mines being missed. There exist very few references in the literature that consider platform uncertainty in the general path planning problem and almost none that consider it in a coverage framework.

A distinction must be made between path planning algorithms that randomly sample Q_{free} and those that explicitly consider robotic platform uncertainty. These sampling-based methods are considered probabilistic because the locations of the nodes in a graph or roadmap are randomly generated, but it should be emphasized that the uncertainty of the pose of the robotic platform itself is usually not considered. In this case, the path planning problem is often solved using a PRM. The uncertainty in the robot pose can be incorporated into the calculation of the the heuristic function to determine a probability that a path generated will be suboptimal [156]. In this approach, the workspace is discretized into a cost map, where each value in the costmap is no a longer deterministic scalar, but a RV.

Some start-to-goal path planning methods have considered the platform location uncertainty in the context of designing paths that remain a 'safe' distance from obstacles. In [130] this is achieved through a set-bounding approach where the platform's 3σ ellipse should never come in contact with an obstacle. As noted, in the context of simultaneous localization and mapping (SLAM), there are uncertainties associated with the platform pose and with the obstacles' locations that should both be accounted for. An alternative approach to defining the safe distance based on pose uncertainty that is more closely related to the approach taken here is to define a probability of collision with an obstacle. Paths can be chosen that have the right combination of optimality and risk [100]. In [34], a user can specify a maximum allowable probability of collision and then an optimal path is found subject to this so-called 'chance constraint.' In [219], a similar approach is taken that bridges the gap between planning and control. A linear-quadratic Gaussian motion planner is combined with the rapidly exploring random trees planner to determine a path that minimizes the probability of collision with obstacles. The path is planned in advance assuming accurate stochastic models for motion dynamics and sensor measurements, as well as obstacle locations. One approach to exploration in the presence of uncertainty is the network of paths concept presented in [210]. In this approach, localization error at the goal is minimized by reversing along the known path whenever a dead-end, or untraversable path, is reached.

The literature is extremely scarce with respect to coverage algorithms that even acknowledge the fact that state estimation is probabilistic (examples of publications where vehicle uncertainty is mentioned in passing in the CPP context include [36], [135]). Recently, a 'probably approximately correct' measure of performance for stochastic coverage was defined in [53]. This measure defines the probability of coverage of a given fraction of the workspace based on the platform pose uncertainty. However, this approach uses the assumption that platform localization error is constant.

2.2 Information Theory

The origins of information theory date back to 1948 and the seminal paper "A Mathematical Theory of Communication" [200]. Originally used to study the fundamental limits of data compression, communication and storage, the ideas have now been applied to such diverse fields as statistical inference, language processing, cryptography, network science and data fusion [101].

2.2.1 Shannon Information

The Shannon entropy of a continuous random variable (RV) X, with probability density $p(x)$, is defined as:

$$\begin{aligned} H(X) &= -E\{\log p(x)\} \\ &= -\int p(x) \log p(x) dx, \end{aligned} \tag{2.6}$$

or similarly for a discrete RV:

$$H(X) = -\sum_x p(x) \log p(x) \tag{2.7}$$

and represents a measure of the compactness of a distribution [88].

As an example consider a binary RV $X \in \{0, 1\}$. The Shannon entropy can be evaluated as:

$$\begin{aligned} H(X) &= -\, p(X=1) \log p(X=1) - p(X=0) \log p(X=0) \\ &= -\, p(X=1) \log p(X=1) - (1 - p(X=1)) \log (1 - p(X=1)) \end{aligned} \tag{2.8}$$

which demonstrates that $H(X)$ can be represented as a function of $p(X = 1)$ in this case. If we graph the relation we get Fig. 2.11. Important to note is that the entropy has its maximum when the $p(X = 1) = p(X = 0) = 0.5$. In this case the value of X is at its most uncertain since it has an equal chance of being a 1 or a 0. If we subsequently were given some further information about X that allows us to guess one outcome as being more likely than the other, then our uncertainty, or entropy, about X would be reduced. Also of note in Fig. 2.11 is that the curve is symmetric about the $p(X = 1) = 0.5$ axis. The entropy values are in no way related to the values in the distribution (there is no x anywhere in (2.6), only $p(x)$). If we are given information that increases the chance of X being 0 or being 1 it is equivalent from an entropy standpoint.

Now consider two RVs X and Z. The *conditional entropy* of X given Z is defined as:

$$H(X|Z) = -E\{\log p(x|z)\} = -\int p(x|z) \log p(x|z) dx. \tag{2.9}$$

Now we can define the *mutual information*, or *entropy reduction* as :

$$\Delta H(X|Z) = H(X) - H(X|Z). \tag{2.10}$$

The mutual information defines a scalar quantity that represents the amount of information about X that is contained in Z, or how much the entropy of X will be reduced by knowledge of Z.

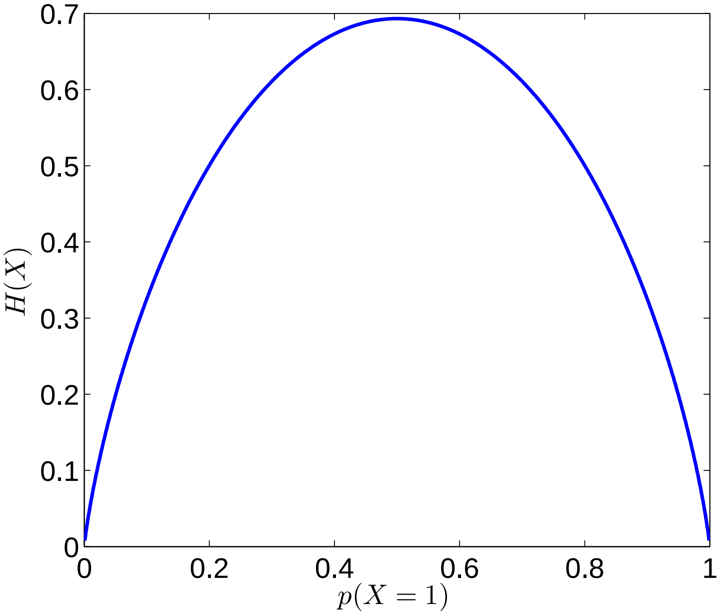


Figure 2.11: The Shannon entropy of a binary RV X plotted as a function of $p(X=1)$

For example, if X is the location of a target and Z is a RV that represents some uncertain sensor measurement, then $\Delta H(X|Z)$ represents the amount of information that we are getting about the location of that target by performing the measurement. From a control perspective, this becomes a useful way of evaluating the utility of actions. However, we usually don't know what the measurement will be before we make it, so we cannot evaluate the mutual information directly since we can't evaluate $p(x|z)$ directly until the measurement is actually made. To compensate, if we have a good idea of how our sensor works, then we can perform an average over a distribution of measurements that could be made. As a result, we can evaluate the expected entropy reduction, or information gain, which represents the expected reduction in entropy of X that will come about by making measurement Z:

$$E_Z[\Delta H(X|Z)] = H(X) - E_Z[H(X|Z)], \tag{2.11}$$

where E_Z denotes expectation over Z. One extremely useful aspect of the expected entropy reduction formulation is that information can be combined *additively*.

Although the concept of Shannon entropy was originally developed to represent communication over a noisy channel, it readily extends to exploration, which is closely related to coverage. From [121]: "Exploration can be viewed as a communication process where the object of interest is the transmitter, the sensor is the receiver, the sensing process is the noisy communication channel, and the sensed signal carries information about the object of interest."

2.2.2 Fisher Information

The Fisher information is an alternate representation of information that is only defined for continuous RVs. It also differs from Shannon information in that for a joint distribution $p(\mathbf{x})$ over a array of RVs $\mathbf{X}$ it produces a matrix rather than a scalar value. The Fisher information is defined by:

$$\boldsymbol{\Omega}(\mathbf{X}) = \frac{d^2}{d\mathbf{x}^2} \log p(\mathbf{x}) \tag{2.12}$$

where $\boldsymbol{\Omega}(\mathbf{X})$ is referred to as the *information matrix* and is a measure of the surface of a bounding region containing the distribution $p(\mathbf{x})$ [88].

An interesting case is when $\mathbf{X}$ is a multivariate n-dimensional Gaussian or normal distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$:

$$p(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma}|}} \exp\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\}. \tag{2.13}$$

If we apply (2.12) to (2.13) then we arrive at the result that:

$$\boldsymbol{\Omega}(\mathbf{X}) = \boldsymbol{\Sigma}^{-1}. \tag{2.14}$$

We can also define an *information vector*, $\boldsymbol{\xi}$:

$$\boldsymbol{\xi} = \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu} \tag{2.15}$$

and the information vector and information matrix can be used as an alternative parameterization of the Gaussian distribution (often referred to as "canonical representation"):

$$p(\mathbf{x}) = \mathcal{N}^{-1}(\boldsymbol{\xi}, \boldsymbol{\Omega}) = -\frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Omega}|}} \exp\{\frac{1}{2}\mathbf{x}^T \boldsymbol{\Omega} \mathbf{x} + \boldsymbol{\xi}^T \mathbf{x}\}. \tag{2.16}$$

which is used for recursive state estimation with the information filter. The additivity of information is exploited in the information filter in the "update" stage where new sensor data is processed additively.

Finally, applying the equation for Shannon entropy (2.6) to the Gaussian distribution a relationship between the Shannon and Fisher information forms can be derived as:

$$H(\mathbf{X}) = -\frac{1}{2} \log\{(2\pi e)^n |\boldsymbol{\Omega}(\mathbf{X})|\} \tag{2.17}$$

where $|\ |$ is the determinant operation. With (2.17) we can calculate the Shannon entropy from the Fisher information matrix but we cannot recover a unique Fisher information matrix from a Shannon entropy value.

2.2.3 Information-Theoretic Approaches to Path Planning

In the realm of robotics path planning, the information concepts can be applied in a number of different ways.

One of the most common is for target localization. The approach is based on optimal sensor placement theory developed in [42]. Autonomous vehicle paths are planned to provide optimal position estimation of a fixed or moving target. In [41], and later in [234], information gain is used to determine paths from a start point to a goal point where some points in the environment are characterized as targets that should be classified. Paths are chosen based on how much information they provide about the targets. These methods are commonly used for target localization from an unmanned aerial vehicle (UAV). For example, in [92] a target is localized using a team of quadrotor helicopters. The pose of the target is estimated using a particle filter, and a formulation for evaluating the entropy of a particle set is introduced. A similar approach for a fixed-wing UAV using particle filters and information gain is presented in [195]. In [121], multiple UAVs cooperatively perform exploration and gather information about objects of interest. An optimization of vehicle heading is performed to minimize mission time using the Shannon channel capacity equation to quantify information. In [216], fixed-wing UAVs are used to localize a target where the utility of actions is formulated as information gain or probability of detection. In this case the search is planned over the set of possible trajectories and it is assumed that the relationship between control inputs and outputs is perfect and deterministic. This assumption implies that once the search over a finite horizon, or look-ahead time, is complete, the UAV is able to *exactly* follow the optimal trajectory. A similar approach is presented in [206] where the complexity of information theoretic planning over a fixed control horizon is also noted. Also related is [70] where an optimization is performed over vehicle headings to minimize the probability of misclassification of a target as an alternative to an information gain objective function. The problem of classification of objects of interest at known locations is related to the coverage problem studied here. For example, a similar possible solution is presented in [102] where Shannon information is used to plan paths. In this thesis, no assumption is made about *a priori* knowledge of the locations of objects of interest and instead the mission is formulated as an area coverage problem. The relationship between these two problems is akin to the difference between the travelling salesman problem and Chinese postman problem [149]. The travelling salesman problem attempts to find the shortest path that visits all nodes in a graph and returns back to the starting location. In the chinese postman problem, the path has to follow every edge in the graph rather than visit every node.

The problem of optimal planning of angles of viewing, or “looks” of rectangular objects at known location but unknown orientation is investigated in [38]. References to AUV planning based on information are relatively scarce. In [32], mutual information is used as the benefit metric in the objective function, combined with a recursive greedy planner. The proposed grid decomposition results in very restrictive paths, and the paths designed apply more readily to a underwater glider.

Another related application of information is for the task of adaptive exploration. Again, originally used on ground robots [44] and more recently used for planetary exploration [213], the problem can be posed through the use of information as an optimal control problem [134]. For example, in [122], an algorithm using information is presented to define optimal paths for a UAV to navigate through a region and obtain information about objects of interest. Exploration and coverage of an unknown environment are closely related [86]. However, as noted in [215], a pure information gain solution to the exploration task results in a greedy solution, and therefore suffers from reduced performance in the case that the mission objective is complete coverage.

Information is also used in the SLAM literature as active localization where control actions are selected that support the localization objective based on how much information they provide about the robot’s location given the map. For example, in [110] multiple quadrotors use information theory to collectively localize using a linearly scalable framework. In [87] information is used to control multiple platforms for localization using bearing-only sensing.

Finally, information can be used to trade off the benefits of exploration and localization [37]. For example in [39], information is used to generate online paths for a UAV in a SLAM context and a detailed observability analysis is provided.

2.3 Sidescan Sonar Sensor

Many underwater MCM missions are conducted with a side-looking sensor: either a synthetic aperture sonar (SAS) or a sidescan sonar sensor (SSS). In this research the SSS has been used. The SSS uses the returns from emitted high frequency sound to generate an image of the seabed. An object sitting on the seabed will cast a sonar shadow that can be analyzed to determine if the shape is suggestive of a mine. The on-board SSS gathers data as the AUV moves forward in rectilinear motion and leaves a narrow channel of unscanned seabed directly beneath it. An AUV path and corresponding SSS coverage swath are shown in Fig. 2.12.

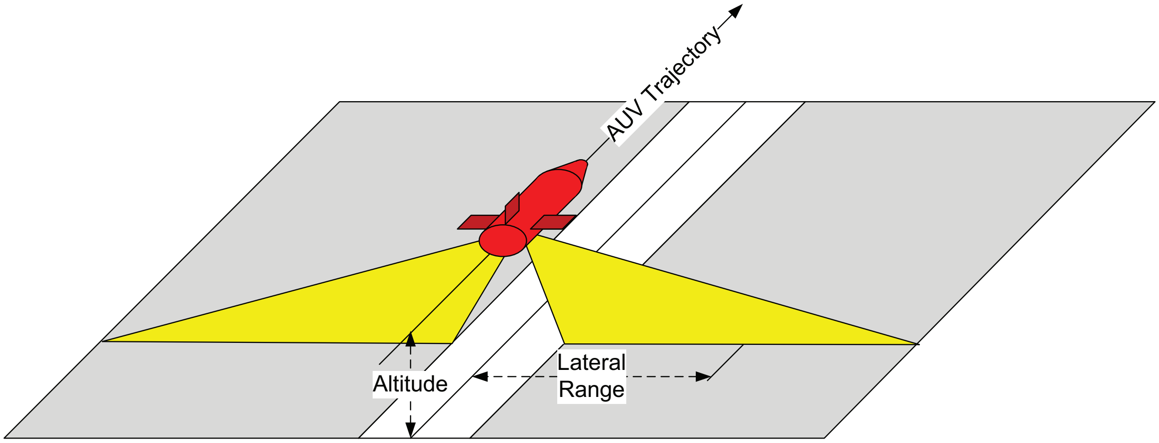


Figure 2.12: An example of the AUV trajectory and corresponding area covered by its SSS.

SSS returns are combined with navigation data to provide geo-referenced mosaicks of the seabed. An example of a mosaicked image is shown in Fig. 2.13. The target is highlighted. Often targets will be detected using automated systems [233], which have recently been implemented on-board AUVs operating in real-time [229]. When the sonar makes sharp turns, areas on the outside of the turn are missed or distorted due to the finite ping rate of the sonar. As a result, it becomes very difficult for systems that rely on template matching to identify targets in these areas [45] and consequently these data are usually completed discarded.

The angle of incidence of the sonar beam with the seabed has a significant effect on the size of the shadow cast by an object and therefore the probability of successful mine detection and classification.

In 2006, there was an effort by the NATO Undersea Research Centre to build a model to quantify minehunting performance with respect to all factors that can influence the probability of detecting a target in sonar data. The result was the Extensible Performance and Evaluation Suite for Sonar (ESPRESSO) [54] which is used in this thesis to model the coverage performance of the sonar.

The program takes as its input a set of parameters and outputs a lateral range curve, which is defined as follows [54]:

Definition 2.3.1. *The **lateral range curve** $P(y)$ defines the probability that a target at a specified lateral distance, y from the robot's track will be detected somewhere along that track.*

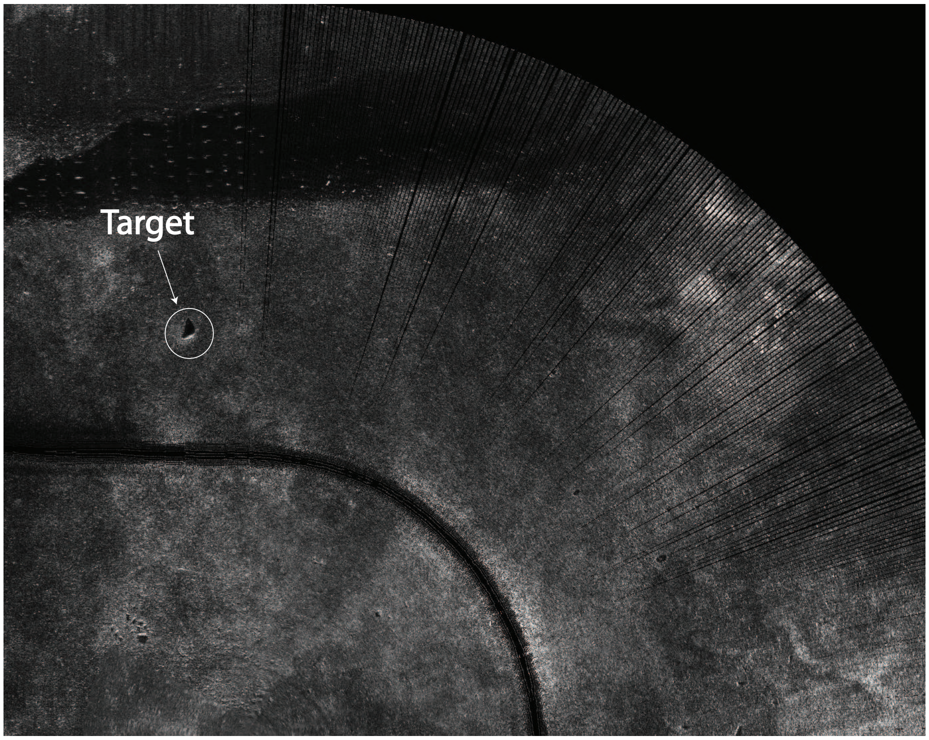


Figure 2.13: A portion of mosaicked imagery after stitching together the data from a sidescan sonar survey with a towed body. Image obtained from Defense R&D Canada.

Parameter values that affect the generation of the lateral range curve are described in Table 2.1 along with the general way in which these parameters are determined. In ESPRESSO, these parameters are set by the user to generate a sensor performance characteristic (lateral range curve) [54]. The parameters fall into four general categories that will affect detection: environmental, target, sonar, and platform.

Fig. 2.14 shows the lateral range curves generated by ESPRESSO for three different seabed types: cobble, sand, and clay, all at a depth of 10 m. In the figure, the "confidence" at a lateral range y represents the probability that if a mine is present it will be detected in the subsequent "Mine Detection / Object Classification" stage of the AUV MCM pipeline shown in Fig. 1.1. This definition will be more formally described in Chapter 3.

It should be explicitly stated that the purpose of this work is not to verify the ESPRESSO model, but rather to plan paths based on the model. Any underwater sonar sensor's performance will be affected by some or all of the parameters described in Table 2.1 and generally only half of these parameters are known beforehand. In this work we evaluate the benefit of potential actions using the ESPRESSO model to represent the sensor characteristic but without assuming known parameters.

If some of the parameters in Table 2.1 are not exactly known at the time of mission planning the plans have to be made overly conservative. Alternatively, a preliminary survey called a rapid environmental assessment can be performed to determine the parameters over the area of interest, but this adds to the total time for the mission.

In Chapter 3, a formulation based on uncertain sonar performance is developed and optimal (in the sense of information gathering) plans can be developed notwithstanding the uncertainty in the lateral range curve.

Category	**Parameters**	**Method of Detection**
Environmental, E	Seabed type	Sonar or camera imagery
	Water salinity and temperature	CT sensor
	Water clarity	Camera
	Water depth	Sonar
Target, F	Mine type, size and configuration	Only available through prior surveys of the area
Sonar, S	Frequency and range of sensor	Known beforehand from sensor specifications
Platform V	Speed	Doppler velocity log
	Depth	Pressure sensor

Table 2.1: Parameters affecting sonar performance characteristics.

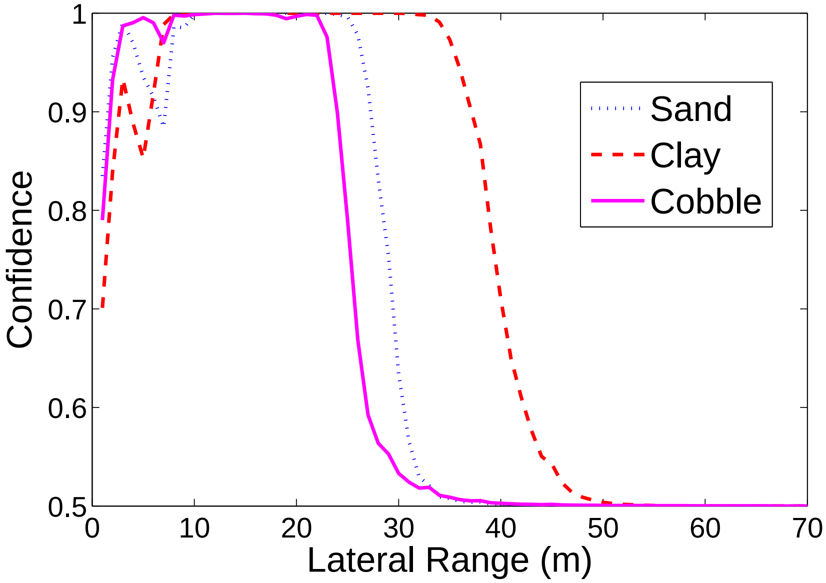


Figure 2.14: Three sample lateral range P(y) curves generated by ESPRESSO [54]. The AUV is equipped with sonar sensors on both sides of the vehicle, this plot shows only one side.

2.4 Autonomous Underwater Vehicle Localization

AUV localization is a challenging problem due primarily to the limited bandwidth and rapid attenuation of higher frequency acoustic signals and the unstructured nature of the undersea environment[3]. Above water, most autonomous systems rely on radio frequency communications and global positioning. However, underwater such signals propagate only short distances and acoustic based sensors and communications perform better. Acoustic communications still suffer from many shortcomings such as:

- Small bandwidth, which means that multiple pairs of communicating nodes have to use time division multiple access (TDMA) techniques to share the communication channel,
- Low data rate, which constrains the volume of data that can be transmitted,
- High latency since the speed of sound in water is only 1500m/s (five orders of magnitude slower than electromagnetic signals),
- Variable sound speed due to variable water temperature, density, and salinity,
- Multi-path transmissions due to the presence of an upper (free surface) and lower (sea bottom) boundary coupled with variable sound speed
- Unreliability, resulting in the need for a communications system designed to handle frequent data loss in transmissions.

An alternative to acoustic communication are optical, or laser, communications techniques that can achieve higher reliability but suffer from reduced range, usually on the order of 100m.

Notwithstanding these significant challenges, research in AUV localization has exploded in the last ten years. The field is in the midst of a paradigm shift from old technologies, such as long baseline (LBL) and ultra short baseline (USBL), which require pre-deployed and localized assets, towards dynamic multi-agent system approaches that allow for rapid deployment and flexibility with minimal infrastructure. In addition, SLAM techniques developed for above water robotics applications are being increasingly applied to underwater systems. The result is that bounded error and accurate localization for AUVs is becoming possible with less cost and overhead.

2.4.1 Overview

AUV localization techniques can be categorized according to Fig. 2.15.

In general, these techniques fall into one of three main categories:

- **Inertial / Deduced Reckoning**: Inertial navigation uses accelerometers and gyroscopes for increased accuracy to propagate the current state. Nevertheless, all of the methods in this category have position error growth that is unbounded.
- **Acoustic transponders and modems**: Techniques in this category are based on measuring the time-of-flight (TOF) of signals from acoustic beacons or modems.

[3] The review presented in this section has been accepted for publication [174].

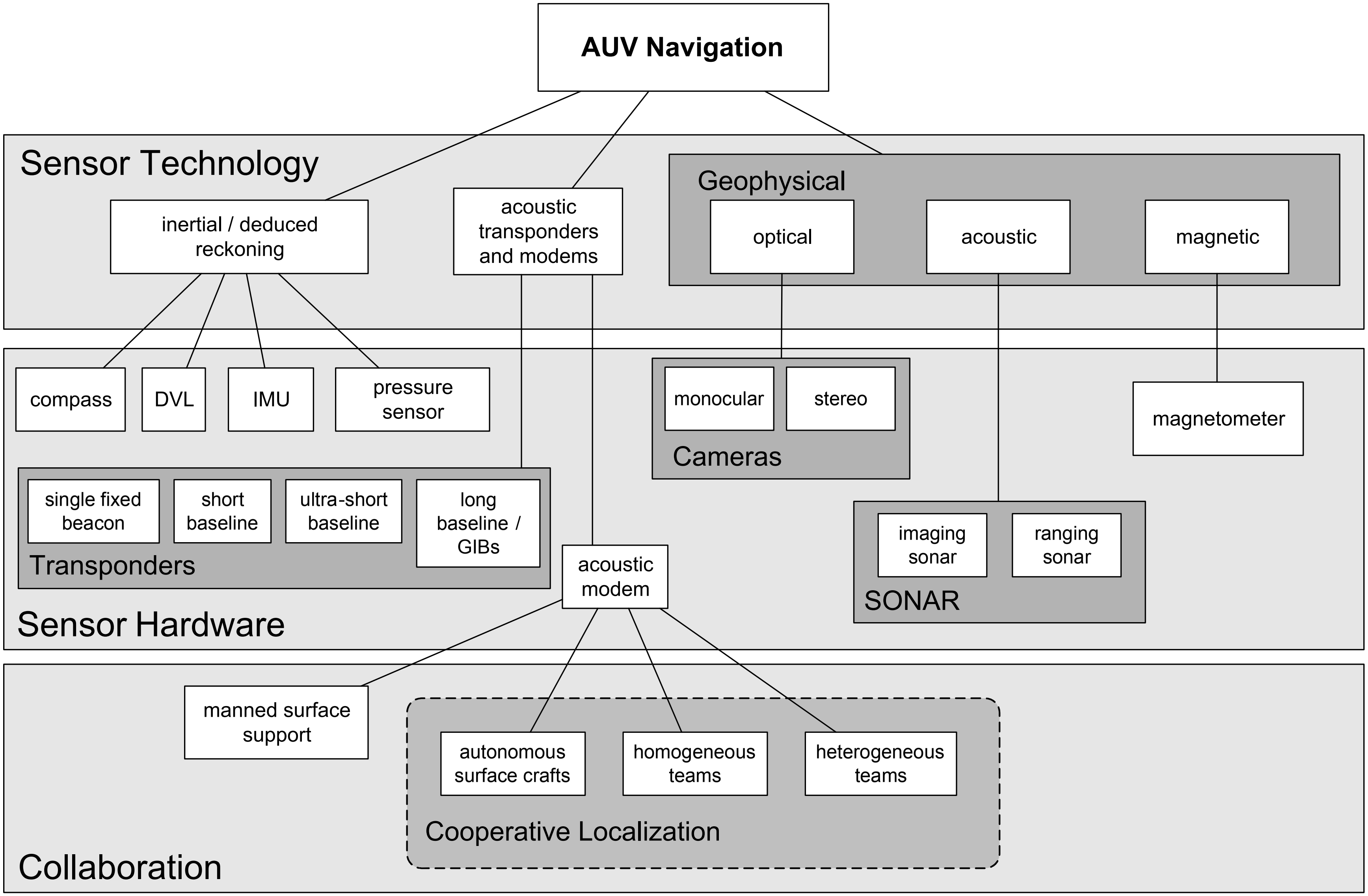


Figure 2.15: Outline of underwater navigation classifications. These methods are often combined in one system to provide increased performance (figure appears in joint publication [174]).

- **Geophysical**: Techniques that use external environmental information as references. This must be done with sensors and processing that are capable of detecting, identifying, and classifying some environmental features.

The type of navigation system used is highly dependent on the type of operation or mission and in many cases different systems can be combined to yield increased performance. The most important considerations are the size of the region of interest and the desired localization accuracy.

Past reviews on this topic include [211], [137], and [118]. Significant advances have been made since these reviews both in previously established technologies, and in new areas. In particular, the development of acoustic communications through the use of underwater modems has lead to the development of new algorithms. In addition, advancements in SLAM research have been applied to the underwater domain in a number of new ways.

2.4.2 Commonly Used Sensors

Table 2.2 describes some commonly used sensors for underwater navigation.

	Description	**Performance**	**Cost**
3-axis Compass	A compass provides a globally bounded heading reference. A typical magnetic compass does so by measuring the magnetic field vector. This type of compass is subject to bias in the presence of objects with a strong magnetic signature and points to the earth's magnetic north pole. More common in marine applications, a gyrocompass measures heading using a fast spinning disc and the rotation of the earth. It is unaffected by metallic objects and points to true north.	Accuracy within 1^o to 2^o for a modestly priced unit.	On the order of hundreds of dollars US.
Pressure Sensor	Underwater depth can be measured with a barometer or pressure sensor.	Since the pressure gradient is much steeper underwater (10m = 1 atmosphere) we can achieve high accuracy ≈ 0.1m.	≈ \$100 to 200 USD

Doppler Velocity Log	The Doppler velocity log uses acoustic measurements to capture bottom tracking and determine the velocity vector of an AUV moving across the seabed. It determines the AUV surge, sway, and heave velocities by transmitting acoustic pulses and measuring the Doppler shifted returns from these pulses off the seabed. Doppler velocity logs will typically consist of four or more beams. Three beams are needed to obtain a 3D velocity vector.	Nominal standard deviation on the order of 0.3cm/s to 0.8cm/s.	≈ $20k to 80k USD
Sonar	A sonar is a device for remotely detecting and locating objects in water using sound. Passive sonars are listening devices that record the sounds emitted by objects in water. Active sonars are devices that produce sound waves of specific, controlled frequencies, and listen for the echoes of these emitted sounds returned from remote objects in the water. Active sonars can be categorized as either imaging sonars that produce an image of the seabed, or ranging sonars which produce bathymetric maps. More details of specific active sonar devices are presented in Table 2.3 in Section 2.4.6.2.	Along-track image resolution for an imaging side-scan sonar is a function of many factors such as range, sonar frequency, and water conditions, however cross-track resolution is independent of range. For example, a Klein 5000 side-scan operating at 455kHz can achieve an along track resolution of 10cm at 38m range and 61cm at the maximum 250m range and a Klein 5900 sidescan operating at 600kHz can achieve along track resolution 5cm at 10m and 20cm resolution at the maximum 100m range. In both cases nominal cross-track resolution is 3.75cm. Resolution for a bathymetry sonar is on the order of $\approx 0.4^o$ to 2^o along track and ≈ 5 to 10cm cross-track.	Prices vary widely from 20k to 200k USD or more.

GPS	GPS can be used for surface vehicles. Position is estimated using the TOF of signals from synchronized satellites.	Many factors can influence the accuracy of a GPS reading, including type of GPS technique used, atmospheric conditions, number of satellites in view, and others. Precisions for different GPS systems are: common commercial off-the-shelf GPS - 10m, Wide Area Differential GPS - 0.3 to 2m, Real-Time Kinematic - 0.05 to 0.5m, Post processed - 0.02 to 0.25m.	From hundreds to thousands of dollars.
Inertial Measurement Unit (IMU)	Use a combination of accelerometers and gyroscopes (and sometimes magnetometers) to estimate a vehicle's orientation, velocity, and gravitational forces. • Gyroscope: Measures angular rates. For underwater applications. Commonly-used types include ring laser and MEMS Since a gyroscope measures angular rates, there will be a drift in the estimated Euler angles as a result of integration. • Accelerometer: Measures the force required to accelerate a proof mass. Common designs include pendulum, MEMS, and vibrating beam among others.	• Gyroscope - Drift extremely variable from $0.0001^o/hr$ (ring laser) to $60^o/hr$ or more for MEMS [56]. • Accelerometer - Bias range from $0.01mg$ (MEMS) to $0.001mg$ (Pendulum) [56].	Extremely variable. From hundreds of dollars for a MEMS IMU to hundreds of thousands of dollars for a commercial grade ring-laser or fibre optic system.

Table 2.2: Some on-board AUV sensors used for state estimation.

2.4.3 Inertial Systems

When the AUV positions itself autonomously, with no acoustic positioning support from a ship or acoustic transponders, it dead reckons. With deduced (ded) reckoning, the AUV advances its position based upon knowledge of its orientation and velocity or acceleration vector. Traditional ded reckoning is not considered a primary means of navigation but modern navigation systems, which depend upon ded reckoning, are widely used in AUVs. The disadvantage of ded reckoning is that errors are cumulative. Consequently, the error in the AUV position grows unbounded with distance traveled.

One simple method of ded reckoning pose estimation, for example if heading is available from a compass and velocity is available from a Doppler velocity log (DVL), is achieved by using the following kinematic equations:

$$\begin{aligned} \dot{x} &= v^u \cos\psi + v^v \sin\psi \\ \dot{y} &= v^u \sin\psi + v^v \cos\psi \\ \dot{\psi} &= 0 \end{aligned} \tag{2.18}$$

where (x, y, ψ) is the position and heading in the standard North-East-Down coordinate system, and v^u, and v^v are the body frame forward and starboard velocities. In this model it is assumed that roll and pitch are zero and that depth is measured accurately with a depth sensor.

An inertial system aims to improve upon the ded reckoning pose estimation by integrating measurements from accelerometers and gyroscopes. Inertial proprioceptive sensors are able to provide measurements at a much higher frequency than acoustic sensors that are based on the TOF of acoustic signals. As a result, these sensors can reduce the growth rate of pose estimation error, although it will still grow without bound.

One problem with inertial sensors is that they drift over time. One common approach, for example used in [153], is to maintain the drift as part of the state space. Slower rate sensors are then used to calibrate the inertial sensors. In [153], the authors also track other possible sources of error such as the variable speed of sound in water to reduce systematic noise. These noise sources are propagated using a random walk model, and then updated from DVL or LBL sensor inputs. Their inertial navigation system (INS) is implemented with an inertial measurement unit that runs at 150Hz.

The basic kinematics model (2.18) is incomplete if the local water current is not accounted for. The current can be measured with an acoustic Doppler current profiler (ADCP). For implementations with ADCP see [84] [89]. A DVL is usually able to calculate the velocity of the water relative to the AUV, v_b and the velocity of the seabed relative to the AUV, v_g. Then the ocean current can be calculated easily as $v_c = v_g$ - v_b. The ocean current can also be obtained from an ocean model, for example in [94] where ocean currents are predicted using the regional ocean modeling system [202] combined with a Gaussian process regression [184]. If access to the velocity over the seabed is not available, then the current can be estimated from a transponder on a surface buoy as in [24]. In [24], the authors analyzed the power spectral density to remove the low frequency excitation on the buoy due to the waves to estimate the underwater current.

The performance of an INS is largely determined by the quality of its inertial measurement units. In general, the more expensive the unit, the better its performance. However, the type of state estimation also has an effect. The most common filtering scheme is the extended Kalman filter (EKF), but others have been used to account for the linearization and Gaussian assumption shortcomings of the EKF. For example, in [112] an unscented Kalman filter (UKF) is used and in [99] a particle filter (PF) application is presented.

Improvements can also be made to INS navigation by modifying (2.18) to provide a more accurate model of the vehicle dynamics. The benefits of such an approach are investigated in [90], particularly in the case that DVL loses bottom lock, for example.

Inertial sensors are the basis of an accurate navigation scheme, and have been combined with other techniques described in subsequent sections. In certain applications, navigation by inertial sensors is the only option. For example, in extreme depths where it is impractical to surface for GPS, an INS is used predominantly, as described in [227].

The best INS can achieve a drift of 0.1% of the distance traveled [74], however, more typical and modestly priced units can easily achieve a drift of 2-5% of the distance traveled.

2.4.4 Acoustic Transponders and Beacons

In acoustic navigation techniques, localization is achieved by measuring ranges from the TOF of acoustic signals. Common methods include:

- **Ultra Short Baseline**: Also sometimes called super short baseline. The transducers on the transceiver are closely spaced with the approximated baseline on the order of less than 10 centimetres. Relative ranges are calculated based on the TOF and the bearing is calculated based on the difference of the phase of the signal arriving at the transceivers. See Fig. 2.16-b.

- **Short Baseline**: Beacons are placed at opposite ends of a ship's hull. The baseline is based on the size of the support ship. See Fig. 2.16-a.

- **Long Baseline and GPS Intelligent Buoys**: Beacons are placed over a wide mission area. Localization is based on triangulation of acoustic signals. See Fig. 2.16-c. In the case of GPS intelligent buoys, the beacons are at the surface whereas for LBL they are installed on the seabed.

- **Single Fixed Beacon**: Localization is performed from only one fixed beacon.

- **Acoustic Modem**: The recent advances with acoustic modems have allowed for new techniques to be developed. Beacons no longer have to be stationary.

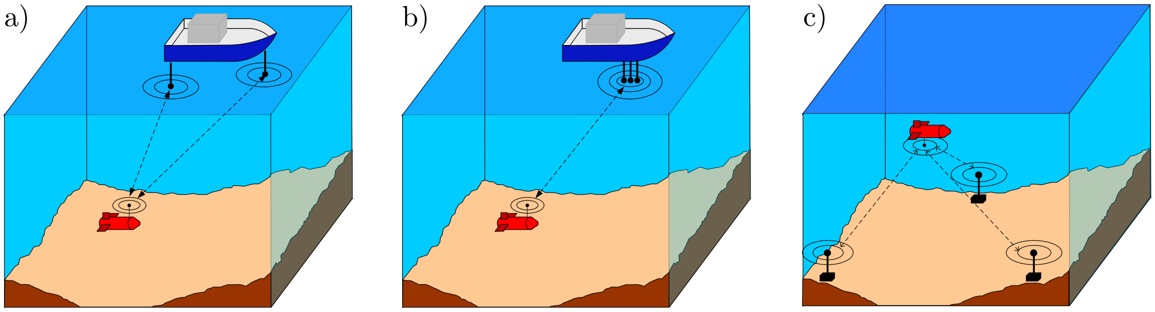


Figure 2.16: (a) Short baseline (b) Ultra-short baseline (c) Long baseline (figure appears in joint publication [174])

Due to the latency of acoustic updates, state estimators are implemented where the ded reckoning proprioceptive sensors provide the predictions and then acoustic measurements provide the updates.

2.4.4.1 Ultra Short and Short Baseline

USBL navigation allows an AUV to localize itself relative to a surface ship. Relative range and bearing are determined by TOF and phase differencing across an array of transceivers, respectively. A typical setup would be to have a ship supporting an AUV. In short baseline, transceivers are placed at either end of the ship hull and triangulation is used.

The major limitation of USBL is the range and of short baseline is that the positional accuracy is dependent on the size of the baseline, i.e. the length of the ship.

In [190] an AUV was developed to accurately map and inspect a hydro dam. A buoy equipped with an USBL and differential GPS helps to improve upon ded reckoning of the AUV which is performed using a motion reference unit, a fibre optic gyro and a DVL. An EKF is used to fuse the data and a mechanical scanning imaging sonar (MSIS) tracks the dam wall and follows it using another EKF. For this application, the USBL is a good choice because the range required for the mission is small. The method proposed in [186] augments [190] by using a delayed-state information filter to account for the time delay in the transmission of the surface ship position.

In [23], sensor based integrated guidance and control is proposed using a USBL positioning system. The USBL is installed on the nose of the AUV while there is an acoustic transponder installed on a known and fixed position as a target. While homing, the USBL sensor listens for the transponder and calculates its range and the bearing based on the time difference of arrival. In [181], USBL is used for homing during the recovery of an AUV through sea ice.

In [113], data from an USBL and an acoustic modem is fused by a particle filter to improve ded reckoning. As a result, the vehicle operates submerged longer as GPS fixes can be less frequent. The simulation and field experiments verify the developed technique.

In [154], a 'tightly-coupled' approach is used where the spatial information of the acoustic array is exploited to correct the errors in the INS.

2.4.4.2 Long Baseline / GPS Intelligent Buoys

In LBL navigation, localization is achieved by triangulating acoustically determined ranges from widely spaced fixed beacons. In most cases, the beacons are globally referenced before the start of the mission by a surface ship [128], a helicopter [104], or even another AUV [220]. In normal operation, an AUV would send out an interrogation signal, and the beacons would reply in a predefined sequence. The two way travel time of the acoustic signals is used to determine the ranges. However, there have been implementations in which synchronized clocks are used to support one-way travel time ranging [51].

GPS intelligent buoys remove the need for the LBL beacons to be installed at the seafloor which can reduce installation costs and the need for recovery of these beacons.

One of the limitations of LBL is the cost and time associated with setting up the network. However, this can be mitigated to some extent if the beacon locations are not globally referenced and either self-localize [50], or the AUV can localize them by performing SLAM. For example, [161] uses a nonlinear least squares implementation, whereas [177] uses a particle filter version of SLAM to determine the location of the fixed beacons during the mission.

A major consideration in an LBL localization network is the treatment of outliers. Methods to account for outliers in LBL systems include hypothesis grids [31] and graph partitioning [166]. Generally, range measurements can fall into one of three categories: direct path, multi-path, or outlier. In [31], a hypothesis grid is built to represent the belief that future measurements from a particular cell will be in a particular category. In graph partitioning, outliers are rejected using spectral analysis. A set of measurements is represented as a graph, then the graph partitioning algorithm is applied to identify sets of consistent measurement [166].

Another consideration is the time difference of arrival of the acoustic responses of the network [33], [59]. The change in vehicle pose between the initial interrogation request and all of the subsequent replies must be explicitly handled. This is often done with a delayed state EKF.

Major drawbacks of LBL are the finite range imposed by the beacons, and the reliance on precise knowledge of the local sound velocity profile of the water column based on temperature, salinity, conductivity, and other factors [128]. However, the LBL systems do overcome these shortcomings to be one of the most robust, reliable, and accurate localization techniques available. For this reason it is often used in high-risk situations such as under-ice surveys [104], [126]. Other implementations include: alignment of Doppler sensors [119] and deep water surveys [232]. An extensive evaluation of the precision of LBL is provided in [30].

2.4.4.3 Single Fixed Beacon

A downside of LBL systems is the cost and time required for installing the beacons and geo-referencing them. It is possible to reduce these infrastructure requirements if only a single fixed beacon is used instead of a network of them. The concept is that the baseline is synthesized by propagating the ranges from a single beacon forward in time until the next update is received. This technique has been referred to as virtual LBL and has been simulated on real world data in [131]. It is noted that the AUV trajectory has a significant effect on the observability of the vehicle state. Long tracks directly towards or away from the single fixed beacon will cause unbounded growth in position error. As a result, tracks or paths for the survey vehicle being localized in this manner should be planned to be tangential to range circles emanating from the transponder.

A visual representation of single beacon navigation is shown in Fig. 2.17. It assumes that the vehicle has prior knowledge of the beacon location. In the figure the vehicle receives three acoustic pings from the beacon at the bottom. Each time, reception of a ping results in a reduction of uncertainty in the direction of the beacon.

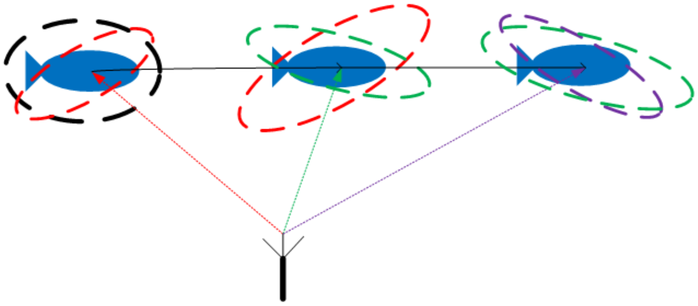


Figure 2.17: An AUV localizing with a single fixed beacon at known location. Uncertainty grows in between updates from the beacon. On reception of an update, uncertainty is reduced in the dimension coinciding with the location of the beacon. (figure appears in joint publication [174])

Single beacon navigation has also been used for homing, for example in [79]. This task is particularly challenging because the AUV will preferentially move in a track directly towards the beacon, violating the observability criterion. As a result, path planning must be designed to avoid this situation. This can be particularly useful for recovering an AUV that has become inoperable, but is still able to transmit pings.

2.4.4.4 Acoustic Modem

Advances in the field of acoustic communications have had a major effect on underwater navigation capabilities. The acoustic modem allows simultaneous communication of small packets and ranging based on TOF. If the position of the transmitter is included in the communicated information, then the receiver can bound its position to a sphere centered on the transmitter. This capability removes the need for beacons to be fixed or localized prior to the mission. In addition, it allows for inter-AUV communication, which means teams of AUVs can cooperatively localize.

Popular acoustic modems are manufactured by Woods Hole Oceanographic Institute, Teledyne Benthos, and Evologics among others. In general communication can either use frequency shift keying with frequency hopping (FH-FSK), which is more reliable but provides lower data rates, or variable rate phase-coherent keying (PSK). Some models also include precise pulse-per-second clocks to allow synchronous ranging. Typically, due to the limited bandwidth underwater, the communication channel is shared using a TDMA scheme. In TDMA, each member in the group is allotted a time slot within which to broadcast information. The major detractor of such a scheme is that the total cycle time grows with group size. At present, achievable bit rates range from 32 bytes per 10s packet with FSK, to several kbit/s in optimal conditions with PSK.

The ability of a modem at the surface to transmit its location to the survey vehicles provides two important benefits over past navigation methods: 1) it removes the necessity to geo-reference the beacons prior to starting the mission, and 2) it allows the beacons to move during the missions. The first advantage saves time and money, and the second allows the mission range to be extended as necessary without re-deploying the sensor network. Many methods have been recently published that exploit one or both of these benefits.

The moving long baseline concept was first demonstrated in [218] using a Sonardyne AvTrack acoustic navigation system. Two manned surface vehicles were used to support one AUV. This concept has proved particularly useful for mapping rivers, such as in [150], where two boats are used to continuously define a cross-section of the river to be mapped near the location of the AUV.

This approach has been extended to a single moving source. A manned surface ship can localize itself with GPS and bound the error of one or more survey AUVs by broadcasting its position. Such an approach is attractive because there is no need for calibration or recovery of beacons.

In [225] and [224], a deep water validation was performed for the single moving beacon concept. It should be noted that localization is done in post-processing. In the proposed approach, an EKF maintains an estimate of the of the position of the survey vehicle as well as the support ship. In a deep water application, the time taken for the acoustic transmission should be accounted for in the filtering algorithm. Here, this is represented as a delayed state EKF. It is accurately noted that the TOF measured is the range between the current position of the receiver and a previous position of the sender. In [224], the performance of the single beacon navigation is compared against an LBL system. Also, there is a more rigorous discussion on sources of error in acoustic range measurements such as errors in sound velocity estimation, acoustic multi-path, and errors in ship GPS. Similarly, in [151] a deep diving AUV is localized in post-processing. In this case, the survey vehicle is required to execute a known closed path after it dives to a depth of 6000m. A similar approach is presented in [80] to map the magnetic signature of a moving vessel.

If the survey vehicles are acting as passive listeners using one-way travel time for range measurement, then the system naturally scales well with number of survey vehicles assuming they stay within range of the surface vehicle. In [66] and [69], a maximum likelihood sensor fusion technique is presented to localize survey vehicles to within 1m over a 100km survey. It is also noted in this paper that over a long survey, the drift of the pulse-per second clock will have a significant effect on the localization performance.

2.4.5 Cooperative Localization

In GPS-denied environments mobile robots can estimate their positions by ded reckoning, integrating velocity or acceleration measurements. The position itself is not observable so the position error will grow without bound.

2.4.5.1 Background

It has been shown that if a team of robots without access to a global reference can share their information and make relative measurements of each other, then the rate of increase of the position error will grow more slowly for every member of the team [192] [96] [185] [81]. This is referred to in the literature as cooperative localization.

Likely the first research done on the subject was presented in [127] where robots in a team take turns acting as stationary landmarks. Since, the restrictions on robots being stationary have been removed and advances have been made in several areas including:

- Comparing different estimators, such as maximum likelihood [96], maximum *a posteriori* (MAP) [159], EKF [97], particle filter [182], nonlinear least squares [235],
- Reducing computational complexity, such as [6]. Full complexity analysis is performed in [60],
- Reducing communication overhead, for example through quantization of data [217, 160], or by careful consideration of which data needs to be transmitted [158] [138].

Another potential advantage of the cooperative localization approach is that increased sensing capabilities of one member can benefit the entire team. For example, in [17], it is shown how one vehicle with a more expensive INS can result in increased localization performance for vehicles that it is able to communicate with. A thorough performance analysis for this case is presented in [155].

In [193], the scalability of cooperative localization is addressed, and it is shown that an upper bound on the rate of increase of position uncertainty is a function of the size of the robot team. The more robots in the team, the better the cooperative localization will be. However, this is subject to the law of diminishing returns: each new additional member improves the location estimation of the team less than the previous. Other important results have been proven, such as that the maximum expected rate of uncertainty increase is independent of the accuracy and number of range measurements and depends only on the accuracy of the proprioceptive sensors on the robot [155].

Another aspect of the cooperative localization scheme that has been extensively investigated is the observability of the system. For example, in [201], the bearing only case is considered for cooperative SLAM, and in [235] the range only cooperative localization case is analyzed for observability properties. In [97], it is shown that the observable subspace of a standard EKF implementation of cooperative localization is of higher dimension than the true nonlinear system, which can lead to overconfidence in the EKF estimate.

Here we are particularly interested in estimating the entire robot trajectory to minimize the position error of objects of interest detected in sonar data. Therefore we are particularly interested in the MAP cooperative localization case, specifically applied to the underwater environment.

A distributed MAP approach is presented in [159]. Here it is shown that, in the case of a fully connected and high bandwidth communication scheme, the calculation of the batch MAP solution can effectively be distributed across members of the team. This could be particularly effective in the case that communications are reliable but on-board computation is lacking. However, the approach is infeasible for underwater implementation since the bandwidth requirements and communication synchronicity are violated.

2.4.5.2 Underwater Cooperative Localization

Although many general cooperative localization papers cite the underwater world as a possible application space, most make either implicit or explicit assumptions which are not applicable to underwater cooperative localization. Nevertheless, advances in underwater cooperative localization have seen a surge in recent years due largely to advances in underwater communications capabilities.

A graphical depiction of multiple AUV cooperative localization is shown in Fig. 2.18. Data is transmitted through the acoustic channel. Upon reception of a data packet, the receiver, vehicle j, can use the TOF of the acoustic signal to determine its range, r_{ij}, from the sender, vehicle i. If the vehicles possess well synchronized clocks, then this range can be determined from the one-way travel time of the acoustic signal, otherwise an interrogation-reply is performed to determine a round-trip range.

The result of these new communications capabilities is that operations that were previously performed with fixed and pre-localized beacons can now be adapted to use nodes with acoustic modems as moving beacons. For example, in [218] two manned surface vehicles are used to support an AUV, referred to as moving long baseline. Once it has been established that vehicles can navigate with the help of manned surface vehicles, a natural progression for the sake of increased autonomy is to move towards unmanned surface vehicles. The first known implementation of autonomous surface crafts (ASCs) being used to support AUVs is presented in [52], which is an extension of the moving long baseline concept. Two ASCs are used to support one Odyssey III AUV in a series of experiments in 2004 and 2005. In [15], two ASCs are used and a general framework is developed for cooperative localization. Bahr et al. developed a framework [16] whereby a single vehicle, termed a communications/navigation aid (CNA) can support a team of AUVs. The CNA can either be a surface craft or and AUV, but all measurements are made relative to the CNA and all communications are marshalled by the CNA. This is further verified by Fallon et al. [75] who used a single autonomous surface craft to aid AUV localization. In both cases, each successful acoustic communication restricts the survey vehicles' locations to an annulus assuming that all vehicles are able to accurately measure their depths.

In [76], the approach is extended to the homogeneous AUV case where any vehicle in the team can surface for a GPS fix to bound the location error of the entire team. In this case a "keyframe" rate is used where multi-vehicle state beliefs are only estimated at some integer fraction of the communication frequency. All other communications are used for

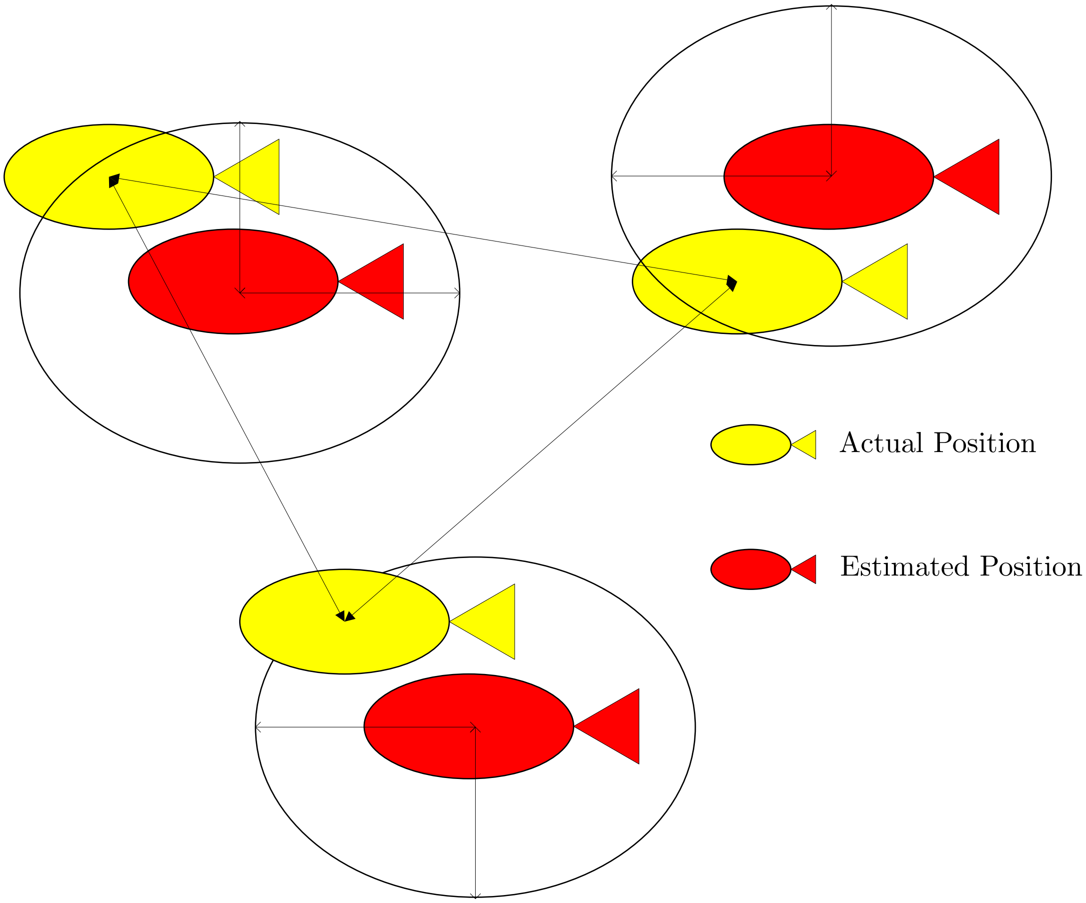


Figure 2.18: Cooperative localization for AUVs: relative ranges are determined from TOF of acoustic communication packets. (figure appears in joint publication [174])

marshalling data.

In [14] an “Interleaved Update” approach is used for a multi-AUV EKF implementation. In this approach, each vehicle maintains a bank of N^2 EKF filters where each filter only includes inter-vehicle measurements from a subset of the rest of the AUV team. On transmission, each vehicle transmits its estimate from all of the N^2 filters.

A key consideration in both [14] and [76] is maintaining consistency of the estimator by correctly accounting for the cross-correlations that are induced when one vehicle estimates its position relative to another. Also, optimizing the use of the communication channel is important since bandwidth is restricted and AUVs must take turns transmitting in a TDMA scheme. Further experimental results with an AUV/CNA cooperation scheme were recently presented in [221]. In this case unknown ocean currents are also explicitly accounted for.

2.4.6 Geophysical Localization

Geophysical localization refers to any method that utilizes external environmental features. Almost all methods in this category that achieve bounded position error use some form of SLAM. Categories include:

- **Magnetic**: It has been proposed to use magnetic field maps for localization. The only known work in this area is by Claus and Bachmayer [12] who have developed a magnetometry system for mapping and navigation using magnetic data. In addition, the team at the University of Idaho has been mapping the magnetic signatures of Navy vessels [8, 7].
- **Optical**: Use of a monocular or stereo camera to capture images of the seabed and then match these images to navigate.
- **Sonar**: Used to acoustically detect then identify and classify features in the environment that could be used as navigation landmarks. With bathymetric sonar features can be extracted almost directly from assembled returns. With sidescan (imaging) sonar, feature extraction is achieved through post-processing of imagery.

2.4.6.1 Optical

Visual odometry is the process of determining the robot pose by analyzing subsequent camera images. This can be achieved through optical flow or structure from motion. Invariant extraction and representation of features is an important consideration. Many previous algorithms have been proposed and applied in ground and air robotics, such as scale-invariant feature transform (SIFT) [142], speeded up robust feature (SURF) [25], amongst many others[4]. Images can be captured with either stereo or monocular cameras. Stereo cameras have the added advantage that full six degree-of-freedom transformations between consecutive image pairs can be found. The major challenge is closing loops in the trajectory by associating non-consecutive images, which is necessary to bound localization error.

Limitations for optical systems in underwater environments include the reduced range of cameras, susceptibility to scattering, and inadequacy of lighting. As a result, visible wavelength cameras are more commonly installed on hovering AUVs because they can get close to objects of interest. In addition, visual odometry and feature extraction rely on

[4] For a review and comparison of some methods, see [152].

the existence of features. Therefore optical underwater navigation methods are particularly well-suited to small-scale mapping of feature-rich environments. Examples include shipwreck and dam inspections.

In [67] and [64], Eustice et al. present an implementation of underwater vision-based SLAM called visually augmented navigation (VAN). The multi-sensor approach combines the benefits of optical and inertial navigation methods and is robust to low overlap of imagery. The approach is a view-based version of EKF-SLAM, where camera-derived relative pose measurements provide the spatial constraints for visual odometry and loop closure. In [68] and [65], the VAN approach is converted to the information form and it is proven that the view-based exact sparse extended information filter SLAM maintains a sparse information matrix without approximations or pruning. This approach was applied to deep water surveying of the RMS Titanic in [68]. The issue of recovering the mean and the covariance from the information form is also addressed. SIFT and Harris extraction points are used to match images. The VAN method has also been applied to ship-hull inspection for the U.S. Navy [116]. In recent work, [95] uses features from both a profiling sonar and a monocular camera in full pose graph formulation for a hovering AUV performing ship hull inspections. Odometry is derived relative to the ship hull using a DVL locked onto the ship. An approach to tackle the problem of data association in feature-poor areas of a ship hull is explored in [115] using a novel online bag-of-words approach to determine inter-image and intra-image saliency. In computer vision, the bag-of-words model can be used to categorize features in imagery.

Feature-based approaches to underwater SLAM have also been proposed. For example, in [62], an augmented EKF is used to generate a topological representation. Non-time-consecutive images are compared and loop closures are made based on observation mutual information. Feature-based EKF SLAM is also applied to the underwater environment in [197] and [198].

A byproduct of accurate localization is that accurate mapping can be achieved. Several authors have shown the ability to compute 2D and 3D reconstructions of the underwater environment based on optical underwater SLAM. For example, in [205], an structure from motion model together with SLAM are used to photomosaic the RMS Titanic as well as a hydrothermal vent area in the mid-Atlantic. In [109], 3D reconstructions of an underwater environment are done using SLAM with a combination of multi-beam sonar and stereo camera.

Pure visual odometry based methods that do not require SLAM can be used for applications such as pipeline tracking [167].

2.4.6.2 Sonar

Sonar imaging of the ocean pre-dates AUVs by decades. As a result, it is a fairly robust technology. Several types of sonars are used for seabed and structure mapping. They are summarized in Table 2.3.

Sonars are designed to operate at specific frequencies [120] depending on the range and resolution required. In all cases, the performance of the SLAM algorithm is dependent on the number and quality of the features present in the environment.

Imaging Sonar

The insonified swath of the sidescan sonar is shown in Fig. 2.19-a. The intensity of the

Sonar		Description	Pros	Trade-Off	Ref
Imaging	SSS	Multiple beams that measure intensity of returns to create a 2D image of sea bed; beams are directed perpendicular to travel direction.	Can work at relatively high speeds (10kt) to give high area coverage.	Resolution inversely proportional to range, e.g. 1.8 MHz produces 40 m range.	[194] [74] [105] [9]
	FLS	Similar in principle to a side-scan sonar only beams are directed forward.	Obstacle avoidance, also as a nadir gap filler.	Limited distance to depth ratio (6:1 max); single angle of view.	[222] [107]
	SAS	Coherent processing of consecutive displaced returns to synthesize a virtual array.	Range independent resolution.	Optimal at low speeds. and deeper water.	[203] [212] [163]
	MSIS	One beam with actuator that scans a swath.	Cheaper than multi-beam.	Slow. Accuracy depends on AUV attitude.	[189], [187], [146], [147], [148], [91], [188]
Ranging	Echo Sounder	Single, narrow beam used to determine depth below the transducer.	Yield representation of seabed and targets between transducer to seabed.	Point measurements in generally one direction.	[72, 73]
	Profiler	Low-frequency echo sounders that penetrate the seabed.	Information on subsurface features.	Penetration depth inversely proportional to resolution.	[145]
	Multi-Beam	TOF from returns to assemble bathymetric maps.	Gathers echo sounding data more efficiently than a single beam.	Resolution inversely proportional to frequency.	[21] [20] [19] [18]

Table 2.3: Sonar imaging and ranging devices used for underwater localization.

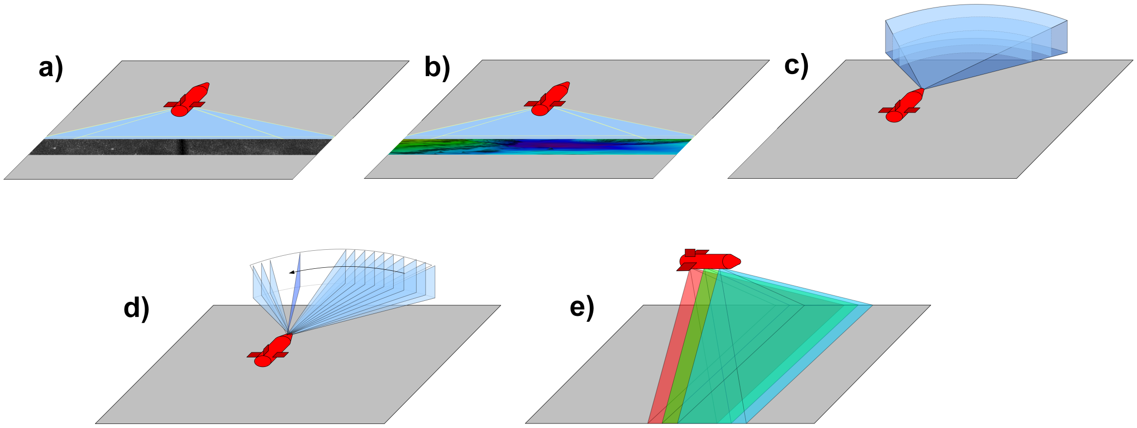


Figure 2.19: Sonar sensor swaths: (a) Sidescan (b) Multibeam (c) Foward looking (d) Mechanical scanning and imaging (e) Synthetic aperture. (figure appears in joint publication [174])

acoustic returns from the seabed of this fan-shaped beam also depends on the bottom type and is recorded in a series of cross-track slices. When mosaicked together along the direction of travel, these assembled slices form an image of the seabed within the swath of the beam. Hard objects protruding from the seabed send a strong return which is represented as a dark image. Shadows and soft areas, such as mud and sand, send weaker returns which are represented as lighter images.

SLAM with the sidescan was first presented in [194] using an augmented EKF and the Rauch-Tung-Striebel smoother. The value of smoothing in the pose estimation is emphasized since all previous poses should be updated when a loop closure event is detected. It is noted that automated feature detection (which is not trivial) and data association are necessary to achieve autonomy for sidescan sonar SLAM. In [74], the approach in [194] is improved using smoothing and incorporates range updates from a CNA. This is the only known research that combines a sonar based SLAM method with acoustic modem ranging. In [105], sidescan sonar SLAM is posed as an interval constraint propagation problem. In [10, 9], the problem was approached using a submap joining algorithm called the selective submap joining SLAM. The work in [9] uses a cascaded Haar classifier for object detection from sidescan imagery and is demonstrated off-line on a gathered dataset. A discussion of which type of features are appropriate for sidescan sonar SLAM is presented in [230]. All of the proposed approaches to sidescan sonar SLAM require post-processing to detect the features for data association.

A depiction of the forward looking sonar (FLS) is shown in Fig. 2.19-c. Based on the transducer geometry, the primary function of this type of sonar device is to map vertical features. As such it is commonly deployed on a hovering AUV capable of approaching man-made underwater structures at very low speeds. Feature-based SLAM based on exact sparse extended information filter has also been implemented with a FLS for ship hull inspections [222]. The issue of online feature extraction from FLS images is addressed in [107].

The MSIS is shown in Fig. 2.19-d. Its operation is similar to the FLS except that instead of multiple beams, a single beam is rotated through the desired viewing angle. Consequently, the update rate is slow. It cannot be assumed that the AUV pose is constant for an entire sensor scan cycle, which increases the complexity of mapping algorithms.

The group at the University of Girona has done significant work with the MSIS as presented in [189, 187, 146, 147, 148, 91, 188]. In [189, 188], online feature extraction and data association and an EKF SLAM implementation is performed. In addition, a submap method is used to reduce computational complexity. Given that the algorithm is based on line feature extraction, the method is well-suited to man made environments with well defined edges and boundaries. In [187], the slow update rate of the MSIS is accounted for with a delayed state EKF SLAM algorithm. In [148, 147, 146, 91], a probabilistic scan matching algorithm is presented that exploits the overlap in the images from the MSIS.

A figure depicting the SAS is shown in Fig. 2.19-e. Synthetic aperture is a methodology that enables high resolution through coherent processing of consecutive displaced returns. Instead of using a large static array of transducers, it uses the sensor's along-track displacement to create a large virtual array. The resulting resolution is on the order of the transducer dimensions and, more importantly, independent of the range between the sensor and the target. Since there is no need to have a small aperture the frequency used can be considerably lower which enables a longer range since lower frequencies propagate further in water. This is at the cost of more complex image processing and the requirement for a tightly prescribed speed. The micronavigation required to attain the tightly prescribed speed and trajectory, in the presence of seas, is an active area of research.

Applications of SAS to AUV navigation represent an area of active investigation. Both [203] and [212] describe a displaced phase center antenna micronavigation technique using SAS and that is good at estimating sway, but poor at estimating yaw. In [163], a constant time SLAM algorithm is developed for use with a SAS. The approach uses a submap approach to maintain scalability. Data association is done off-line and comparisons with ground truth provided by LBL show good results. An additional comment about SAS for navigation is that although it offers remarkable detection capabilities, the SAS payload can significantly change the hydrodynamics and controllability of a small AUV [163].

Ranging Sonar

A depiction of the multi-beam sensor swath is shown in Fig. 2.19-b. With multi-beam instead of just one transducer pointing down there are multiple beams from arrays of transducers arranged in a precise pattern on an AUV hull. The sound bounces off the seafloor at different angles and is received by the AUV at slightly different times. The signals are then processed on-board the AUV, converted into water depths, and arranged as a bathymetric map. The multi-beam resolution achieved depends on its transducer quality, operating frequency, and altitude from the seabed. Multi-beam bathymetry systems have been routinely used to map out large areas of seafloor. Each survey line that the AUV transits collects a corridor of data known as a swath. The multi-beam sonar yields 2.5D bathymetric features (elevation map) whereas the sidescan sonar produces 2D imagery. The former better facilitates feature based navigation as evidenced in the literature [21], [20].

Barkby et al. have proposed a bathymetric SLAM algorithm called BPSLAM that is based on a featureless FastSLAM implementation [21] [20] [19] [18]. In their approach, the need for feature extraction is removed; each particle maintains an estimate of the current vehicle state and the two-dimensional bathymetric map. An important issue with employing a particle filter based system with such a large state space is the computational expense of copying particles' maps during the resampling process. This problem is solved by "distributed particle mapping" where a particle ancestry tree is maintained. Copying of particle maps is avoided by having new particles generated during the resampling process point to their parents' maps rather than copying them. Maps of leaf nodes in the tree are reconstructed by recombining the maps of all ancestors. In [21], the need to store each particle's map is removed completely by storing just the particle's trajectory and linking poses to an entry in a log of bathymetry observations. Maps are then reconstructed as needed using Gaussian process regression, and, as a result, loop closures can be achieved even in the case where there is little or no overlap between sonar images since the regression process is able to make predictions about areas of seabed that have not been directly observed.

Similar to optical SLAM, the higher the quality of the navigation algorithm, the higher the quality of the data. A convenient way of representing 3D data is in octree form as in [71]. In this work, an active localization framework is presented where actions are selected to reduce the entropy of particles. The vehicle pose is estimated with particle filter SLAM.

Using phase only matched filtering for comparing subsequent images is presented in [40] and later [180]. It is shown that this method outperforms the standard iterative closest method to find six degree-of-freedom transformations between subsequent multi-beam scans. A similar approach using planar surface registration is presented in [169].

Chapter 3

Sensor-Driven Online Coverage Planning

3.1 Introduction

This chapter presents an online approach for AUV seabed surveying[1]. Many of the path planning methodologies outlined in Chapter 2.1 require prior knowledge of the presence of obstacles to generate paths. In an AUV seabed surveying application, the challenge is not due to obstacles but rather to the variable performance of the sonar sensor used for seabed coverage that is dependant on environmental factors that can be unknown and variable. An approach is presented that is inspired by sensor-driven [1] and information gain [41] approaches but uniquely tailored to the AUV seabed coverage problem. Adaptivity is achieved by maintaining a coverage map as the vehicle traverses the workspace and then planning paths based on the up-to-date coverage map.

Sensor objectives for the coverage task are particularly hard to define because of the uncertainty of sensor data so information gain is exploited as a merit criterion. However, it is shown that the information gain method alone is not sufficient to achieve global goals when there is incomplete prior knowledge about the environment. To compensate, the concept of branch entropy is proposed. Although the proposed research can be applied to diverse missions or sensors, it is particularly well-suited to AUV MCM missions where the seabed is scanned using a SSS.

In what follows, the term *workspace* will be used to refer to the geometric area of seabed to be covered and the term *environment* will be used to refer to describe the physical world including acoustic propagation conditions, seabed type, currents and other things of that type.

3.1.1 Problem Statement

For UAV systems, detection, identification, and classification of objects and targets can be done *in situ* by transmitting imagery back to a base station for analysis during the mission [70, 175]. There is an advantage to *in situ* processing of the sonar data as the mission can be adapted on-the-fly to address the detected targets. However, at present wireless

[1] The bulk of the work presented in this chapter is published in [171, 172, 170]. Additionally, application of the proposed algorithms to coverage with a fixed-wing unmanned aerial vehicle has been accepted for publication in [175].

transmission of sonar data through water is not feasible because of limited bandwidth and attenuation over relatively short ranges. As a result, data gathering must be completed before object classification and image processing can take place as shown in the AUV MCM pipeline Fig. 1.1. If the data gathered is insufficient for high probability detection, identification and classification, then the AUV will have to be redeployed to fill the gaps that were missed on the first pass. This prolongs the mission.

Historically, paths were planned beforehand (offline) for data collection regardless of unknown parameters that could affect the quality of the resulting sonar imagery. This involved either making overly conservative estimates resulting in suboptimal survey paths or by making repeated passes in the case that the parameters were not as expected. Additionally, sometimes planned paths were not exactly followed, for example due to unexpected disturbances such as currents or seas.

3.1.2 Objectives and Significance

Prior to this work, few if any research proposed online strategies to underwater seabed coverage. Usually AUVs are pre-programmed with waypoints that specify a structured path, such as a zig-zag or lawn mower [164]. In this case, performance will rely heavily on the accuracy of information about the workspace and the ability of the vehicle to exactly follow the prescribed plan. Often plans are made overly conservative in order to compensate for unexpected disturbances or incorrect environmental information.

The objective of the this chapter is to present an algorithm where paths can be planned online during mission execution based on the most up-to-date estimate of the parameters affecting sonar performance and the coverage already achieved over the workspace.

The result is that the first two blocks of Fig. 1.1 given in Chapter 1, “Survey Mission Planning” and “Data Collection,” should be combined into one block, “Adaptive Data Collection,” yielding the alternative pipeline shown in Fig. 3.1.

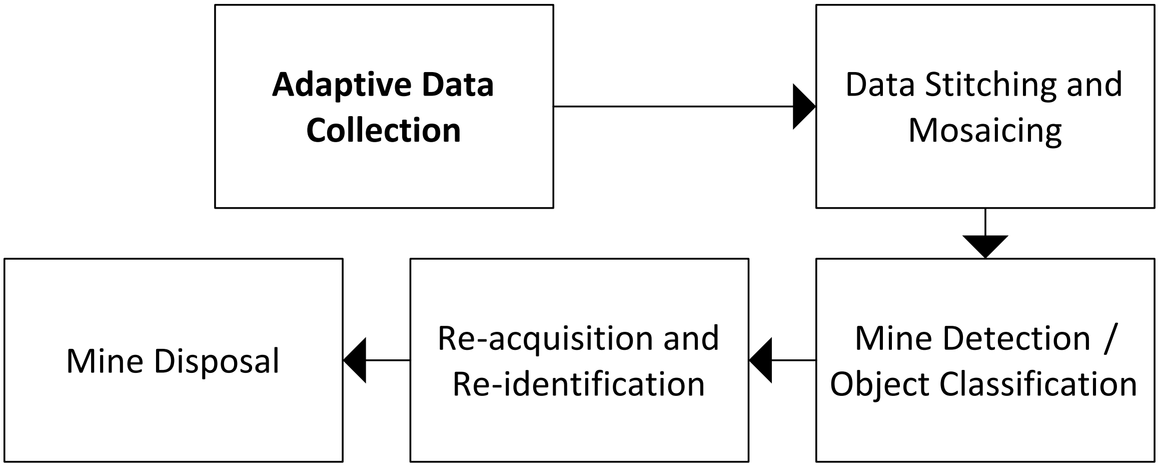

Adaptive Data Collection
Data Stitching and Mosaicing
Mine Detection / Object Classification
Re-acquisition and Re-identification
Mine Disposal

Figure 3.1: Adaptive AUV MCM task pipeline.

This online approach to planning has the advantages that it saves operators’ time as there is no need to predefine waypoints, it removes the need to perform an environmental assessment survey to determine the environment parameters, and the total time to complete the mission is reduced.

3.1.3 Overview of Contribution

In the approach taken here, path planning is achieved through reconciling behaviours that represent the multiple objectives defined for efficient mission completion as the vehicle navigates through the workspace. The proposed approach has the advantages that:

1. The total path lengths and times required to cover a workspace are shorter in many cases compared to traditional structured methods.

2. There is no need for pre-programmed waypoints.

3. It is adaptive to any changes in environmental factors that can be detected *in situ* such as seabed type and water properties.

4. It is able to generate paths for complex and non-convex environment shapes such as would typically be found in harbours closer to the shore.

5. Preference is given to viewing seabed from different insonification angles, which is beneficial for target recognition [78].

In this chapter it is assumed that vehicle pose is exactly known, an assumption that is common in coverage literature [49] but rarely true in reality, particularly for AUV systems. In the following chapter, this assumption is removed.

Due to the adaptive nature of the approach, the paths produced can contain more turns than traditional approaches, although this is avoided where possible.

The performance of the approach is evaluated via simulation and implementation on the Iver2 AUV developed by OceanServer Inc.

3.1.4 Chapter Nomenclature

A table of nomenclature used in the rest of this chapter is given in Table 3.1.

Variable	Description
t	Time index
i	Grid cell index
$\mathbf{c}^i$	Grid cell i at location $[x^i, y^i]^T$ in global coordinate frame
${}^s\mathbf{c}^i_t$	$= [{}^sx^i_t, {}^sy^i_t]^T$, the location of cell $\mathbf{c}^i$ in the sensor coordinate frame
${}^s_g\mathbf{T}_t$	Transformation matrix from global to sensor coordinates
N	Total number of grid cells
$\mathcal{W}$	The workspace or area to be covered
M^i	Binary RV representing the existence or absence of a mine in cell $\mathbf{c}^i$
$\tilde{M}^i_t$	Binary RV represented the detection of a mine in cell $\mathbf{c}^i$ at time t
T^i_t	Binary RV representing confidence (correct classification) at cell $\mathbf{c}^i$ based on all data up to and including t
$\breve{T}^i_t$	Binary RV representing confidence (correct classification) from only data at time t
$\boldsymbol{\mathcal{E}}^i_t$	RV representing set of all parameters that affect mine detection performance

$\mathbf{X_t}$	$= [X_t, Y_t, \Psi_t]$ Vehicle pose at time t (represented as RVs here for consistency with following chapter, although assumed deterministic here: $p(\mathbf{X_t} = \mathbf{x_t}) = \mathbf{1}$)
$P_{\boldsymbol{\varepsilon}_t^i}(y)$	The sensor characteristic curve generated by the ESPRESSO model with the parameters specified by $\boldsymbol{\varepsilon}_t^i$ that defines the confidence as a function of the lateral range from the AUV track y
$\mathcal{S}_t$	The sensor swath: the set of all cells that are covered to some extent by the sonar at time t
α	The acute angle between two "looks" of a cell
$R(\psi)$	The collective objective function
$J(\psi)$	The maintain heading objective function
$B(\psi)$	The information gain objective function
w_B, w_G, w_J	Weights used in the collective objective function
$\tau_\psi(s)$	A straight line path parameterized by s used to evaluate the information gain expected from travelling at a specified heading ψ
Q_{free}	The free configuration space
$G(\psi)$	The branch entropy objective function
a^j	Hexagon cell with index j
N_j	The total number of hexagon cells
$\bar{H}^j$	The average entropy of the underlying grid cells in cell a^j
g_k	Branch entropy value for branch number k

Table 3.1: Chapter 3 nomenclature.

3.1.5 Chapter Outline

The remainder of the chapter is organized as follows: Section 3.2 details how the coverage map is maintained as the AUV traverses the workspace. Section 3.3 describes the proposed online planning solution, including the information gain and branch entropy objective functions. Section 3.4 describes the experimental setup and the simulation framework. Section 3.5 shows simulation and experimental results, while a more in-depth discussion is performed in Section 3.6. Section 3.7 provides a brief summary.

3.2 Maintaining the Coverage Map

The approach presented here differs from most coverage planning methods in that paths for coverage are generated based on a coverage map that is actively maintained as the vehicle traverses the workspace based on the achieved vehicle pose. This allows paths to be generated online and can result in path planning methods that are adaptive to sensor data gathered *in situ*.

The process of building the coverage map comprises two main steps:

1. Estimate the coverage over the workspace resulting from a sonar measurement at time t

2. Combine the estimate from time t with estimates from times prior to t

3.2.1 Problem Formulation

The workspace to be covered, $\mathcal{W}$, is maintained as a grid of N small cells $\mathbf{c}^i = [x^i, y^i]^T$. The size of each cell is sufficiently small that the coverage over the cell can be treated as uniform. For example, the size of the grid cells could be chosen to be equal to the resolution of the sonar sensor.

Define the RV $M^i \in \{mine, \overline{mine}\}$ to represent the presence of a mine in cell $\mathbf{c}^i$, $i = 1..N$ of the discretized workspace. Then, define the RV $\tilde{M}_t^i \in \{mine, \overline{mine}\}$ to represent the mine detection event in $\mathbf{c}^i$ at time t. There are four possible scenarios for each cell:

1. $\tilde{M}_t^i = mine$ and $M^i = mine$: target detected
2. $\tilde{M}_t^i = mine$ and $M^i = \overline{mine}$: false alarm
3. $\tilde{M}_t^i = \overline{mine}$ and $M^i = mine$: missed target
4. $\tilde{M}_t^i = \overline{mine}$ and $M^i = \overline{mine}$: no target and nothing detected

In the common case that object detection, identification, and classification is not being done on-board the AUV, we have no access to either M^i, whether a mine exists at location $\mathbf{c}^i$, or $\tilde{M}_t^i$, whether a mine will be detected in cell $\mathbf{c}^i$. However, the ESPRESSO model [54] described in Section 2.3 was developed to address exactly this problem. In the absence of knowledge of whether there is or is not a mine at location we can express the coverage of cell $\mathbf{c}^i$ using the concept of confidence, which describes the probability that we correctly assess the status (mine or no mine) at cell $\mathbf{c}^i$.

Define a binary RV $\breve{T}_t^i \in \{0, 1\}$ that represents whether cell $\mathbf{c}^i$ will be correctly classified as either containing a mine, or not, at time t:

$$\begin{aligned}
p(\breve{T}_t^i = 1) &= p(\text{target detected}) + p(\text{no target and nothing detected}) \\
&= p(\tilde{M}_t^i = mine, M^i = mine) + p(\tilde{M}_t^i = \overline{mine}, M^i = \overline{mine}) \\
p(\breve{T}_t^i = 0) &= p(\text{false alarm}) + p(\text{missed target}) \\
&= p(\tilde{M}_t^i = mine, M^i = \overline{mine}) + p(\tilde{M}_t^i = \overline{mine}, M^i = mine)
\end{aligned} \tag{3.1}$$

For each $\breve{T}_t^i$, $p(\breve{T}_t^i = 1) \in [0.5, 1]$ represents the confidence, or probability of correct classification of cell $\mathbf{c}^i$ as either containing a target or not, and $p(\breve{T}_t^i = 0) = 1 - p(T_t^i = 1)$ is the probability of incorrect classification, based on the data at time t.

Additionally, for each cell, $\mathbf{c}^i$, associate a binary RV, T_t^i, where $p(T_t^i = 1)$ represents the confidence at cell $\mathbf{c}^i$ by combining all the "looks" of cell c^i from the start of the survey until time t. This collection of RVs is referred to as the confidence map.

At the start of the mission the confidence values are all initialized to 0.5: $p(T_{t_0}^i = 1) = 0.5$, $\forall i = 1..N$, and the objective of the mission is to maximize these confidence values.

3.2.2 Estimating Coverage Based on Current Pose

As the AUV navigates around the workspace the confidence map is updated based on the current AUV pose and the environmental parameters affecting sonar performance described in Table 2.1.

The target detection event can be represented by the Bayesian network shown in Fig. 3.2, where $\mathbf{X}_t \triangleq \{X_t, Y_t, \Psi_t\}$ is the 2D vehicle pose at time t. Any variable with the superscript i varies across the workspace and any variable with the subscript t varies with time.

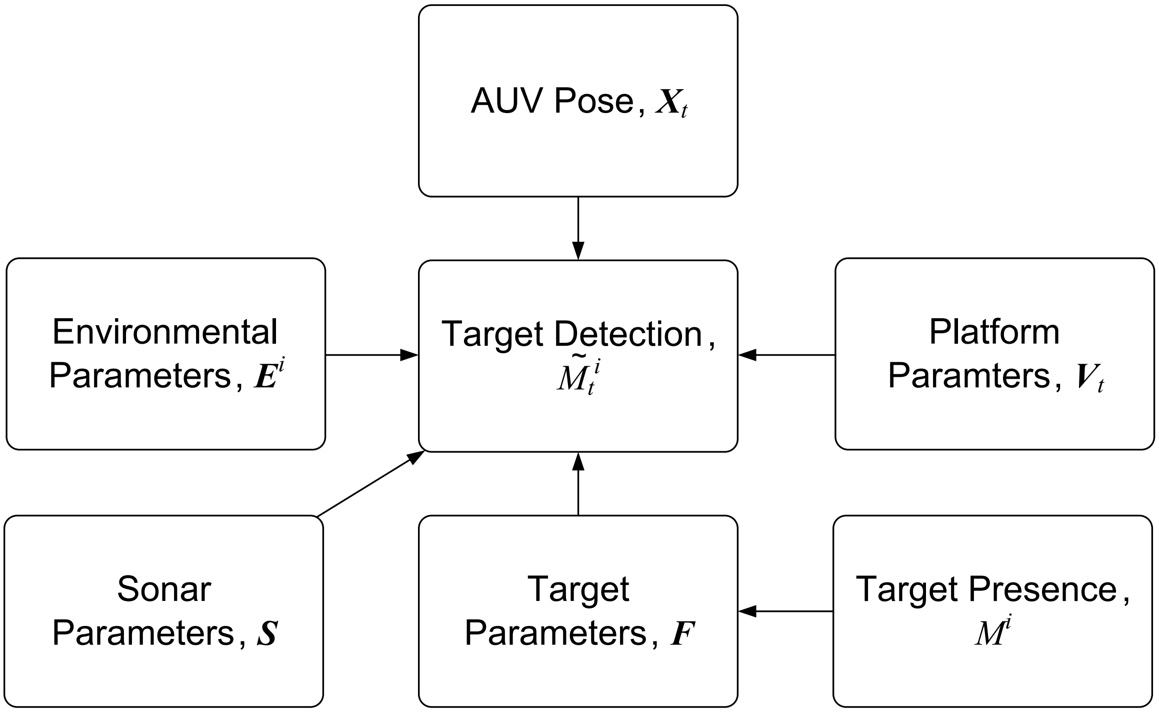

AUV Pose, X_t
Environmental Parameters, E^i
Target Detection, $\tilde{M}_t^i$
Platform Paramters, V_t
Sonar Parameters, S
Target Parameters, F
Target Presence, M^i

Figure 3.2: Bayesian network representing target detection similar to [41]. Arrows represent conditional probabilities.

We can express the joint probability of target detection based on a single observation and combination of sensor, environment, and AUV pose as:

$$\begin{aligned}&p(\tilde{M}_t^i = \tilde{m}_t^i, \mathbf{E}^i = \mathbf{e}^i, \mathbf{S} = \mathbf{s}, \mathbf{F} = \mathbf{f}, \mathbf{V}_t = \mathbf{v}_t, M^i = m^i, \mathbf{X}_t = \mathbf{x}_t) = \\ &p(\tilde{M}_t^i = \tilde{m}_t^i | \mathbf{E}^i = \mathbf{e}^i, \mathbf{S} = \mathbf{s}, \mathbf{F} = \mathbf{f}, \mathbf{V}_t = \mathbf{v}_t, \mathbf{X}_t = \mathbf{x}_t) \times \\ &p(\mathbf{F} = \mathbf{f} | M^i = m^i) p(\mathbf{E}^i = \mathbf{e}^i) p(\mathbf{S} = \mathbf{s}) p(\mathbf{V}_t = \mathbf{v}_t) p(M^i = m^i) p(\mathbf{X}_t = \mathbf{x}_t).\end{aligned} \tag{3.2}$$

However, since we are only interested in estimating the confidence over the workspace and not the actual presence of mines, (3.1) can be used to rewrite the right hand side of (3.2) as:

$$\begin{aligned}&p(\breve{T}_t^i = 1 | \mathbf{E}^i = \mathbf{e}^i, \mathbf{S} = \mathbf{s}, \mathbf{F} = \mathbf{f}, \mathbf{V}_t = \mathbf{v}_t, \mathbf{X}_t = \mathbf{x}_t) \times \\ &p(\mathbf{E}^i = \mathbf{e}^i) p(\mathbf{S} = \mathbf{s}) p(\mathbf{F} = \mathbf{f}) p(\mathbf{V}_t = \mathbf{v}_t) p(\mathbf{X}_t = \mathbf{x}_t).\end{aligned} \tag{3.3}$$

Define $\boldsymbol{\mathcal{E}}_t^i = \{\mathbf{E}^i, \mathbf{S}, \mathbf{F}, \mathbf{V}_t\}$ as the set of all RVs from Table 2.1 at location $\mathbf{c}^i$ and time t and (3.3) can be further reduced to:

$$p(\breve{T}_t^i = 1 | \boldsymbol{\mathcal{E}}_t^i = \varepsilon_t^i, \mathbf{X}_t = \mathbf{x}_t) p(\boldsymbol{\mathcal{E}}_t^i = \varepsilon_t^i) p(\mathbf{X}_t = \mathbf{x}_t) \tag{3.4}$$

Assumption 3.2.1. *As is common in coverage literature, assume that the AUV pose is exactly known ($p(\mathbf{X}_t = \mathbf{x}_t) = 1$). Extension to the case where the AUV configuration is uncertain is the subject of the next chapter.*

Based on Assumption 3.2.1, we can further simplify (3.4) to:

$$p(\breve{T}_t^i = 1|\boldsymbol{\mathcal{E}}_t^i = \boldsymbol{\varepsilon}_t^i, \mathbf{X}_t = \mathbf{x}_t)p(\boldsymbol{\mathcal{E}}_t^i = \boldsymbol{\varepsilon}_t^i). \tag{3.5}$$

Denote the sonar lateral range characteristic generated by the ESPRESSO model with parameters $\boldsymbol{\varepsilon}_t^i$ as $P_{\boldsymbol{\varepsilon}_t^i}(y)$ where y is the lateral range. $p(\breve{T}_t^i = 1|\boldsymbol{\mathcal{E}}_t^i = \boldsymbol{\varepsilon}_t^i, \mathbf{X}_t = \mathbf{x}_t)$ is read directly from the lateral range curve:

$$p(\breve{T}_t^i = 1|\boldsymbol{\mathcal{E}}_t^i = \boldsymbol{\varepsilon}_t^i, \mathbf{X}_t = \mathbf{x}_t) = \begin{cases} P_{\boldsymbol{\varepsilon}_t^i}({}^s y_t^i) & \text{if } \mathbf{c}^i \in \mathcal{S}_t \\ 0.5 & \text{otherwise} \end{cases} \tag{3.6}$$

where the sensor swath, $\mathcal{S}_t$ is the set of cells that are covered to some extent by the sonar sensor measurement taken at time t and ${}^s y_t^i$ is the orthogonal distance from cell $\mathbf{c}^i$ to the AUV track at time t and is computed by transforming the location of the cell in the global coordinate frame into the sensor coordinate frame through the transformation matrix ${}^s_g\mathbf{T}_t$:

$${}^s\mathbf{c}_t^i = [{}^s x_t^i, {}^s y_t^i]^T = {}^s_g\mathbf{T}_t[x^i, y^i]^T. \tag{3.7}$$

We do not require perfect knowledge of the environmental parameters encapsulated by $\boldsymbol{\mathcal{E}}_t^i$. The final confidence resulting from a single sonar sensor reading at time t for cell $\mathbf{c}^i$ can be obtained by marginalizing out the environmental parameters from (3.5):

$$\begin{aligned} p(\breve{T}_t^i = 1) &= \sum_{\boldsymbol{\varepsilon}_t^i} p(\breve{T}_t^i = 1, \boldsymbol{\mathcal{E}}_t^i = \boldsymbol{\varepsilon}_t^i, \mathbf{X}_t = \mathbf{x}_t) \\ &= \sum_{\boldsymbol{\varepsilon}_t^i} p(\breve{T}_t^i = 1|\boldsymbol{\mathcal{E}}_t^i = \boldsymbol{\varepsilon}_t^i, \mathbf{X}_t = \mathbf{x}_t)p(\boldsymbol{\mathcal{E}}_t^i = \boldsymbol{\varepsilon}_t^i) \\ &= E_{\boldsymbol{\mathcal{E}}_t^i}[p(\breve{T}_t^i = 1|\boldsymbol{\mathcal{E}}_t^i = \boldsymbol{\varepsilon}_t^i, \mathbf{X}_t = \mathbf{x}_t)] \\ &= \begin{cases} E_{\boldsymbol{\mathcal{E}}_t^i}[P_{\boldsymbol{\varepsilon}_t^i}({}^s y_t^i)] & \text{if cell } c^i \in \mathcal{S}_t \\ 0.5 & \text{otherwise} \end{cases} \end{aligned} \tag{3.8}$$

where $p(\boldsymbol{\mathcal{E}}_t^i = \boldsymbol{\varepsilon}_t^i)$ is the probability that we are using the correct sensor characteristic curve to evaluate the confidence of cell $\mathbf{c}^i$, and $E_{\boldsymbol{\mathcal{E}}_t^i}$ represents the expectation over the RV $\boldsymbol{\mathcal{E}}_t^i$.

Example 3.2.1. *Consider the case where all parameters are assumed known and correspond to the conditions that generate the lateral range curves in Fig. 2.14 except the seabed type which is assumed to have equal probability of being either "sand", "cobble", or "clay" ($p(\boldsymbol{\mathcal{E}}_t^i = cobble) = p(\boldsymbol{\mathcal{E}}_t^i = sand) = p(\boldsymbol{\mathcal{E}}_t^i = clay) = 1/3$) for the cell $\mathbf{c}^i$ at location $(x^i, y^i) = (4, 30)$. The AUV is currently at location $(x_t, y_t) = (4, 0)$ pointing due north ($\psi_t = 0$). Denote the three lateral range curves in Fig. 2.14 as $P_{clay}(y)$, $P_{cobble}(y)$ and $P_{sand}(y)$. The lateral range of cell $\mathbf{c}^i$ is ${}^s y_t^i = 30$. The confidence after following this single track should be evaluated* *using* (3.8)*:*

$$\begin{aligned} p(\breve{T}_t^i = 1) &= E_{\boldsymbol{\mathcal{E}}_t^i}[p(\breve{T}_t^i = 1|\boldsymbol{\mathcal{E}}_t^i = \boldsymbol{\varepsilon}_t^i, \mathbf{X}_t = \mathbf{x}_t)] \\ &= \sum_{\boldsymbol{\varepsilon}_t^i \in \{clay, sand, cobble\}} p(\breve{T}_t^i = 1|\boldsymbol{\mathcal{E}}_t^i = \boldsymbol{\varepsilon}_t^i, \mathbf{X}_t = \mathbf{x}_t) p(\boldsymbol{\mathcal{E}}_t^i = \boldsymbol{\varepsilon}_t^i) \\ &= P_{clay}(30)(1/3) + P_{cobble}(30)(1/3) + P_{sand}(30)(1/3) \\ &= 0.9985(1/3) + 0.5331(1/3) + 0.6340(1/3) = 0.7219. \end{aligned} \tag{3.9}$$

If there is no knowledge of environmental conditions beforehand, the distribution of $\boldsymbol{\mathcal{E}}_t^i$ can be initialized as uniform across all possible environmental parameter values. As the AUV traverses the workspace, some unknown parameters such as seabed type can be measured *in situ* using sensors as described in Table 2.1.

To obtain a final confidence value at cell $\mathbf{c}^i$ from all data up to and including time t, $p(\breve{T}_t^i = 1)$, must be combined with previous measurements of the same location to obtain the total confidence $p(T_t^i = 1)$ that if a mine exists in cell $\mathbf{c}^i$ it will be detected.

3.2.3 Combining Measurements From Different Looks

Previous work on AUV MCM has either assumed that subsequent observations of the same location should be considered as either statistically dependent [228] or statistically independent [183] regardless of insonification angle. However, there is a rich body of literature to support the fact that viewing a target from different angles is beneficial for target detection and identification [78, 38]. This is largely due to the fact that targets are identified in SSS imagery not by looking at the bright returns caused by the object but by the shadow that is cast behind the object which provides more information about the shape and size of the object.

The combination of subsequent looks can be represented by the Bayes' network shown in Fig. 3.3. A visual depiction of an AUV viewing a cylindrical target twice is shown in Fig. 3.4. Define the acute angle between the two angles of incidence to be α. For the work here, we propose that the angle of viewing should be accounted for when combining confidences of subsequent "looks" because the conditional dependence or independence of these subsequent looks is a function of the amount of overlap in the sonar shadows. The two shadows in Fig. 3.4 should be treated as independent since there is no overlap in the information that they are providing about the shape of the target.

Proposition 3.2.1. *Subsequent looks of the same cell should be combined by taking the maximum confidence in the worst case that looks are assumed conditionally dependent and with De Morgan's law in the best case that looks are assumed conditionally independent:*

$$\begin{aligned} p(T_t^i = 1) \geq& \max\{p(\breve{T}_{t_1}^i = 1), p(\breve{T}_{t_2}^i = 1), ..., p(\breve{T}_{t_K}^i = 1)\} \\ p(T_t^i = 1) \leq& 1 - \prod_{k=1}^{K}(1 - p(\breve{T}_{t_k}^i = 1)) \end{aligned} \tag{3.10}$$

Proof. See appendix B. □

It is argued here the level of correlation between two looks represented by $\breve{T}_{t_1}^i = 1)$ and $p(\breve{T}_{t_2}^i = 1)$ is directly related to the overlap in the shadows produced in the sonar imagery.

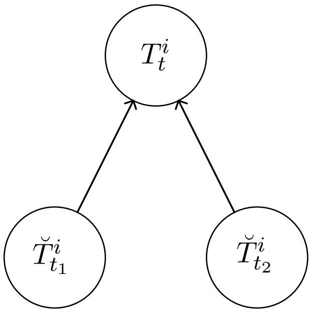


Figure 3.3: Bayes' network describing the generation of T_t^i from two previous "looks" or "views" of cell $\mathbf{c}^i$ at time t_1 and t_2 represented by $\breve{T}_{t_1}^i$ and $\breve{T}_{t_2}^i$ respectively.

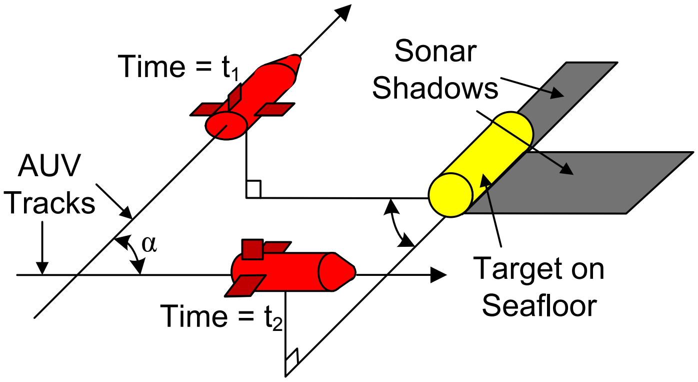


Figure 3.4: Different views of targets will produce different shadows. It is desirable for target recognition to view targets at different angles to get more information about the shape of the target.

Further analysis of the correlation between looks as a function of the incidence angle would be an interesting avenue for future work, but is outside of the scope here.

In the case that the two observations are parallel then $\alpha = 0$ and the two shadows are exactly overlapping and the max function should be used:

$$p(T_t^i = 1) = \max\{p(\breve{T}_{t_1}^i = 1), p(\breve{T}_{t_2}^i = 1)\} = p(\breve{T}_{t_1}^i = 1). \tag{3.11}$$

where we assume without loss of generality that the look at time t_1 yields a higher confidence: $p(\breve{T}_{t_1}^i = 1) \geq p(\breve{T}_{t_2}^i = 1)$.

In the case that the two observations are perpendicular then $\alpha = \pi/2$ and the two shadows will have no overlap (as shown in Fig. 3.4) and the two looks are independent and should be combined using:

$$p(T_t^i = 1) = 1 - [(1 - p(\breve{T}_{t_1}^i = 1))(1 - p(\breve{T}_{t_2}^i = 1)) \tag{3.12}$$

If the angle $0 < \alpha < \pi/2$ then the resulting confidence, $p(T_t^i = 1)$, should be somewhere in between based on the amount of overlap in the sonar shadows which is a function of the angle α. Estimating the exact overlap as a function of the angle α would be difficult since it is dependent on the target geometry and its orientation relative to the insonification angle. As an estimation, a linear relation between (3.11) and (3.12) is used:

$$p(T_t^i = 1) \approx \frac{2\alpha p(\breve{T}_{t_2}^i = 1)}{\pi}(1 - p(\breve{T}_{t_1}^i = 1)) + p(\breve{T}_{t_1}^i = 1), \tag{3.13}$$

While this is an approximation and detailed verification of this relation could be supported by future work, it is important that the benefit of viewing a target from different insonification angles will result in a higher probability of correct classification (confidence). This will be reflected in the coverage map and additionally in the the information gain objective function which will preferentially choose to view areas of seabed at different angles because they will result in more information gain.

3.3 Path Planning Approach Based on Multi - Objective Optimization

As the AUV moves about the workspace, the confidence map is updated based on the achieved vehicle path and our estimates about the parameters that affect sonar performance over the workspace using the process described in Section 3.2. To satisfy the main objective of this work, which is to develop online planning strategies for coverage, we define a multi-objective function that uses the up-to-date coverage map to evaluate the benefit of different actions.

This is formulated as an optimization that produces a desired heading ψ_d by maximizing an objective function that is evaluated over the domain of all possible headings: $\psi = [0, 2\pi]$:

$$\psi_d = \underset{\psi}{\operatorname{argmax}}\, R(\psi) = w_B B(\psi) + w_G G(\psi) + w_J J(\psi), \tag{3.14}$$

where $R(\psi)$ is the total utility, $B(\psi)$ is the information gain, $G(\psi)$ is the branch entropy, $J(\psi)$ is the benefit of maintaining the current heading, and w_B, w_G, and w_J are the respective weights. Tuning of the weights is an important consideration. In the present implementation, trial and error has been used to tune the weights, however, it would be simple to optimize them with some meta heuristic method such as genetic algorithms or particle swarm optimization [139]. All functions will be explicitly defined, but, in general, the function $B(\psi)$ prioritizes headings that cover the most area in the short term, the function $G(\psi)$ prioritizes over headings that will help the robot complete its coverage mission in the longer term, and the function $J(\psi)$ prioritizes over closest headings to avoid unnecessary turns.

The functions $J(\psi)$, $B(\psi)$ and $G(\psi)$ will be described in detail in Sections 3.3.1, 3.3.2 and 3.3.3 respectively.

The optimization takes place over the heading reference only and it is assumed that desired speed and depth are generated by some other method. In this case speed and altitude reference are held constant and tracked by inner loop controllers. The reference depth can be calculated from the reference altitude using known bathymetry or data from on-board sensors and and is usually chosen based on the desired resolution of the data to be gathered.

The evaluation of the multi-objective function is done using interval programming (IvP) in the mission-oriented operating suite (MOOS) framework [28] described in Appendix A and collectively referred to as MOOS-IvP. Each term in the objective function is defined as a behaviour which generates a piecewise linear objective function at each iteration of the outer-loop controller. Accuracy of the underlying objective functions can be traded off against computation time by specifying the number of pieces in the piecewise linear approximation. As a result, the domain is discretized. However, the discretization does not need to be consistent over all objective functions and also need not be uniform.

Each objective function is scaled such that the maximum utility is 100. As a result, the units of the individual functions can be disregarded.

An AUV is shown in a workspace in Fig. 3.5. This snapshot in time will be used as an example for each of the objective functions below.

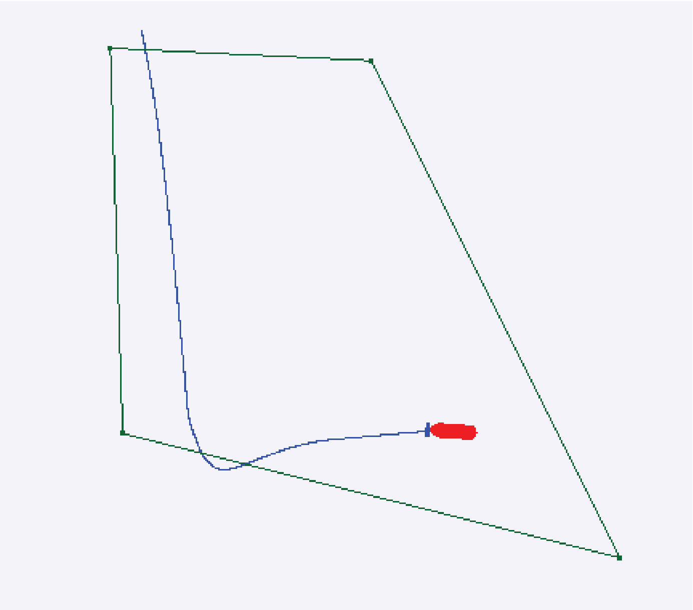


Figure 3.5: A simulated path.

3.3.1 The Maintain Heading Objective Function

The function $J(\psi)$ is the simplest objective function and is designed to have peak utility at the present heading and then rapidly decay based on the allowable yaw rate for valid sonar data mosaicking. A plot of $J(\psi)$ for the snapshot shown in Fig. 3.5 is shown in Fig. 3.6.

3.3.2 The Information Gain Objective Function

Information theory is used to quantify utility over the short term to define the function $B(\psi)$ from (3.14).

It is proposed here that Shannon entropy reduction is an appropriate and novel way of adaptively achieving coverage.

Recall from (2.8) that the Shannon entropy of a binary RV T_t^i can be evaluated as:

$$H(T_t^i) = -p(T_t^i = 1)\log(p(T_t^i = 1)) - (1 - p(T_t^i = 1))\log(1 - p(T_t^i = 1)), \tag{3.15}$$

and from (3.15) and Fig. 2.11:

$$\lim_{p(T_t^i=1)\to 1} H(T_t^i) = 0, \tag{3.16}$$

which implies that maximizing the confidence over the workspace minimizes the entropy of T_t^i for all i. As a result, an information gain objective function formulated in terms of gaining information about T_t^i will result in convergence towards complete coverage.

The expected entropy reduction (EER):

$$\Delta H(T_t^i|\boldsymbol{\mathcal{E}}_t^i, \mathbf{X}_t) \triangleq H(T_t^i) - E_{\boldsymbol{\mathcal{E}}_t^i}[H(T_t^i|\boldsymbol{\mathcal{E}}_t^i, \mathbf{X}_t)], \tag{3.17}$$

defines a scalar quantity that represents the *a priori* expected amount of information that will be gained about T_t^i by making the sonar measurement from pose $\mathbf{X}_t$ at time t with

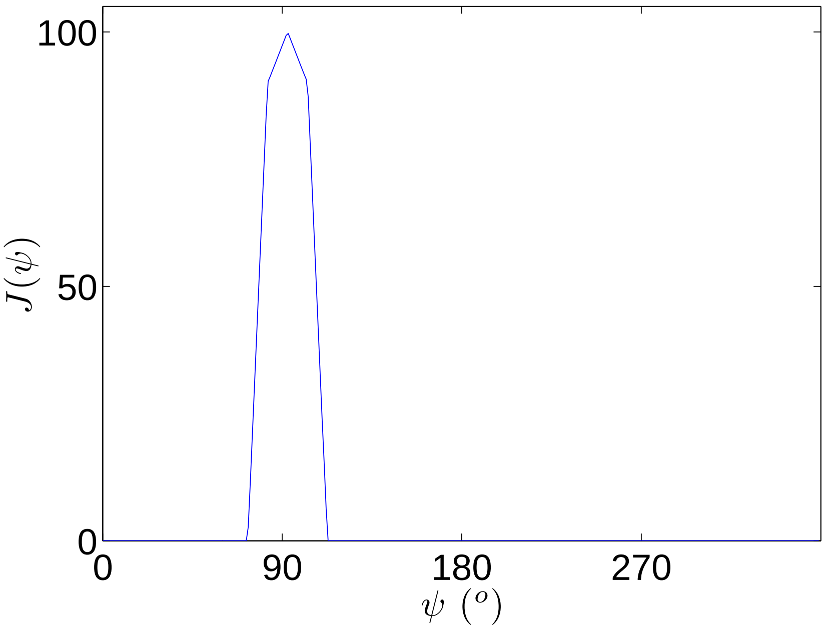


Figure 3.6: The "maintain heading" objective function, $J(\psi)$, where the current vehicle heading is $\psi_t = 92^o$.

(possibly uncertain) environmental parameters $\boldsymbol{\mathcal{E}}_t^i$.

The second term on the right-hand side of (3.17), $E_{\boldsymbol{\mathcal{E}}_t^i}[H(T_t^i|\boldsymbol{\mathcal{E}}_t^i, \mathbf{X}_t)]$, is evaluated by taking an expectation of the conditional Shannon entropy over $\boldsymbol{\mathcal{E}}_t^i$:

$$\begin{aligned} &E_{\boldsymbol{\mathcal{E}}_t^i}[H(T_t^i|\boldsymbol{\mathcal{E}}_t^i, \mathbf{X}_t)] \\ &= \sum_{\boldsymbol{\varepsilon}_t^i} p(\boldsymbol{\mathcal{E}}_t^i = \boldsymbol{\varepsilon}_t^i)[-p(T_t^i = 1|\boldsymbol{\mathcal{E}}_t^i = \boldsymbol{\varepsilon}_t^i, \mathbf{X}_t = \mathbf{x}_t) \log p(T_t^i = 1|\boldsymbol{\mathcal{E}}_t^i = \boldsymbol{\varepsilon}_t^i, \mathbf{X}_t = \mathbf{x}_t) \\ &- (1 - p(T_t^i = 1|\boldsymbol{\mathcal{E}}_t^i = \boldsymbol{\varepsilon}_t^i, \mathbf{X}_t = \mathbf{x}_t)) \log(1 - p(T_t^i = 1|\boldsymbol{\mathcal{E}}_t^i = \boldsymbol{\varepsilon}_t^i, \mathbf{X}_t = \mathbf{x}_t))] \end{aligned} \tag{3.18}$$

where $p(T_t^i = 1|\boldsymbol{\mathcal{E}}_t^i = \boldsymbol{\varepsilon}_t^i, \mathbf{X}_t = \mathbf{x}_t)$ is the confidence at cell $\mathbf{c}^i$ after the sonar measurement at time t which is generated by using (3.8) to first generate $\breve{T}_t^i$ which is then combined with previous measurements at cell $\mathbf{c}^i$ using the process in Section 3.2.3. $p(\boldsymbol{\mathcal{E}}_t^i = \boldsymbol{\varepsilon}_t^i)$ is the probability that $\boldsymbol{\varepsilon}_t^i$ are the correct parameters.

This definition specifies a way of combining the potential benefits of sensor observations additively. We can compute the total expected entropy reduction at time t as the sum of the EERs for every cell in the sensor swath:

$$\Delta H(\mathbf{T}_t|\boldsymbol{\mathcal{E}}_t^{1:N}, \mathbf{X}_t) = \sum_{i:\mathbf{c}^i \in \mathcal{S}_t} \Delta H(T_t^i|\boldsymbol{\mathcal{E}}_t^i, \mathbf{X}_t) \tag{3.19}$$

where $\mathbf{T}_t \triangleq [T_t^1, T_t^2, ..., T_t^N]$.

To evaluate the information gain resulting from travelling at different headings, straight line paths are generated that begin at the current AUV position, (x_t, y_t), and traverse a fixed distance r at the heading to be evaluated, ψ.

The potential paths to be evaluated are represented by the variable τ_ψ and parameterized by s. As s moves from 0 to 1, the two endpoints of the line, $\tau_\psi(s)$ moves along the straight line path in the free configuration space Q_{free}:

$$\begin{aligned} \tau_\psi : [0,1] \rightarrow Q_{free}, s \rightarrow \tau_\psi(s) \\ \tau_\psi(s) = (x_t + \frac{r}{s}\cos(\psi), y_t + \frac{r}{s}\sin(\psi), \psi) \end{aligned} \tag{3.20}$$

where the set of paths to be evaluated are defined by varying the value of ψ from 0 to 2π. To evaluate the information gain of traveling at heading ψ, we integrate the EERs at each pose along the line:

$$B(\psi) \triangleq \int_0^1 \Delta H(\mathbf{T}_t|\boldsymbol{\mathcal{E}}_t^{1:N}, \tau_\psi(s))ds \tag{3.21}$$

It is important to stress that the confidence map is not updated during this searching process. The confidence map is only updated based on the achieved AUV poses as the AUV is in motion in the workspace.

The information gain objective function for the snapshot shown in Fig. 3.5 is shown in Fig. 3.7. Note that the highest utility for the information gain objective function in this case is approximately 92^o, the direction that is being traveled, and the lowest utility is the reverse direction, 268^o, because almost no new information would be gained from moving over the path that was just traveled. The local minima at angles 47^o, 137^o, 227^o, and 317^o represent other paths that produce sensor swaths that overlap with the previously achieved coverage.

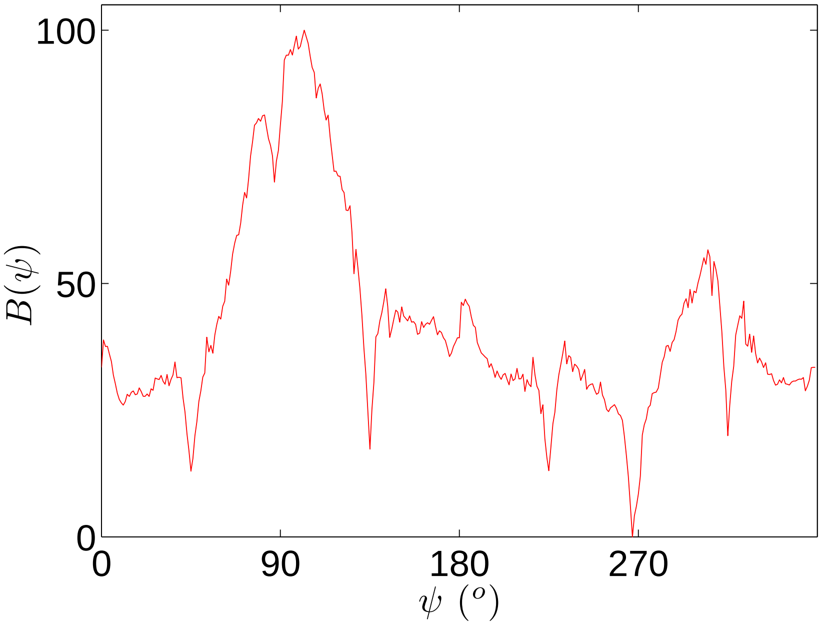


Figure 3.7: The "information gain" objective function, $B(\psi)$.

3.3.3 The Branch Entropy Objective Function

In this section, the $G(\psi)$ term of the objective function (3.14) will be motivated and derived.

3.3.3.1 Motivation

The information gain method was shown to be effective for solving the path planning problem when *a priori* knowledge of the environment, obstacles, and targets are available [41]. However, the approach taken here removes this requirement and is larger in scope than the short term goals. In the sensor-driven approach, the information gain $B(\psi)$ is useful for evaluating the benefits of each of the potential next moves, but when complete coverage is the goal, this approach reduces to a greedy-first search.

It is necessary to include a parameter in the objective function that helps the AUV achieve its global goal - complete coverage. This term is referred to as the branch entropy and can be thought of as akin to a potential function for coverage. The benefits of including the branch entropy in the objective function are:

- It helps the AUV complete the coverage of sections before it leaves them.
- It allows the AUV to find the areas of the workspace that have not been covered and are out of range of the information gain behaviour.
- It acts as a tie-breaker so the AUV does not oscillate between headings of equal information gain. Instead, this behaviour operates over the entire workspace so that coverage will converge towards complete.

3.3.3.2 Overview of Approach

A block diagram showing an overview of the proposed approach for generating the branch entropy objective function is shown in Fig. 3.8.

At initialization, the workspace is decomposed into equal sized hexagon cells where each hexagon cell contains many grid cells. This cell decomposition is used to generate a directed acyclic graph (DAG) (directed graph with no cycles)with the cell that the AUV currently occupies as the root. The average entropy values of the grid cells in the hexagon cells are used as inputs to a formula where each neighbour of the cell currently occupied by the AUV is given a value representing the benefit of heading towards that particular cell. The value is determined by how much entropy there is down that branch of the DAG (explained further in the following section), with priority given to high entropy areas that are nearby so that the AUV will finish covering sections of the workspace before it leaves them. The result is that, by simply applying a formula on the decomposition, and without performing an exhaustive search, the AUV can determine what areas of the map are left to be explored. Finally, the branch entropies of each of the neighboring cells are combined into one function over all headings that represents the branch entropy objective function.

After the AUV moves, the average entropy of the hexagon cells must be recomputed to incorporate the new information. If the AUV has moved from one cell to a neighbouring cell, then the DAG must also be updated to reflect the fact that there is a new root.

3.3.3.3 Exact Hexagon Decomposition

Cell decomposition, described in Chapter 2.1.5, is an effective way to abstract the path planning problem into a graph searching operation [48]. Normally, the cells are either

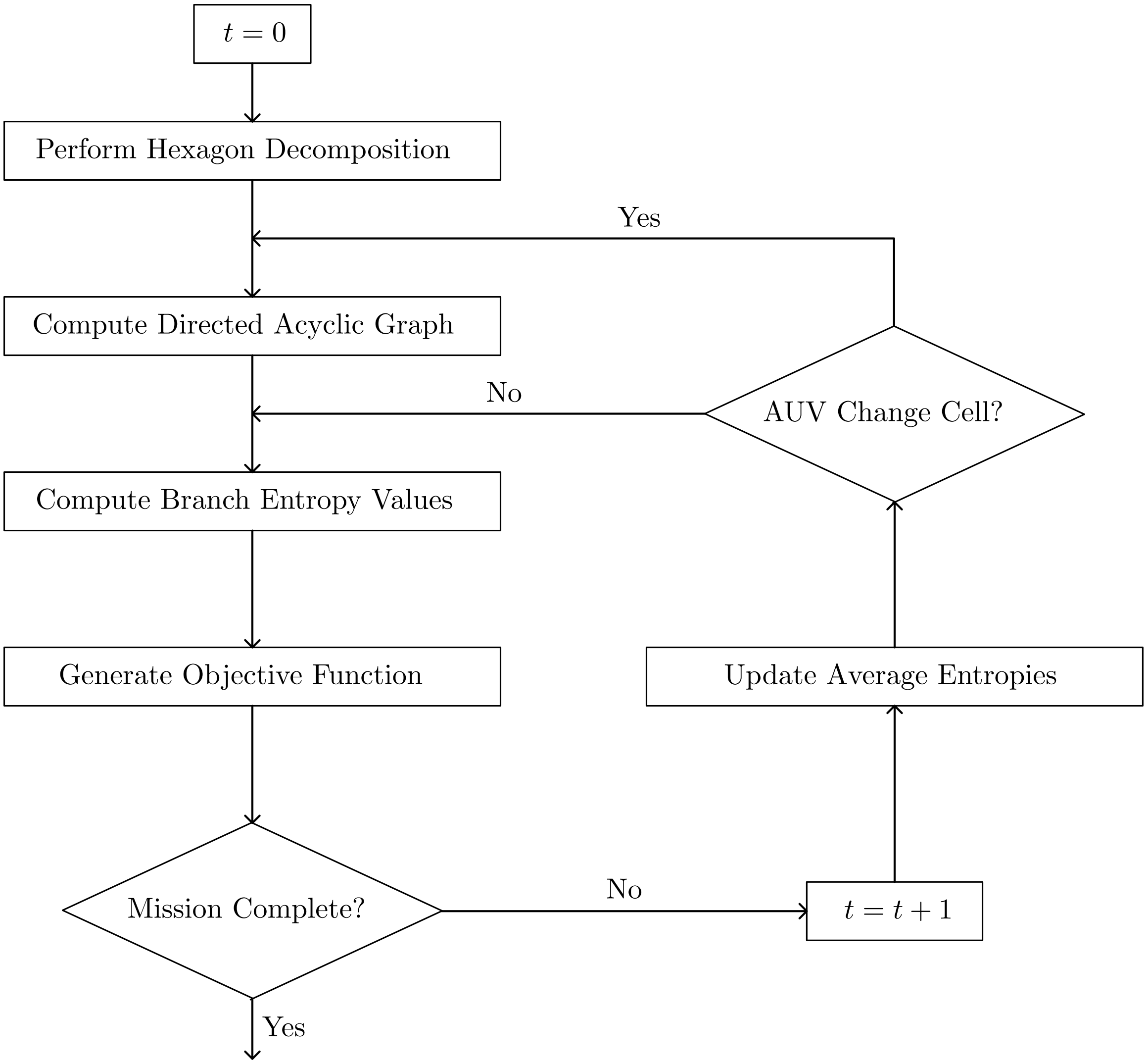

$t = 0$
Perform Hexagon Decomposition
Yes
Compute Directed Acyclic Graph
No
AUV Change Cell?
Compute Branch Entropy Values
Generate Objective Function
Update Average Entropies
Mission Complete?
No
$t = t + 1$
Yes

Figure 3.8: Flow diagram depicting the generation of the branch entropy objective function.

exactly or approximately decomposed into rectangles, although other polygonal shapes have been proposed [165]. In this case we select a hexagonal decomposition. A main benefit of the hexagon decomposition is that the distance from the center of any cell to the center of any adjacent cell is a constant.

A hexagon decomposition is performed such that the union of all cells, $a^j, j = 1..N_j$, where N_j is the total number of hexagon cells, covers the entire workspace, $\mathcal{W}$:

$$\mathcal{W} \subseteq \bigcup_{j=1}^{N_j} a^j. \tag{3.22}$$

Associated with each hexagon cell is an average entropy, $\bar{H}^j$, which represents a measure of the average uncertainty over the area of the workspace that falls within that cell:

$$\bar{H}^j = \frac{1}{\eta} \sum_{i:\mathbf{c}^i \in \{a^j \bigcap \mathcal{W}\}} H(T_t^i), \tag{3.23}$$

where η is the number of grid cells in hexagon cell a^j.

This averaging operation on the entropy of the grid cells is a valid operation since summing of the entropies is equivalent to the entropy of the joint distribution in the case that RVs are independent:

$$\begin{aligned} H(X_1, X_2) &= E[-\log p(x_1, x_2)] \\ &= E[-\log p(x_1)p(x_2)] \\ &= E[-\log p(x_1)] + E[-\log p(x_2)] \\ &= H(X_1) + H(X_2) \end{aligned} \tag{3.24}$$

So this average entropy can be interpreted as a normalized joint entropy over all grid cells in the hexagon.

Each cell is also assigned a level, l, which is the minimum number of cells that must be traversed to reach the presently occupied cell a^p, which is the root of the tree, and a list of children, which are all neighbours in level $l + 1$.

A hexagon decomposition of a workspace is shown in Fig. 3.9. The workspace is the pink polygon area underneath the hexagons. The hexagon on the right shows the numbering convention for the neighbours. Each cell that is neighboring the cell that the the AUV currently occupies represents a branch. The branch number follows the same convention as the numbering for neighbours. For example, the cells in branch 0 in Fig. 3.9 are indicated by the bold green outline.

3.3.3.4 The Directed Acyclic Graph

The hexagon decomposition is subsequently converted into a directed acyclic graph (DAG). Every cell a^j appears only once in the graph, and is at level l. There can be several paths from a^p to a^j but they must all be the same minimum length. The hexagon decomposition geometry is exploited in that center of every hexagon cell at level l is the same Euclidean distance from the center of the current hexagon cell, a^p.

Each neighbour of a^p becomes a child in the DAG. The neighbours of those nodes become children provided they are not already in the DAG at a higher level. This process continues

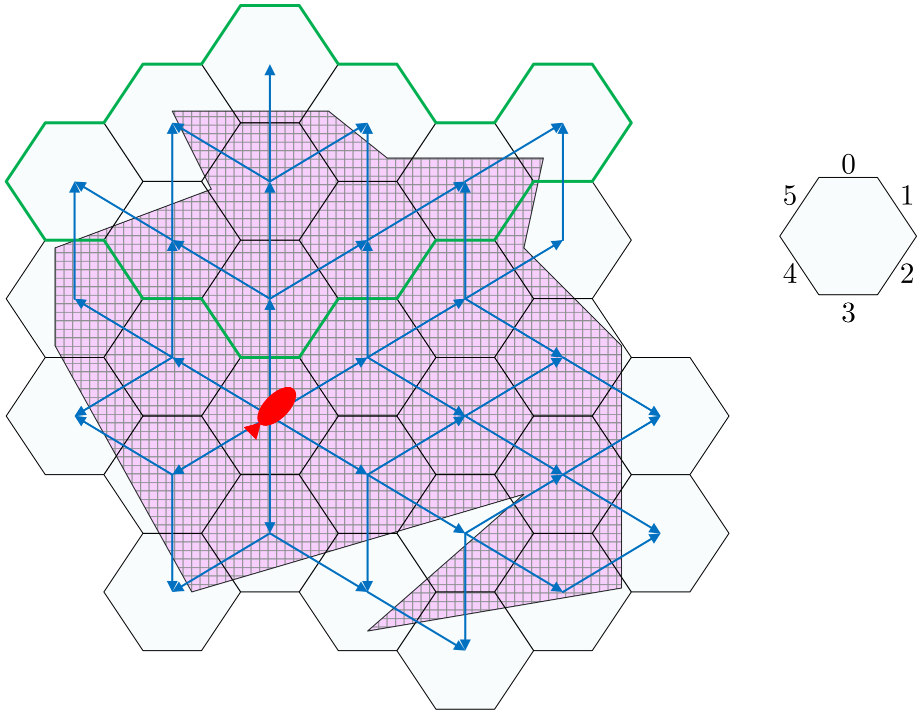


Figure 3.9: A workspace with a hexagon cell decomposition. The workspace to be covered is shown in pink and is also decomposed into smaller grid cells. The hexagon cells that will be in branch 0 have been outlined in green.

until all hexagon cells in the workspace are included in the DAG.

Algorithm 7 details the process of building the DAG. The inputs are a^p, the current cell, and $\{a^j | j = 1..N_j\}$, the set of all other cells.

3.3.3.5 The Branch Entropy Equation

The branch entropy is used to evaluate how much entropy there is down each branch of the DAG to select a path that takes the AUV towards an unsurveyed area of $\mathcal{W}$. Also, priority will be given to paths that have more unsurveyed area nearer to the current position so that the AUV does not leave an area before it is completely surveyed.

There will be a value of branch entropy for each neighbour of the current cell a^p as each neighbour has its own branch in the DAG. In order for the branch entropy to provide the benefits desired, cells that are at higher levels in the graph must be given more weight. For each neighbour, $k = 0..5$, of a^p, the branch entropy, g_k, for a DAG with a total of L levels is defined as:

$$g_k \triangleq \frac{\displaystyle\sum_{l=2}^{L} (L - l + 1) \frac{\displaystyle\sum_{j \in \mathcal{C}_{l,k}} \bar{H}^j}{m_{l,k}}}{\displaystyle\sum_{l=1}^{L-1} l}. \tag{3.25}$$

where $m_{l,k}$ is the number of nodes in level l of branch k and $\mathcal{C}_{l,k}$ is a list containing the indices of the hexagons in level l of branch k. In (3.25), the closer cells are weighted higher

Algorithm 7 Build_DAG($\{a^j\}_{j=1..N_j}$,a^p)

```
$DoneList \leftarrow a^p$
$level \leftarrow 1$
while $DoneList \neq \{a^j\}_{j=1..N_j}$ do
  $level \leftarrow level + 1$
  for $n \leftarrow$ Each node in $level - 1$ do
    $CurrentList \leftarrow \emptyset$
    $n.children \leftarrow \emptyset$
    for $k \leftarrow$ All neighbours of $n$ do
      if $k \notin DoneList$ then
        $n.children \leftarrow n.children \cup k$
        $k.value \leftarrow C^k.entropy$
        if $k \notin CurrentList$ then
          $CurrentList \leftarrow CurrentList \cup k$
        end if
      end if
    end for
  end for
  $DoneList \leftarrow DoneList \cup CurrentList$
end while
```

using an inverse linear function. Other weighting functions, such as exponential decay could have been used, and the impact of this choice could be explored further in future work. The denominator is used for normalization.

Example 3.3.1. *Fig. 3.10 shows the transformation from hexagon cells to DAG. The cell labeled a^p is the cell that the AUV is currently in, and the values in all of the other cells represent their average entropies.*

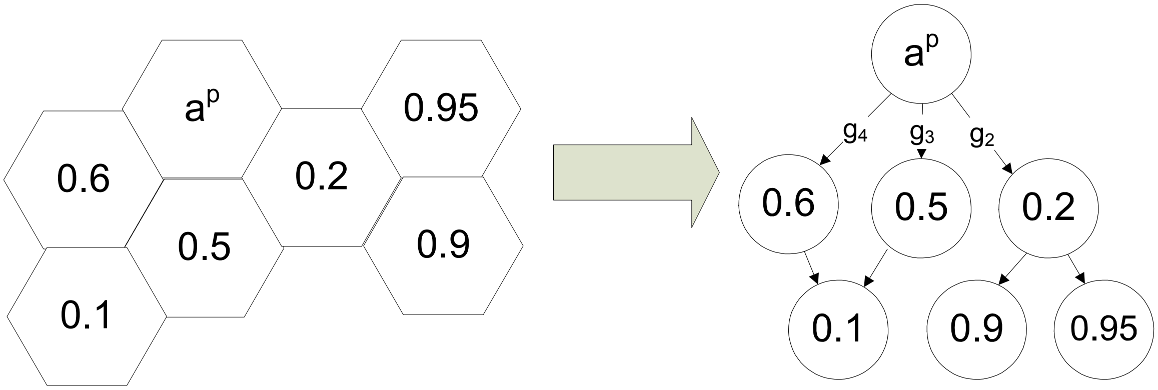


Figure 3.10: A transformation from hexagon decomposition to DAG. (numbers in cells/nodes represent average cell entropy)

The cell a^p has three neighbours: 2, 3, and 4. Consequently there will be three branch entropy values: g_2, g_3, and g_4 ($g_0 = g_1 = g_5 = 0$).

For branch 2, $L = 2, m_{1,2} = 1, m_{2,2} = 2$ and the average entropy in level one is 0.2 and *in level two are 0.9 and 0.95:*

$$g_2 = 1/3((2)(0.2) + (1)(1/2)(0.95 + 0.90)) = 0.442,$$

for branch 3, $L = 2, m_{1,3} = 1, m_{2,3} = 1$ *and the average entropy in level one is 0.5 and in level two is 0.1:*

$$g_3 = 1/3((2)(0.5) + (1)(0.1)) = 0.367,$$

and for branch 4: $L = 2, m_{1,4} = 1, m_{2,4} = 1$ *and the average entropy in level one is 0.6 and in level two is 0.1:*

$$g_4 = 1/3((2)(0.6) + (1)(0.1)) = 0.433.$$

In this case g_2 *is the highest, which reflects that fact that there is much more uncertainty down the branch 2.*

3.3.3.6 Building the Branch Entropy Objective Function

Once the values of branch entropy are computed, they are combined into one function over all headings, ψ, that is used in the multi-objective optimization.

The values of branch entropy are treated as samples of the underlying objective function and are connected by straight lines. The six headings, ψ of known utility are $\frac{\pi}{3}k, k = 0..5$, which corresponds to the headings that pass through the midpoints of the neighbouring hexagon faces. The corresponding points used to generate the objective function for $G(\psi)$ are $(\pi k/3, g_k), k = 0..5$. A general equation for the objective function, $G(\psi)$ is derived that parameterizes each of the connecting lines:

$$G(\psi) = \frac{3}{\pi}(g_k - g_{k+1})\psi + g_k(1 - k) + g_{k+1}. \tag{3.26}$$

where

$$k = \lfloor \frac{3\psi}{\pi} \rfloor. \tag{3.27}$$

Note that for consistency define $g_6 = g_0$.

The branch entropy objective function for the snapshot from Fig. 3.5 is shown in Fig. 3.11. The branch entropy behaviour is maximum at 0 and π as these headings point to the areas of the map that have unfinished areas.

It is important to stress that the branch entropy objective function is used in conjunction with the information gain objective function where the branch entropy function uses the hexagon decomposition over the entire space to achieve the global coverage objective in the long term and the information gain is used to maximize coverage in the short term.

3.3.4 The Full Multi-Objective Function

According to (3.14), the final utility, $R(\psi)$, is the weighted sum of the objective functions. In Fig. 3.12 the objective functions at a snapshot are shown together with the collective with $w_B = 1.0$, $w_G = 1.0$, and $w_J = 0.8$. In this case, the collective objective function selects the heading at 94^o to be the best desired heading.

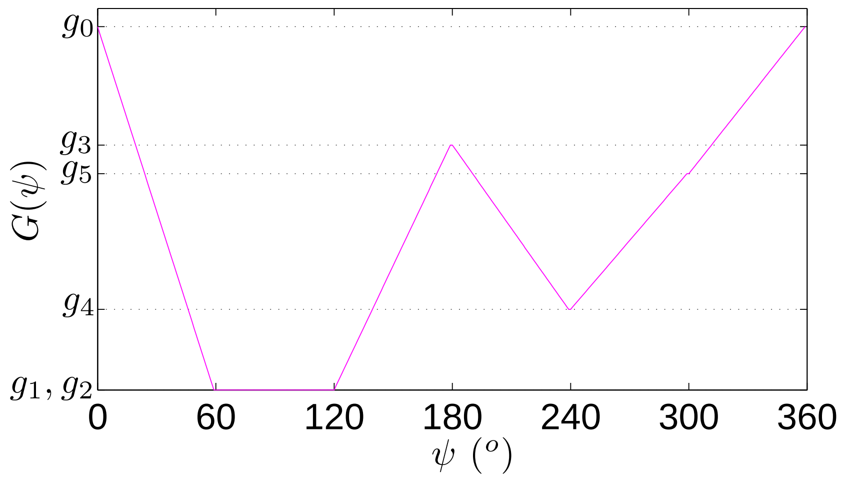


Figure 3.11: The "branch entropy" objective function for the snapshot shown in Fig. 3.5. The outputs of the branch entropy equation (3.25) are connected together in a piecewise linear fashion to define generate the objective function.

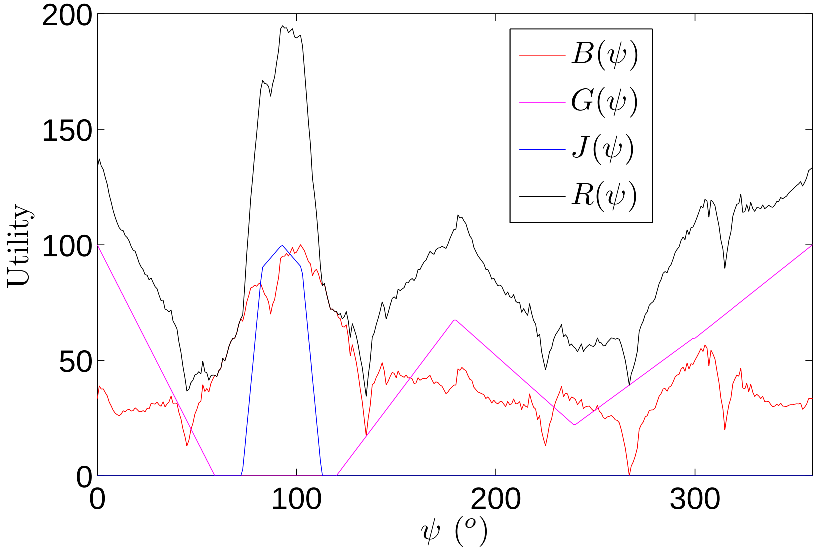


Figure 3.12: The information gain, branch entropy, maintain heading, and collective objective functions corresponding to the path shown in Fig. 3.5.

3.4 Experimental Setup

In order to test the proposed algorithms, missions were performed with MOOS both in simulation and in water. For a background of MOOS-IvP refer to Appendix A.

A block diagram of the MOOS communities is shown in Fig. 3.13[2], where the blocks labeled "Coverage Map", "Information Gain", "Branch Entropy", and "Maintain Heading" were custom built applications that were not included in the open-source distribution of MOOS-IvP. The Iver2 AUV on which hardware tests were done is set up in a frontseat/backseat configuration with two Intel ATOM 1.6GHz CPUs which communicate through a serial connection [61]. All the sensors and actuators are interfaced through the frontseat computer. When underway, the backseat computer can take control either by specifying actuator values directly, or by specifying outer loop references which are tracked by an inner loop controller on the frontseat. The backseat has access to the raw sensor data and also a filtered state estimate from the frontseat. In the work in this chapter, the filtered state estimate is used directly from the frontseat. In the next chapter the raw sensor data is used on the backseat to perform navigation.

The simulation is setup to mimic the actual hardware so the backseat MOOS community can remain untouched when the transition is made to hardware. In simulation, an additional frontseat community is used for sensing, actuation, navigation, inner loop control, and vehicle kinematics.

The backseat MOOS community contains three essential components:

1. The frontseat/backseat interface which is responsible for communicating with the frontseat (specifically *iOceanServerComms* [61]).

2. The custom "Coverage Map" application which subscribes to AUV state data and produces a coverage map based on the method described in Section 3.2.

3. The IvP Helm [28] which reconciles behaviours into an outer loop heading reference at each iteration. In this case each term in the objective function (3.14) is represented by a behaviour shown in Fig. 3.13 as "Information Gain", "Branch Entropy", and "Maintain Heading".

A description of all simulated components and actual hardware is also given in Table. 3.2. Also on-board the AUV but not simulated was a Neil Brown CT sensor to gather water conductivity and temperature information.

3.5 Results

3.5.1 Simulation

The system is tested using the simulation setup. The first simulation done is on a simple square workspace where environmental parameters are assumed known and fixed (assumed to be cobble seabed at 10m depth to produce the curve in Fig. 2.14). The resulting path is shown in Fig. 3.14.

The planner produces a spiral-type path that efficiently covers the entire area based on the known sonar performance. Without variable or unknown environmental parameters

[2] There are other MOOS applications running that are doing logging, mission monitoring, and other tasks that are part of the open source distribution.

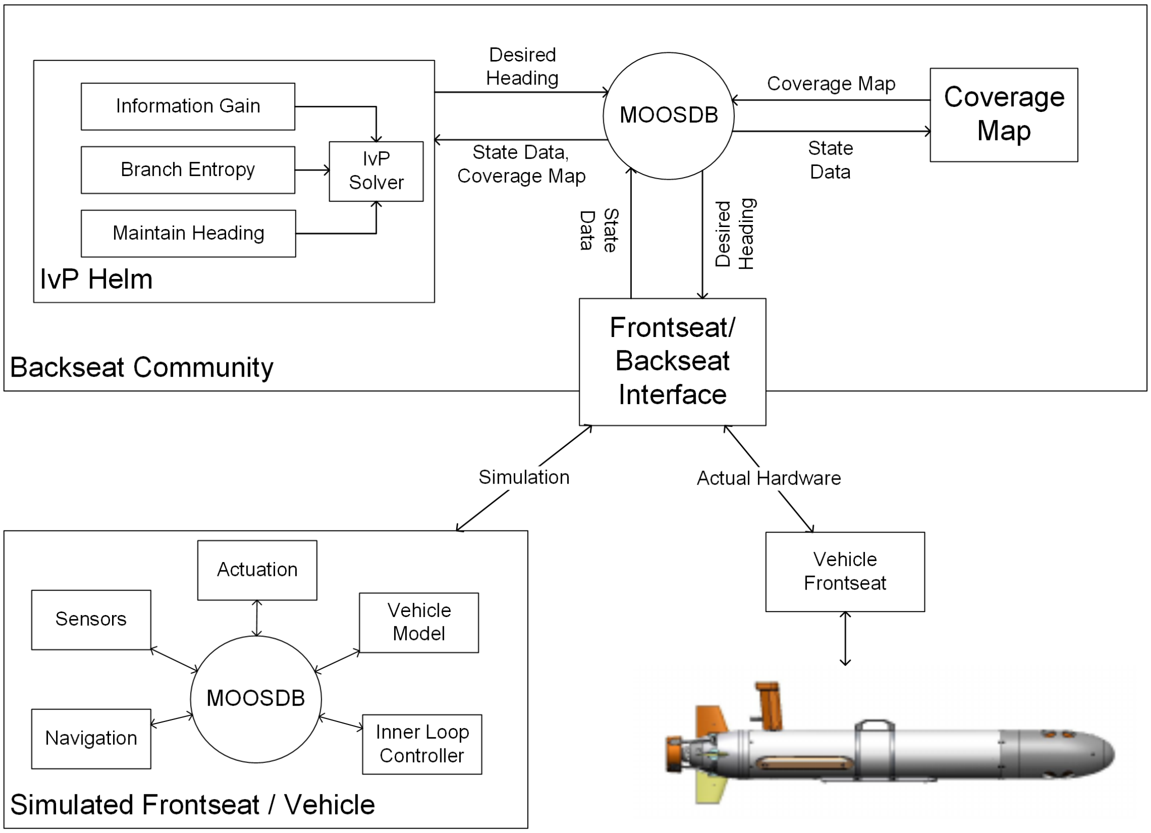


Figure 3.13: Simulation and actual hardware setup with MOOS. State data is passed from frontseat to backseat.

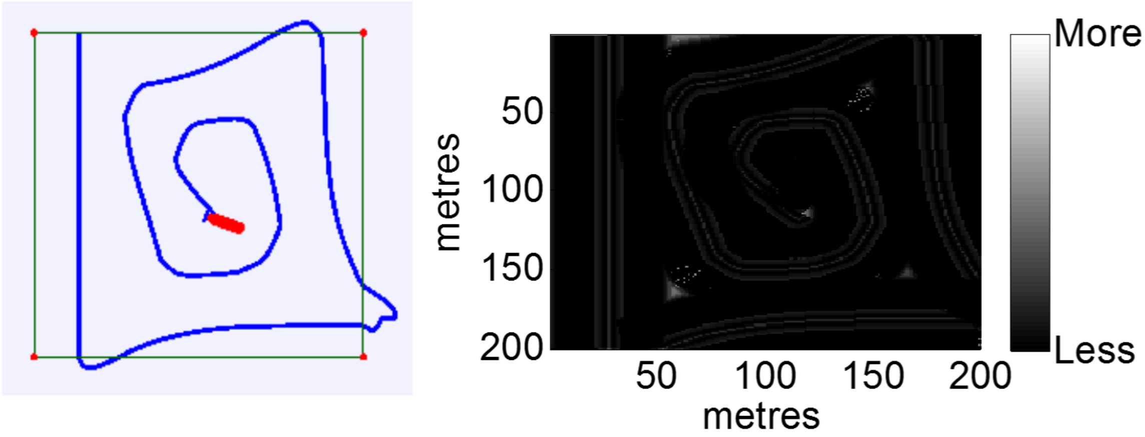


Figure 3.14: **Left:** The path planned by the proposed planner for a rectangular workspace with constant and *a priori* known parameters. **Right:** The resulting confidence map. Darker areas indicate higher confidence.

Component	**Simulation**	**Actual Hardware**
GPS Sensor	iGPS [162]	UBlox LEA-5H
Compass Sensor	iCompass [162]	OS5000-USG
Depth Sensor	iDepth [162]	MSI Depth sensor
DVL	iDVL [162]	SonTek/YSI DVL
Sensor Drivers	"iSensor" applications [162]	Iver2 frontseat
Inner Loop Control	pMarinePID [162]	Iver2 frontseat
Actuator Drivers	included in iActuation [162]	Iver2 frontseat
Propellor	iActuation [162]	130 Watt, 4000 RPM Brushless DC motor
Frontseat Computer	Dell Dual Core 3GHz	Intel 1.6 GHz ATOM processor
Frontseat - Backseat Communication	pMOOSBridge [162]	iOceanServerComms [61]
Backseat Computer	Dell Dual Core 3GHz	Intel 1.6 GHz ATOM processor
Outer Loop Control	IvP Helm [26]	IvP Helm [26]
Navigation	pEKF	Iver2 frontseat
High Level Planning	Behaviours	Behaviours

Table 3.2: Description of components used for the simulation and implemented on the IVER2 for the in-water trials.

and using a simple rectilinear workspace, the planner converges to a structured path. This limiting case is a good test of the algorithm.

Since in general the system is stochastic, a Monte Carlo style simulation is conducted to compare the performance for a developed random track algorithm, the information gain behaviour alone (IG), the branch entropy alone (BE), and information gain with branch entropy (IG/BE) by repeating the simulation 36 times with random initial conditions. The results are tested against the *a priori* scripted lawn mower pattern for a simple workspace. Results for three different levels of desired confidence are shown in Table 3.3 where the data from the Monte Carlo simulation is presented with a mean and a covariance (values in square brackets). In the generation of the results in Table 3.3, it is assumed that environmental and target parameters for the ESPRESSO model are unknown beforehand. As a result, the lawn mower tracks are based on the most pessimistic assumption for the unknown parameters. The lawn mower path length required to obtain 98% coverage is significantly higher because the tracks must be placed closely enough that the areas that are missed directly underneath the tracks are covered by subsequent tracks. For lawn mower paths this is referred to as swath overlap and has a large effect on the length of the resultant path. It is clear from the results that the information gain approach alone is inefficient. The mean path lengths are considerably longer. Also note that the variances are also much larger, particularly for the 95% confidence case. This is characteristic of a greedy approach because sometimes it may get 'lucky' and find a good path quickly, but when it is 'unlucky' it has a very difficult time completing the mission and the path length becomes long. Additionally, the branch entropy approach alone is not as effective as the combination of information gain with branch entropy. It is most efficient to use the combination of information gain for achieving coverage in the short term and branch entropy to guide the AUV to uncovered areas of the workspace.

		Desired Coverage		
		.90	.95	.98
Search Method	Lawn Mower	1275	1545	2355
	Random	1279 [446]	1915 [460]	2299 [677]
	IG	1488 [362]	2429 [817]	3307 [730]
	BE	1260 [146]	1738 [232]	2634 [292]
	IG/BE	1088 [105]	1458 [150]	1761 [160]

Table 3.3: Performance of lawn mower, random walk, information gain, branch entropy, and information gain with branch entropy algorithms for different desired levels of coverage. Values in the table are in metres and are mean results from 36 simulation runs with variance in brackets.

It can also be challenging to design lawn mower paths when the workspace is oddly shaped. For example consider the plots in Fig. 3.15 where the workspace is shown as the dark green outline. In this case environmental parameters are considered fixed and known (cobble seabed at depth of 10m). For the case of low desired confidence thresholds, the lawn mower planner performs well. However, in the case that high confidence is desired, which is common, the lawn mower tracks must be designed so that the channel left by one track is covered by the next. Indeed this has been noted in the past as a shortcoming of the lawn mower method [164]. The IG/BE planner proposed produces a path that occasionally crosses itself, but results in less total sensor swath overlap allowing it to achieve high coverage faster than the lawn mower pattern. As the workspace shape becomes more irregular in geometry, the benefit of the proposed planner will increase.

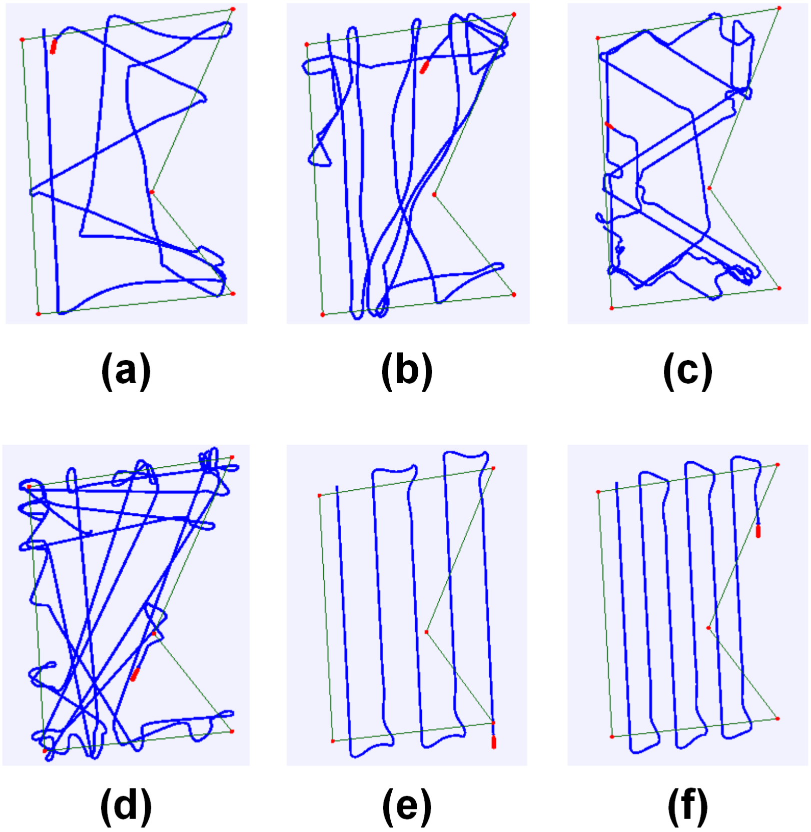


Figure 3.15: A comparison of sample paths planned by the various approachs. **(a)** Proposed IG/BE planner for final confidence greater than 98%. **(b)** IG alone for final confidence greater than 98%. **(c)** BE alone for final confidence greater than 98%. **(d)** Path planned by random track planner for final confidence greater than 98%. **(e)** Deterministic lawn mower path for final confidence of 95%. **(f)** Deterministic lawn mower path for final confidence of 99%. In order for the lawn mower path to obtain coverage greater than 95%, the valleys under the AUV track must be covered resulting in high sensor swath overlap between subsequent tracks in the lawn mower survey.

In the case that the environmental parameters are known but vary over the workspace, the simplest way to construct the lawn mower path is to place tracks closely enough that coverage will be obtained even in the worst case over the workspace. In Fig. 3.16 the parameters are assumed to be known beforehand where the seabed type varies between cobble, sand, and clay. The sensor characteristic curves for the three areas of the worksapce are shown in Fig. 2.14. In order to ensure coverage, the lawn mower tracks must be placed closely enough to guarantee coverage in the case that the seabed is cobble, which is the worst case. The proposed IG/BE planner maintains the confidence map as the AUV traverses the workspace and is therefore better able to exploit the better sonar performance obtained in the case that the seabed type is clay.

The algorithm computation scales constantly with time and the size of the workspace after an initialization, meaning that the amount of computation required to generate the

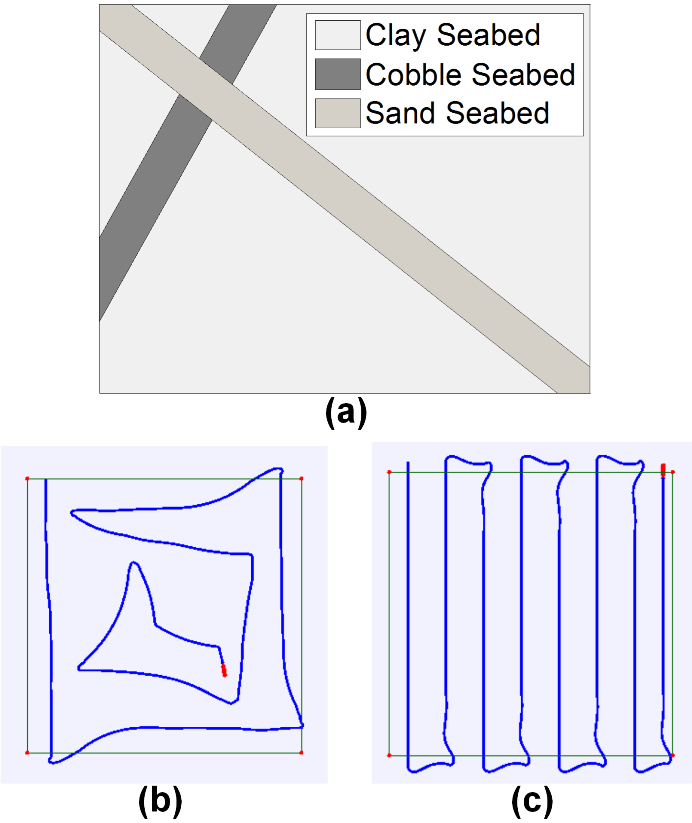


Figure 3.16: **(a)** A 300m by 300m square workspace with variable parameters, in this case three different types of seabed, which are assumed to be known *a priori*. **(b)** The path planned by the proposed IG/BE planner. Path length to achieve 97% confidence is 1085m. The AUV automatically devotes more time to the areas of seabed with poorer sensor performance **(c)** Deterministic path for a lawn mower pattern. Path length to achieve 97% confidence is 1185m. Note that if a higher coverage threshold was desired then the tracks would have to be significantly closer as described in Fig. 3.15.

information gain and branch entropy objective functions is independent of time and the size of the workspace. A path planned on a larger workspace is shown in Fig. 3.17. The total length of the path planned to cover the $118750m^2$ environment is $3794m$. The corresponding path length for the random track approach was 7937.39m (path not shown). Therefore, in this case there is a factor of two reduction in the proposed path length and likely time to perform the mission.

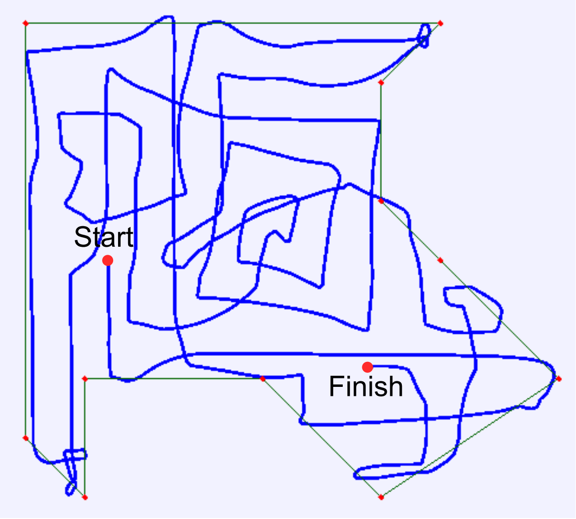


Figure 3.17: Path planned for coverage of a large environment up to 99.5% confidence. Area of environment is $118750m^2$. Length of path planned is 3794m.

3.5.2 In-Water Trials

Tests were performed on OceanServer's Iver2 AUV in the Bedford Basin in Nova Scotia, Canada.

The AUV was able to successfully cover three environments within the limited operating region. A plot of a sample path taken in a simple convex workspace and the corresponding final confidence map are shown in Fig. 3.18. The runs were stopped when confidence values reached 95%. A comparison lawn mower mission was also performed where the overlap is sufficient to guarantee the 95% coverage; see Fig. 3.19.

When comparing the two paths from Fig. 3.18 and Fig. 3.19, it is interesting to note that, although the desired tracks for the lawn mower pattern are straight lines, the actual path oscillates about these desired paths. This is largely due to the inability of the frontseat controller to track the desired path in the presence of currents. It should be noted that the currents on the day when this trial was conducted (Sept. 1 2011) were extremely small, on the order of 0.2 knots at most. The proposed planner has, as a component of the multi-objective function, an objective to explicitly maintain headings rather than track paths. The result is that the oscillation in heading can be suppressed to some extent.

A non-convex environment test was also conducted with results shown in Fig. 3.20. By comparison with Fig. 3.15, it is shown that the results from simulation and from in-water trials are very similar, confirming the validity of the simulations.

Finally, a third and more irregularly shaped workspace was tested. In addition, the threshold for the coverage mission to be considered complete was higher at 98% confidence.

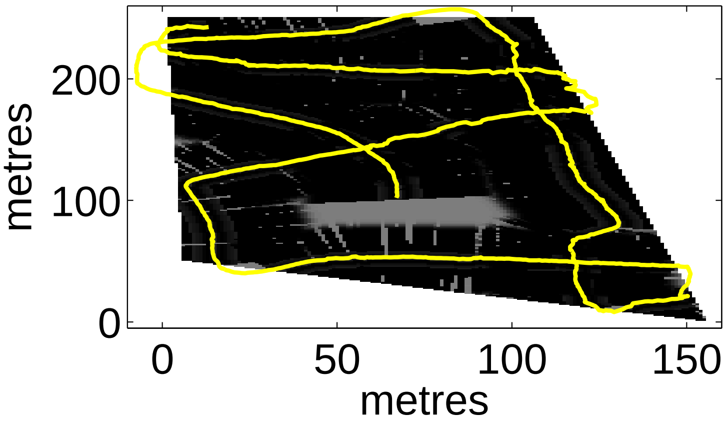


Figure 3.18: Path taken by AUV in-water trial overlayed on final confidence map. In the confidence map, darker areas indicate higher confidence. Final confidence was 95%.

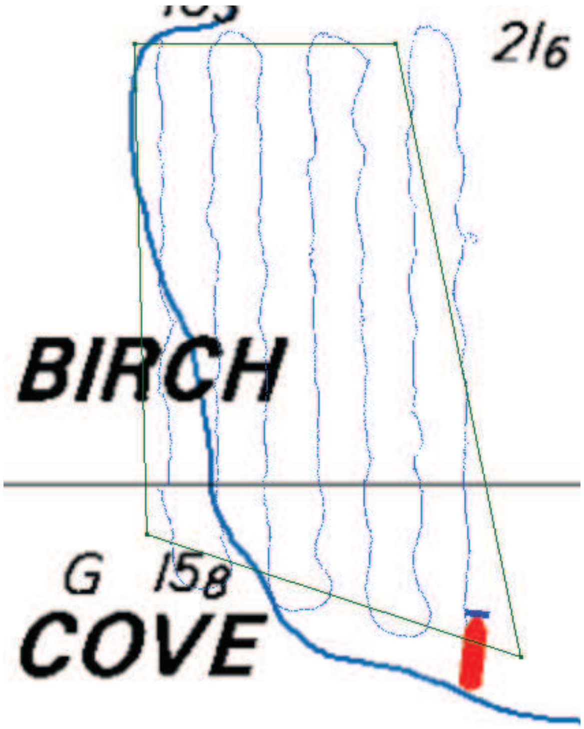


Figure 3.19: Lawn mower path to cover the same workspace as Fig. 3.18 to 95%.

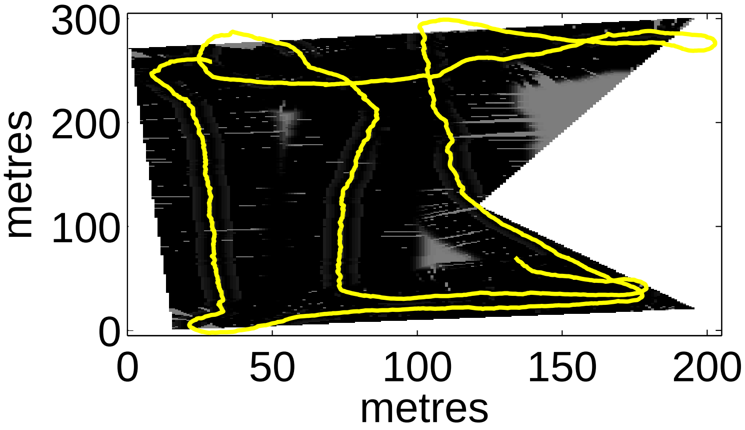


Figure 3.20: Path taken by AUV in non-convex workspace during in-water trial overlayed on resulting confidence map (darker areas are higher confidence). Final confidence 95%.

	Path Length	Workspace Area	Confidence
Proposed planner (Fig. 3.18) Lawn mower (Fig. 3.18)	1203 m 1580 m	28 000 m^2	95%
Proposed planner (Fig. 3.20)	1661 m	41 250 m^2	95%
Proposed planner (Fig. 3.21)	1717 m	41 200 m^2	98%

Table 3.4: Sample path lengths for paths planned during hardware trials.

The resulting path and confidence map are shown in Fig. 3.21.

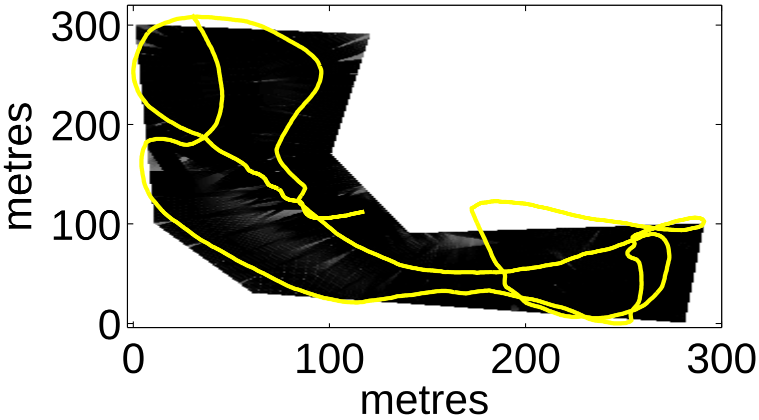


Figure 3.21: Path taken by AUV in very oddly shaped environment during in-water trial overlayed on resulting confidence map (darker areas are higher confidence). Final confidence 98%.

The path lengths for the trials are shown in Table 3.4.

3.6 Discussion

As discussed, the status quo for AUV sidescan seabed surveys is to perform a structured search, either a lawn mower or zig-zag type pattern. The waypoints that define the path are either input by a human operator or somehow optimized beforehand using a method such as [228]. The method proposed here is fundamentally different. The simulation and experimental results illustrate that the proposed planner is able to find shorter paths under many conditions. However, the benefits of the approach extend beyond simply shorter path lengths. In order to further compare the method presented against the standard lawn mower method, an empirical comparison is presented in Table 3.5.

	Proposed Planner	**Lawn Mower Approach**
Coverage overlap	While it is acknowledged that there is some overlap as the path sometimes crosses itself, the actual amount of coverage overlap is reduced evidenced by the fact that coverage swaths can be more accurately estimated and accounted for online.	Guaranteeing high coverage requires tight spacing of lawn mower tracks which results in high coverage overlap, but in some cases when desired coverage is sufficiently low, overlap can be minimized.
Level of autonomy	Extremely high. One button solution.	Usually requires operator to specify waypoints to define tracks. In complex environments the performance is subject to the judgement and skill of the survey designer.
Total energy consumption	Fairly low since trajectories are smooth.	Requires sharp turns at the end of tracks and also high energy requirements to follow the track as shown by the jagged path in Fig. 3.18.
Online vs. off-line	All planning takes place online. The main advantage as has been stated is that mission plans can be adaptive to environmental parameters and stochastic sensor observations.	All planning is done offline. There is an inherent assumption with this type of planning that the vehicle trajectory will exactly follow the plan and that all environmental parameters will be as predicted.
Deterministic vs. stochastic	Stochastic in that it is capable of adapting to the stochastic nature of state estimation and sensor input.	Deterministic.
Completeness	Probabilistically complete - due to the branch entropy behaviour, the AUV is guaranteed to find the areas of the map that are not covered.	Although the motion plan can have a guarantee of completeness, there is no actual guarantee of that the entire workspace will be covered in reality.
Path tracking	Optimizes an objective function over heading so paths are not tracked.	Required to follow track between waypoints.
Level of feedback	Closed loop - sensor feedback used to update plan.	Open loop.
Computational requirements	High - requires simulation over headings for IG behaviour but only a simple numerical calculation for the branch entropy behaviour.	Low.

Optimality	Provides no guarantee of optimality.	In certain cases where the environmental parameters are known and the desired coverage is low can provide provably optimal plans.
Quality of sonar data	Mosaicking can be difficult if swath overlap is not achieved or if paths contain curves.	In general can be better since the survey planner has direct control over the amount of swath overlap at the planning stage.
Overall performance	Results in path with shorter path in general if environmental parameters are not known *a priori* and particularly in the case of complex environment geometries.	Paths tend to be longer if worst case environmental conditions are assumed. However, in ideal case with full prior knowledge, simple environment geometry and low required confidence threshold can provide better solution.

Table 3.5: Comparison of proposed method and standard lawn mower.

3.7 Summary

This chapter presents an online sensor-driven robotics path planner with particular application to seabed coverage with a sidescan sonar sensor and an AUV. The approach combines information theory with a new concept referred to as branch entropy to efficiently cover areas of seabed. Simulation results and in-water trials illustrate the benefit of this approach over standard lawn mower planners. These advantages are: the total path length and time to cover an environment are shorter in many cases, heading is better maintained for data mosaicking, there is no need for predetermined waypoints, factors affecting sensor performance can be accounted for, the planner is able to autonomously handle very complex shaped environments, and the planner preferentially views the seabed from different insonification angles which is preferable for target recognition.

In the following two chapters, this approach will be extended to incorporate uncertainty in the vehicle pose, and then to multiple AUVs.

Chapter 4

Probabilistic Area Coverage: Accounting for Sensor Location Uncertainty

4.1 Introduction

In this chapter two fields of robotics research, area coverage and state estimation, are linked at a level that has not been seen to date. As will be shown, this improves the ability of a robot to achieve its coverage objective in the real world. First the theory is developed for a general coverage problem and then specifically applied to AUV seabed coverage.

4.1.1 Problem Statement

The problem of coverage can be defined as taking a sensor measurement of, or passing an end effector over, an area of interest [48]. A great deal of literature exists on planning paths for complete coverage that can generate plans that are provably complete [1, 3]. However, as is noted in [48] "The term complete is used in the motion planning sense, not in the operating research field sense." Indeed it is acknowledged that while a plan might achieve provable coverage there is no guarantee of actual area coverage when the degree of coverage cannot be directly sensed.

Historically, coverage is determined by assuming that there is perfect knowledge about the location of the sensor. Since coverage is a function of the configuration of the sensor at the time measurements are taken and the sensor is attached to a robot platform with uncertain pose, the measurement, and consequently the area covered by the measurement, will be uncertain.

For example, consider Fig. 4.1 which represents field data of an AUV covering an area of seabed with a SSS. In this case, the mission was run on the surface for access to the GPS data for ground truth, but the GPS data is not used when the vehicle is inside of the workspace (yellow box) to simulate an AUV submerging to perform the survey and then surfacing at the end of each leg. Position was estimated on-board the Iver2 AUV using an EKF which fused velocity measurements from the DVL and orientation from the compass. Fig. 4.1-a shows the region to be covered, the pre-planned lawn mower tracks designed to cover the area with the SSS, the AUV position estimates from the EKF, and the actual trajectory from the GPS data. Fig. 4.1-b shows the desired coverage that would be obtained if the AUV were able to exactly follow the pre-planned tracks, Fig. 4.1-c shows the believed coverage based on the EKF position estimates, and Fig. 4.1-d shows the coverage based on the GPS data. In this case the desired coverage was 96% (b), the estimated coverage was 80% (c), and the actual coverage was 67% (d).

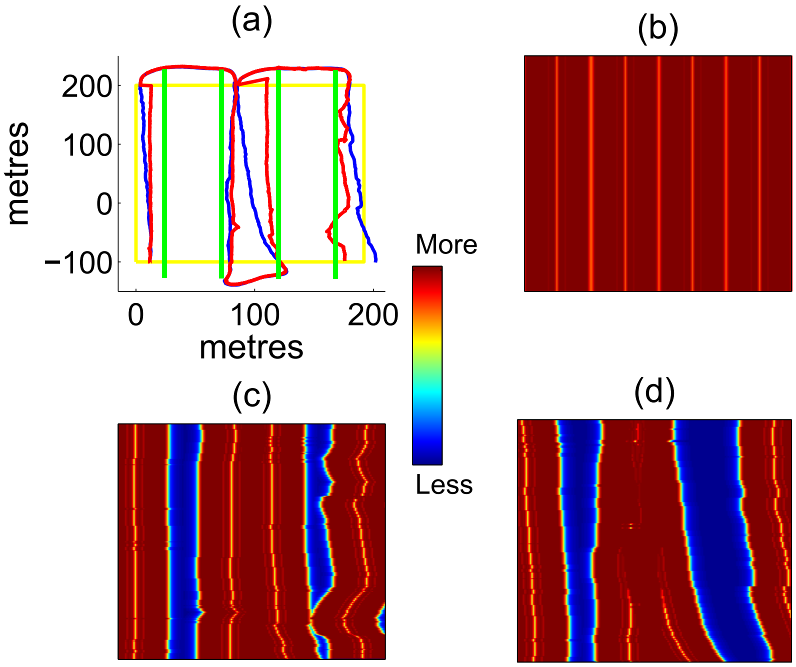


Figure 4.1: **(a)** Plot of workspace to be covered (yellow) with desired lawn mower tracks (green), EKF estimated path (red), and achieved path (blue). **(b)** Desired coverage over workspace based on exactly following the lawn mower tracks. **(c)** Estimated coverage from EKF pose estimates. **(d)** Achieved coverage based on GPS ground truth data.

In the case that seabed coverage is being done for a safety critical task such as MCM, this represents a large discrepancy between the desired risk (b), the perceived risk (c), and the real risk (d) associated with moving an asset and personnel over this area of seabed.

The actual vehicle trajectory can deviate from the plan for a number of reasons including: 1) The plan may not be feasible based on limitations of the non-holonomic vehicle, 2) The tracking controller on-board the vehicle may not be able to exactly track the planned path due to unexpected external disturbances such as currents, or 3) The vehicle location estimate contains uncertainty. In the previous chapter an approach is proposed that maintains a map of the coverage as the AUV traverses the workspace and then plans paths based on the achieved coverage, which overcomes the first two issues. However, based on Assumption 3.2.1, the third issue has not been addressed since we assumed perfect knowledge of the AUV location. In this chapter that assumption is removed.

Location uncertainty effects have been fairly extensively considered in general start-to-goal path planning literature as summarized in Chapter 2.1.8. However, work that accounts for any of these in a coverage context is fairly scarce. For example, in [36], a heuristic method is developed that combines three types of behaviours: inward spiral, shifting spiral, and greedy. The application is for a lawn mowing robot and the kinematic constraints of the vehicle are explicitly considered. The coverage map is maintained, and after the initial pass with the inward spiral and shifting spiral behaviours, the missing areas are covered using a greedy approach. Localization uncertainty is naively accounted for by planning paths to have some sensor overlap.

Perhaps the most closely related work is presented in [53], where a "probably approximately correct" measure of coverage performance is developed that accounts for localization uncertainty. Das et. al. argue correctly that once platform localization error is significant, the definition of the coverage completion criterion must be modified to become probabilistic. However, the approach only considers simple coverage sensor characteristics and provides no adaptive approach to account for the uncertainty in vehicle pose. Also, it is assumed that the localization variance reaches a steady-state. This may be reasonable if the state estimate is derived from a low-precision GPS, but will not be the case when the vehicle is ded reckoning and experiences unbounded position uncertainty growth.

Some methods proposed to adjust existing heuristics so that platform uncertainty may be accounted for (e.g. move tracks or spirals closer to each other) [36]. However this can produce paths with unnecessary sensor overlap in the case that pose uncertainty is low. Here we propose to maintain the "probability of coverage" over the entire workspace and use this to plan paths.

4.1.2 Objectives and Significance

There are three main objectives in this chapter:

1. Develop a framework whereby platform location uncertainty can be incorporated within the coverage model.

2. Propose a state estimation algorithm that is well-suited to the coverage problem.

3. Extend the planning algorithms proposed in the previous chapter to operate within this probabilistic model of coverage.

Finally these methods will be applied to the AUV seabed surveying problem. The probabilistic area coverage framework is ideally suited to AUV surveying for a number of reasons. First, the localization uncertainty is often not negligible due to the difficulties in estimating pose underwater. Second, trajectory estimation is essential to correct past pose beliefs after the vehicle obtains accurate localization information such as GPS or acoustic updates from a beacon. Third, in the case that the survey is being done for the purposes of MCM, it is extremely important that an accurate representation of the coverage be obtained due to the safety-critical nature of the application. AUV localization critically impacts the location of targets and the subsequent re-acquisition and disposal. A location of seabed that is falsely identified as covered could contain a mine and put lives and assets at risk.

4.1.3 Overview of Contribution

This research bridges the gap between area coverage and state estimation that has only been mentioned in passing in previous literature. The work here achieves this by developing a probabilistic coverage framework that can accurately represent the probability of coverage. In the presence of significant platform uncertainty, mission completion must be redefined as achieving a specified probability of coverage. For example, "we want to be 80% sure that we have covered at least 90% of the workspace" could be a valid mission objective but "we want to cover at least 85% of the workspace" is not.

This is the first known work that explicitly provides a general framework to combine the robot pose uncertainty resulting from state estimation from noisy pose sensors with a model of the area coverage. Once the link has been made, it becomes clear that the quality of the state estimation will have an impact on the coverage. The coverage over the workspace is a function of the entire vehicle trajectory, so it is proposed that the entire trajectory should be estimated (sometimes referred to as smoothing) as opposed to estimating only the pose at the present time (filtering). However, since the computation required for estimating the entire trajectory will grow with time, a new adaptive sliding window approach is proposed that approximates the best full trajectory estimate. The window size is adaptive based on the information gain along the trajectory resulting from global updates.

Once the framework has been developed, a planning algorithm is presented that operates within this new richer and more descriptive probabilistic coverage model. Many classical deterministic CPP methods no longer apply due to the stochastic nature of the problem. The planning algorithm presented in the previous chapter must be adapted to account for the new uncertainty associated with the coverage. To summarize, the contributions of the present work are the following:

1. A probabilistic coverage model that uses stochastic state estimates to produce a more descriptive representation of the coverage in the sense that it accounts for the uncertainty in the system.

2. An adaptive sliding window method for estimating the AUV trajectory to obtain a close approximation to the optimal coverage estimate without requiring more computation as time passes or the size of the workspace grows.

3. A path planning framework to achieve coverage within the probabilistic coverage model.

4. An application with field results of an AUV being deployed for underwater seabed coverage for MCM.

4.1.4 Chapter Nomenclature

An overview of the nomenclature used in this chapter is given in Table. 4.1.

4.1.5 Chapter Outline

The general framework for probabilistic coverage is presented in Section 4.2. In Section 4.3, it is motivated that a trajectory estimation, or smoothing, technique is preferable from a coverage standpoint and a sliding window approach is proposed. The proposed framework and planning algorithms from the previous chapter are applied to the problem of seabed coverage for an AUV in Section 4.4. Section 4.5 describes the experimental setup. In-water results are presented in Section 4.6. Finally, some conclusions are presented in Section 4.7.

Variable	Description
t	Time index
$\mathcal{S}_t$	Coverage sensor swath
$\mathcal{W}$	Workspace to be covered
g, r, s	Global, robot, and sensor frame respectively
$\mathbf{c}^i, i = 1..N$	N small grid cells
w	Coverage level
$\mathbf{X}_t$	RV for robot pose (position and orientation)
$\mathbf{U}_t$	RV for control input or odometry
$\mathbf{Z}_t$	RV for localization sensor readings
${}^s\mathbf{C}_t^i$	RV for location of cell i in sensor frame
$\mathcal{S}_t'$	Increased sensor swath incorporating all cells that have a "reasonable" chance of being covered
$\mathcal{H}({}^s\mathbf{c}_t^i)$	Function to map a cell in the sensor frame to a coverage value (coverage sensor model)
${}^s_g\mathbf{T}_t$	Transformation matrix to map a cell location in the global frame to the sensor frame
$\breve{W}_t^i$	RV for coverage of cell i resulting from coverage sensing from only time t
W_t^i	RV for coverage of cell i resulting from all coverage sensing for all time up to and including t

Table 4.1: Chapter 4 nomenclature.

4.2 Probabilistic Coverage Framework

4.2.1 Problem Formulation

Consider a robot with platform geometry $\mathcal{A}$ and coverage sensor swath $\mathcal{S}$ (Fig. 4.2) [41]. Define the workspace $\mathcal{W}$ as the area that is to be covered.

Similarly to the last chapter, the workspace is decomposed into N small grid cells: $\mathbf{c}^i, i = 1..N$ where the size of the cells is sufficiently small that the coverage can be treated as uniform over the cell. This cell size could be determined by the resolution of the coverage sensor. Each cell $\mathbf{c}^i$ is represented by its position in the workspace. For example, for the 2D case, $\mathbf{c}^i \triangleq [x^i, y^i]^T$ where x^i and y^i are the x and y locations respectively of cell $\mathbf{c}^i$ in the global frame.

If the robot takes a sensor reading at time t, then the set of all cells that are at least partially covered by the measurement defines the coverage sensor swath, $\mathcal{S}_t$.

The position of a cell $\mathbf{c}^i$ in the sensor frame is obtained by performing a transformation from the global frame to the sensor frame through the robot body frame:

$$ {}^s\mathbf{c}_t^i \triangleq [{}^s x_t^i, {}^s y_t^i]^T = {}^s_r\mathbf{T}\ {}^r_g\mathbf{T}_t\ {}^g\mathbf{c}^i = {}^s_g\mathbf{T}_t\ {}^g\mathbf{c}^i \tag{4.1} $$

where ${}^s\mathbf{c}_t^i$ and ${}^g\mathbf{c}^i$ are the location of the cell in the sensor and the global frame respectively and ${}^b_a\mathbf{T}$ is a homogeneous transformation (rotation and translation) from frame a to frame b. In the absence of any specified frame (preceding superscript) assume the global frame.

We can associate with each cell $\mathbf{c}^i$, $i = 1..N$ a value w_t^i that represents the level to which cell $\mathbf{c}^i$ is covered at time t. The collection of values, $\mathbf{w}_t \triangleq [w_t^1, w_t^2, ..., w_t^N]$ is referred to as

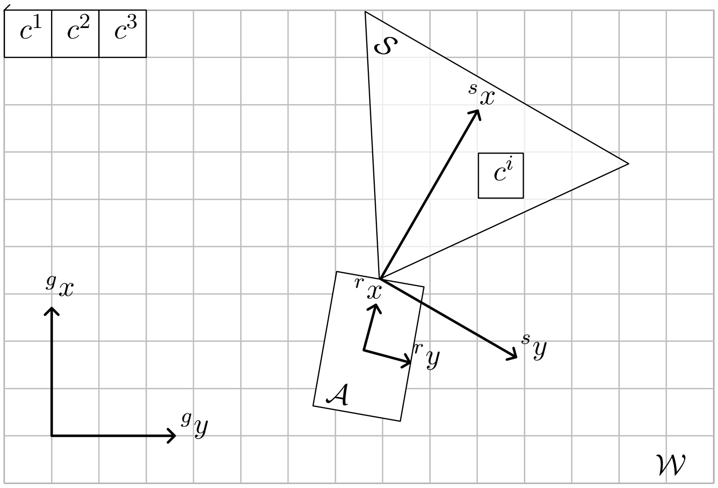


Figure 4.2: The workspace $\mathcal{W}$ is decomposed into N cells. As the robotic platform, $\mathcal{A}$, moves about, $\mathcal{W}$ becomes covered in a manner consistent with the coverage sensor geometry $\mathcal{S}$. The cells should be chosen small enough such that the coverage can be considered as constant over the cell and cells are never partially covered. Here the sensor geometry is shown as a 2D swath for the general robotics coverage problem. In the case of AUV seabed coverage, the swath is a 1D line perpendicular to the direction of AUV motion.

the coverage grid map (CGM).

There are two important assumptions that are common for coverage problems that should be explicitly stated:

Assumption 4.2.1. *The workspace being covered is static (for the case an AUV MCM mission this would mean that no mines are being added or removed during the survey). As a result the level of coverage can only increase with time during the mission.*

Assumption 4.2.2. *The coverage level cannot be directly sensed. The coverage level must be inferred through the coverage sensor model, such as ESPRESSO [54].*

Based on a detailed investigation of the coverage sensor, such as ESPRESSO [54] for the case of AUV seabed surveying, the CGM cells are updated according to:

$$w_t^i = \mathcal{H}({}^s\mathbf{c}_t^i), \tag{4.2}$$

where $\mathcal{H}({}^s\mathbf{c}^i)$ represents the coverage sensor model and defines how to update the CGM based on the locations of the grid cells in the sensor frame.

If we assume a perfect and uniform coverage sensor then $w \in \{0.5, 1\}$, meaning that each cell is either covered or not ($w = 0.5$: not covered, $w = 1$: covered).

In this case we state the complete coverage objective, assuming known poses, as requiring

$$w_t^i = 1, \forall i = 1..N. \tag{4.3}$$

However, in many cases, such as MCM sonars, the coverage sensor performance is not uniform as is elicited by the lateral range curves shown in Fig. 2.14. In the case of a non-uniform sensor characteristic, $w \in [0.5, 1]$. In this case, (4.3) can become difficult or impossible to achieve depending on the coverage sensor model. We can consider the mission complete either when a proportion of cells, P_1, have been covered to a specified level w_{thresh}:

$$\frac{1}{N}\#\{i|i = 1..N, w_t^i > w_{thresh}\} > P_1, \tag{4.4}$$

where $\#\{\cdot\}$ is the cardinality operator on a set, or when we have achieved a desired average coverage, w_{avg} over the entire workspace:

$$\frac{1}{N}\sum_{i=1}^{N} w_t^i > w_{avg}, \tag{4.5}$$

which was the criterion for mission completion used in the previous chapter.

4.2.2 Coverage with Uncertain Poses

One of the main conjectures in this work is that if we remove Assumption 3.2.1 and consider the AUV pose uncertainty then these mission objectives (4.4), (4.5) are not sufficient criteria for mission completion because they do not encapsulate the uncertainty in the coverage. In general it is impossible to exactly localize a mobile robot so there will always be some uncertainty about which areas of the workspace have been covered.

As an example, consider the simple circular shape floor vacuuming robot in Fig. 4.3. In this case, the size of the end effector is identical to the robot's footprint . For an indoor application the robot might estimate its position from only its two wheel encoders. Since the encoder does not give a global position measurement, only a change in position estimate, as time passes the error in the encoder data will accumulate causing the position estimate to drift. In the figure, the robot drift results in some areas being inadvertently covered whereas other areas that were meant to be covered are not. In Fig. 4.4 we consider the same robot at a snapshot in time. The robot pose has some uncertainty and the areas of higher probability of coverage are represented as lighter in the figure. In this case, we can be fairly certain that cell $\mathbf{c}^1$ is covered and fairly certain that cell $\mathbf{c}^3$ is not covered. But for cell $\mathbf{c}^2$ there is uncertainty about its coverage status.

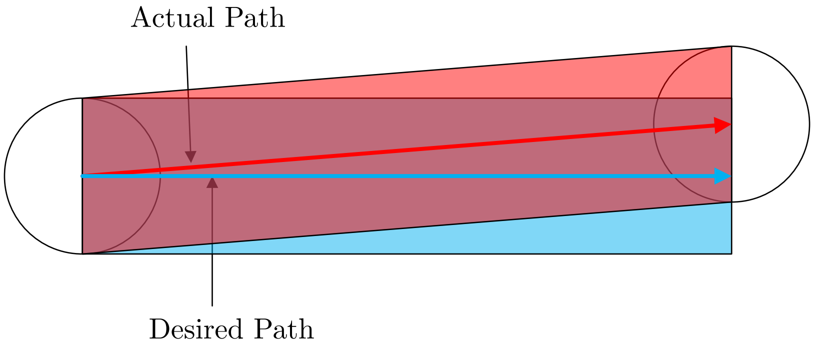


Figure 4.3: A simple circular robot moving with drift. The blue arrow and associated coverage area corresponds to the desired path and coverage. The red arrow and area are the actual path and coverage. Some areas are not covered that were meant to be while others were inadvertently covered.

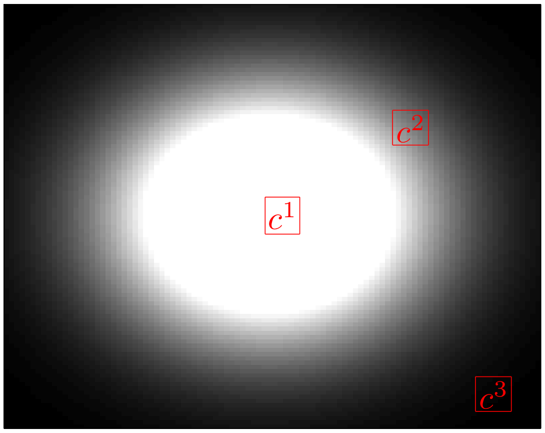


Figure 4.4: The robot from Fig. 4.3 at a snapshot in time. The lighter areas indicate areas with a higher probability of being covered than darker areas. The cells $\mathbf{c}^1$, $\mathbf{c}^2$ and $\mathbf{c}^3$ will have different probabilities of being covered.

4.2.3 Accounting for Pose Uncertainty During Coverage

Consider a robot whose pose distribution at time t is given by $\mathbf{X}_t$. The goal of recursive state estimation is to estimate the belief distribution of the state $\mathbf{X}_t$ given by [215]:

$$\begin{aligned} &bel(\mathbf{X}_t = \mathbf{x}_t) \triangleq bel(\mathbf{x}_t) = \\ &p(\mathbf{X}_t = \mathbf{x}_t | \mathbf{U}_{1:t} = \mathbf{u}_{1:t}, \mathbf{Z}_{1:t} = \mathbf{z}_{1:t}, \mathbf{X}_0 = \mathbf{x_0}) \triangleq \mathbf{p}(\mathbf{x_t} | \mathbf{u_{1:t}}, \mathbf{z_{1:t}}, \mathbf{x_0}) \end{aligned} \tag{4.6}$$

where $\mathbf{U}$ is some control input or odometry [215] and $\mathbf{Z}$ is a measurement used for localization.

In filtering, the state at time t is recursively estimated through an approximation of the Bayes' filter which operates in a predict-update cycle. Prediction is given by [215]:

$$\bar{bel}(\mathbf{x}_t) = \int p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) bel(\mathbf{x}_{t-1}) d\mathbf{x}_{t-1} \tag{4.7}$$

and update is given by:

$$bel(\mathbf{x}_t) = \eta p(\mathbf{z}_t | \mathbf{x}_t) \bar{bel}(\mathbf{x}_t). \tag{4.8}$$

where η is a normalization factor. Central to this formulation is the Markov assumption, that only the most recent state estimates, input and measurements need to be considered to generate the estimate of the next state.

The pose belief at time t is used to generate a probability of coverage for all grid cells in the workspace. Now, instead of each cell having one value that represents its coverage, it has a whole distribution representing the probability of coverage to different levels. The uncertainty in the robot pose transfers to an uncertainty in the position of cell i in the sensor frame: ${}^s\mathbf{c}_t^i$ caused by the uncertain transformation ${}^r_gT_t$ in (4.1). The position of cell i in the sensor frame is now uncertain and is represented by a RV ${}^s\mathbf{C}_t^i$ whose distribution is calculated by mapping the position of the cell in the global frame through the uncertain

Algorithm 8 Iterative coverage distribution estimation

```
Predict the current pose belief using (4.7)
Update the current pose belief using the prediction and the localization measurement using (4.8)
for all cells in the effective coverage sensor swath do
    Map the cell location to the sensor frame using (4.9)
    Generate the coverage distribution for only the current coverage sensor reading using (4.14)
    Update the current coverage distribution with the previous coverage distribution using (4.22)
end for
```

transformation from global to sensor frame where:

$$p({}^{s}\mathbf{C}_t^i = {}^{s}\mathbf{c}_t^i) = p({}^{s}_{g}\mathbf{T}_t \, {}^{g}\mathbf{c}^i = {}^{s}\mathbf{c}_t^i) \triangleq p({}^{s}\mathbf{c}_t^i) \tag{4.9}$$

is the probability that cell i sits at location ${}^{s}\mathbf{c}_t^i$ in the sensor frame at time t.

The uncertain location of the cell, in the sensor frame, results in an uncertain coverage. Consequently, the coverage is now represented by a RV W_t^i where $p(W_t^i = w_t^i)$ represents the probability that cell $\mathbf{c}^i$ is covered to a coverage level w at time t considering all past robot states.

Similarly to the previous chapter, it is useful to define a RV $\breve{W}_t^i$ that represents the probability that cell $\mathbf{c}^i$ is covered to a level w_t^i resulting from *only* the sensor reading at time t.

The robot pose uncertainty will result in unintended cells having some probability of being covered. As a result, we define the effective coverage sensor swath, $\mathcal{S}_t'$ to be all cells that have a "reasonable" chance of being covered at time t:

$$\mathcal{S}_t' = \{\mathbf{c}^i | p(\mathbf{c}^i \in \mathcal{S}_t) > \epsilon, i = 1..N\} \tag{4.10}$$

where ϵ is some arbitrarily small threshold.

The probabilistic coverage estimation is represented by the Bayes' network shown in Fig. 4.5 which is an extension of the standard Bayes' network used for robot pose estimation [215].

The mapping from pose to cell location in the sensor frame is achieved with (4.9) (yellow arrows). The uncertain cell location is then mapped to a coverage distribution for a single timestep (red arrows) and is described further in Section 4.2.4. Finally, the coverage estimate from a single timestep is combined with the previous coverage estimates (blue arrows) and is described further in Section 4.2.5. The iterative generation of the coverage distribution is briefly summarized in Algorithm 8. In Fig. 4.5, it is assumed that the coverage distribution is Markovian (the coverage estimate at time t only depends on the estimate at time $t-1$), an assumption that will be revisited in Section 4.2.5.

Consider again the simple vacuum cleaner robot shown in Fig. 4.4 where three cells are shown after a single sensor swath at time t. For cell $\mathbf{c}^1$ we can say that it is almost certainly covered:

$$p(\breve{W}_t^1 = 1) \approx 1. \tag{4.11}$$

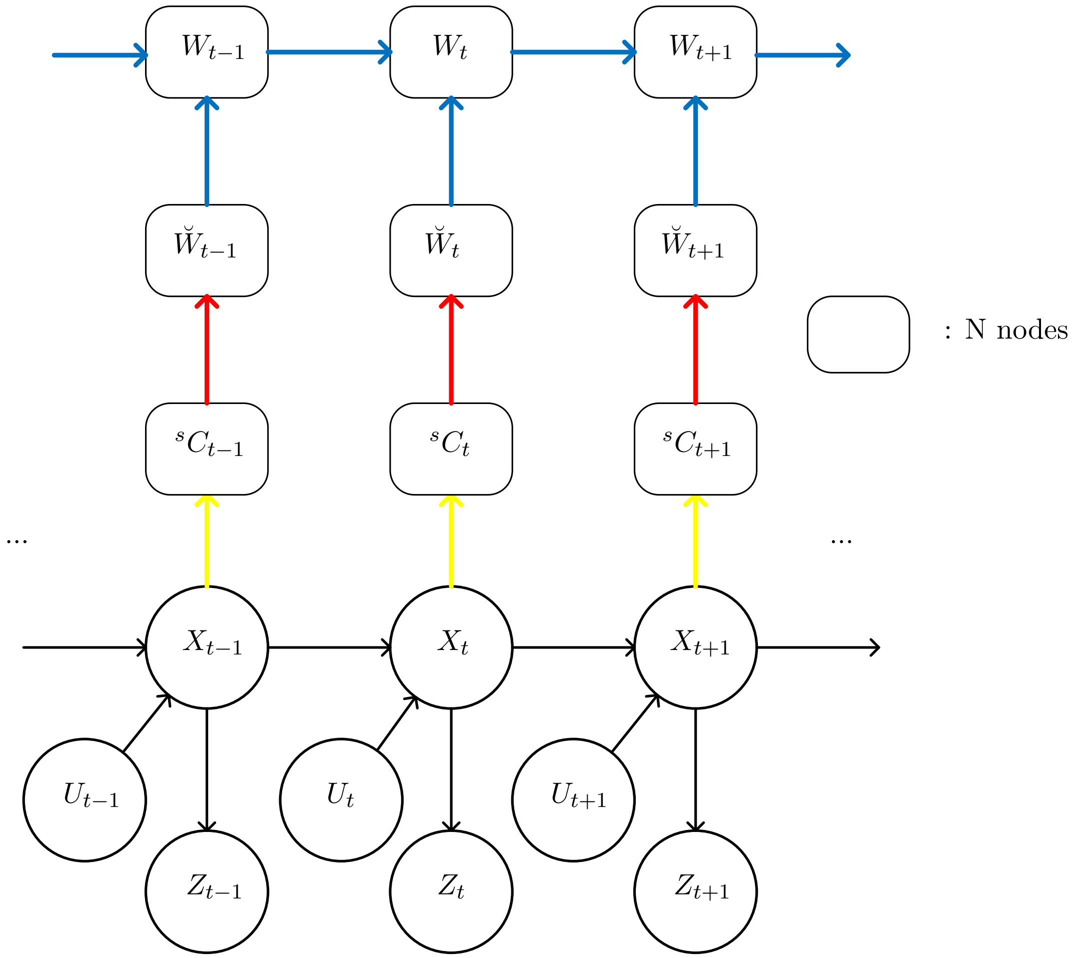


Figure 4.5: Bayes' network describing how to estimate the coverage distribution recursively.
↑: Map the uncertain pose $\mathbf{X}_t$ through the transformation using (4.1) to determine the uncertain locations of the cells in the sensor frame.
↑: Map the uncertain cell location through the coverage sensor model (e.g. sonar lateral range curve for AUV MCM)to obtain the coverage resulting from only time t using (4.14).
↑: Combine the coverage at time t with the previous coverage distribution using (4.22)

Similarly for cell $\mathbf{c}^3$ we are fairly certain that it was not covered:

$$p(\breve{W}_t^3 = 1) \approx 0. \tag{4.12}$$

However for cell $\mathbf{c}^2$ we are uncertain about its coverage status, but we need a formal and accurate representation of what the probability of coverage is based on the uncertain robot pose.

4.2.4 Propagating Robot Pose Uncertainty to Coverage Distribution

Here we describe the process for estimating the coverage distribution resulting from a single coverage sensor reading from an uncertain location. The distribution $p(\breve{W}_t^i = w)$, can be determined by propagating the uncertain cell location in the sensor frame through the coverage sensor model, $\mathcal{H}$:

$$\breve{W}_t^i = \mathcal{H}({}^s\mathbf{C}_t^i). \tag{4.13}$$

The distribution of $\breve{W}_t^i$ can be determined based on the theorem of total probability:

$$p(\breve{W}_t^i = w_t^i) = \int p(\breve{W}_t^i = w_t^i | {}^s\mathbf{C}_t^i = {}^s\mathbf{c}_t^i) p({}^s\mathbf{C}_t^i = {}^s\mathbf{c}_t^i) d^s\mathbf{c}_t^i \tag{4.14}$$

In the implementation, the value of $p(\breve{W}_t^i = w_t^i)$ can be found by applying the "FUNDAMENTAL THEOREM" ([168] p.93) for mapping RVs through functions:

$$p(\breve{W}_t^i = w_t^i) = \frac{p({}^s\mathbf{C}_t^i = {}^s\mathbf{c}_t^i[1])}{\mathcal{H}'({}^s\mathbf{c}_t^i[1])} + \frac{p({}^s\mathbf{C}_t^i = {}^s\mathbf{c}_t^i[2])}{\mathcal{H}'({}^s\mathbf{c}_t^i[2])} + ... + \frac{p({}^s\mathbf{C}_t^i = {}^s\mathbf{c}_t^i[n])}{\mathcal{H}'({}^s\mathbf{c}_t^i[n])} \tag{4.15}$$

where ${}^s\mathbf{c}_t^i[1], {}^s\mathbf{c}_t^i[2], ..., {}^s\mathbf{c}_t^i[n]$ are the n solutions found when solving $w_t^i = \mathcal{H}({}^s\mathbf{c}_t^i)$:

$$w_t^i = \mathcal{H}({}^s\mathbf{c}_t^i[1]) = \mathcal{H}({}^s\mathbf{c}_t^i[2]) = ... = \mathcal{H}({}^s\mathbf{c}_t^i[n]) \tag{4.16}$$

and $\mathcal{H}'(c)$ is the derivative of $\mathcal{H}(c)$.

Eqs (4.14) and (4.15) give us an explicit way of evaluating the coverage distribution of each coverage grid cell from a single uncertain pose. $\mathcal{H}({}^s\mathbf{c}_t^i)$ is evaluated from the coverage sensor model (4.2). $p({}^s\mathbf{c}_t^i)$ is generated by mapping the cell location in the global frame through the uncertain transformation ${}^s_g\mathbf{T}_t$ defined in (4.1).

Referring back to Fig. 4.4 to the simple case with uniform sensor characteristics, we can now formally represent the coverage status of the grid cell at location 2 by $p(\breve{W}_t^2 = 0.5)$ is the probability that cell c^2 is not covered and $p(\breve{W}_t^2 = 1)$ is the probability that it is covered from one sensor reading at time t.

Example 4.2.1. *For an example with a non-uniform coverage sensor model consider Fig. 4.6 which represents a 1D ($\mathbf{c}^i = [x^i]$) lawn mower. Shown at the top is a lawn mower with angled blades The angling of the blades produces the non-uniform coverage sensor model. In this case coverage is defined by the length of the grass with the coverage sensor characteristic shown in the top right. The distribution of the location of cell i in the sensor frame is shown in the bottom right. This distribution is propagated through the coverage sensor model to obtain the coverage distribution shown at the left.*

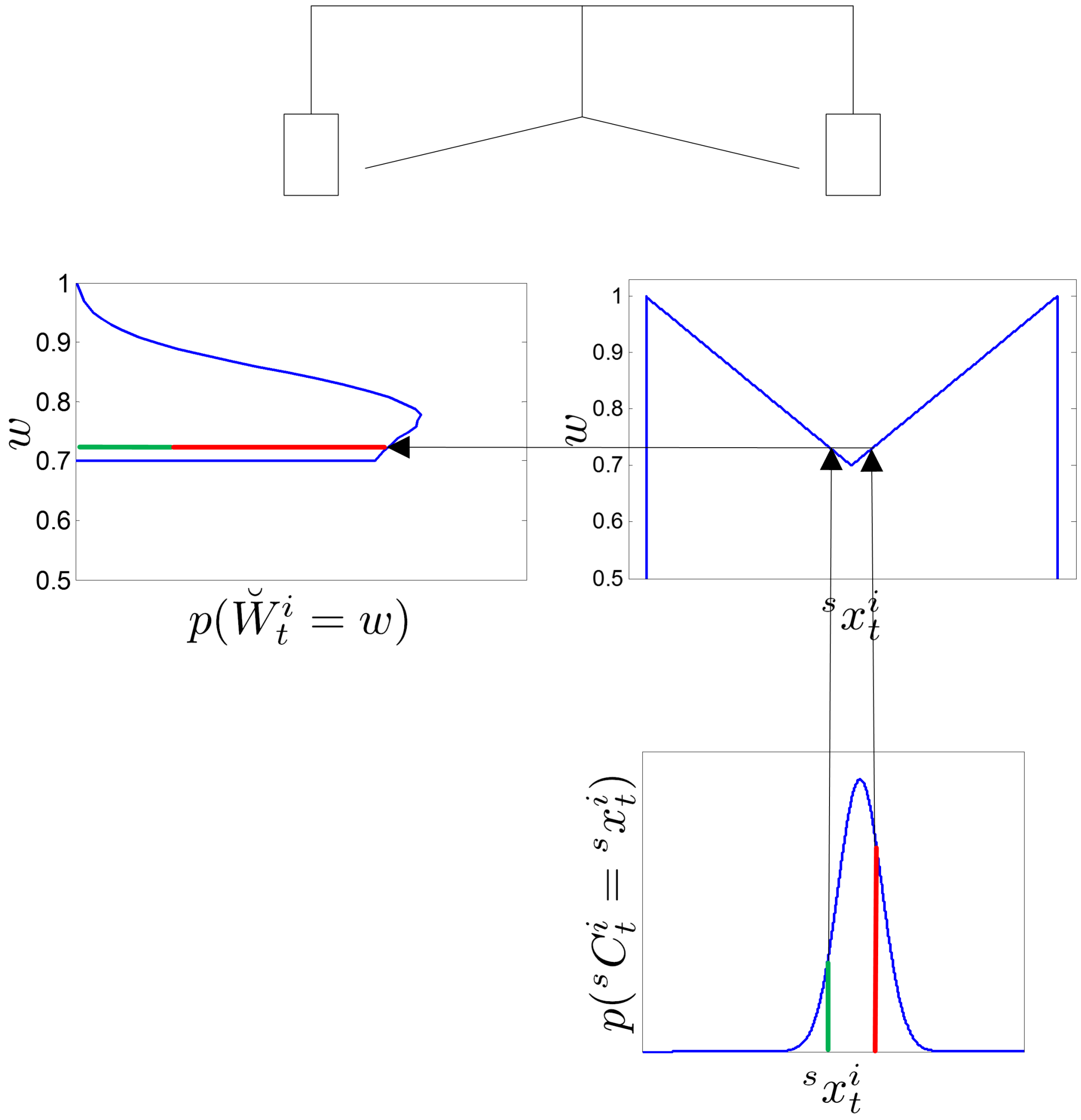


Figure 4.6: The uncertain cell location, ${}^s x_t^i$ (bottom right) is propagated through the coverage sensor model (middle right) to obtain the coverage distribution resulting from the only time t (middle left) for the lawn mower with uneven blades (top). Due to the non-uniform nature of the coverage model, the generation of one value in the coverage distribution is shown as the sum of the probabilities of two possible cell locations, shown as the red and green lines.

4.2.5 Generating the Coverage Distribution Recursively from Subsequent Measurements of the Same Location

Equation (4.14) provides a method for us to propagate the pose uncertainty through the sensor characteristic to obtain an estimate of the coverage distribution from a single sensor pose. Over the course of a coverage mission, the robot will move around the workspace and make many sensor readings. We require a way of combining the coverage distribution from time t with the previous coverage distribution.

First we will describe the process for combining coverage measurements assuming that the poses are completely known exactly, then the approach will be extended to the uncertain pose case.

4.2.5.1 Assuming Certain Poses

In the previous chapter (Proposition 3.2.1), it was proposed that the most optimistic way of fusing measurements, of the same location but at different times, is by assuming that they are statistically independent, which yields:

$$w^i = 1 - \prod_{l=1}^{L} (1 - \breve{w}^i_{t_l}). \tag{4.17}$$

where L is the total number of sensor readings of cell $\mathbf{c}^i$, and $\breve{w}^i_{t_l}$ is the coverage value for a single timestep $t = t_l$. Alternately, the most pessimistic assumption is to assume that all sensor readings are dependent, in which case they should be combined using the maximum function:

$$w^i = \max_{l=1\ldots L} \{\breve{w}^i_{t_l}\}. \tag{4.18}$$

How the values should be combined is dependent on the type of coverage problem. For example, for the lawn mowing case, the grass can be no shorter than the shortest cut, so using the max function would be appropriate. Regardless of the coverage problem, we can use the max function to obtain a lower bound on the coverage resulting in the following recursive formulation:

$$w^i_t = \max(\breve{w}^i_t, w^i_{t-1}) \tag{4.19}$$

where w^i_t now contains the coverage level for cell $\mathbf{c}^i$ incorporating all previous measurements of that cell.

4.2.5.2 Assuming Uncertain Poses

Using the maximum assumption, we can recursively define the coverage distribution using the maximum function on RVs.

It is known in probability theory that if X, Y, and Z are RVs with probability density functions $p(X = x)$, $p(Y = y)$, and $p(Z = z)$ respectively and $Z = \max(X, Y)$ then ([168] p.141):

$$p(Z = z) = p(X = z) \int_{-\infty}^{z} p(Y = \epsilon) d\epsilon + p(Y = z) \int_{-\infty}^{z} p(X = \epsilon) d\epsilon. \tag{4.20}$$

We can recursively define a lower bound on the coverage distribution at time t as a function of the coverage distribution at time $t-1$ and the coverage resulting from only time t as given by the following:

$$W_t^i = \max(\breve{W}_t^i, W_{t-1}^i) \tag{4.21}$$

where the distribution of W_t can be evaluated through the following relation:

$$\begin{aligned} p(W_t^i = w) = p(\breve{W}_t^i = w) \int_0^w p(W_{t-1}^i = \epsilon) d\epsilon \\ + p(W_{t-1}^i = w) \int_0^w p(\breve{W}_t^i = \epsilon) d\epsilon \end{aligned} \tag{4.22}$$

Based on this maximum function formulation for combining coverage sensor measurements for a given cell $\mathbf{c}^i$, we have defined a way of updating the coverage distribution at time t based only on the pose at time t (used to generate $\breve{W}_t^i$) and the coverage distribution at time $t-1$, W_{t-1}^i, meaning that the Markov assumption can be applied to the coverage sensor measurements and pose information can be discarded once the coverage distribution at time t has been updated based on the Bayes' network in Fig. 4.5.

Example 4.2.2. *Continuing on from Ex. 4.2.1, we now consider the case where a cell is covered twice by the lawnmower with angled blades. Fig. 4.7 shows a cell $\mathbf{c}^i$ that is covered at two times, t_1 and t_2, for the angled lawnmower example. At t_1 the vehicle state covariance is larger and mean location of cell $\mathbf{c}^i$ is closer to the middle of the lawnmower. At time t_2 the state covariance is smaller with mean location of cell $\mathbf{c}^i$ is closer to the right wheel. The distribution of cell location in the sensor frame (row **(a)**) is mapped through the coverage sensor model (row **(b)**) to obtain two instantaneous coverage distributions for times t_1 and t_2 represented by the distributions of the RVs $\breve{W}_{t_1}^i$ and $\breve{W}_{t_2}^i$ (row **(c)**). These two distributions are used in (4.22) to obtain the resulting RV W_t^i (row **(d)**).*

The algorithm for iterative coverage distribution estimation is summarized in Algorithm 9 which is a more detailed version of Algorithm 8. The first two lines constitute the initialization procedure. In line 1, the coverage distributions are initialized to have all weight at 0.5 ($p(W_0^i = 0.5) = 1$). In line 2, the pose state is initialized based on some prior state distribution. The algorithm then enters the main execution loop line 4-13 until mission completion. In lines 5 and 6 the vehicle pose is estimated. Then each cell in the effective sensor swath is processed in the loop from lines 7 to 11. The distribution of the cell location in the sensor frame is determined in line 8. In line 9 the uncertain state is propagated through the coverage sensor characteristic using (4.14). In line 10 the coverage distribution from time $t-1$ is combined with the coverage obtained at the current time t using (4.22). Lines 8, 9 and 10 are repeated for all cells in the effective sensor swath $\mathcal{S}_t'$.

4.2.6 New Definition of Mission Completion

It is argued here that the original definitions of mission completion that can be used for non-probabilistic coverage defined in (4.4) and (4.5) are not appropriate to describe mission completion now that coverage over the workspace is described stochastically. To compensate, new definitions of coverage are proposed to define mission completeness.

The first original criterion (4.4) is to require that a proportion P_1 of the area is covered up to at least a level of w_{thresh}, is updated to (4.23) which requires that a proportion P_1 of the area has a probability of being covered to at least a level w_{thresh} of at least P_2:

$$\frac{1}{N} \#\{i | p(W_t^i \geq w_{thresh}) \geq P_2, i = 1..N\} > P_1. \tag{4.23}$$

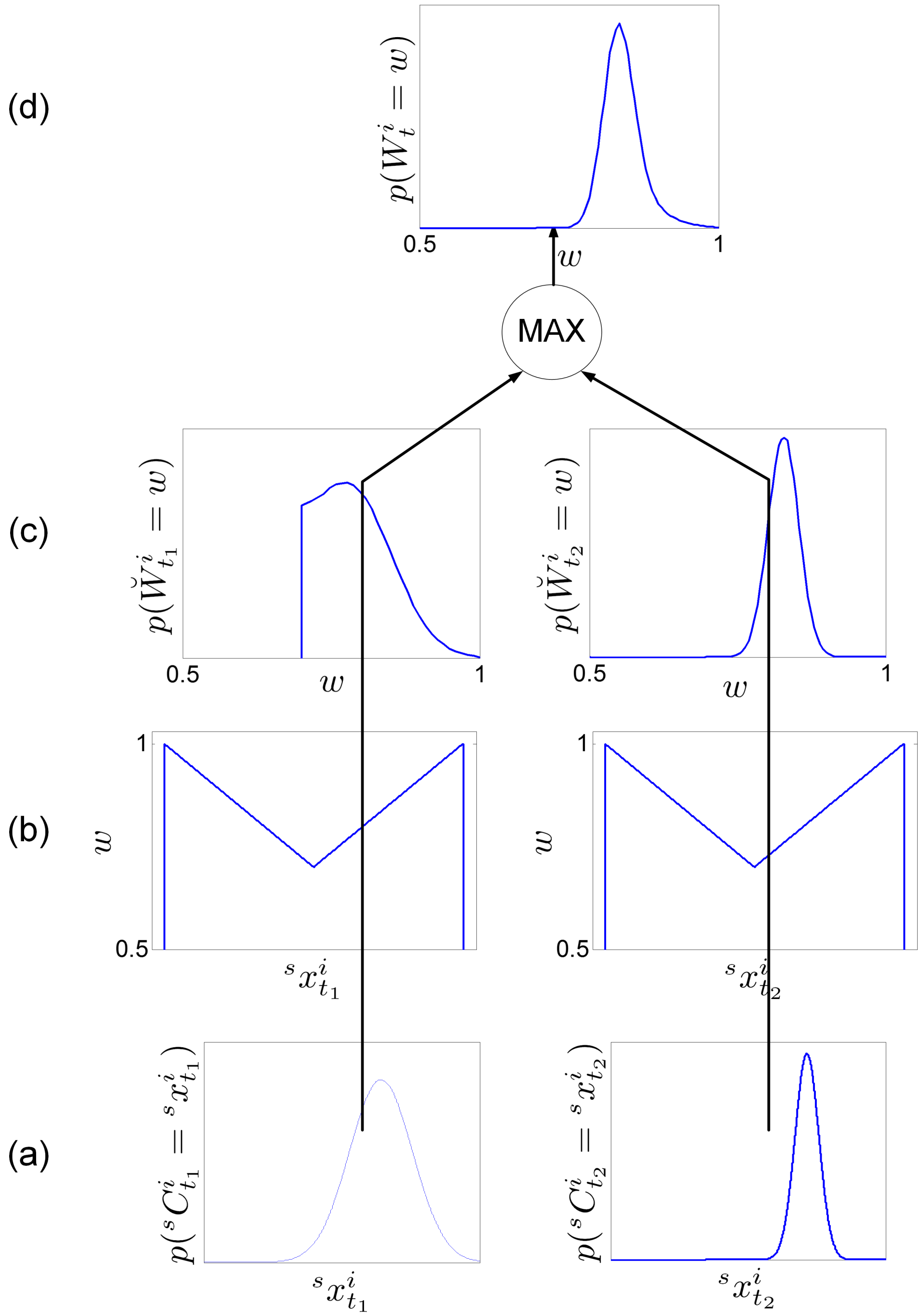


Figure 4.7: Cell $\mathbf{c}^i$ is covered twice by the uneven lawnmower. In each case, the distribution of the cell location in the sensor frame (row **(a)**) is mapped through the coverage sensor model (row **(b)**) to obtain an instantaneous coverage distribution (row **(c)**). These two distributions are combined using the max operation on RVs to obtain the final distribution (row **(d)**) that represents the coverage distribution for cell $\mathbf{c}_i$ resulting from both coverage readings.

Algorithm 9 Iterative coverage distribution estimation (detailed)

```
$p(W_0^i = 0.5) = 1\ \forall i = 1...N$
$bel(x_0) \leftarrow$ distribution of prior pose state estimate
$t \leftarrow 1$
**repeat**
    $\overline{bel}(\mathbf{x}_t) = \int p(\mathbf{x}_t|x_{t-1}, u_t)bel(x_{t-1})dx_{t-1}$
    $bel(\mathbf{x}_t) = \eta p(z_t|\mathbf{x}_t)\overline{bel}(\mathbf{x}_t)$
    **for all** $i$ such that $\mathbf{c}_i \in \mathcal{S}'_t$ **do**
        $p({}^s\mathbf{c}_t^i) = p({}^s_g\mathbf{T}_t\ {}^g\mathbf{c}_t^i = {}^s\mathbf{c})$
        $p(\breve{W}_t^i = w) = \int p(\breve{W}_t^i = w|{}^s\mathbf{C}_t^i = {}^s\mathbf{c}_t^i)p({}^s\mathbf{C}_t^i = {}^s\mathbf{c}_t^i)d^s\mathbf{c}_t^i$
        $p(W_t^i = w) = p(\breve{W}_t^i = w)\int_0^w p(W_{t-1}^i = \epsilon)d\epsilon$
          $+\, p(W_{t-1}^i = w)\int_0^w p(\breve{W}_t^i = \epsilon)d\epsilon$
    **end for**
    $t \leftarrow t + 1$
**until** mission completion
```

The second original criterion (4.5) is to have achieved a desired average coverage, w_{avg} over the entire workspace, can be updated to (4.24) requiring an expected average coverage greater than w_{avg}

$$\frac{1}{N}\sum_{i=1}^{N} E[W_t^i] = E[\frac{1}{N}\sum_{i=1}^{N} W^i] > w_{avg}. \tag{4.24}$$

In the case where the coverage sensor characteristic is uniform such as the robot vacuum example, $P_2 = 1$ and (4.23) reduces to the "probably approximately complete" measure of completeness presented in [53].

4.3 Adaptive Sliding Window Filter for Optimal Coverage Estimation

The previous section provides an explicit method of accounting for pose uncertainty in the coverage model. This framework is independent of the state estimation method. Coverage over the workspace is dependant on the entire sensor trajectory, therefore to obtain the best estimate of the coverage distribution using the proposed framework, we require the best estimate of the entire robot trajectory (i.e. the robot pose for all time steps, $1..t$) as opposed to just the present state, t. This is sometimes referred to as smoothing rather than filtering [57].

This approach is particularly applicable to the case when a robot is receiving intermittent global position updates and needs to optimize its trajectory between these position updates as is the case for AUV seabed coverage.

4.3.1 Motivation

The same lawn mowing mission as was shown in Fig. 4.1 where robot state estimation was performed with an EKF was repeated with a maximum *a posteriori* estimator [111] to optimize the entire AUV trajectory and the results are shown in Fig. 4.8. Once again, the robot dead reckons while inside the workspace, and then obtains GPS updates at the end of each track. In this case, the area that we believe we have covered (Fig. 4.8-b) and the area that we have actually covered (Fig. 4.8-c) have a much closer match than in Fig. 4.1. Fig. 4.8-d shows the mean probability of coverage over the workspace determined using the probabilistic framework presented in the previous section.

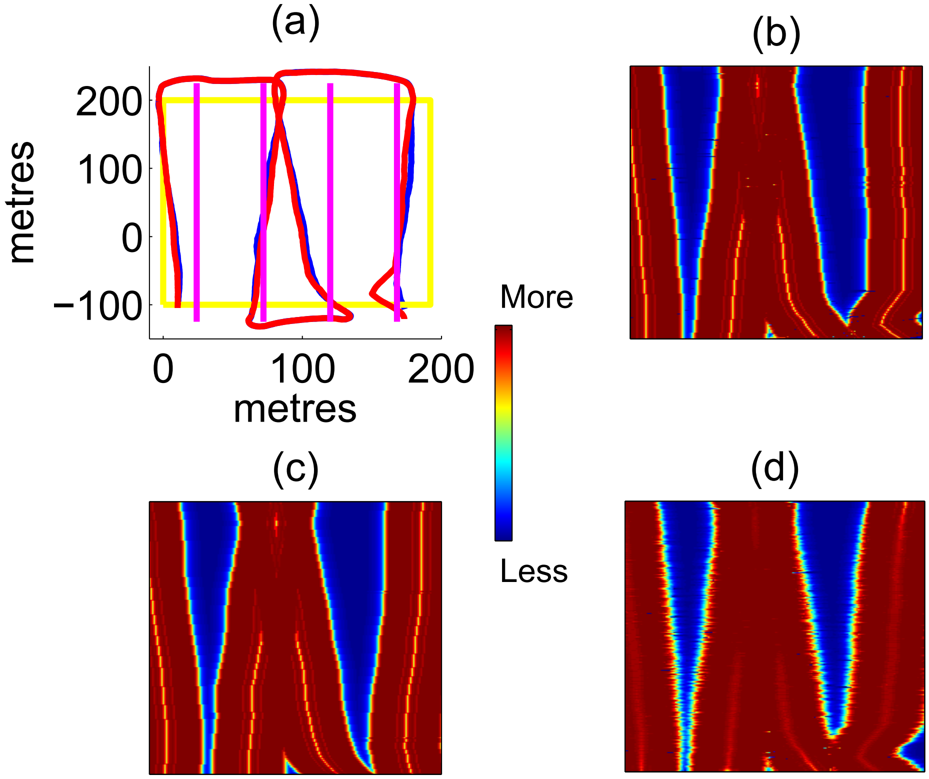


Figure 4.8: **(a)** Plot of workspace to be covered with desired tracks, estimated path, and actual path. (**b**) Estimated coverage. (**c**) Actual coverage. **(d)** Mean probability of coverage.

Not only does smoothing the trajectory give the user a more accurate representation of the coverage over the workspace, but in most cases reducing the uncertainty of past state estimates through smoothing the trajectory will increase the area coverage in the probabilistic framework. For a more rigorous treatment of the relationship between robot pose uncertainty and area coverage refer to Appendix **??**. If the robot can obtain an estimate of the smoothed trajectory online, then it can make decisions based on the optimal estimate of the current coverage.

Since minimizing the robot pose uncertainty will in general provide the most efficient completion of the coverage task, if we obtain some information at time t that allows us to improve our estimate of (and consequently reduce our uncertainty about) the robot pose at some previous time $t' < t$ then we can increase the total effective area of the workspace that has been covered by recalculating the coverage based on the updated pose estimate. In the implementation, every time the robot receives a global location update, either through a GPS sensor, cooperative localization, or loop closure in the case of SLAM, then the entire trajectory should be re-optimized and the coverage over the workspace should be recalculated based on the new optimized trajectory.

However, such an approach is problematic in that, since the coverage is a function of the entire trajectory, every time the trajectory is re-optimized, the entire coverage map must be re-computed from the start of the mission. As missions become longer the problem will become computationally intractable. To compensate for this, a adaptive sliding window filter approach is proposed whereby only the more recent robot poses that are still being updated by new information received at the present time are maintained in the state space.

4.3.2 Adaptive Sliding Window Criterion

In a sliding window filter [204], [108], a fixed number of past poses are maintained in the state space. In some cases it becomes unnecessary to maintain extremely old poses in the state space since they will no longer be appreciably updated by new incoming sensor data. The advantage of the sliding window filter approach is that required computation does not increase with time. Past sliding window filter approaches such as [204], [108], [98], use fixed and predetermined sizes for the sliding window.

Here we propose an adaptive sliding window criterion based on information gain whereby the size of the sliding window can grow and shrink based on the localization sensor data that is received. Each time a new global update, Z_t, is received, the criterion is run for all poses in the state space to find the oldest pose that is still updated significantly (quantified in terms of entropy reduction) by the new sensor data. The time of this oldest pose is referred to as the sliding window time t_{sw} and and poses from before the sliding window time are marginalized out and no longer maintained in the filter.

The adaptive sliding window criterion is expressed as:

$$t_{sw} = \underset{t' \in [0,t]}{\operatorname{argmin}} \{ H(\mathbf{X}_{t'}) - H(\mathbf{X}_{t'} | \mathbf{Z}_t) \} > \epsilon \tag{4.25}$$

where $H(X)$ is the Shannon entropy equation defined in (2.6) and ϵ is a tunable threshold that will determine the adaptive window size.

In the common case that the states are being estimated as Gaussian distributions, then the relationship between the Shannon and Fisher informations given in (2.17) can be used to calculate the Shannon entropies in (4.25).

After performing the sliding window criterion (4.25), the probabilistic coverage map at the sliding window time, $\mathbf{W}_{t_{sw}} \triangleq [W^1_{t_{sw}}, W^2_{t_{sw}}, ..., W^N_{t_{sw}}]$ is stored. On subsequent updates, there is no need to calculate the coverage map based on the entire trajectory but rather only for the parts of the trajectory that still in the state space (i.e. the ones that are inside the sliding window). Finally, all poses from times before t_{sw} can be marginalized out.

4.4 Application to Autonomous Underwater Vehicle Seabed Surveying

Localization error cannot be considered as negligible in most AUV seabed surveys since the AUV has no access to a global position reference when the vehicle is submerged. The only known paper that considers the robot pose uncertainty when performing seabed coverage is [176]. In this case, the pose distribution is simply convolved with the sensor characteristic. Such an approach is equivalent to maintaining only the mean values of the coverage of the coverage map variables proposed here. On the first pass, these mean values will be correct, but on subsequent passes, the coverage will be under estimated because the max function does not distribute across the expectation operation. This will be elaborated upon further in Section 4.4.4.

4.4.1 Overview of Approach

The framework described in Section 4.2 and 4.3 is applied to estimate the coverage distribution over an area of seabed.

When the AUV is submerged it "dead reckons" using velocity estimates from the DVL and orientation estimates from the compass. The motion model is given by the set of equations (2.18). In this mode we perform the iterative coverage estimation process described by Algorithm 9.

At some interval, such as when the robot surfaces for a GPS fix or receives an acoustic localization update, the sliding window criterion (4.25) is applied and the coverage map is regenerated from the sliding window time as described in Section 4.3.

4.4.2 Propagating AUV Location Uncertainty to Coverage Distribution

Since the SSS can only generate useful mosaicked data when the AUV is in rectilinear motion, the following simplifying assumption is proposed:

Assumption 4.4.1. *Only the cross-track uncertainty in the AUV location is considered as it will have a larger affect on the coverage uncertainty in the case that the AUV is moving in a straight line.*

This assumption is applied to simplify the formulations here, however, an investigation of the impact of this assumption would be an interesting avenue of future work.

As a result, the pose distribution is projected onto the line orthogonal to the direction of AUV motion as shown in Fig. 4.9.

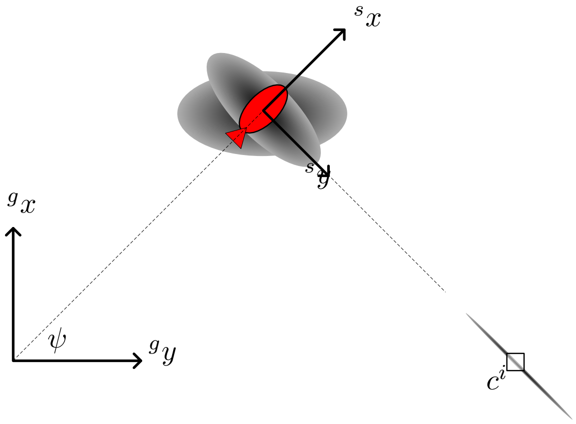


Figure 4.9: Projection of 2D position distribution onto 1D line orthogonal to the direction of AUV motion

It is also assumed here that DVL, GPS, and compass noise are normally distributed (for justification see [224]) so that the robot pose can be approximated as following a Gaussian distribution. In this case the 2D position is estimated as a bivariate Gaussian distribution of the RVs (X_t, Y_t):

$$\begin{aligned} bel(X_t = x, Y_t = y) &\sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{X}_t}, \boldsymbol{\Sigma}_{\mathbf{X}_t}) \\ =& \frac{1}{2\pi\sigma_{x_t x_t}\sigma_{y_t y_t}\sqrt{1-\rho_t^2}} \exp(-\frac{1}{2(1-\rho_t^2)}[\frac{(x-\mu_{x_t})^2}{\sigma_{x_t x_t}^2} \\ &+ \frac{(\mathrm{y}-\mu_{y_t})^2}{\sigma_{y_t y_t}^2} - \frac{2\rho_t(x-\mu_{x_t})(y-\mu_{y_t})}{\sigma_{x_t x_t}\sigma_{y_t y_t}}]) \end{aligned} \tag{4.26}$$

where $\boldsymbol{\mu}_{\mathbf{X}_t} = [\mu_{x_t}, \mu_{y_t}]^T$ and

$$\boldsymbol{\Sigma}_{\mathbf{X}_t} = \begin{bmatrix} \sigma_{x_t x_t}^2 & \rho_t\sigma_{x_t x_t}\sigma_{y_t y_t} \\ \rho_t\sigma_{x_t x_t}\sigma_{y_t y_t} & \sigma_{y_t y_t}^2 \end{bmatrix} \tag{4.27}$$

and ρ_t is the bivariate correlation coefficient [168]:

$$\rho_t = \frac{E[(\mathbf{X}_t - \mu_{x_t})(\mathbf{Y}_t - \mu_{y_t})]}{\sigma_{x_t x_t}\sigma_{y_t y_t}} \tag{4.28}$$

The cell location in the global frame is transformed through ${}^s_g T_t$ into a distribution in the sensor frame $p({}^s\mathbf{C}_t^i = {}^s\mathbf{c}_t^i)$ and then the along-track $({}^s x_t^i)$ uncertainty is marginalized out to yield a distribution for cell i in the cross-track direction: $p({}^sY_t^i = {}^s y_t^i) \sim \mathcal{N}(\mu_{{}^s y_t^i}, \sigma_{{}^s y_t^i {}^s y_t^i})$ with:

$$\mu_{{}^s y_t^i} = (x^i - \mu_{x_t})\sin\psi_t + (y^i - \mu_{y_t})\cos\psi_t, \tag{4.29}$$

and:

$$\sigma_{{}^s y_t^i {}^s y_t^i} = \sigma_{x_t x_t}^2 \sin^2\psi_t + \rho_t\sigma_{x_t x_t}\sigma_{y_t y_t}\sin 2\psi_t + \sigma_{y_t y_t}^2\cos^2\psi_t. \tag{4.30}$$

(for derivation see Appendix C).

The coverage sensor model, $\mathcal{H}$, originally defined in (4.2) is now given by:

$$\breve{W}_t^i = \mathcal{H}({}^s\mathbf{C}_t^i) = P({}^sY_t^i) \tag{4.31}$$

where only the cells that are perpendicular to the AUV pose at time t are in the sensor swath, and those cells are updated by mapping the orthogonal distance from the cell to the AUV track through the sonar lateral range curve $P(y)$ based on (4.14).

Fig. 4.10 shows how the distribution of $\breve{W}_t^i$ is generated by mapping an uncertain cell location through the sensor characteristic. The distribution of the orthogonal distance of the cell from the sensor is shown in red (bottom right). The sonar coverage sensor model is shown in blue (top right). The cell location distribution is mapped through the coverage sensor model to obtain the coverage distribution resulting from this one single measurement (top left).

Finally, a lower bound on the coverage can be obtained recursively by applying the max operation on RVs defined in (4.22) to determine the coverage distribution W_t^i.

4.4.3 Sliding Window Filter Estimation

The coverage distributions are recomputed from the sliding window time whenever a global update is received. In the implementation, this takes place when the vehicle surfaces for a GPS reading or receives an acoustic update from a stationary or moving beacon.

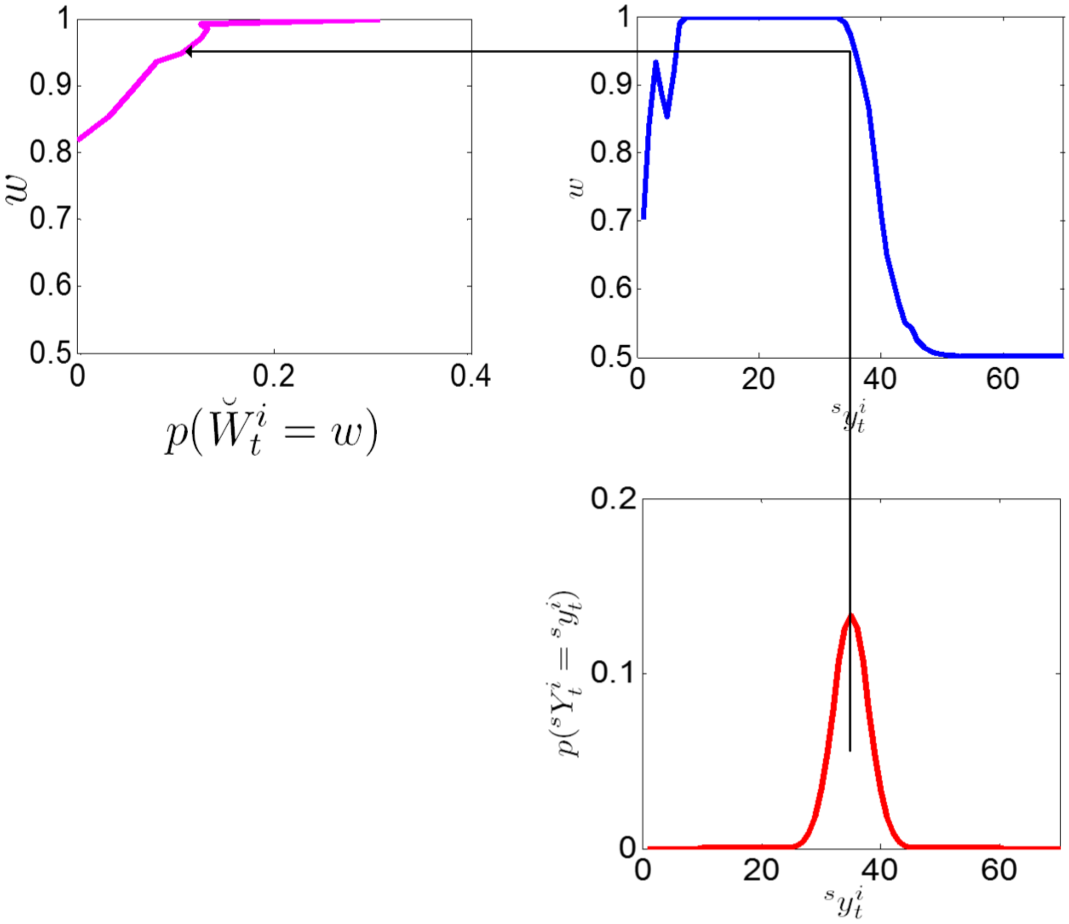


Figure 4.10: The distribution of the cell in the sensor frame (bottom right) is mapped through the $P(y)$ curve (top right) to obtain a coverage distribution (top left).

Example 4.4.1. *The purpose of this example is to further clarify the effect of smoothing on the confidence distribution. In this simulation, the robot is made to follow a pre-defined path. In addition, the vehicle is diving and then surfacing twice, only having access to GPS when it is at the surface. The path of the robot is shown in Fig. 4.11 (top) with five time steps* $t_{1:5}$. *One point is highlighted in red, defined here as cell* $\mathbf{c}^i$. *The coverage distribution of* $\mathbf{c}^i$, W^i_t *is shown for the five time steps,* $t = \{t_1, t_2, t_3, t_4, t_5\}$. *in Fig. 4.11 (bottom). Time* t_1 *is the start of the simulation, time* t_2 *is the time when cell* $\mathbf{c}^i$ *is first observed,* t_3 *is the time when the robot first surfaces, at time* t_4 $\mathbf{c}^i$ *is observed a second time, and then the robot finally surfaces again at* t_5. *For clarity of the example, the sonar performance is assumed constant.*

Several points are important to note from Fig. 4.11. Note that the distribution of $W^i_{t_3}$ *has reduced entropy compared to time* t_2 *because the robot has surfaced and the trajectory has been re-estimated based on the GPS reading as is noted by looking at the distributions in Fig. 4.11(bottom). When* $\mathbf{c}^i$ *is passed a second time, the two looks producing* $\breve{W}^i_{t_2}$ *and* $\breve{W}^i_{t_4}$ *are combined resulting in* $W^i_{t_4}$. *The coverage distribution is once again recalculated at* t_5 *when the vehicle surfaces again to give the final distribution that shows an extremely high total probability of coverage.*

4.4.4 Path Planning Within the Probabilistic Framework

In order to fully exploit the benefits of the richer description of coverage, path planners must be developed within the framework so that mission objectives can be achieved efficiently.

Past approaches to AUV seabed surveying have planned structured paths offline and assumed that the tracks are precisely followed. A summary of some of the approaches was presented in Chapter 4.4.4.

In the previous chapter we presented a sensor-driven approach to adaptive coverage planning. Recall from the Bayes' network in Fig. 3.2 we arrived at the expression (3.4) for the joint distribution of confidence at cell $\mathbf{c}^i$ from one sensor reading $\breve{T}^i_t$, the environmental parameters, $\boldsymbol{\mathcal{E}}^i_t$, and the vehicle pose $\mathbf{X}_t$:

$$\begin{aligned}&p(\breve{T}^i_t = 1, \boldsymbol{\mathcal{E}}^i_t = \varepsilon^i_t, \mathbf{X}_t = \mathbf{x}_t) = \\ &p(\breve{T}^i_t = 1 | \boldsymbol{\mathcal{E}}^i_t = \varepsilon^i_t, \mathbf{X}_t = \mathbf{x}_t) p(\boldsymbol{\mathcal{E}}^i_t = \varepsilon^i_t) p(\mathbf{X}_t = \mathbf{x}_t)\end{aligned} \tag{4.32}$$

In Chapter 3, we made the assumption that the pose was certain and $p(\mathbf{X}_t = \mathbf{x}_t) = 1$. Here, this assumption is removed. For simplicity of formulation, we assume here that the environmental parameters $\boldsymbol{\mathcal{E}}^i_t$ are known ($p(\boldsymbol{\mathcal{E}}^i_t = \varepsilon^i_t) = 1$) however the extension to the case where both the environmental parameters and pose are uncertain is straightforward.

Similar to (3.8), we can evaluate the confidence resulting from a single sonar sensor reading at time t for cell $\mathbf{c}^i$ by marginalizing out the pose:

$$\begin{aligned} p(\breve{T}^i_t = 1) &= \sum_{\mathbf{x}_t} p(\breve{T}^i_t = 1, \boldsymbol{\mathcal{E}}^i_t = \varepsilon^i_t, \mathbf{X}_t = \mathbf{x}_t) \\ &= \sum_{\mathbf{x}_t} p(\breve{T}^i_t = 1 | \boldsymbol{\mathcal{E}}^i_t = \varepsilon^i_t, \mathbf{X}_t = \mathbf{x}_t) p(\mathbf{X}_t = \mathbf{x}_t) \\ &= E_{\mathbf{X}_t}[p(\breve{T}^i_t = 1 | \boldsymbol{\mathcal{E}}^i_t = \varepsilon^i_t, \mathbf{X}_t = \mathbf{x}_t)] \\ &= \begin{cases} E_{{}^sY^i_t}[P_{\varepsilon^i_t}({}^sY^i_t)] & \text{if cell } c^i \in \mathcal{S}'_t \\ 0.5 & \text{otherwise} \end{cases} \end{aligned} \tag{4.33}$$

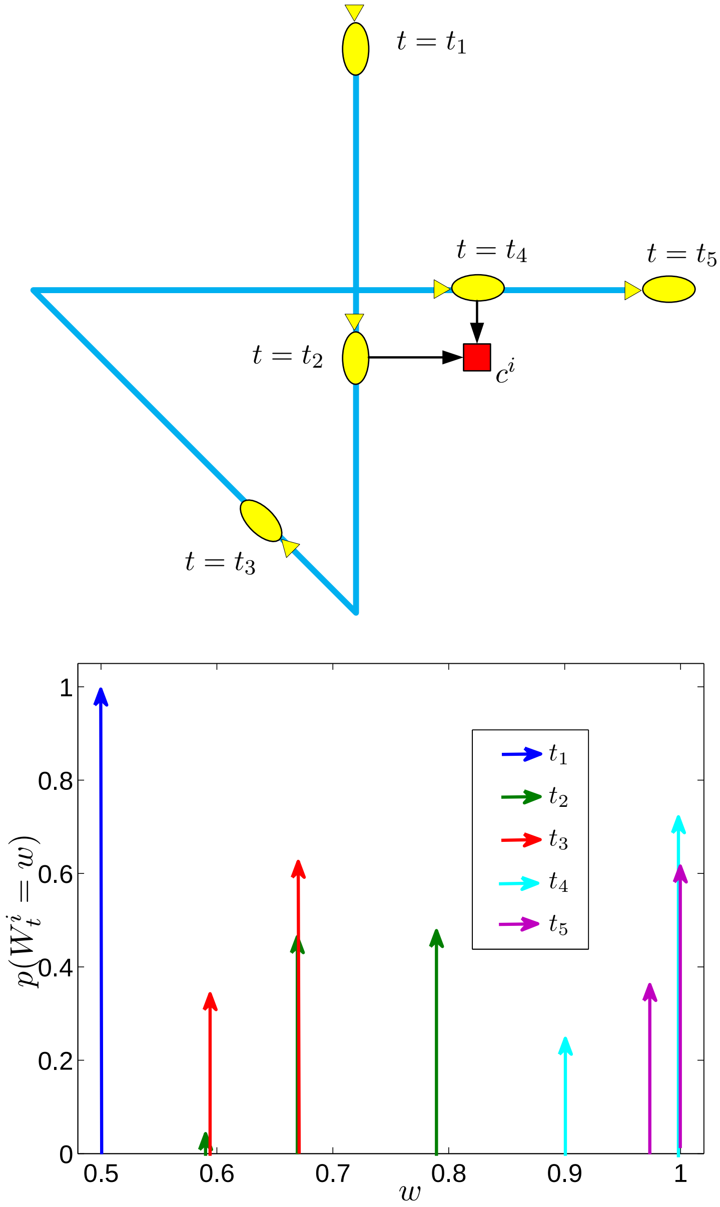


Figure 4.11: Coverage distributions at five time steps. **(top)** Trajectory of AUV with five time steps shown. **(bottom)** The coverage distribution of $\mathbf{c}^i$ for five time steps. The vehicle begins at time t_1, first passes the cell of interest at t_2, surfaces for GPS at t_3, passes the cell a second time at t_4 and finally surfaces again at t_5.

where the sonar lateral range curve is a specific representation of the general coverage sensor function $\mathcal{H}$ defined in (4.2). So (4.33) can be rewritten as:

$$E_{{}^sY_t^i}[P_{\boldsymbol{\mathcal{E}}_t^i}({}^sY_t^i)] = E_{{}^s\mathbf{C}_t^i}[\mathcal{H}({}^s\mathbf{C}_t^i)] = E[\breve{W}_t^i]. \tag{4.34}$$

So the confidence from one look $p(\breve{T}_t^i = 1)$ in the case of the AUV seabed coverage problem is evaluated as the mean of the coverage distribution from the probabilistic framework. For the cells that are in the sensor swath:

$$p(\breve{T}_t^i = 1) = E[\breve{W}_t^i] \tag{4.35}$$

Similarly, the final confidence after combining n looks can be expressed as:

$$p(T_t^i = 1) = E[W_t^i] = E[\max\{\breve{W}_{t_1}^i, \breve{W}_{t_2}^i, ..., \breve{W}_{t_n}^i\}] \tag{4.36}$$

where the necessity to maintain the entire coverage distributions as opposed to just the means is highlighted by the fact that the expectation operator, E cannot distribute across the max operator: $E[\max\{\breve{W}_{t_1}^i, \breve{W}_{t_2}^i\} \neq \max\{E[\breve{W}_{t_1}^i], E[\breve{W}_{t_2}^i]\}$.

Now the sensor-driven path planning approaches presented in the previous chapter over the confidence variables $\mathbf{T}_t$ can be applied within the probabilistic framework. Simulation results are shown in Fig. 4.14.

However, an additional, more structured approach to path planning within the probabilistic framework based on information is also presented in the following section.

4.4.4.1 Structured Approach

In this approach, the location of the next track is optimized at the end of each track. The placement of the next track to be followed should maximize the information to be gained by following the track subject to the constraint that no gaps are left.

One proposed method to define structured paths within the probabilistic framework is described by Fig. 4.12.

The optimization is over the domain of d subject to the constraint that the area A satisfy mission completion. Define the track as τ_d and parameterized by s:

$$\begin{aligned} \tau_d : [0,1] &\rightarrow Q_{free}, s \rightarrow \tau_d(s) \\ \tau_d(0) &= (d, x_{min}) \\ \tau_d(1) &= (d, x_{max}) \end{aligned} \tag{4.37}$$

where x_{min} and x_{max} are the minimum and maximum x values of the track. Note that this approach easily extends to surveys at any angle but for simplicity a vertical track formulation is shown.

In this case we selected (4.24) as the criterion for mission completion.

So the optimization now becomes to select the value of d that will maximize the information gained similar to (3.21):

$$B(d) \triangleq \int_0^1 \Delta H(\mathbf{T}_t | \boldsymbol{\mathcal{E}}_t^{1:N}, \tau_d(s))ds \tag{4.38}$$

subject to the constraint:

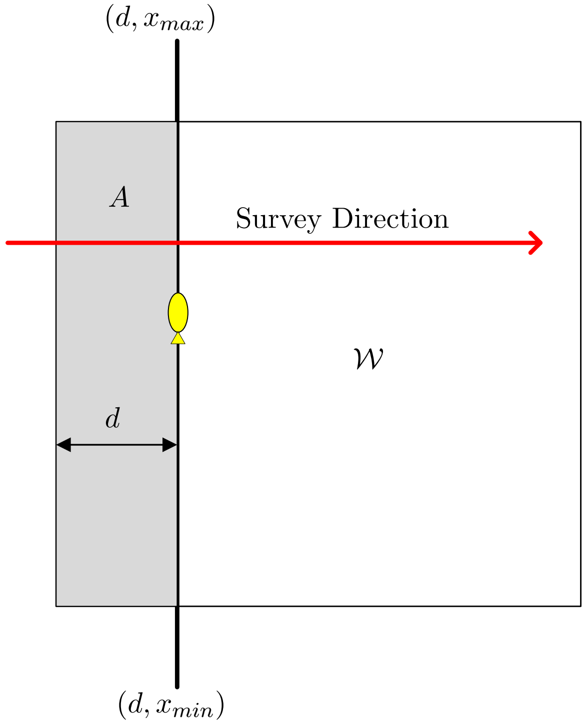


Figure 4.12: The value of d should be optimized such that the portion of the map defined by A satisfies the mission completion criterion

$$\frac{1}{A}\sum_{i:\mathbf{c}^i \in A}^{N} E[W_t^i] > w_{avg} \tag{4.39}$$

The advantage of this approach over the unstructured one is that it will produce straight transects across the workspace. The disadvantage is that it could be more inefficient for more complex shaped workspaces.

4.5 Experimental Setup

The methods are tested in a simulation and in water trials similarly to the previous chapter. The adapted backseat MOOS community is shown in Fig. 4.13. The "Adaptive Sliding Window Filter" application uses raw sensor data from the frontseat to estimate the state and trajectory. The "Probabilistic Coverage Map" is an enhanced version of the "Coverage Map" application described in Chapter 3 where now coverage values are probabilistic. The "Behaviours" are similar to the ones described in Chapter 3 but enhanced to operate within the new probabilistic framework. All other applications are left unchanged and are part of the open-source distribution of MOOS-IvP.

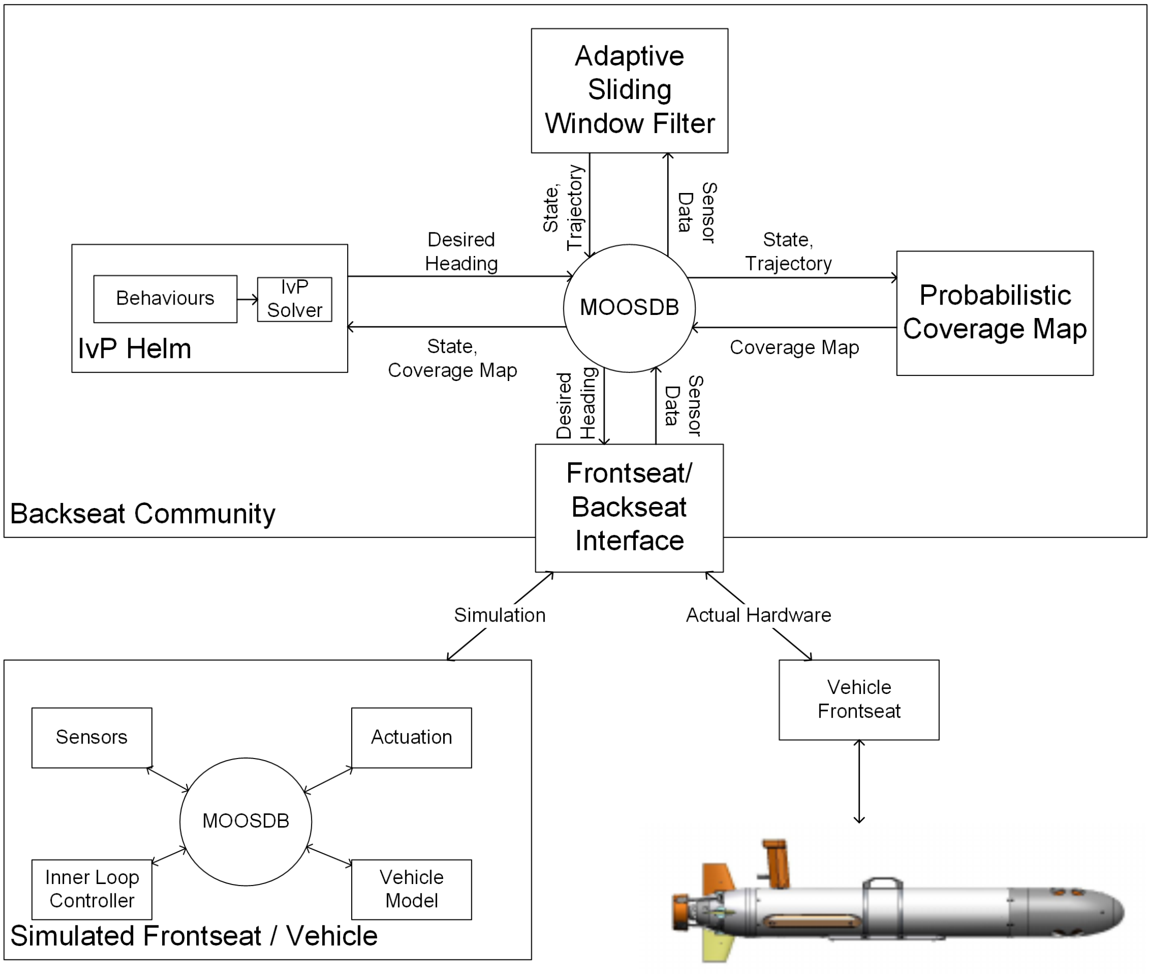


Figure 4.13: MOOS communities for simulation and actual vehicle implementation. State estimation is now performed on the backseat and raw sensor data is passed from the frontseat.

4.6 Results

4.6.1 Simulation

The simulation is used to validate the proposed approach. The results from the unstructured planner are shown in Fig. 4.14. In the figure, the vehicle surfaces for a GPS fix eight times, and each time it surfaces the sliding window criterion is applied. The portion of the trajectory that is being maintained in the sliding window filter after each of the updates is shown in red.

The final total smoothed path and actual path from the simulator are shown in Fig. 4.15. The mean probability of coverage for the smoothed planner is shown in Fig. 4.16 yielding an expected coverage of 98%. The actual coverage is shown in Fig. 4.17 and is 98%.

4.6.2 In-Water Trials

Many field experiments have been performed to test the proposed method. One such example is shown in Fig. 4.18 and Fig. 4.19. This experiment was performed on the same workspace that was shown in Fig. 4.1 and Fig. 4.8, which is a rectangular box approximately 200m by 300m. The vehicle runs on the surface but the GPS data is not incorporated into the state estimate unless the vehicle is at the beginning or end of a track (when a GPS fix might normally occur). Each time a track is completed, the optimization (4.38) is re-run until convergence. In some cases acquiring the GPS fix and computing the optimal track takes some time, so the vehicle executes a loiter pattern while this is ongoing.

In this case, the criterion for mission completion is given by:

$$\frac{1}{N}\sum_{i=1}^{N} E[W_t^i] > 0.985 \tag{4.40}$$

or mean average probability of 98.5%.

In Fig. 4.18, the yellow box represents the workspace, the red plot represents the achieved AUV position from the GPS logged data, and the blue represents the trajectory that is currently being estimated in the sliding window filter. Each time a new GPS update is received, the sliding window criterion is run and old poses are eliminated.

The final trajectory and actual coverage map from the GPS data are shown in Fig. 4.19. The actual average coverage in this case was 98.9% when the mission was terminated. It is important to note in this case is that even though the ability of the AUV to follow the tracks is extremely poor, it is still able to achieve its coverage objective.

4.7 Summary

In this chapter we form an an explicit link between area coverage and state estimation for a general robotics area coverage problem. The framework is then applied specifically to the AUV seabed coverage problem.

We have presented a probabilistic framework within which paths can be generated that will guarantee coverage based on new probabilistic coverage criteria, notwithstanding the uncertainty in vehicle pose. In the proposed method, the pose belief is used to generate a coverage distribution that is maintained as the vehicle navigates the workspace. It is also shown that trajectory estimation should be used since minimizing the uncertainty over the

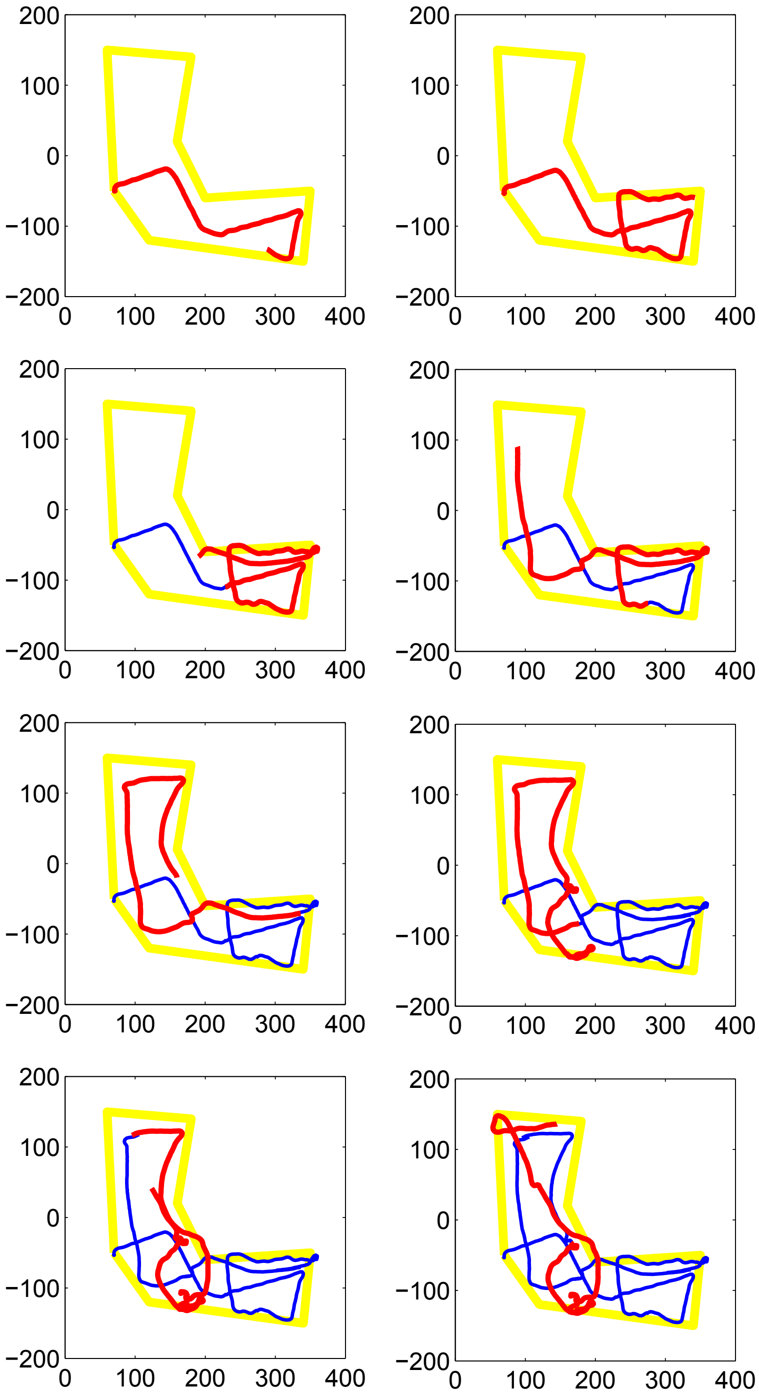


Figure 4.14: The eight plots show the state of the system after each GPS update **yellow:** The area to be covered. **red:** The portion of the trajectory that is being estimated by the adaptive sliding window filter. **blue:** The portion of the trajectory that has been smoothed but is no longer being updated. Figures progress from left to right and from top to bottom with time. Units are in metres and sensor characteristic is considered fixed and is the lateral range curve for cobble seabed and 10m depth shown in Fig. 2.14.

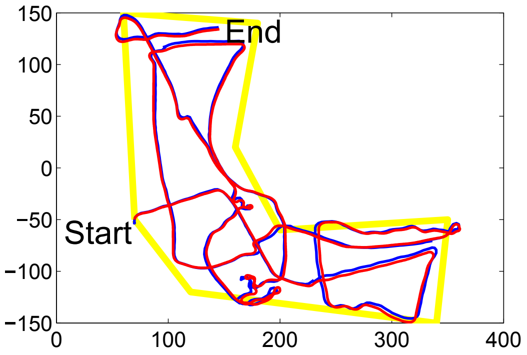


Figure 4.15: The achieved (red) and smoothed (blue) trajectories at mission completion. The smoothing approach yields a better estimate of the entire robot trajectory than a standard filtering approach. Units in the plot are in metres.

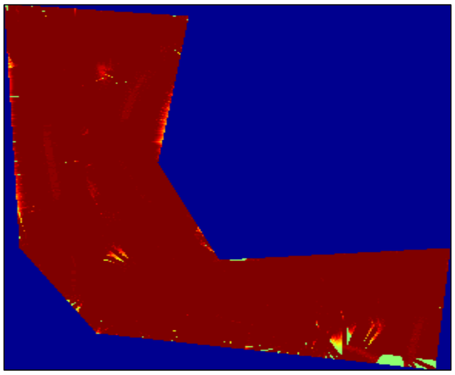


Figure 4.16: Smoothed coverage estimate (Mean probability of coverage = 98%)

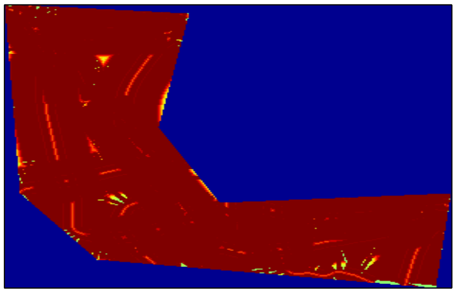


Figure 4.17: Achieved coverage (98%)

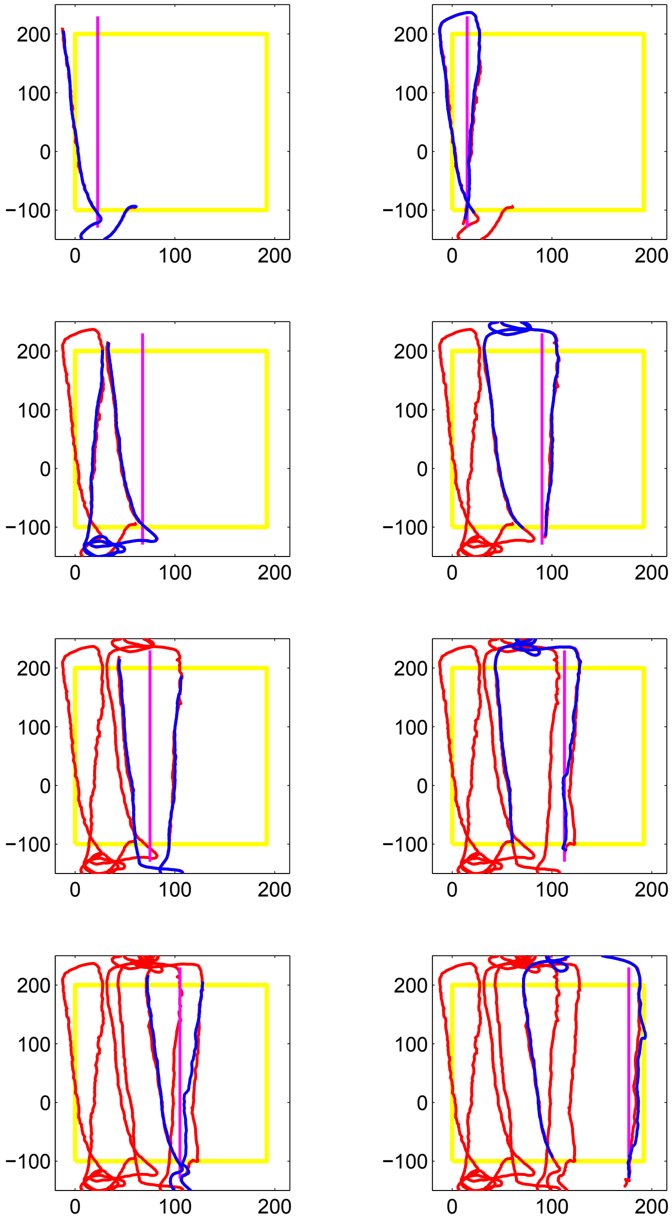


Figure 4.18: The eight plots show the state of the system after GPS updates are obtained nearing the end of each track. Time progresses from left to right and top to bottom. Units are in metres. The sensor characteristic used is the 10m depth cobble seabed shown in Fig. 2.14. In each plot: **yellow:** The workspace to be covered. **pink:** The track that has just been followed. **red:** The achieved trajectory that has been followed to that point based on the GPS data. **blue:** The portion of the trajectory that is being estimated by the sliding window filter. Each time a track is finished the location of the next track is optimized based on (4.38) so that the workspace is guaranteed to be covered. The AUV is able to cover the area to 98.8% even though it is drifting more than 20m over a 200m track.

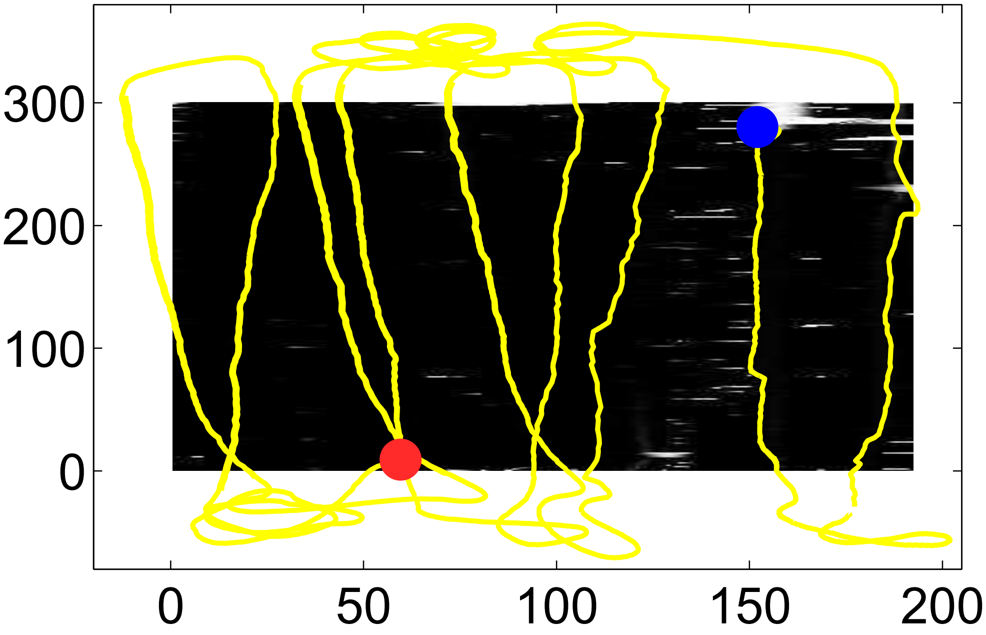


Figure 4.19: The final estimated trajectory overlayed on the coverage map (darker area indicates higher coverage). Path starts at red dot and ends at blue dot. Final path length is 4959 metres.

entire trajectory maximizes the effective coverage in most cases. To maintain scalability of computation with time, a sliding window filter approach is utilized. A criterion is defined for which old poses can be removed from the state space based when their information gain resulting from global position updates is sufficiently low. Upon removal, the coverage map at the time of the oldest pose still being estimated is stored and used as the starting point for subsequent updates.

The method is then applied to the autonomous seabed coverage where it is shown that with this adaptive approach we can guarantee that the seabed coverage mission objective is achieved even in cases of very poor ded reckoning. In the case of a safety critical application such as MCM, it is crucial that we have the most accurate estimate possible of the confidence of coverage of the workspace to prevent the possibility of missed mines.

Chapter 5

Cooperative AUV Trajectory Estimation

5.1 Introduction

Using multiple survey robots has many potential benefits, such as increased efficiency and robustness. In the absence of any coordination or cooperation, a team of robots could cover an area of seabed more quickly by partitioning the workspace into smaller areas with each covering a smaller piece. An additional possible benefit of using multiple robots is that, in the case that they are able to communicate and make relative measurements of each other's range, localization can be improved [192]. However, operating with multiple robots does come with the added overhead required to deploy, control and recover the extra platforms, so it is important to maximize the potential benefits.

In the previous chapter the link was made between coverage and state estimation and it was motivated that full trajectories should be estimated rather than just present states. In this chapter, cooperation amongst AUVs is exploited for trajectory estimation to increase the efficiency of coverage for a multi-AUV survey team.

It has been shown theoretically and experimentally that a team of agents without access to a global location reference can localize themselves more accurately if they can make measurements of range relative to one another and communicate those measurements [192]. This is referred to in the literature as cooperative localization, for a review see Chapter 2.4.5.2.

In the underwater scenario, AUVs can make range measurements relative to each other using the TOF of acoustic transmissions in water and the known speed of sound in water (~ 1500m/s):

$$\text{relative range} = \text{time of flight} \times \text{speed of sound}, \tag{5.1}$$

and these relative range measurements can be used to perform cooperative localization. Performing cooperative localization in an underwater environment is challenging because inter-vehicle communications are challenging. Higher frequency signals attenuate rapidly with range in water, so for robots to communicate any appreciable distance requires using acoustic signals. The amount of raw data to be shared amongst robots in order to recover a consistent centralized multi-robot state estimate incorporating the inter-robot measurements is large and the communications bandwidth on the acoustic channel is limited.

A depiction of the AUV cooperative localization setup is abstractly shown in Fig. 5.1. Bandwidth is so limited that robots must take turns transmitting data at the same carrier frequency. In the figure, AUV 1 transmits at time t_1, AUV 2 transmits at some later time t_2 and then finally AUV 3 transmits at time t_3. If the AUVs are equipped with precisely synchronized clocks (for example by synchronizing to the GPS time signal at the surface before submerging) and transmit at known times, then, upon reception of an acoustic data packet, the receiver can also calculate its range relative to the sender. Each reception of a data packet combined with the associated relative range measurement allows the receiving robot to reduce its position uncertainty in the direction relative to the sender.

In AUV cooperative localization, the job of the system designer (and the topic of this chapter) is how to most effectively select information to be transmitted to other AUVs to achieve the cooperative localization objective.

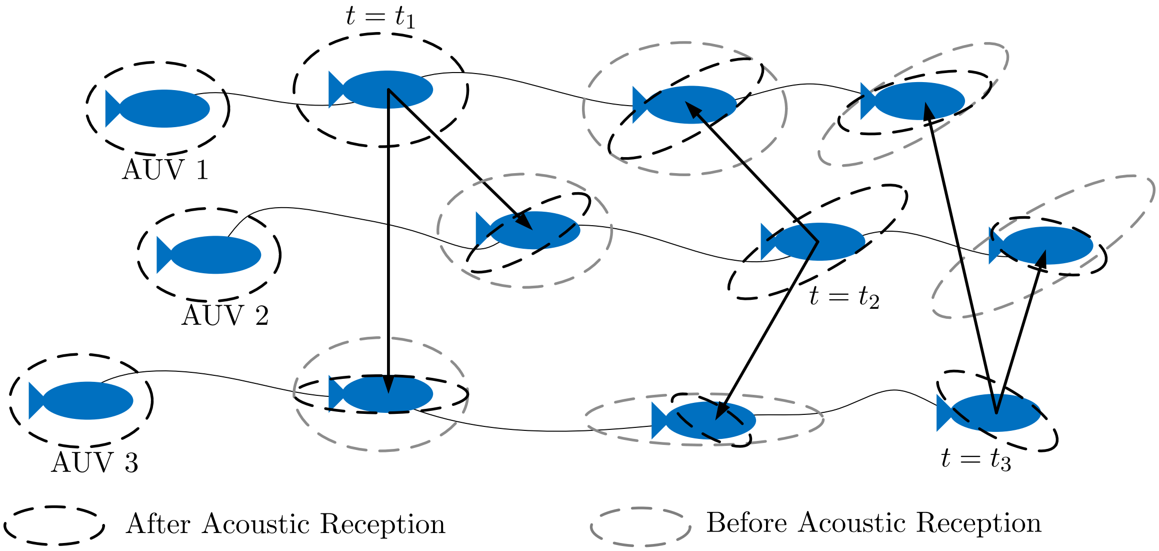


Figure 5.1: Depiction of a three AUV team performing cooperative localization. Each AUV in the team takes a turn to make an acoustic transmission (dark arrows). As robots move, their position uncertainty (indicated by dashed ellipses) grows. Upon reception of an acoustic transmission the receiver can calculate the TOF of the acoustic signal from the source and use knowledge of the sender's location (which was contained in the data packet itself) to reduce its position uncertainty in the direction of the sender (gray ellipse to dark ellipse).

5.1.1 Problem Statement

It is desirable that a team of AUVs use their ability to communicate and make relative range measurements to improve their trajectory estimates while performing a seabed survey mission.

It is infeasible to transmit all raw sensor data through the limited and unreliable underwater acoustic channel. The challenge is, therefore, to devise a method in which each AUV can use the relative range measurements of other members of the team to improve own-vehicle localization, without becoming overconfident about localization accuracy, using the underwater acoustic communications channel.

5.1.1.1 Acoustic Modem

Additional to the on-board sensor suite described in Table 3.2, each robot is equipped with an acoustic modem that is capable of sending and receiving acoustic data packets ($\sim$ 192 byte packet every 10s). The acoustic modems have precise clocks that can be synchronized on the surface before submersion using the GPS time signatures. These clocks drift on the order of one ms over a 14 hour period [224], resulting in about a 1.5m error range. For surveys that are significantly shorter than 14 hours the drift in the synchronized clocks can be considered negligible. Transmitting robots initiate communications precisely at the start of a second and receiving robots can calculate the inter-robot range measurement based on TOF which is calculated as the fractional portion of the second on arrival.

5.1.1.2 Restrictions of Acoustic Communications

Communicating through water using acoustics imposes several constraints that do not exist in radio-frequency communications. While detailed acoustic communications theory is outside of the scope of this thesis, a brief description is given here on the aspects of communicating though water that make it particularly challenging:

1. Reduced bandwidth: Bandwidth is so limited that only one node (robot) can transmit at a time. As a result, a TDMA medium access control (MAC) protocol is used so that the channel may be shared. Robots are allocated, for example, 10s slots in a cycle that repeats. In this work, the slots in the TDMA cycle are assigned before the mission, but other MAC schemes are possible that would allow for robots to dynamically join and leave the network.

2. Reduced throughput: Allowable data packet size for one transmission is 192 bytes (rate #1 of the Woods Hole Oceanographic Institute micromodem [83]).

3. Unacknowledged: Acknowledgment of individual packets is generally prohibitive. Adding new nodes or robots to the system increases the duration of the TDMA cycle. If nodes acknowledge individual packets, then the length of the TDMA cycle scales $\sim \mathcal{O}(N^2)$ where N is the number of nodes, which quickly becomes unwieldy even for small team sizes. If communications are broadcast without acknowledgment, or if acknowledgments are included in the normally scheduled transmissions, then the length of the TDMA cycle scales as only $\sim \mathcal{O}(N)$.

4. Low Reliability: In field tests we have experienced packet drop rates from 20-50% depending on conditions, relative orientation of sender and receiver, ambient noise, etc. AUV cooperative localization algorithms should make no assumptions about the success rate of the acoustic communications.

The challenges listed above are certainly severe, but acoustic communications does come with a few benefits: 1) The fact that data packets travel at the speed of sound in water ($\sim$ 1500m/s) makes accurate ranging possible based on the TOF and 2) Communications are broadcasted so a transmission from any node in the network could be received by all other nodes within acoustic range.

5.1.1.3 Inter-Robot Range Measurements Through Time-of-Flight of Acoustic Data Packets

For a group of N robots, TDMA cycle c and slot duration in the cycle of T seconds, a transmission will be initiated at time $t = kT \triangleq t_k$ where k is an integer indexing the transmission number. If we assume that the first transmission is made by robot 1 in cycle $c = 0$, then the transmission at time kT will be made by robot $k - Nc$. Each broadcast transmission has the potential to result in $N - 1$ range measurements. For example, if a range measurement of robot i by robot j at time t_k is denoted by $r_{t_k}^{i,j}$, a transmission by robot i at time t_k will result in the $N - 1$ range measurements $r_{t_k}^{i,j}, j = 1..N, j \neq i$ if all transmissions are successfully received.

The distinguishing aspect of underwater cooperative localization is that the inter-robot measurements (ranges) are a result of the inter-robot communications and therefore ranging and communications happen concurrently. This differentiates it from cooperative localization performed above-water where measurements and communications are not concurrent.

5.1.1.4 Selecting the Right Information to Transmit

Cooperative localization in the presence of perfect communications amounts to a distributed sensor fusion problem and can be solved using a centralized filtering [192] or smoothing [159] approach.

In the general cooperative localization literature, the underwater domain is often cited as a potential application space. However, the vast majority of algorithms for above-water cooperative localization in the literature make one or more assumptions that violate the constraints of the underwater acoustic channel listed above.

For AUV cooperative localization, there is simply too much data compared to the bandwidth available to send everything. The essential aspect of any AUV cooperative localization algorithm is how this problem is overcome.

5.1.2 Objectives and Significance

In this work, we assume that the team of AUVs are homogeneous, meaning that all of them are equipped the same suite of sensors. The overall objective is to utilize cooperative localization amongst this homogeneous team of AUVs to estimate the AUV trajectories and consequently achieve the seabed coverage mission as efficiently as possible.

Cooperating AUVs will have a reduced rate of position uncertainty growth while submerged due to their ability to cooperatively localize. This is beneficial from a seabed coverage standpoint for two main reasons:

1. AUVs can surface less for GPS fixes and thus have more efficient coverage missions and reduce energy consumption.

2. AUV localization error will be reduced over the entire trajectory, resulting in more rapid convergence towards coverage, based on the probabilistic coverage framework presented in the previous chapter.

After surfacing for GPS position fixes or receiving acoustic packets, the entire trajectory is re-optimized to provide the best estimate of the coverage over the workspace in a manner similar to the approach in Chapter 4 but extended to multiple AUVs. In the absence of communications failures the proposed algorithm provides optimal estimates, but in some cases communication failures will result in loss of optimality without significant impacts.

The area to be covered should be equally partitioned amongst the robots. Although it is possible to use the state updates from other robots to update the local coverage map, this coverage map would only be, at best, an approximation of the actual global coverage since state updates are only intermittent.

5.1.2.1 Localization Overconfidence

A consideration in any cooperative localization algorithm design is avoiding overconfidence in the location estimate. As described in [191], if cross correlations induced by the estimation of one robot position relative to another are not considered, the result will be an inconsistent or overconfident estimate (i.e. the estimated covariance will be less than the actual or optimal covariance).

Given that accurate localization is a critical requirement for successful MCM, overconfidence in the state estimate should be avoided. It leads to an overly optimistic representation of the coverage achieved over the workspace, making the real risk associated with the area higher than estimated.

5.1.3 Overview and Contribution

Cooperative localization is formulated here as a least-squares problem and an algorithm is proposed that operates within this framework to estimate the entire trajectory of AUVs.

The contribution of the chapter can be summarized as follows:

1. Formulation of cooperative localization as a least-squares optimization problem in Section 5.2.

2. An algorithm that provides near-optimal estimates of entire AUV trajectories onboard each robot with a communication throughput that scales linearly with the number of robots in the team and constantly with time. This is the first known algorithm where packet size scales linearly with the number of robots in the team and is not affected by communications failures.

The benefit of the proposed approach for AUV seabed surveying is that robots will have to surface less often resulting in AUVs being able to spend a higher percentage of their time in water actually performing the survey. In addition, the smoothing approach is beneficial since, as was shown in the previous chapter, optimizing the entire trajectory allows for more rapid convergence towards coverage meaning that mission completion is achieved more rapidly.

5.1.4 Chapter Nomenclature

A table showing the nomenclature used in the rest of the chapter is presented in Table 5.1.

5.1.5 Chapter Outline

The remainder of the chapter will be organized as follows: in Section 5.2 AUV cooperative trajectory estimation is formulated as a least-squares optimization problem. In Section 5.3 the algorithm for decentralized multi-robot trajectory estimation is presented. In Section

Variable	Description
t	Time index
t_k	Time of the kth acoustic transmission
t_k^i	Time of the kth acoustic transmission either transmitted or received by robot i
c	Number of TDMA cycles that have been completed
N	Total number of AUVs
T	Length of a slot in seconds in the TDMA cycle
$\mathbf{X}_t^i$	RV for state of robot i at time t. Pose given by RVs $[X_t^i, Y_t^i, \Psi_t^i]$
$\mathbf{R}_{t_k}^{i,j}$	RV for range measurement by AUV j of AUV i at time t_k
$\mathbf{U}_t^i$	RV for control of robot i at time t
$\mathbf{Z}_t^i$	Proprioceptive measurement on-board robot i at time t
${}^G\mathbf{Z}_t^i$, ${}^C\mathbf{Z}_t^i$	GPS and compass measurement RVs
V_t^u, V_t^v	Linear velocity readings from the DVL in the bow (u) and starboard (v) directions
$\boldsymbol{\zeta}_t^i$	Additive Gaussian noise associated with process model on robot i at time t with mean 0 and covariance $\mathbf{Q}_t^i$
${}^G\boldsymbol{\delta}_t^i$, ${}^C\boldsymbol{\delta}_t^i$, ${}^R\boldsymbol{\delta}_t^i$	Additive Gaussian noise associated with GPS, compass, and inter-robot range measurements respectively with associated covariances Λ^G, Λ^C, and Λ^R
$f()$	Process function
$h^G()$,$h^C()$, $h^R()$	Measurement functions for GPS, compass, and inter-robot range respectively
ϕ	Factor function in factor graph

Table 5.1: Chapter 5 nomenclature.

5.4 the multi-AUV experimental setup is described. In Section 5.5 results are presented. Finally, in Section 5.6 some general discussion and conclusions are given.

5.2 Cooperative AUV Trajectory Estimation as a Least-Squares Problem

In this section AUV cooperative trajectory estimation is formulated as a least-squares optimization problem. First, cooperative trajectory estimation is presented in two different but related forms: the Bayes' network and the factor graph. The relationship between these representations is outlined and finally a derivation of the least squares formulation for AUV trajectory estimation is obtained which draws on elements of each.

5.2.1 Problem Formulation

We formulate the problem in two dimensions since AUV depth can be accurately measured. Extension to higher dimensions is straightforward. Let the pose of robot i at time t be represented by the array of RVs: $\mathbf{X}_t^i = [X_t^i, Y_t^i, \Psi_t^i]^T$. Each robot generates an estimate of its own pose through control values, $\mathbf{U}_t^i$, and proprioceptive measurements: either GPS, ${}^G\mathbf{Z}_t^i$, or compass ${}^C\mathbf{Z}_t^i$.

Robots communicate with each other using the acoustic modem in a TDMA scheme as described previously. AUV j sends an acoustic transmission at time $t_k \triangleq t_k^j$ and it is received on robot i at time $t_k + \Delta_k^{i,j} \triangleq t_k^i$ where $\Delta_k^{i,j}$ is the time of flight of the acoustic packet initiated by robot j at time t_k^j and received by robot i at time t_k^i. Robot i can estimate its relative range at time t_k^i from j at time t_k^j when the packet was sent. This measurement is represented by the RV $\mathbf{R}_{t_k}^{i,j}$. (It should be noted that in reality the acoustic transmission is sent from point to point in 3D space. The measured range is actually the result of projecting the 3D range onto a 2D plane which requires knowledge of both robots depths since depths can be easily an accurately measured using pressure sensors as described in Section 2.4).

The process model is given by:

$$\mathbf{X_t} = f(\mathbf{X}_{t-1}^i, \mathbf{U}_t^i) + \boldsymbol{\zeta}_t^i \tag{5.2}$$

where the additive noise term, $\boldsymbol{\zeta}_t^i$, is assumed to be Gaussian with mean 0 and covariance $\mathbf{Q}_t^i$.

The GPS measurement model is given by:

$${}^G\mathbf{Z}_t^i = h^G(\mathbf{X}_t^i, {}^G\boldsymbol{\delta}_t^i) \tag{5.3}$$

where ${}^G\boldsymbol{\delta}_t^i \sim \mathcal{N}(0, \mathbf{\Lambda}^G)$ is zero mean Gaussian noise associated with the GPS sensor reading (covariance assumed constant with time and across all robots).

The compass measurement model is given by:

$${}^C\mathbf{Z}_t^i = h^C(\mathbf{X}_t^i, {}^C\boldsymbol{\delta}_t^i) \tag{5.4}$$

where ${}^C\boldsymbol{\delta}_t^i \sim \mathcal{N}(0, \mathbf{\Lambda}^C)$ is zero mean Gaussian noise associated with the compass measurement ${}^CZ_t^i$ (covariance assumed constant with time and across all robots).

The range measurement model is given by:

$$\mathbf{R}_{t_k}^{i,j} = h^R(\mathbf{X}_{t_k^i}^i, \mathbf{X}_{t_k^j}^j) + {}^R\boldsymbol{\delta}_{t_k}^{i,j} \tag{5.5}$$

where ${}^R\boldsymbol{\delta}_{t_k}^{i,j} \sim \mathcal{N}(0, \boldsymbol{\Lambda}^R)$ is zero mean Gaussian noise associated with the inter-robot range measurements, whose covariance is assumed to be constant with time and independent of range, a claim experimentally validated in [224]

For completeness, details of all process and measurement models are included in Appendix D.

The process and measurement equations can be represented as conditional probabilities by collecting all non-noise terms on the left hand side and utilizing the assumption that noise distributions are Gaussian:

$$\begin{aligned}
\mathbf{X}_t^i - f(\mathbf{X}_{t-1}^i, \mathbf{U}_t^i) = \boldsymbol{\zeta}_t^i &\rightarrow p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i, \mathbf{u}_t^i) = \frac{1}{\nu_1}\exp\{-\frac{1}{2}||f(\mathbf{x}_{t-1}^i, \mathbf{u}_t^i) - \mathbf{x}_t^i||_{\mathbf{Q}_t^i}^2\} \\
{}^G\mathbf{Z}_t^i - h^G(\mathbf{X}_t^i) = {}^G\boldsymbol{\delta}_t^i &\rightarrow p({}^G\mathbf{z}_t^i|\mathbf{x}_t^i) = \frac{1}{\nu_2}\exp\{-\frac{1}{2}||h^G(\mathbf{x}_t^i) - {}^G\mathbf{z}_t^i||_{\boldsymbol{\Lambda}^G}^2\} \\
{}^C\mathbf{Z}_t^i - h^C(\mathbf{X}_t^i) = {}^C\boldsymbol{\delta}_t^i &\rightarrow p({}^C\mathbf{z}_t^i|\mathbf{x}_t^i) = \frac{1}{\nu_3}\exp -\{\frac{1}{2}||h^C(\mathbf{x}_t^i) - {}^C\mathbf{z}_t^i||_{\boldsymbol{\Lambda}^C}^2\} \\
\mathbf{R}_{t_k}^{i,j} - h^R(\mathbf{X}_{t_k^i}^i, \mathbf{X}_{t_k^j}^j) = {}^R\boldsymbol{\delta}_{t_k}^{i,j} &\rightarrow p(\mathbf{r}_{t_k}^{i,j}|\mathbf{x}_{t_k^i}^i, \mathbf{x}_{t_k^j}^j) \\
&= \frac{1}{\nu_4}\exp\{-\frac{1}{2}||h^R(\mathbf{x}_{t_k^i}^i, \mathbf{x}_{t_k^j}^j) - \mathbf{r}_{t_k}^{i,j}||_{\boldsymbol{\Lambda}^R}^2\}
\end{aligned} \tag{5.6}$$

where $||e||_\Sigma^2 = e^T\Sigma^{-1}e$ is the Mahalanobis distance notation and the variables $\nu_{1:4}$ are scaling constants.

The goal is to generate an estimate of the entire trajectory of all AUVs in the team, $\mathbf{X}_{1:t}^{1:N}$, on-board each vehicle conditional on all control ($\mathbf{U}_{1:t}^{1:N}$), proprioceptive sensor (${}^G\mathbf{Z}_{1:t}^{1:N}$ and ${}^C\mathbf{Z}_{1:t}^{1:N}$), and inter-robot range ($\mathbf{R}_{t_1:t_k}^{1:N,1:N}$) data. This is expressed by the following posterior distribution:

$$p(\mathbf{x}_{1:t}^{1:N}|\mathbf{u}_{1:t}^{1:N}, {}^G\mathbf{z}_{1:t}^{1:N}, {}^C\mathbf{z}_{1:t}^{1:N}, \mathbf{r}_{t_1:t_k}^{1:N,1:N}, \mathbf{x}_0^{1:N}) \tag{5.7}$$

which is distinguished from filtering where we only wish to estimate the current AUV poses, $\mathbf{x}_t^{1:N}$:

$$p(\mathbf{x}_t^{1:N}|\mathbf{u}_{1:t}^{1:N}, {}^G\mathbf{z}_{1:t}^{1:N}, {}^C\mathbf{z}_{1:t}^{1:N}, \mathbf{r}_{t_1:t_k}^{1:N,1:N}, \mathbf{x}_0^{1:N}). \tag{5.8}$$

5.2.2 The Information Flow Graph

A useful topological representation of the communication and range measurement between robots is the "information flow graph" [138]. An example of such a structure is shown in Fig. 5.2. First AUV 1 transmits at time t_1. Upon reception, AUVs 2 and 3 are able to measure their relative range to AUV 1. Then AUV 2 transmits at time t_2 and so on.

5.2.3 Cooperative AUV Trajectory Estimation as a Bayes' Network

The standard representation for conditional dependencies in robotics state estimation is through a Bayes' network (BN) [215]. The BN graphical representation for AUV cooperative localization, here between two AUVs, is shown in Fig. 5.3. In the figure, AUV 1 transmits at time t_1 and AUV 2 transmits at time t_2.

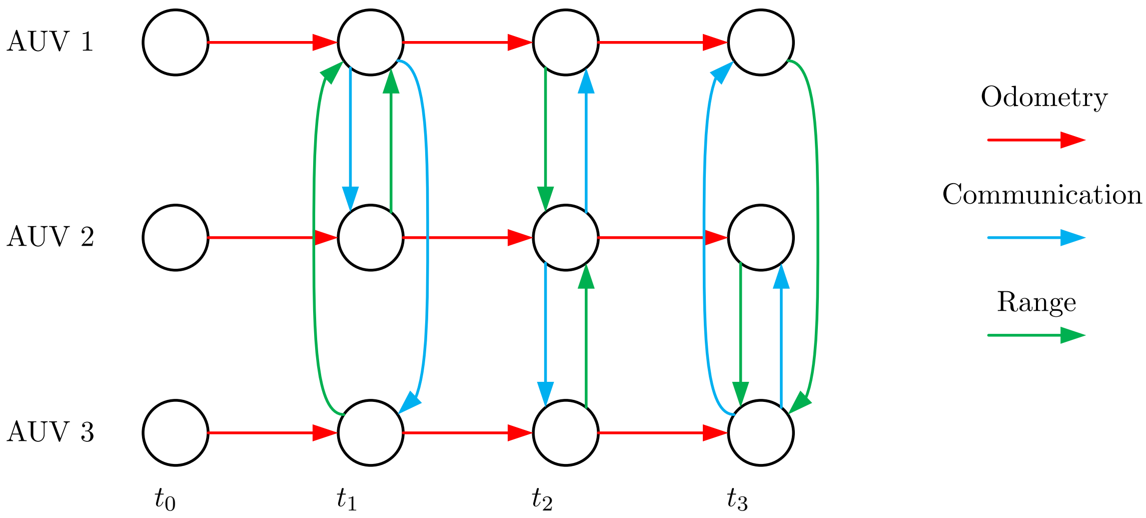


Figure 5.2: Information flow graph for three AUVs depicting inter-robot communications and measurements as a function of time.

Each arrow in Fig. 5.3 represents a conditional dependance. As a result we can estimate the joint probability distribution at time t over all variables in the network as a product of priors, process conditionals and compass, GPS, and range measurement conditionals:

$$\begin{aligned} & p(\mathbf{x}^{1:N}_{1:t}, \mathbf{u}^{1:N}_{1:t}, {}^{G}\mathbf{z}^{1:N}_{1:t}, {}^{C}\mathbf{z}^{1:N}_{1:t}, \mathbf{r}^{1:N,1:N}_{t_1:t_k}, \mathbf{x}^{1:N}_{0}) \propto \\ & \prod_{i=1}^{N} p(\mathbf{x}^i_0) \\ \times & \prod_{t'=1}^{t}\prod_{i=1}^{N} p(\mathbf{x}^i_{t'}|\mathbf{x}^i_{t'-1}, \mathbf{u}^i_{t'}) \\ \times & \prod_{t'=1}^{t}\prod_{i=1}^{N} p({}^{C}\mathbf{z}^i_{t'}|\mathbf{x}^i_{t'}) \\ \times & \prod_{t'=1}^{t}\prod_{i=1}^{N} p({}^{G}\mathbf{z}^i_{t'}|\mathbf{x}^i_{t'}) \\ \times & \prod_{k'=1}^{k}\prod_{i=1}^{N} p(\mathbf{r}^{i,k'-Nc}_{t_{k'}}|\mathbf{x}^i_{t^i_{k'}}, \mathbf{x}^j_{t^{k'-Nc}_{k'}}) \end{aligned} \tag{5.9}$$

where $\propto$ denotes proportionality and $k' - Nc$ is the index of the transmitting robot at time t_k as described in Section 5.1.1 (Recall N is the total number of robots and c is the TDMA cycle number).

In what follows we omit the priors $p(\mathbf{x}^i_0)$ since in reality all robots start on the surface and their state is initialized through compass and GPS sensor measurements.

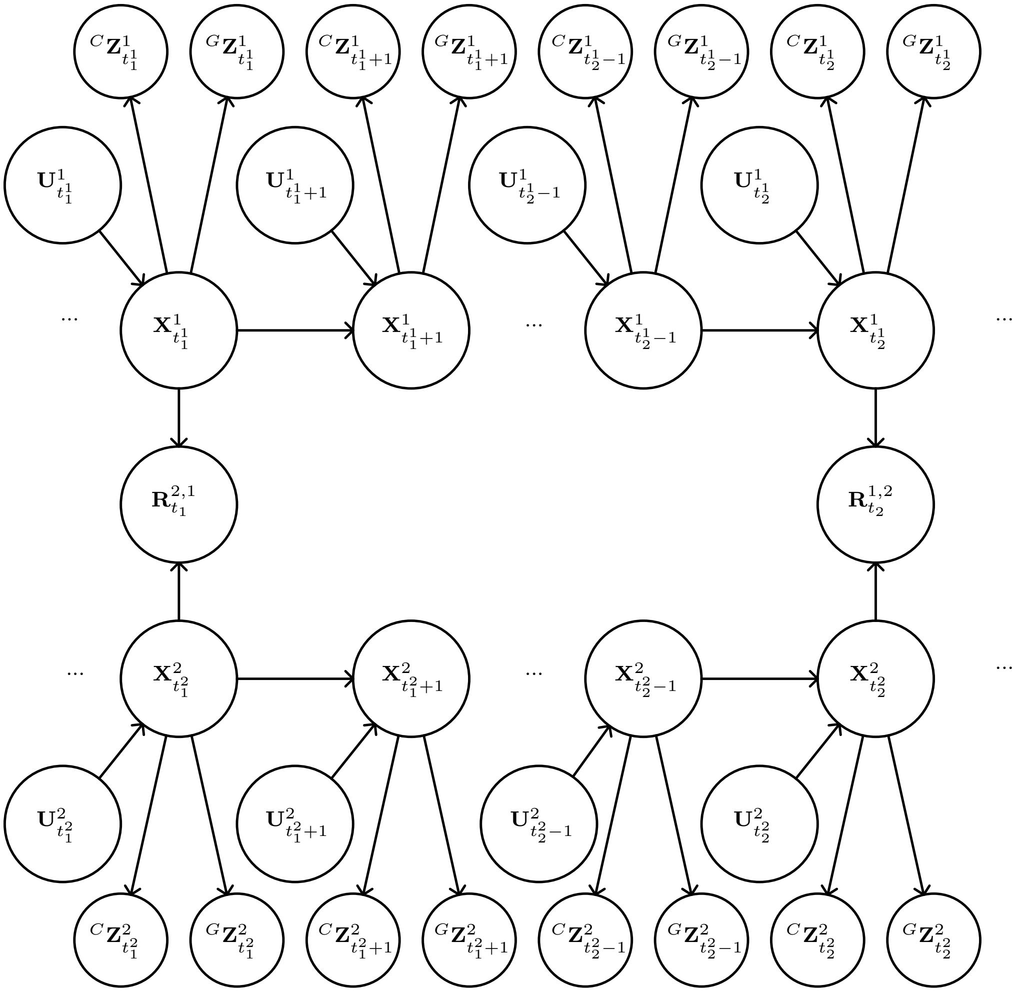

${}^{C}\mathbf{Z}^{1}_{t^1_1}$
${}^{G}\mathbf{Z}^{1}_{t^1_1}$
${}^{C}\mathbf{Z}^{1}_{t^1_1+1}$
${}^{G}\mathbf{Z}^{1}_{t^1_1+1}$
${}^{C}\mathbf{Z}^{1}_{t^1_2-1}$
${}^{G}\mathbf{Z}^{1}_{t^1_2-1}$
${}^{C}\mathbf{Z}^{1}_{t^1_2}$
${}^{G}\mathbf{Z}^{1}_{t^1_2}$
$\mathbf{U}^{1}_{t^1_1}$
$\mathbf{U}^{1}_{t^1_1+1}$
$\mathbf{U}^{1}_{t^1_2-1}$
$\mathbf{U}^{1}_{t^1_2}$
...
$\mathbf{X}^{1}_{t^1_1}$
$\mathbf{X}^{1}_{t^1_1+1}$
...
$\mathbf{X}^{1}_{t^1_2-1}$
$\mathbf{X}^{1}_{t^1_2}$
...
$\mathbf{R}^{2,1}_{t_1}$
$\mathbf{R}^{1,2}_{t_2}$
...
$\mathbf{X}^{2}_{t^2_1}$
$\mathbf{X}^{2}_{t^2_1+1}$
...
$\mathbf{X}^{2}_{t^2_2-1}$
$\mathbf{X}^{2}_{t^2_2}$
...
$\mathbf{U}^{2}_{t^2_1}$
$\mathbf{U}^{2}_{t^2_1+1}$
$\mathbf{U}^{2}_{t^2_2-1}$
$\mathbf{U}^{2}_{t^2_2}$
${}^{C}\mathbf{Z}^{2}_{t^2_1}$
${}^{G}\mathbf{Z}^{2}_{t^2_1}$
${}^{C}\mathbf{Z}^{2}_{t^2_1+1}$
${}^{G}\mathbf{Z}^{2}_{t^2_1+1}$
${}^{C}\mathbf{Z}^{2}_{t^2_2-1}$
${}^{G}\mathbf{Z}^{2}_{t^2_2-1}$
${}^{C}\mathbf{Z}^{2}_{t^2_2}$
${}^{G}\mathbf{Z}^{2}_{t^2_2}$

Figure 5.3: Bayes' network representation of cooperative trajectory estimation with two AUVs. There are two inter-robot observations shown. Implicit in this network is the assumption that measurements at time t only observe states at time t, and that states at time t are only depend on control at time t and states at time $t-1$ (the Markov assumption).

5.2.4 Cooperative AUV Trajectory Estimation as a Factor Graph

An alternate representation of cooperative AUV trajectory estimation from the BN is as a factor graph [124]. A factor graph is a bipartite graph where the measurements ${}^{C}\mathbf{Z}_{1:t}^{1:N}$, ${}^{G}\mathbf{Z}_{1:t}^{1:N}$ and $\mathbf{R}_{t_1:t_k}^{1:N,1:N}$ and control $\mathbf{U}_{1:t}^{1:N}$ are represented by factors nodes, and the variables to be estimated, $\mathbf{X}_{1:t}^{1:N}$ are variable nodes. Edges connect factor nodes to variable nodes where an edge indicates that the factor operates on that variable.

In a factor graph each factor represents a potential function, denoted here by ϕ_p over some of the unknowns, usually one or two.

The joint density in (5.9) can then be factored as a product of these potential functions or factors:

$$p(\mathbf{X}_{1:t}^{1:N}, \mathbf{U}_{1:t}^{1:N}, {}^{C}\mathbf{Z}_{1:t}^{1:N}, {}^{G}\mathbf{Z}_{1:t}^{1:N}, \mathbf{R}_{t_1:t_k}^{1:N,1:N}) = \phi(\mathbf{X}_{1:t}^{1:N}) = \prod_p \phi_p(\mathbf{X}_{1:t}^{1:N\{p\}}) \tag{5.10}$$

where $\mathbf{X}_{1:t}^{1:N\{p\}}$ represents the subset of variables of $\mathbf{X}_{1:t}^{1:N}$ that are adjacent in the graph to factor ϕ_p.

The factor graph representation has a close relationship with the measurement Jacobian A as each row in A is representative of one factor as will be derived below. The structure of the measurement Jacobian is included in Fig. 5.4 where shaded elements are non-zero. The information matrix $\mathbf{\Omega}$ is related to the measurement Jacobian through $\mathbf{\Omega} = A^T A$.

The factors in (5.10) are proportional to the elements of the joint distribution given in (5.9).

It is important to stress that these two BN and factor graph representations are inherently different views of the same system.

5.2.5 Cooperative AUV Trajectory Estimation as a Least Squares Optimization

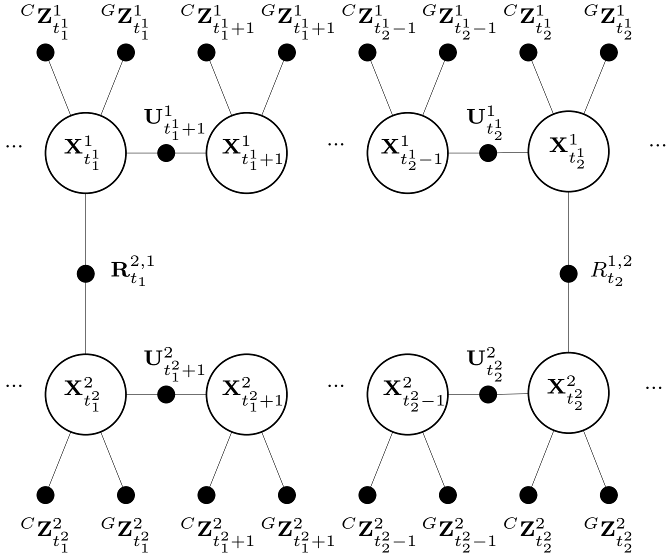
In this section we use properties of the BN and factor graph representations of cooperative localization just shown to derive a general least-squares optimization form for the cooperative localization problem. Once the general form is derived, any new measurement models, or factors, can easily be incorporated into the framework without changing the fundamental underlying procedure of how the least-squares problem is solved to yield an estimate.

The maximum *a posteriori* estimate of $\mathbf{X}_{1:t}^{1:N}$, denoted $\mathbf{x}_{1:t}^{*1:N}$, is the value of $\mathbf{x}_{1:t}^{1:N}$ that maximizes the joint probability [57]: $p(\mathbf{x}_{1:t}^{1:N}, {}^{C}\mathbf{z}_{1:t}^{1:N}, {}^{G}\mathbf{z}_{1:t}^{1:N}, \mathbf{u}_{1:t}^{1:N}, \mathbf{r}_{t_1:t_k}^{1:N,1:N})$

$$\mathbf{x}_{1:t}^{*1:N} = \underset{\mathbf{x}_{1:t}^{1:N}}{\operatorname{argmax}}\, p(\mathbf{x}_{1:t}^{1:N}, {}^{C}\mathbf{z}_{1:t}^{1:N}, {}^{G}\mathbf{z}_{1:t}^{1:N}, \mathbf{u}_{1:t}^{1:N}, \mathbf{r}_{t_1:t_k}^{1:N,1:N}) \tag{5.11}$$

The probability $p(\mathbf{x}_{1:t}^{1:N}, {}^{C}\mathbf{z}_{1:t}^{1:N}, {}^{G}\mathbf{z}_{1:t}^{1:N}, \mathbf{u}_{1:t}^{1:N}, \mathbf{r}_{t_1:t_k}^{1:N,1:N})$ is a multi-variate Gaussian, therefore maximizing it is equivalent to minimizing the negative log [57]:

$$\mathbf{x}_{1:t}^{*1:N} = \underset{\mathbf{x}_{1:t}^{1:N}}{\operatorname{argmin}}\{-\log p(\mathbf{x}_{1:t}^{1:N}, \mathbf{z}_{1:t}^{1:N}, \mathbf{u}_{1:t}^{1:N}, \mathbf{r}_{t_1:t_k}^{1:N,1:N})\} \tag{5.12}$$



${}^{C}\mathbf{Z}^1_{t^1_1}$
${}^{G}\mathbf{Z}^1_{t^1_1}$
${}^{C}\mathbf{Z}^1_{t^1_1+1}$
${}^{G}\mathbf{Z}^1_{t^1_1+1}$
${}^{C}\mathbf{Z}^1_{t^1_2-1}$
${}^{G}\mathbf{Z}^1_{t^1_2-1}$
${}^{C}\mathbf{Z}^1_{t^1_2}$
${}^{G}\mathbf{Z}^1_{t^1_2}$
...
$\mathbf{X}^1_{t^1_1}$
$\mathbf{U}^1_{t^1_1+1}$
$\mathbf{X}^1_{t^1_1+1}$
...
$\mathbf{X}^1_{t^1_2-1}$
$\mathbf{U}^1_{t^1_2}$
$\mathbf{X}^1_{t^1_2}$
...
$\mathbf{R}^{2,1}_{t_1}$
$R^{1,2}_{t_2}$
...
$\mathbf{X}^2_{t^2_1}$
$\mathbf{U}^2_{t^2_1+1}$
$\mathbf{X}^2_{t^2_1+1}$
...
$\mathbf{X}^2_{t^2_2-1}$
$\mathbf{U}^2_{t^2_2}$
$\mathbf{X}^2_{t^2_2}$
...
${}^{C}\mathbf{Z}^2_{t^2_1}$
${}^{G}\mathbf{Z}^2_{t^2_1}$
${}^{C}\mathbf{Z}^2_{t^2_1+1}$
${}^{G}\mathbf{Z}^2_{t^2_1+1}$
${}^{C}\mathbf{Z}^2_{t^2_2-1}$
${}^{G}\mathbf{Z}^2_{t^2_2-1}$
${}^{C}\mathbf{Z}^2_{t^2_2}$
${}^{G}\mathbf{Z}^2_{t^2_2}$

	$\mathbf{X}^1_{t^1_1}$	$\mathbf{X}^1_{t^1_1+1}$	$\mathbf{X}^1_{t^1_2-1}$	$\mathbf{X}^1_{t^1_2}$	$\mathbf{X}^2_{t^2_1}$	$\mathbf{X}^2_{t^2_1+1}$	$\mathbf{X}^2_{t^2_2-1}$	$\mathbf{X}^2_{t^2_2}$
$\mathbf{U}^1_{t^1_1}$	▨							
$\mathbf{U}^1_{t^1_1+1}$	▨	▨						
$\mathbf{U}^1_{t^1_2-1}$		▨	▨					
$\mathbf{U}^1_{t^1_2}$			▨	▨				
$\mathbf{U}^2_{t^2_1}$					▨			
$\mathbf{U}^2_{t^2_1+1}$					▨	▨		
$\mathbf{U}^2_{t^2_2-1}$						▨	▨	
$\mathbf{U}^2_{t^2_2}$							▨	▨
${}^{C,G}\mathbf{Z}^1_{t^1_1}$	▨							
${}^{C,G}\mathbf{Z}^1_{t^1_1+1}$		▨						
${}^{C,G}\mathbf{Z}^1_{t^1_2-1}$			▨					
${}^{C,G}\mathbf{Z}^1_{t^1_2}$				▨				
${}^{C,G}\mathbf{Z}^2_{t^2_1}$					▨			
${}^{C,G}\mathbf{Z}^2_{t^2_1+1}$						▨		
${}^{C,G}\mathbf{Z}^2_{t^2_2-1}$							▨	
${}^{C,G}\mathbf{Z}^2_{t^2_2}$								▨
$\mathbf{R}^{2,1}_{t_1}$	▨					▨		
$\mathbf{R}^{1,2}_{t_2}$				▨			▨	

Figure 5.4: Factor graph representation of BN in Fig. 5.3. The factor graph has a natural representation with the measurement Jacobian where each new factor adds a new row and operates over the variables that are adjacent to it. Shaded elements of the measurement Jacobian are non-zero with non-shaded elements being 0.

which allows the joint density defined in (5.9) as a product to be represented as a sum:

$$\begin{aligned}\mathbf{x}^{*1:N}_{1:t} = \underset{\mathbf{x}^{1:N}_{1:t}}{\operatorname{argmin}}\, & const+ \\ & \sum_{t'=1}^{t}\sum_{i=1}^{N}\log\{p(\mathbf{x}^i_{t'}|\mathbf{x}^i_{t'-1},\mathbf{u}^i_{t'})\}+ \\ & \sum_{t'=1}^{t}\sum_{i=1}^{N}\log\{p({}^{C}\mathbf{z}^i_{t'}|\mathbf{x}^i_{t'})\}+ \\ & \sum_{t'=1}^{t}\sum_{i=1}^{N}\log\{p({}^{G}\mathbf{z}^i_{t'}|\mathbf{x}^i_{t'})\}+ \\ & \sum_{k'=1}^{k}\sum_{i=1}^{N}\log\{p(\mathbf{r}^{i,k'-Nc}_{t_{k'}}|\mathbf{x}^i_{t^i_{k'}},\mathbf{x}^{k'-Nc}_{t^{k'-Nc}_{k'}})\}.\end{aligned} \tag{5.13}$$

where *const* is the sum of the logs of the normalization constants $\nu_i, i = 1..4$ and can be neglected since it would not affect the argmin operation. Replacing the conditional probabilities in (5.13) with the distributions derived in (5.6), the logs and exponents cancel out to yield:

$$\begin{aligned}\mathbf{x}^{*1:N}_{1:t} = \underset{\mathbf{x}^{1:N}_{1:t}}{\operatorname{argmin}}\{ & \\ & \sum_{t'=1}^{t}\sum_{i=1}^{N}\frac{1}{2}||f(\mathbf{x}^i_{t'-1},\mathbf{u}^i_{t'})-\mathbf{x}^i_{t'}||^2_{\mathbf{Q}^i_{t'}}+ \\ & \sum_{t'=1}^{t}\sum_{i=1}^{N}\frac{1}{2}||h^G(\mathbf{x}^i_{t'})-{}^{G}\mathbf{z}^i_{t'}||^2_{\mathbf{\Lambda}^G}+ \\ & \sum_{t'=1}^{t}\sum_{i=1}^{N}\frac{1}{2}||h^C(\mathbf{x}^i_{t'})-{}^{C}\mathbf{z}^i_{t'}||^2_{\mathbf{\Lambda}^C}+ \\ & \sum_{k'=1}^{k}\sum_{i=1}^{N}\frac{1}{2}||h^R(\mathbf{x}^i_{t^i_{k'}},\mathbf{x}^{k'-Nc}_{t^j_{k'}})-\mathbf{r}^{i,k'-Nc}_{t_{k'}}||^2_{\mathbf{\Lambda}^R}\}.\end{aligned} \tag{5.14}$$

which is a non-linear least squares problem. Such a problem can be solved with a non-linear optimization method such as Gauss-Newton or Levenberg-Marquardt algorithm [58]. In either case, linearization via first order Taylor expansion is performed on (5.14) around a linearization point, ${}^{m}\mathbf{x}^{1:N}_{1:t}$, where m is the iteration number of the non-linear optimization.

Analytical derivations of the Jacobians required for the first order Taylor expansion are given in Appendix D.

Once the nonlinear functions in the exponents have been replaced by linear approximations (i.e. matrices), the Mahalanobis distances can be removed from (5.14) by a change of variable based on [57]:

$$||e||^2_{\Sigma} = e^T\Sigma^{-1}e = (((\Sigma^{-1})^{1/2})^T e)^T(((\Sigma^{-1})^{1/2})^T e) = ||(((\Sigma^{-1})^{1/2})^T e)||^2. \tag{5.15}$$

Finally, we collect all the Jacobians into one matrix A and all the residuals into a "right-hand side" vector $\mathbf{b}$ and obtain the linear least-squares form:

$$\mathbf{x}_{1:t}^{*1:N} = \underset{\mathbf{x}_{1:t}^{1:N}}{\operatorname{argmin}}\{||A(\mathbf{x}_{1:t}^{1:N} - {}^{m}\mathbf{x}_{1:t}^{1:N}) - \mathbf{b}||^2\} \tag{5.16}$$

which can be solved by differentiating with respect to $\mathbf{x}_{1:t}^{1:N} - {}^{m}\mathbf{x}_{1:t}^{1:N}$ and equating to zero to yield:

$$A^T A(\mathbf{x}_{1:t}^{1:N} - {}^{m}\mathbf{x}_{1:t}^{1:N}) = A^T\mathbf{b} \tag{5.17}$$

A is the measurement Jacobian shown in Fig. 5.4.

Recently, effort has gone into finding efficient incremental solutions to (5.16) in the context of SLAM by factoring the measurement Jacobian A into a square root form using either QR [111] or Cholesky [57] factorization. A key consideration in these algorithms is choosing the ordering of the variables being estimated to maintain sparseness of the square-root matrix, R. It has been shown that obtaining the optimal variable ordering is an NP-hard problem [111], heuristics have been proposed that can provide near-optimal solutions very efficiently, for example "COLAMD" [55].

5.2.6 Data Throughput for Optimal Centralized Estimate

The optimal (in terms of minimum mean square error and minimum covariance) centralized multi-robot trajectory estimate is obtained by solving the full least squares equation given in (5.14). From inspection of (5.14), it is clear that to solve this equation requires knowledge of all control and measurement data from all robots for all time. Unfortunately, it is generally infeasible for any robot to obtain all raw data for all robots up to the present time t due to the restrictions of the underwater acoustic channel.

5.2.6.1 Best Case

In this context, "best case" refers to 100% communications packet success rate. If the DVL and compass frequencies are 10Hz and each piece of data can be encoded with 1byte (8 bits) and the TDMA slot length is 10s and the number of robots in the team is N, then each robot would potentially need to transmit (8bits/piece of data*30 pieces of data /second * 10seconds/slot * N slots)*N robots, which would be 21.6Kbits of data per transmission for a modest team size of $N = 3$. While this may be feasible using radio frequency communications, such throughput rates are unachievable in water.

5.2.6.2 Worst Case

In the inevitable case there are communications dropouts, the data required to be transmitted is unbounded and grows linearly with time. In the worst case all robots would need to transmit all their sensor data from the start of the mission.

5.3 Multi-Robot Trajectory Estimation

We showed in Chapter 4 that maintaining the entire robot trajectory in the state space is advantageous from the standpoint of coverage efficiency. Specifically, if an AUV receives some information at time t that allows it to improve its estimate of a past state $\mathbf{X}_{t'}, t' < t$, then this information can be used to update the estimate of the coverage of the workspace.

In this section, an algorithm is presented whereby all robots in the AUV team can cooperatively estimate their entire trajectories, which is useful for coverage estimation, and also has better scalability of data throughput required on the underwater acoustic channel.

5.3.1 Overview

In overview, on each transmission a robot transmits its accumulated change in position and associated uncertainty from times of known previous contact with the other members of the team to guarantee that the multi-robot trajectory graph will remain connected and consistent.

The following two definitions are useful for bookkeeping purposes

Definition 5.3.1. *An* ***incoming contact point****, denoted $\mathcal{C}^i_{in}[j]$, is the time of the last successful communication from robot j to robot i known to robot i.*

Definition 5.3.2. *An* ***outgoing contact point****, denoted $\mathcal{C}^i_{out}[j]$, is the time of the last successful communication from robot i to robot j known to robot i.*

Each robot i maintains two sets representing the incoming ($\mathcal{C}^i_{in}$) and outgoing ($\mathcal{C}^i_{out}$) contact point times. At data transmission, the accumulated change in position to each of these contact point times are calculated and sent. This data along with any other GPS data received or inter-robot ranges calculated locally comprise the data packet contents.

To summarize the salient aspects of the proposed algorithm:

- Each robot estimates its own trajectory as well as the positions of other robots at the times of inter-robot measurements.
- Through the acknowledgment bits, each robot tracks its last *outgoing* and *incoming* contact points (times of most recent successful communications), and transmits accumulated change in position to these contact points to ensure connectivity and consistency of the estimator.
- In the case that a robot has no knowledge about whether a previous communication was successful or not, it is always assumed that it failed. It will be shown that this assumption has little to no impact on the performance since the receiving robot can recover the correct data in the case that the assumption is incorrect. This assumption is the key that allows for packet size to grow linearly with the team size and be constant with respect to time even in the case of communications failures.

The proposed method has the following advantages:

- Data throughput remains constant with time.
- On-board computation required does not grow with time.
- Data throughput scales linearly with AUV team size, $O(N)$, where N is the number of robots in the team.
- Data packet size is not correlated with communications success rate.
- Able to produce a consistent estimate of multi-robot trajectory in the presence of any combination of communications failures. In the worst case (all failures) the estimate reduces to non-cooperative trajectory estimation.

- Robust to complete robot failure.

An overview of the algorithm is given in Algorithm 10.

Algorithm 10 Cooperative AUV trajectory estimation

```
while Coverage mission not complete do
    Add control factor to least-squares formulation
    Add proprioceptive measurement factors to least-squares formulation
    if Scheduled to transmit then
        Algorithm 11
    end if
    if Data received then
        Algorithm 12
    end if
    Perform least-squares optimization
    Update coverage map and plan path to achieve coverage based on method presented
    in Chapter 4.
end while
```

In the remainder of this section, it is shown how the proposed method can be implemented with the least-squares framework described in Section 5.2. Further detail of exactly what data should be communicated is given and algorithms are presented for data transmission and reception. Finally, case studies are described to show the performance of the algorithm under varying circumstances.

5.3.2 Least-Squares Formulation

Consistent with the previous section, the proposed method will be formulated as a least-squares optimization problem.

The change in position for robot i at time t is given by:

$$\mathbf{X}_t^i = \mathbf{X}_{t-1}^i + \mathbf{\Delta X}_t^i + \tilde{\boldsymbol{\zeta}}_t^i \tag{5.18}$$

where:

$$\mathbf{\Delta X}_t^i \triangleq \begin{bmatrix} \Delta X_t^i \\ \Delta Y_t^i \end{bmatrix} = \Delta t \begin{bmatrix} \cos \Psi_t^i & \sin \Psi_t^i \\ -\sin \Psi_t^i & \cos \Psi_t^i \end{bmatrix} \begin{bmatrix} V_t^u \\ V_t^v \end{bmatrix} \tag{5.19}$$

is a RV that represents the change in position from time $t-1$ to t and Δt is the reciprocal of the frequency of the DVL sensor. $\tilde{\boldsymbol{\zeta}}_t^i \sim \mathcal{N}(0, \tilde{\mathbf{Q}}_t^i)$ is the Gaussian noise with mean 0 and covariance $\tilde{\mathbf{Q}}_t^i$ associated with the change in position that results from the fact that velocity sensor data from the DVL (V_t^u and V_t^v) contain noise.

The accumulated change in position from time t_1 to time t_2 can be calculated by summing all changes in position between time t_1 and time t_2:

$$\mathbf{\Delta X}_{t_1 \to t_2}^i \triangleq \sum_{t=t_1}^{t_2} \mathbf{\Delta X}_t^i \tag{5.20}$$

The additive noise term for the accumulated change in position can be similarly calculated as:

$$\tilde{\boldsymbol{\zeta}}^i_{t_1\to t_2} \triangleq \sum_{t=t_1}^{t_2} \tilde{\boldsymbol{\zeta}}^i_t \sim \mathcal{N}(0, \tilde{\mathbf{Q}}^i_{t_1\to t_2}) \tag{5.21}$$

where the covariance $\tilde{\mathbf{Q}}^i_{t_1\to t_2}$ is calculated as:

$$\tilde{\mathbf{Q}}^i_{t_1\to t_2} = \sum_{t=t_1}^{t_2} \tilde{\mathbf{Q}}^i_t \tag{5.22}$$

Finally we can write an equation for the accumulated change in position for robot i from time t_1 to time t_2 as:

$$\mathbf{X}^i_{t_2} = \mathbf{X}^i_{t_1} + \mathbf{\Delta X}^i_{t_1\to t_2} + \tilde{\boldsymbol{\zeta}}^i_{t_1\to t_2} \tag{5.23}$$

The same procedure as described in Section 5.2 is followed by collecting non-noise terms on the left hand side to generate the following least-squares formulation (5.14) in this case for robot j:

$$\begin{aligned}
\mathbf{x}^{*1:N}_{1:t} = \underset{\mathbf{x}^{1:N}_{1:t}}{\operatorname{argmin}}\{ \\
&\sum_{t'=1}^{t} \frac{1}{2}||f(\mathbf{x}^j_{t'-1}, \mathbf{u}^j_{t'}) - \mathbf{x}^j_{t'}||^2_{\mathbf{Q}^j_{t'}} + \\
&\sum_{k_1=1}^{k} \sum_{k_2=k_1}^{k} \sum_{\substack{i=1..N \\ i\neq j}} \frac{1}{2}||\mathbf{x}^i_{t_{k_1}} + \mathbf{\Delta x}^i_{t_{k_1}\to t_{k_2}} - \mathbf{x}^i_{t_{k_2}}||^2_{\tilde{\mathbf{Q}}^i_{t_{k_1}\to t_{k_2}}} + \\
&\sum_{t'=1}^{t} \sum_{i=1}^{N} \frac{1}{2}||h^G(\mathbf{x}^i_{t'}) - {}^G\mathbf{z}^i_{t'}||^2_{\mathbf{\Lambda}^G} + \\
&\sum_{t'=1}^{t} \frac{1}{2}||h^C(\mathbf{x}^j_{t'}) - {}^C\mathbf{z}^j_{t'}||^2_{\mathbf{\Lambda}^C} + \\
&\sum_{k'=1}^{k} \sum_{i=1}^{N} \frac{1}{2}||h^R(\mathbf{x}^i_{t^i_{k'}}, \mathbf{x}^{k'-Nc}_{t_{k'}}) - \mathbf{r}^{i,k'-Nc}_{t_{k'}}||^2_{\mathbf{\Lambda}^R}\}.
\end{aligned} \tag{5.24}$$

which is identical to (5.14) except that the control factors have been re-organized into own-robot (first term) and other-robot accumulated changes in position (second term) and that the compass factors have been removed for all other robots (fourth term).

This new least squares equation can be solved using the same method as described in Section 5.2.

5.3.3 Bookkeeping - Contact Points

Central to the approach is that each robot should maintain a set of $N - 1$ incoming and outgoing confirmed contact points. These contact points are represented as the times of most recent confirmed communications and rangings. An incoming contact point is the time of last data *reception* from a given robot, and an outgoing contact point is the time of most recent successful data *transmission* to a given robot.

As an example, for the information flow graph of Fig. 5.2, the incoming contact point time sets after the communication at time t_3 are given by:

$$\begin{aligned}
\mathcal{C}_{in}^1 &= \{-, t_2^1, t_3^1\} \\
\mathcal{C}_{in}^2 &= \{t_1^2, -, t_3^2\} \\
\mathcal{C}_{in}^3 &= \{t_1^3, t_2^3, -\}
\end{aligned} \tag{5.25}$$

and the outgoing contact point times sets are given by:

$$\begin{aligned}
\mathcal{C}_{out}^1 &= \{-, t_1^1, t_1^1\} \\
\mathcal{C}_{out}^2 &= \{t_0^2, -, t_2^2\} \\
\mathcal{C}_{out}^3 &= \{t_0^3, t_0^3, -\}
\end{aligned} \tag{5.26}$$

where a '$-$' represents the entry in the set that corresponds to the robot on which it resides ($\mathcal{C}_{in}^i[i] = \mathcal{C}_{out}^i[i] =$ '$-'$). All contact points in this case were initialized to t_0. For example, the outgoing contact point times for $\mathcal{C}_{out}^3$ are still t_0 because AUV 3 has no knowledge if the communications that it sent out at time t_3 were successful and won't until it gets a confirmation through the acknowledgement bits on the next cycle, at time t_4 from AUV 1 and time t_5 from AUV 2.

Incoming contact points are easily detectable based on the times at which communications are received. However, outgoing contact points necessitate the use of communicated acknowledgment bits that are sent in subsequent data packet transmissions. In the case that an acknowledgement communication also fails, the contact point time will not be updated, in essence assuming that the previous outgoing communication from that robot failed. However, it will be shown that, in the case that this assumption is incorrect, the receiving robot will still be able to recover the appropriate factor.

These contact points are important because they determine which change in position factors need to be sent so that other vehicles in the cooperative can build the constraints for their own least-squares optimizations.

5.3.4 Data Transmission

In this subsection the details of what data should be sent and how it is generated locally before transmission, are presented. The process is summarized in Algorithm 11.

Consider the case where robot i makes an acoustic transmission at time t. The following data should be included in the data packet:

- The accumulated change in position factors from incoming and outgoing contact point times to the present time: $\mathbf{\Delta X}^i_{\mathcal{C}_{in}^i[j]\to t}$ and $\mathbf{\Delta X}^i_{\mathcal{C}_{out}^i[j]\to t}$ for all $j = 1..N, j \neq i$ and the associated covariances $\tilde{\mathbf{Q}}^i_{\mathcal{C}_{in}^i[j]\to t}$ and $\tilde{\mathbf{Q}}^i_{\mathcal{C}_{out}^i[j]\to t}$.

- Range data associated with each of the incoming contact point times: $\mathbf{R}^{i,j}_{\mathcal{C}_{in}^i[j]}$ for all $j = 1..N, j \neq i$.

- A GPS measurement if one has been made since the oldest contact point and an accumulated change in position to the time of the GPS measurement.

- A set of $N-1$ acknowledgment bits: 1 indicates success reception and 0 indicates failure on the last cycle.

Algorithm 11 Generating data packet for acoustic transmission on-board robot i

```
Transmission queue is empty
for all $j = 1..N, j \neq i$ do
    Calculate $\mathbf{\Delta X}^i_{\mathcal{C}^i_{in}[j]\to t}$ and $\mathbf{\Delta X}^i_{\mathcal{C}^i_{out}[j]\to t}$ and associated covariances $\tilde{\mathbf{Q}}^i_{\mathcal{C}^i_{in}[j]\to t}$ and $\tilde{\mathbf{Q}}^i_{\mathcal{C}^i_{out}[j]\to t}$ and add them to transmission queue.
    Add range measurement $\mathbf{R}^{i,j}_{\mathcal{C}^i_{in}[j]}$ to transmission queue.
end for
if GPS update, ${}^G\mathbf{Z}^i_{t_g}$ received since oldest contact point time ($t_g < \min\{\mathcal{C}^i_{in}, \mathcal{C}^i_{out}\}$) then
    Add ${}^G\mathbf{Z}^i_{t_g}$, $\mathbf{\Delta X}^i_{t_g\to t}$, and $\tilde{\mathbf{Q}}^i_{t_g\to t}$ to the transmission queue.
end if
Add acknowledgement bits to the transmission queue.
Push transmission queue to the modem hardware for transmission.
```

5.3.5 Data Reception

Upon reception of an acoustic communication containing the data just described, the receiver must generate the correct terms to compute the maximum *a posteriori* estimate by solving (5.24). Generating the correct accumulated change in position factors that relate the positions of other robots to own robot poses can possibly require using the subtraction property of the accumulated change in position factors in the case that a previous transmission that was assumed to have failed was actually successful. For example, consider the case where AUV i receives two accumulated change in position factors from robot j at time t, $\mathbf{\Delta X}^j_{t_1\to t}$ and $\mathbf{\Delta X}^j_{t_2\to t}$ with $t_1 < t_2$ then $\mathbf{\Delta X}^j_{t_1\to t_2}$ can be recovered using:

$$\mathbf{\Delta X}^j_{t_1\to t_2} = \mathbf{\Delta X}^j_{t_1\to t} - \mathbf{\Delta X}^j_{t_2\to t} \tag{5.27}$$

An overview of the method for processing the received data is given in Algorithm 12.

In line 1 the new range factor is generated by calculating the range based on the TOF of the acoustic transmission and the assumed lock-step synchronization of the on-board clocks. The incoming accumulated change in position data is sorted into a set $\mathcal{X}$ with the most recent first, and then are processed individually in the for-loop (line 4 to line 28). For each accumulated change in position in the set, we see if we can decompose it using an already existing accumulated change in position in the factor graph that shares the same end time but a later start time (if statement at line 8) or shares the same start time and has an earlier end time (if statement at line 14). In either case we can decompose the accumulated change in position using the subtractive property defined in (5.27). The decomposition process (while loop from line 6 to line 20) continues until the accumulated change in position can be decomposed no further. At this point, any new position states that need to be generated are created and added to the state space (lines 22 and 25) and then finally the decomposed accumulated change in position constraint can be added that operates on these two states (line 27). Subsequently, the range and GPS factors can be added (lines 29 and 30) and the contact point times can be updated (lines 31 and 33). Finally, the least squares optimization is performed to generate a new estimate over the newly expanded state space containing own-robot poses and other-robot positions.

The essential aspect of this algorithm is that robot i can recover what is needed for its own

Algorithm 12 Reception of acoustic data packet on AUV i from AUV j at time t_k

```
Build range constraint $R^{i,j}_{t_k}$ based on TOF of clocks and covariance $\mathbf{\Lambda}^R$.
Collect the $P$ received accumulated change in position factors into a set $\mathcal{X} \triangleq \{\mathcal{X}[p]\}_{p=1..P}$ where $\mathcal{X}[p] \triangleq \mathbf{\Delta X}^j_{t^p_1 \to t^p_2}$ and similarly with the associated covariances into a set $\mathcal{Q}$
Order $\mathcal{X}$ and $\mathcal{Q}$ in terms of decreasing $t_1$: $t_1^{p_1} < t_1^{p_2}$ if $p_1 < p_2$
for all $p = 1..P$ do
  repeat_flag $\leftarrow$ true
  while repeat_flag do
    repeat_flag $\leftarrow$ false
    if there already exists some $\mathbf{\Delta X}^j_{t \to t^p_2}$ with $t > t^p_1$ in the factor graph then
      $\mathcal{X}[p] \leftarrow \mathbf{\Delta X}^j_{t^p_1 \to t} = \mathbf{\Delta X}^j_{t^p_1 \to t^p_2} - \mathbf{\Delta X}^j_{t \to t^p_2}$
      $\mathcal{Q}[p] \leftarrow \tilde{\mathbf{Q}}^j_{t^p_1 \to t} = \tilde{\mathbf{Q}}^j_{t^p_1 \to t^p_2} - \tilde{\mathbf{Q}}^j_{t \to t^p_2}$
      $t^p_2 \leftarrow t$
      repeat_flag $\leftarrow$ true
    end if
    if there already exists some $\mathbf{\Delta X}^j_{t^p_1 \to t}$ with $t < t^p_2$ then
      $\mathcal{X}[p] \leftarrow \mathbf{\Delta X}^j_{t \to t^p_2} = \mathbf{\Delta X}^j_{t^p_1 \to t^p_2} - \mathbf{\Delta X}^j_{t^p_1 \to t}$
      $\mathcal{Q}[p] \leftarrow \tilde{\mathbf{Q}}^j_{t \to t^p_2} = \tilde{\mathbf{Q}}^j_{t^p_1 \to t^p_2} - \tilde{\mathbf{Q}}^j_{t^p_1 \to t}$
      $t^p_1 \leftarrow t$
      repeat_flag $\leftarrow$ true
    end if
  end while
  if $\mathbf{X}^j_{t^p_1}$ not in state space then
    Add $X^j_{t^p_1}$ to state space
  end if
  if $\mathbf{X}^j_{t^p_2}$ not in state space then
    Add $\mathbf{X}^j_{t^p_2}$ to state space
  end if
  Add accumulated change in position as a constraint $(\mathcal{X}[p], \mathcal{Q}[p])$ between nodes $\mathbf{X}^j_{t^p_1}$ and $\mathbf{X}^j_{t^p_2}$.
end for
Add all range constraints (both the ones received in the packet and the one just generated)
Add GPS constraint $({}^G\mathbf{Z}_{t_g}, \mathbf{\Lambda}^G)$ if we received one
$\mathcal{C}^i_{in}[j] \leftarrow t$
if acknowledgment bit received for last transmission time $t'$ then
  $\mathcal{C}^i_{out}[j] \leftarrow t'$
end if
```

local factor graph and the appropriate range, GPS, and accumulated changes in position can be added to the factor graph that are guaranteed to stay connected to yield an estimate of the own-robot's entire trajectory.

5.3.6 Data Throughput Analysis

The major advantage of the proposed method is the reduced data throughput on the underwater acoustic channel and robustness to communications failures.

Maximum total data to be transmitted is $2 * (N - 1)$ accumulated change in position factors (comprising value and associated covariance), $N - 1$ range factors, one GPS factor with associated accumulated change in position, and $N - 1$ acknowledgment bits regardless of communications failures, for a total of $4N - 3$. Scaling is linear with respect to the size of the AUV team N and constant with respect to time t even in the worst case of communications dropouts. This represents an improvement over any AUV cooperative localization algorithm presented in the literature.

5.3.7 Case Studies

In this section three examples are presented to further illustrate the operation of the proposed algorithm.

5.3.7.1 2 Robots with No Communications Failures

An example of the evolution of the factor graphs is shown in Table 5.2.

First AUV 1 transmits at time t_1 and on reception AUV 2 is able to calculate $\mathbf{R}^{2,1}_{t_1}$ and add and connect nodes for AUV 1 at time t_0 and t_1. AUV 2 transmits at time t_2 and on reception AUV 1 is able to build the complete pose graph with two range factors in it. In this way the cycle continues.

In this case there are no communication failures and each robot is able to obtain the full centralized equivalent estimate (the estimate that would be obtained if a centralized agent had access to all sensor data obtained on-board all of the vehicles).

5.3.7.2 3 Robots with No Communications Failures

The two robot case is quite trivial. For a more complex example, consider the case of three robots to elaborate on the scalability issue. Fig. 5.5 is the local multi-robot factor graph on robot 1 after a full cycle of successful range transmissions in the order [1,2,3]. It is now time t_4 and robot 1 is scheduled for transmission.

The incoming contact points set is $\mathcal{C}^1_{in} = \{-, t^1_2, t^1_3\}$ and the outgoing contact points set is $\mathcal{C}^1_{out} = \{-, t^1_1, t^1_1\}$.

Based on the contact point times, robot 1 should transmit $\mathbf{\Delta X}^1_{t_4}$, $\mathbf{\Delta X}^1_{t_3 \to t_4}$, and $\mathbf{\Delta X}^1_{t_2 \to t_4}$ along with the associated covariances $\tilde{\mathbf{Q}}^1_{t_4}$, $\tilde{\mathbf{Q}}^1_{t_3 \to t_4}$, and $\tilde{\mathbf{Q}}^1_{t_2 \to t_4}$ and the range measurements at the incoming contact point times $\mathbf{R}^{1,2}_{t_2}$ and $\mathbf{R}^{1,3}_{t_3}$ as well as the acknowledgement bits $[1, 1]$ denoting success at the last cycle. It is interesting to note that in this case we only need to send three accumulated change in position factors as opposed to the maximum four (since $2(N - 1) = 4$) since the outgoing contact point times coincide for both robots. This will be a common situation so we will only need to transmit N accumulated change in position factors.

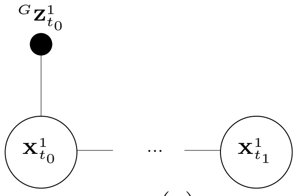
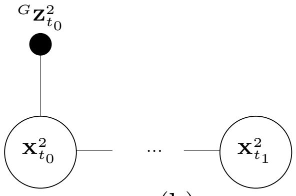
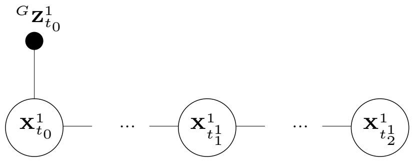
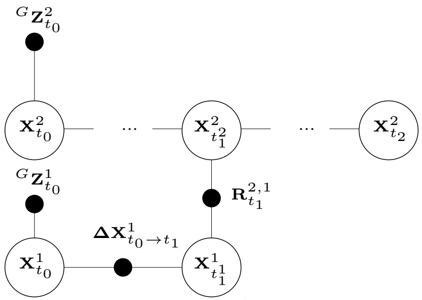
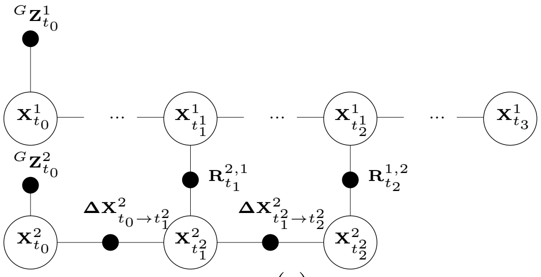
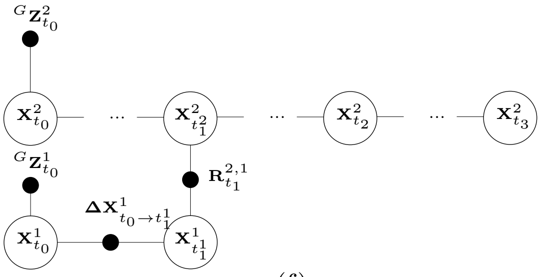
AUV 1	AUV 2
 ${}^G\mathbf{Z}^1_{t_0}$ $\mathbf{X}^1_{t_0}$ ... $\mathbf{X}^1_{t_1}$ (a)	 ${}^G\mathbf{Z}^2_{t_0}$ $\mathbf{X}^2_{t_0}$ ... $\mathbf{X}^2_{t_1}$ (b)
 ${}^G\mathbf{Z}^1_{t_0}$ $\mathbf{X}^1_{t_0}$ ... $\mathbf{X}^1_{t_1^1}$ ... $\mathbf{X}^1_{t_2^1}$ (c)	 ${}^G\mathbf{Z}^2_{t_0}$ $\mathbf{X}^2_{t_0}$ ... $\mathbf{X}^2_{t_1^2}$ ... $\mathbf{X}^2_{t_2}$ ${}^G\mathbf{Z}^1_{t_0}$ $\mathbf{R}^{2,1}_{t_1}$ $\mathbf{\Delta X}^1_{t_0 \to t_1}$ $\mathbf{X}^1_{t_0}$ $\mathbf{X}^1_{t_1^1}$ (d)
 ${}^G\mathbf{Z}^1_{t_0}$ $\mathbf{X}^1_{t_0}$ ... $\mathbf{X}^1_{t_1^1}$ ... $\mathbf{X}^1_{t_2^1}$ ... $\mathbf{X}^1_{t_3}$ ${}^G\mathbf{Z}^2_{t_0}$ $\mathbf{R}^{2,1}_{t_1}$ $\mathbf{R}^{1,2}_{t_2}$ $\mathbf{\Delta X}^2_{t_0 \to t_1^2}$ $\mathbf{\Delta X}^2_{t_1^2 \to t_2^2}$ $\mathbf{X}^2_{t_0}$ $\mathbf{X}^2_{t_1^2}$ $\mathbf{X}^2_{t_2^2}$ (e)	 ${}^G\mathbf{Z}^2_{t_0}$ $\mathbf{X}^2_{t_0}$ ... $\mathbf{X}^2_{t_1^2}$ ... $\mathbf{X}^2_{t_2}$ ... $\mathbf{X}^2_{t_3}$ ${}^G\mathbf{Z}^1_{t_0}$ $\mathbf{R}^{2,1}_{t_1}$ $\mathbf{\Delta X}^1_{t_0 \to t_1^1}$ $\mathbf{X}^1_{t_0}$ $\mathbf{X}^1_{t_1^1}$ (f)

Table 5.2: Evolution of two multi-robot factor graphs while robots take turns transmitting. In this case there are no communication failures and estimates are optimal. AUV 2 receives transmission from AUV 1 at time t_1 containing ${}^G\mathbf{Z}^1_{t_0}$, $\mathbf{\Delta X}^1_{t_0 \to t_1}$ and is able to add range factor $\mathbf{R}^{2,1}_{t_1}$ from calculation of the TOF. AUV 1 receives transmission from AUV at time t_2 containing ${}^G\mathbf{Z}^2_{t_0}$, $\mathbf{\Delta X}^2_{t_0 \to t_1}$, $\mathbf{\Delta X}^2_{t_1 \to t_2}$, $\mathbf{R}^{2,1}_{t_1}$ and is able to add range factor $\mathbf{R}^{1,2}_{t_2}$ from calculation of the TOF.

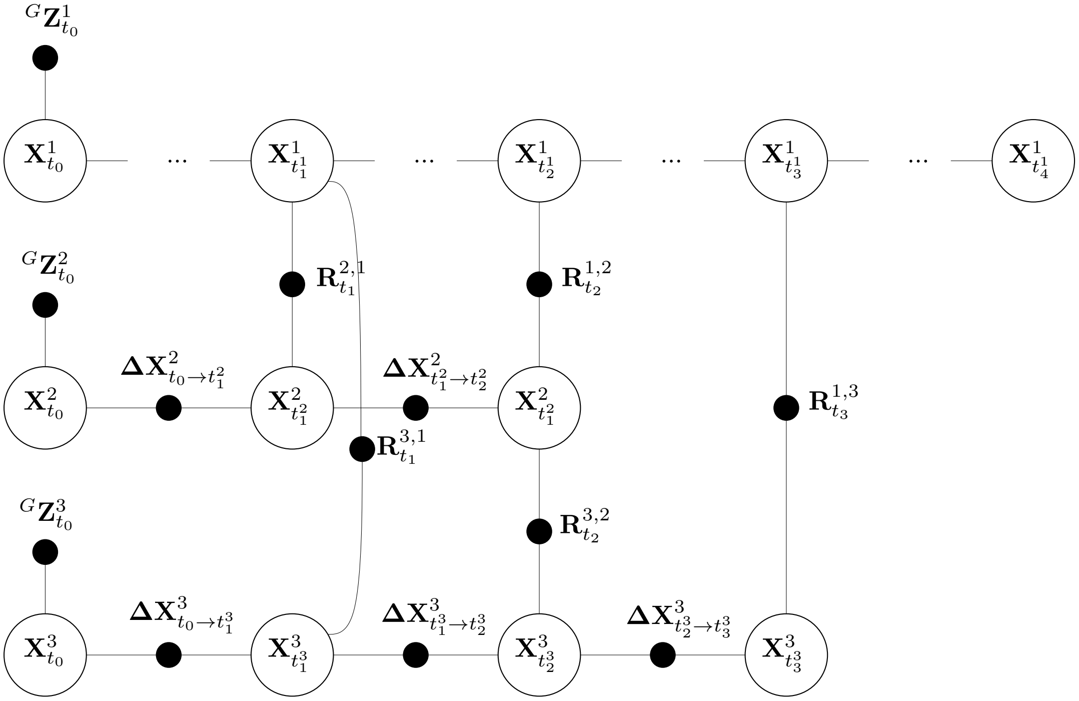


Figure 5.5: Local multi-robot factor graph on robot 1 at time t_4 after a full cycle of successful communications.

5.3.7.3 3 Robots with Random Communications Failures

For a final example, consider the case where three AUVs are cooperatively localizing, but there is a communication failure. In this case, the transmissions from AUV 2 at time t_2 to both AUVs 1 and 3 fail as shown in Fig. 5.6. The resulting local multi-robot factor graph on robot 1 at time t_4 is shown in Fig. 5.7. Robot 1 has never received any successful communications from robot 2 so there are no nodes corresponding to robot 2. At this time robot 1 is scheduled to transmit.

The incoming contact points set is given by $\mathcal{C}^1_{in} = \{-, t^1_0, t^1_3\}$ and the outgoing contact points set is given by $\mathcal{C}^1_{out} = \{-, t^1_0, t^1_1\}$. Note in the outgoing contact point set entry for AUV 2 is t_0 even though there was a successful outgoing communication at time t_1 because AUV 1 has no knowledge that it was successful (transmission of the acknowledgment bit failed).

As a result, to be transmitted should be the accumulated change in position factors ($\mathbf{\Delta X}^1_{t_0 \to t_4}$, $\mathbf{\Delta X}^1_{t_1 \to t_4}$, $\mathbf{\Delta X}^1_{t_3 \to t_4}$, $\tilde{\mathbf{Q}}^1_{t_0 \to t_4}$, $\tilde{\mathbf{Q}}^1_{t_1 \to t_4}$, $\tilde{\mathbf{Q}}^1_{t_3 \to t_4}$), the GPS measurement (${}^G\mathbf{Z}^1_{t_0}$), the range measurement ($\mathbf{R}^{1,3}_{t_3}$) and the acknowledgment bits ([0,1]) denoting failure to receive from AUV 2 and successful reception from AUV 3 on the last cycle.

However, AUV 2 did actually receive $\mathbf{\Delta X}^1_{t_0 \to t_1}$ and $\tilde{\mathbf{Q}}^1_{t_0 \to t_1}$ when they were transmitted at time t_1 and therefore can determine the appropriate accumulated change in position ($\mathbf{\Delta X}^1_{t_1 \to t_4}$, $\tilde{\mathbf{Q}}^1_{t_1 \to t_4}$) from $\mathbf{\Delta X}^1_{t_0 \to t_4} - \mathbf{\Delta X}^1_{t_0 \to t_1}$ and $\tilde{\mathbf{Q}}^1_{t_0 \to t_4} - \tilde{\mathbf{Q}}^1_{t_0 \to t_1}$ (in this case it happened to be actually sent because AUV 3 required it).

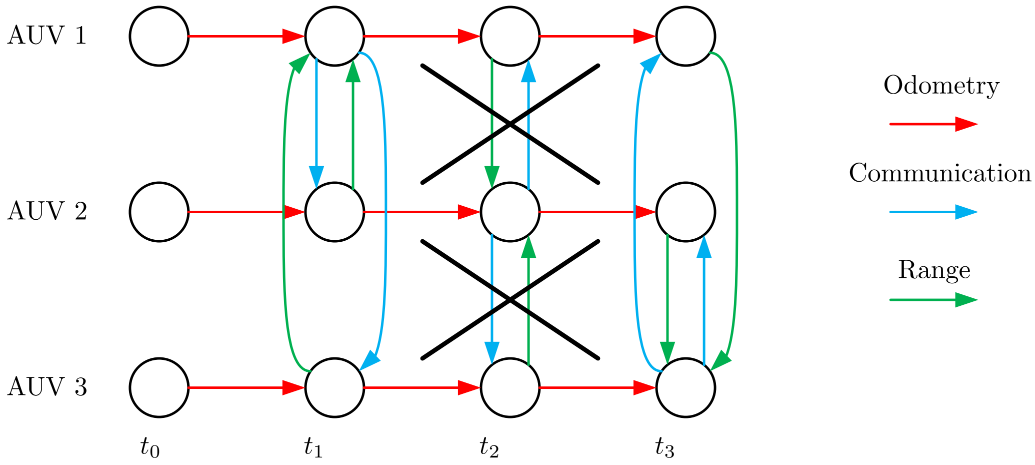


Figure 5.6: Information flow graph with three AUVs. Transmissions from AUV 2 both failed.

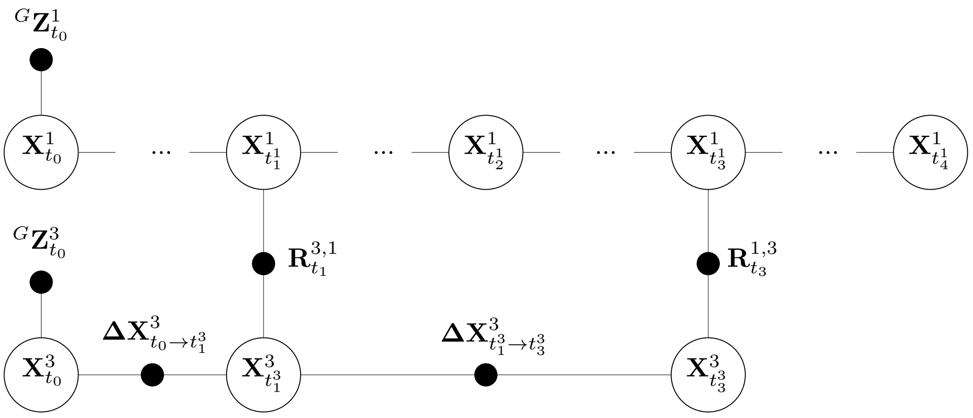


Figure 5.7: Local multi-robot pose for AUV 1 at time t_4 and scheduled for transmission. At time t_2 the transmissions from AUV 2 to both AUVs 1 and 3 failed.

5.3.8 Comparison with Other Methods

In this section we compare the proposed method with three other methods in the literature: [76] and [14] which were specifically developed for AUV cooperative localization, and then [138] which is a general framework for cooperative localization.

In [76], a method of cooperative trajectory estimation for a homogeneous team of AUVs is presented. Data from all robots is sent in a packet meaning that data can be transmitted to robot i from robot j through a non-direct route. This approach is more robust to data failures, but results in poorer scalability of the approach (in general $\mathcal{O}(N^2)$ where N is the size of the AUV team and also will backlog if communications are poorer than expected. To compensate for backlogging of data caused by poor scalability and communication failures, a "keyframes" approach is proposed, which is essentially downsampling. Only a fraction (the keyframe rate) of the data transmissions are used for inter-robot measurements, all others are used to marshal data. The multi-robot estimate is only made at the keyframe frequency, which is an integer fraction of the communications frequency. For example, for a keyframe rate of 4, 3/4 of the inter-robot range measurements are discarded resulting in suboptimality. This keyframe rate must be chosen prior to mission start based on beliefs or expectations of the system communication throughput rate. The advantage of the approach proposed here is that a much higher fraction of the inter-robot measurements can be used for estimation, and the rate that data is discarded is adaptive to the communications throughput *in situ* rather than specified beforehand. The communication channel will be used more opportunistically to exploit favorable conditions and achieve better performance. In addition, the choice to transmit all sensor data from all robots results in poorer scalability, both as a function of the team size, and as a function of time in the event that communications are worse than expected.

In [14] a framework for consistent filtering amongst a team of AUVs is presented. A bank of EKFs is maintained on-board each robot that provide estimates that are independent of each combination of AUVs in the team. This is referred to as an "interleaved update" approach. While the main objective of guaranteeing consistency of the estimates is achieved, it is at the expense of being overly conservative. Estimated covariances are actually much larger than the covariances that would be obtained by a centralized equivalent estimator. In stark contrast to [76] where only raw sensor data is transmitted, in this case only state estimates are transmitted. However, the estimates from all N^2 filters have to be transmitted to guarantee consistency.

A last comparison is made with [138], a general framework for optimal delayed-state filtering for cooperative localization is presented. In the approach, when all data from all robots is received up to some time, termed the checkpoint time, then the raw data can be safely replaced with a multi-robot estimate. The advantage of the approach is that it guarantees optimality of the estimate at the checkpoint time. However, this is at the expense of having to communicate the full filtered covariance matrix at the checkpoint time as well as any sensor data since the checkpoint time. Graphs highlighting the data packet size scalability of the proposed algorithm compared with [76], [14], and [138] are shown in Figs. 5.8 and 5.9. In Fig. 5.8 the packet size is expressed as function of the number of consecutively failed communications attempts for a team size of three AUVs. A key advantage of the proposed approach is that the data packet size is constant in this case, although the contents of the packets are changing. In Fig. 5.9, the packet size is expressed as a function of the size of the AUV team, N, assuming no communications failures. The proposed algorithm is the only one that scales linearly.

A reasonable assumption is that eight bits can be used to represent each floating point number, so the vertical axis in Figs. 5.8 and 5.9 can also be represented as a number of bytes. It is also important to note that these issues will compound, meaning that as team size grows and/or consecutive communications dropouts occur, data will quickly backlog and the 192byte limit per packet will be easily exceeded.

In comparison with [138] and [76], it should be noted that the reduced data throughput does have an impact on the performance of the estimator. Both [138] and [76] can guarantee optimal filtered estimates. When communications dropouts occur, we can no longer guarantee optimality, however, as shown Section 5.5, in all but the most severe dropout conditions, estimates remain close to optimal for the entire trajectory.

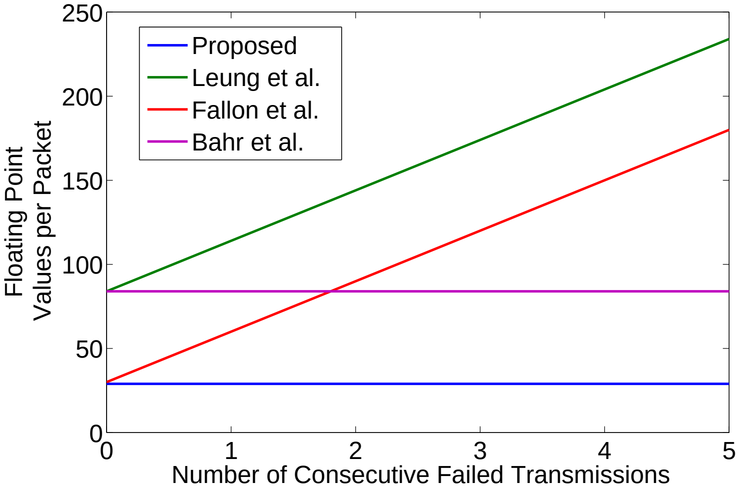


Figure 5.8: The size of data packets as a function of consecutive failed communications for the algorithm proposed here and three other comparable algorithms.

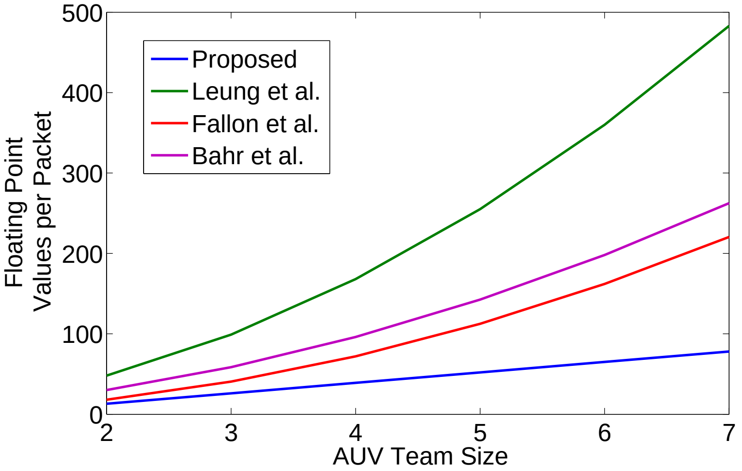


Figure 5.9: The size of data packets as a function of AUV team size for the algorithm proposed here and three other comparable algorithms.

5.4 Experimental Setup

The experimental setup is similar to that of the previous chapters, except that now the acoustic modem has to be added. On the Iver2 AUV the acoustic modem, in this case the micromodem developed at Woods Hole Oceanographic Institute, is interfaced directly to the backseat computer. The micromodem driver and also tools for simulating the modem and interfacing it with MOOS are provided by the Goby suite of tools [199]. Of particular importance is the 'pAcommsHandler' application which handles the interfacing between the MOOSDB and the actual or simulated modem. As in the setup for the experiments in Chapter 4 raw sensor data is passed from the frontseat to the backseat.

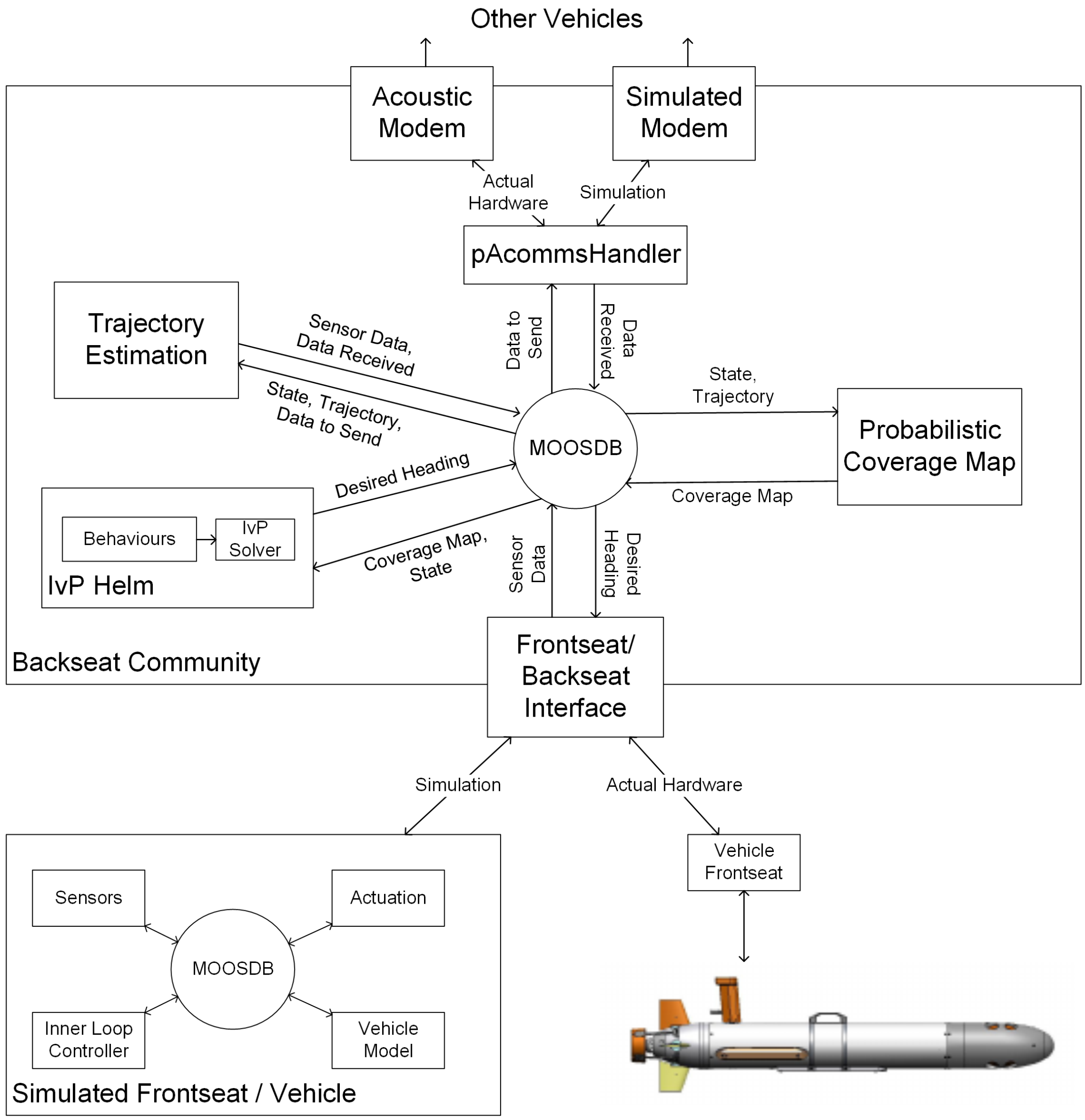


Figure 5.10: Overview of multiple robots mission setup within MOOS.

5.5 Results

5.5.1 Tests with Logged Data

In order to evaluate the effect of varying the quality of the communications channel in the most realistic way, real sensor data gathered from the field was "played back" using the MOOS application 'uPlayback' and the communications channel was simulated and controlled. In this way the effect of the communications channel performance can be directly related to the quality of the cooperative AUV trajectory estimation because in each simulated run the same real sensor data from the DVL, GPS, and compass was used.

In the presentation of the results, I first refer back to the two stated objectives in Section 5.1.2 which are restated here. Coverage with multiple AUVs will be more efficient if

1. AUVs minimize surfacing for GPS fixes,
2. Uncertainty is reduced over the entire AUV trajectory.

This should be achieved without violating the constraints imposed by the underwater acoustic channel.

5.5.1.1 Objective 1: Less Frequent Surfacing for GPS Fixes

Normally, AUVs surface when their location uncertainty reaches a threshold. Therefore, the longer that an AUV's location uncertainty can be maintained below the threshold, the less frequently it needs to surface.

This objective is achieved in the proposed method in two ways that compound.

First, the fact that robots are communicating and making relative measurements causes their instantaneous location uncertainty to grow more slowly. Consider Fig. 5.11 which shows the filtered location estimate uncertainties for one of two robots using the proposed algorithm. Throughout the simulation the robots are communicating acoustically and measuring their relative ranges. Each reception of an acoustic transmission reduces the instantaneous location uncertainty, resulting in the sawtooth pattern shown. From the figure, even for a 20% success rate of data communications (red plot) there is a significant advantage over no communication (100% failure). In addition, for a success rate of 50% or higher the robot is able to recover a near-optimal estimate (optimal is represented by 100% success) after a successful communication.

One of the essential advantages of the proposed algorithm is that the amount of data to be transmitted does not increase in the event of failed communications. This is certainly not the case for other algorithms proposed in the literature [76, 14].

Second, the surfacing of one robot for a GPS fix can bound the localization error of all robots in the team. For illustration of this refer to Fig. 5.12. In this plot communications was assumed 100% successful. The plot shows both the instantaneous, or filtered, covariances and also the smoothed covariances obtained by re-optimizing the trajectory at time $t = 500s$. The uncertainty on the vertical axis of Fig. 5.12 is represented as the sum of the autocovariances: $\sigma^2_{x_t x_t} + \sigma^2_{y_t y_t}$. AUV 1 surfaces for GPS twice at which point the instantaneous location uncertainty is reduced to $\approx 3m$. In between the GPS fixes, the uncertainties of the two robots are similar, meaning that AUV 2 obtained most of the benefit of surfacing without ever having to do so, thus saving energy.

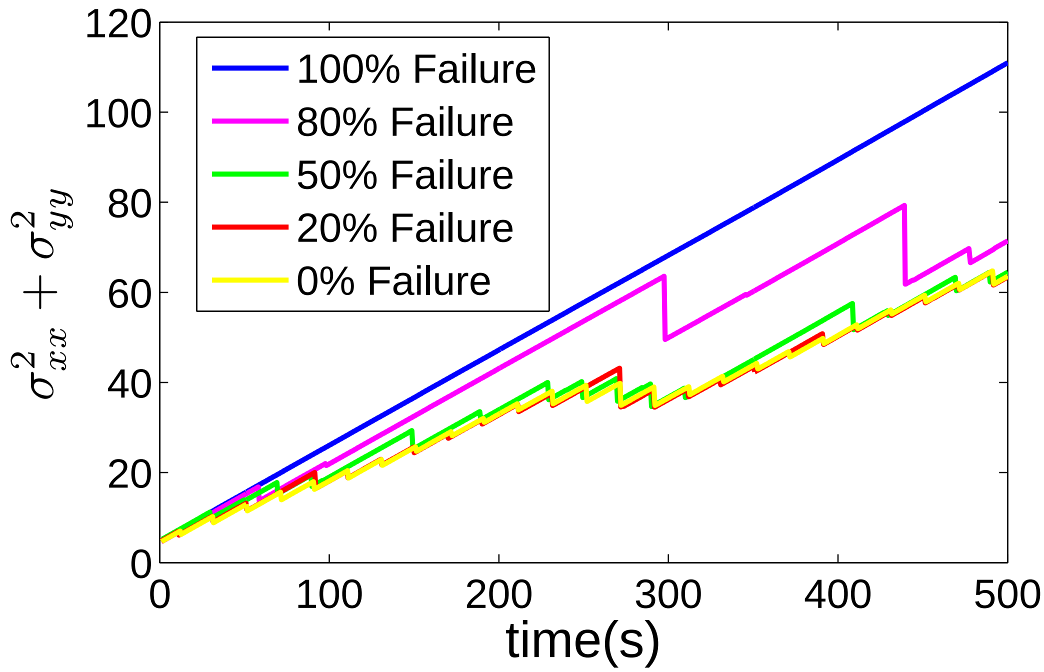


Figure 5.11: The instantaneous (filtered) location uncertainty of AUV 1 using the proposed method for different dropout rates.

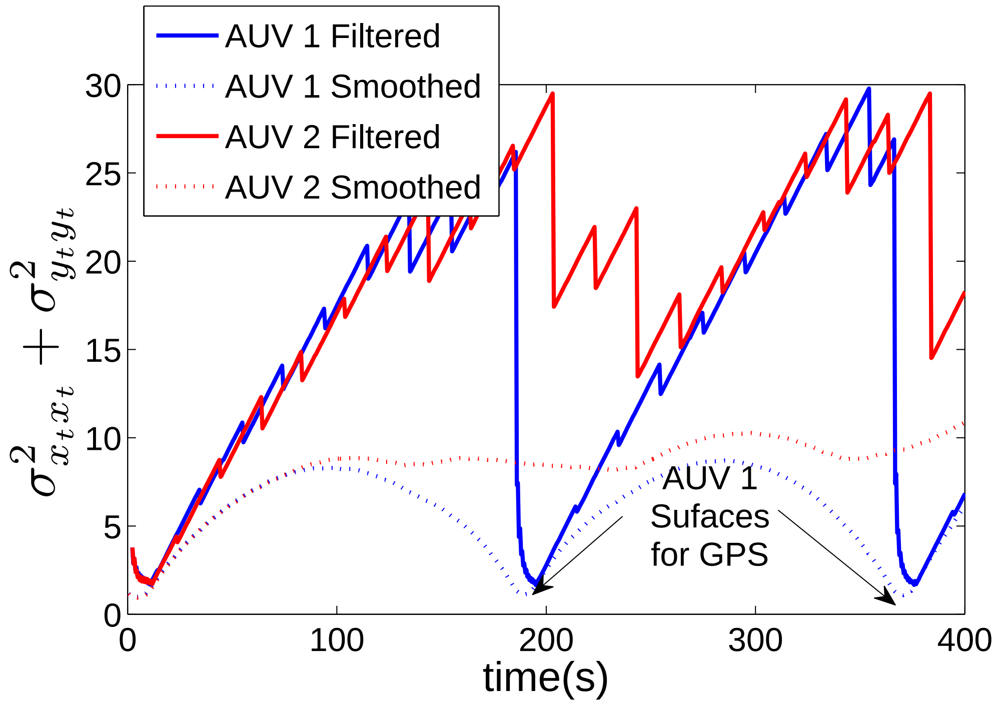


Figure 5.12: The instantaneous (filtered) and smoothed uncertainties of two robots cooperatively localizing using the proposed method. AUV 1 surfaces for GPS twice thus bounding the uncertainty growth for both robots.

5.5.1.2 Objective 2: Uncertainty Reduced over Entire Trajectory of all AUVs

The benefits of estimating the entire robot trajectory were explicitly shown in Chapter 4. As a result, one of the main design objectives was to use inter-AUV measurements to estimate an entire trajectory. The benefit of full trajectory illustration is shown in Fig. 5.12 by comparing the instantaneous location uncertainty (solid lines) with the smoothed estimate uncertainty (broken lines). In this case, since inter-AUV measurements are intermittent and provide information about the AUV location directly, smoothing has a large effect.

Fig. 5.13 shows the smoothed estimate uncertainties for the cases from Fig. 5.11 that are computed at time $t = 500s$. Fig. 5.13 shows that, even for a success rate of only 50%, the smoothed estimate uncertainties are very close to the optimal case (100% success).

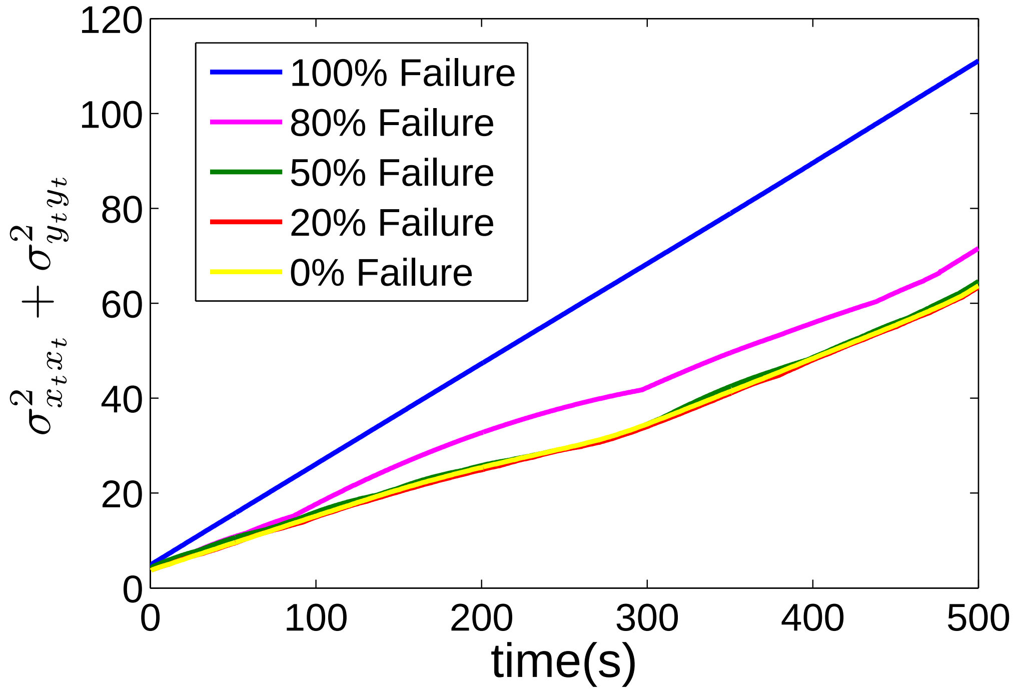


Figure 5.13: The location uncertainty of the smoothed estimate at time $t = 500s$ of AUV 1 using the proposed method for different dropout rates.

5.6 Summary

In this chapter we have presented a least-squares optimization framework as the basis for a cooperative AUV trajectory estimation algorithm. The essential advantage this algorithm has over others proposed in the literature is that data throughput scales linearly with the number of robots in the team and is independent of the dropout rate on the unreliable acoustic channel. The algorithm is able to produce estimates of full robot trajectories that are near-optimal even in the case of a 50% dropout rate.

For multi-AUV coverage this has two main benefits: 1) That robots minimize surfacing for GPS fixes and 2) Coverage estimates improve due to the reduced uncertainty over the entire robot trajectory.

The result is that time can be saved, covertness can be increased, and the time to reacquire targets can be reduced through better localization.

Chapter 6

Conclusion

6.1 Summary

We have presented a novel approach to AUV seabed surveying. Historical approaches to plan these surveys use simple heuristics such as lawn-mower or zig-zag patterns. These plans are made prior to surveying and are not adapted while the vehicle is underway.

Seabed coverage with an AUV is posed here as a sensor coverage problem. What makes it particularly challenging is that the performance of the coverage sensor, the sidescan sonar, is highly nonlinear and variable with respect to many factors. Since these factors are not necessarily known beforehand the result is that preplanned paths can be suboptimal.

In cases of coverage on land or in air, it is often reasonable to assume that the sensor is fairly accurately localized and so the uncertainty in the location of the sensor can be neglected. Once an AUV submerges, it loses access to a global position reference and its position uncertainty starts to grow. At some point, this uncertainty is no longer negligible and must be considered in the coverage model. For a mission like MCM this is an important consideration.

To bound the location uncertainty, the AUV will periodically surface for a GPS fix. This is very time-consuming. By using multiple AUVs to perform the survey we can greatly reduce the need for each of them to surface as frequently.

6.1.1 Overview of Contributions

A review of the contributions of the thesis include:

- A sensor-driven approach to AUV seabed coverage planning that is demonstrated through an AUV MCM example,
- An explicit link between area coverage and state estimation for robotics, including a novel adaptive sliding window approach for estimating the AUV trajectory that is well-suited to the coverage problem, and online planning strategies that operate within the new probabilistic model of coverage. The algorithms are applied to the AUV seabed surveying problem.
- An algorithm for multi-AUV cooperative trajectory estimation that scales well with the cooperative size (number of AUVs) and is suitable for multiple AUVs performing seabed surveying operations.

6.2 Avenues for Future Work

The following extensions to the work presented here would be very interesting:

- Including uncertainty with obstacle locations into the probabilistic coverage framework within a full SLAM scenario would be a natural progression,
- Evaluating the effects of marginalizing out the along track error and heading uncertainty from the AUV state estimate in Chapter 4 would be interesting,
- Developing multi-AUV planning methods where the surveys are in the same workspace and can track the coverage of other team members would be extremely challenging but could be possible.

6.3 Final Comments

Underwater surveying, and particularly MCM is inherently a very dangerous task. Improvements to the efficiency of AUV surveys result in less time required at sea and less personnel in harms way. It is critical for decision makers to get accurate estimates of the risks associated with moving personnel or assets through a body of water. The worst case scenario is that the perceived risk is lower than the actual risk, in which case decisions can be made that put people and assets at unnecessary risk.

References

[1] Ercan U. Acar and Howie Choset, *Sensor-based coverage of unknown environments: Incremental construction of Morse decompositions*, International Journal of Robotics Research **21** (2002), 345–367.

[2] Ercan U. Acar, Howie Choset, Alfred A. Rizzi, Prasad N. Atkar, and Douglas Hull, *Morse decompositions for coverage tasks*, I. J. Robotic Res. **21** (2002), no. 4, 331–344.

[3] Ercan U. Acar, Howie Choset, Yangang Zhang, and Mark J. Schervish, *Path planning for robotic demining: Robust sensor-based coverage of unstructured environments and probabilistic methods*, I. J. Robotic Res. **22** (2003), no. 7-8, 441–466.

[4] E.U. Acar, H. Choset, and Ji Yeong Lee, *Sensor-based coverage with extended range detectors*, Robotics, IEEE Transactions on **22** (2006), no. 1, 189–198.

[5] A. Alvarez, A. Caiti, and R. Onken, *Evolutionary path planning for autonomous underwater vehicles in a variable ocean*, Oceanic Engineering, IEEE Journal of **29** (2004), no. 2, 418–429.

[6] R. Aragues, L. Carlone, G. Calafiore, and C. Sagues, *Multi-agent localization from noisy relative pose measurements*, Robotics and Automation (ICRA), 2011 IEEE International Conference on, 2011, pp. 364–369.

[7] B. Armstrong, J. Pentzer, D. Odell, T. Bean, J. Canning, D. Pugsley, J. Frenzel, M. Anderson, and D. Edwards, *Field measurement of surface ship magnetic signature using multiple AUVs*, OCEANS 2009, MTS/IEEE Biloxi - Marine Technology for Our Future: Global and Local Challenges, Oct. 2009, pp. 1–9.

[8] B. Armstrong, E. Wolbrecht, and D.B. Edwards, *AUV navigation in the presence of a magnetic disturbance with an extended Kalman filter*, OCEANS 2010 IEEE - Sydney, May 2010, pp. 1–6.

[9] J. Aulinas, A. Fazlollahi, J. Salvi, X. Llado, Y. Petillot, J. Sawas, and R. Garca, *Robust automatic landmark detection for underwater SLAM using side-scan sonar imaging*, Proceedings of the 11th International Conference on Mobile Robots and Competitions (ROBOTICA'11) (Lisboa, Portugal), Apr. 2011.

[10] J. Aulinas, X. Llado and, J. Salvi, and Y.R. Petillot, *Selective submap joining for underwater large scale 6-DOF SLAM*, Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, Oct. 2010, pp. 2552–2557.

[11] F. Aurenhammer, *Voronoi diagrams - a survey of a fundamental geometric structure*, ACM Computing Surveys **23** (1991), 345–405.

[12] Claus B and Bachmayer R, *Development of a magetometry system for an underwater glider*, UUST, 2011, pp. 1–7.

[13] Sanghoon Baek, Tae-Kyeong Lee, O. H. Se-Young, and Kwangro Ju, *Integrated on-line localization, mapping and coverage algorithm of unknown environments for robotic vacuum cleaners based on minimal sensing*, Advanced Robotics **25** (2011), no. 13-14, 1651–1673.

[14] A. Bahr, M.R. Walter, and J.J. Leonard, *Consistent cooperative localization*, Robotics and Automation, 2009. ICRA '09. IEEE International Conference on, May 2009, pp. 3415–3422.

[15] Alexander Bahr and John J. Leonard, *Cooperative localization for autonomous underwater vehicles*, International Symposium on Experimental Robotics, 2006, pp. 1–10.

[16] Alexander Bahr, John J. Leonard, and Maurice F. Fallon, *Cooperative localization for autonomous underwater vehicles*, The International Journal of Robotics Research **28** (2009), no. 6, 714–728.

[17] Tim Bailey, M. Bryson, Hua Mu, J. Vial, L. McCalman, and H. Durrant-Whyte, *Decentralised cooperative localisation for heterogeneous teams of mobile robots*, Robotics and Automation (ICRA), 2011 IEEE International Conference on, 2011, pp. 2859–2865.

[18] S. Barkby, S. Williams, O. Pizarro, and M. Jakuba, *An efficient approach to bathymetric SLAM*, Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, Oct. 2009, pp. 219–224.

[19] ______, *Incorporating prior maps with bathymetric distributed particle SLAM for improved AUV navigation and mapping*, OCEANS 2009, MTS/IEEE Biloxi - Marine Technology for Our Future: Global and Local Challenges, Oct. 2009, pp. 1–7.

[20] Stephen Barkby, Stefan B. Williams, Oscar Pizarro, and Michael V. Jakuba, *A featureless approach to efficient bathymetric SLAM using distributed particle mapping*, Journal of Field Robotics **28** (2011), no. 1, 19–39.

[21] Stephen Barkby, Stefan B Williams, Oscar Pizarro, and Michael V Jakuba, *Bathymetric particle filter SLAM using trajectory maps*, International Journal of Robotics Research **31** (2012), no. 12, 1409–1430.

[22] J. Barraquand, B. Langlois, and J.-C. Latombe, *Numerical potential field techniques for robot path planning*, Systems, Man and Cybernetics, IEEE Transactions on **22** (1992), no. 2, 224–241.

[23] P. Batista, C. Silvestre, and P. Oliveira, *A sensor-based controller for homing of underactuated AUVs*, Robotics, IEEE Transactions on **25** (2009), no. 3, 701–716.

[24] Pedro Tiago Martins Batista, Carlos Silvestre, and Paulo Jorge Ramalho Oliveira, *Optimal position and velocity navigation filters for autonomous vehicles.*, Automatica (2010), 767–774.

[25] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool, *Speeded-up robust features (SURF)*, Comput. Vis. Image Underst. **110** (2008), 346–359.

[26] Michael Benjamin, Paul Newman, Henrik Schmidt, and John Leonard, *An overview of MOOS-IvP and a brief users guide to the IvP Helm autonomy software*, http://dspace.mit.edu/bitstream/handle/1721.1/45569/MIT-CSAIL-TR-2009-028.pdf, Jun. 2009.

[27] ______, *An overview of MOOS-IvP and a brief users guide to the IvP Helm autonomy software*, http://dspace.mit.edu/bitstream/handle/1721.1/45569/MIT-CSAIL-TR-2009-028.pdf, June 2009.

[28] Michael R. Benjamin, Henrik Schmidt, Paul M. Newman, and John J. Leonard, *Nested Autonomy for Unmanned Marine Vehicles with MOOS-IvP*, Journal of Field Robotics **27** (2010), no. 6, 834–875.

[29] J. Biggs and W. Holderbaum, *Optimal kinematic control of an autonomous underwater vehicle*, Automatic Control, IEEE Transactions on **54** (2009), no. 7, 1623–1626.

[30] B. Bingham, *Predicting the navigation performance of underwater vehicles*, Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, Oct. 2009, pp. 261–266.

[31] Brian Bingham and W. Seering, *Hypothesis grids: improving long baseline navigation for autonomous underwater vehicles*, Oceanic Engineering, IEEE Journal of **31** (2006), no. 1, 209–218.

[32] J. Binney, A. Krause, and G.S. Sukhatme, *Informative path planning for an autonomous underwater vehicle*, Robotics and Automation (ICRA), 2010 IEEE International Conference on, May 2010, pp. 4791–4796.

[33] A.N. Bishop, B. Fidan, B.D.O. Anderson, K. Dogancay, and P.N. Pathirana, *Optimal range-difference-based localization considering geometrical constraints*, Oceanic Engineering, IEEE Journal of **33** (2008), no. 3, 289–301.

[34] L. Blackmore, M. Ono, and B.C. Williams, *Chance-constrained optimal path planning with obstacles*, Robotics, IEEE Transactions on **27** (2011), no. 6, 1080–1094.

[35] M. Blanco and P.A. Wilson, *Autonomous underwater vehicle minimum-time navigation in a current field*, OCEANS 2010, Sept 2010, pp. 1–4.

[36] M. Bosse, N. Nourani-Vatani, and J. Roberts, *Coverage algorithms for an under-actuated car-like vehicle in an uncertain environment*, Robotics and Automation, 2007 IEEE International Conference on, april 2007, pp. 698–703.

[37] F. Bourgault, A.A. Makarenko, S.B. Williams, B. Grocholsky, and H.F. Durrant-Whyte, *Information based adaptive robotic exploration*, Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on, vol. 1, 2002, pp. 540–545.

[38] F. Bourque and B.U. Nguyen, *Optimal sensor configurations for rectangular target detection*, Control and Automation (ICCA), 2011 9th IEEE International Conference on, 2011, pp. 455–460.

[39] M. Bryson and S. Sukkarieh, *Observability analysis and active control for airborne SLAM*, Aerospace and Electronic Systems, IEEE Transactions on **44** (2008), no. 1, 261–280.

[40] Heiko Bülow and Andreas Birk, *Spectral registration of noisy sonar data for underwater 3D mapping*, Auton. Robots **30** (2011), 307–331.

[41] Cheghui Cai and Silvia Ferrari, *Information-driven sensor path planning by approximate cell decomposition*, IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics **39** (2009), no. 3, 672–689.

[42] Alec Cameron and Hugh Durrant-Whyte, *A Bayesian approach to optimal sensor placement*, The International Journal of Robotics Research **9** (1990), no. 5, 70–88.

[43] J.F. Canny and M.C. Lin, *An opportunistic global path planner*, Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on, May 1990, pp. 1554–1559.

[44] A.R. Cassandra, L.P. Kaelbling, and J.A. Kurien, *Acting under uncertainty: discrete Bayesian models for mobile-robot navigation*, Intelligent Robots and Systems '96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on, vol. 2, nov 1996, pp. 963–972.

[45] Philip Chapple, *Automated detection and classification in high-resolution sonar imagery for autonomous underwater vehicle operations*, DSTO-GD-0537, Maritime Operations Division DSTO Defence Science and Technology Organisation, December 2002.

[46] Chi-Tsun Cheng, K. Fallahi, H. Leung, and C.K. Tse, *An AUVs path planner using genetic algorithms with a deterministic crossover operator*, Robotics and Automation (ICRA), 2010 IEEE International Conference on, may 2010, pp. 2995–3000.

[47] H. Choset, W. Burgard, S. Hutchinson, G. Kantor, L. E. Kavraki, K. Lynch, and S. Thrun, *Principles of robot motion: Theory, agorithms, and implementation*, MIT Press, June 2005.

[48] Howie Choset, *Coverage of known spaces: the Boustrophedon cellular decomposition*, Autonomous Robots **9** (2000), 247–253.

[49] ______, *Coverage for robotics - a survery of recent results*, Annals of Mathematics and Artificial Intelligence **31** (2001), 113–126.

[50] P. Corke, C. Detweiler, M. Dunbabin, M. Hamilton, D. Rus, and I. Vasilescu, *Experiments with underwater robot localization and tracking*, Robotics and Automation, 2007 IEEE International Conference on, Apr. 2007, pp. 4556–4561.

[51] B.P. Crosbie, M. Anderson, E. Wolbrecht, J. Canning, and D.B. Edwards, *Synchronous navigation of AUVs using WHOI micro-modem 13-bit communications*, OCEANS 2010, Sep. 2010, pp. 1–5.

[52] 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*, OCEANS, 2005. Proceedings of MTS/IEEE, 2005, pp. 730–735.

[53] Colin Das, Aaron Becker, and Timothy Bretl, *Probably approximately correct coverage for robots with uncertainty*, Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, sept. 2011, pp. 1160–1166.

[54] G. Davies and E. Signell, *ESPRESSO scientific user guide*, NURC-SP-2006-003, NATO Underwater Research Centre, 2006.

[55] Timothy A. Davis, John R. Gilbert, Stefan I. Larimore, and Esmond G. Ng, *A column approximate minimum degree ordering algorithm*, ACM Trans. Math. Softw. **30** (2004), no. 3, 353–376.

[56] M. De Agostino, A.M. Manzino, and M. Piras, *Performances comparison of different MEMS-based IMUs*, Position Location and Navigation Symposium (PLANS), 2010 IEEE/ION, 2010, pp. 187–201.

[57] Frank Dellaert and Michael Kaess, *Square root SAM: Simultaneous localization and mapping via square root information smoothing*, International Journal of Robotic Research **25** (2006), no. 12, 1181–1203.

[58] J. E. Dennis, Jr. and Robert B. Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations (Classics in Applied Mathematics, 16)*, Soc for Industrial & Applied Math, 1996.

[59] Carrick Detweiler, John Leonard, Daniela Rus, and Seth Teller, *Passive mobile robot localization within a fixed beacon field*, 7th International Workshop on Algorithmic Foundations of Robotics, 2006, pp. 1–6.

[60] Y. Dieudonne, O. Labbani-Igbida, and F. Petit, *Deterministic robot-network localization is hard*, Robotics, IEEE Transactions on **26** (2010), no. 2, 331–339.

[61] D.P. Eickstedt and S.R. Sideleau, *The backseat control architecture for autonomous robotic vehicles: A case study with the Iver2 AUV*, Marine Technology Society Journal **44** (2010), no. 4, 42–54.

[62] Armagan Elibol, Nuno Gracias, and Rafael Garcia, *Augmented state extended Kalman filter combined framework for topology estimation in large-area underwater mapping*, Journal of Field Robotics **27** (2010), no. 5, 656–674.

[63] B. Englot and F. Hover, *Inspection planning for sensor coverage of 3D marine structures*, Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, Oct. 2010, pp. 4412–4417.

[64] R.M. Eustice, O. Pizarro, and H. Singh, *Visually augmented navigation for autonomous underwater vehicles*, Oceanic Engineering, IEEE Journal of **33** (2008), no. 2, 103–122.

[65] R.M. Eustice, H. Singh, and J.J. Leonard, *Exactly sparse delayed-state filters for view-based SLAM*, Robotics, IEEE Transactions on **22** (2006), no. 6, 1100–1114.

[66] R.M. Eustice, L.L. Whitcomb, H. Singh, and M. Grund, *Experimental results in synchronous-clock one-way-travel-time acoustic navigation for autonomous underwater vehicles*, Robotics and Automation, 2007 IEEE International Conference on, Apr. 2007, pp. 4257–4264.

[67] Ryan Eustice, *Large-area visually augmented navigation for autonomous underwater vehicles*, Ph.D. thesis, Massachusetts Institute of Technology and the Woods Hole Oceanographic Institution, 2005.

[68] Ryan M. Eustice, Hanumant Singh, John J. Leonard, and Matthew R. Walter, *Visually mapping the RMS Titanic: Conservative covariance estimates for SLAM information filters*, The International Journal of Robotics Research **25** (2006), no. 12, 1223–1242.

[69] 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** (2011), no. 1, 121–136.

[70] M. Faied, P. Kabamba, B. Hyun, and A. Girard, *Path planning for optimal classification*, Decision and Control (CDC), 2012 IEEE 51st Annual Conference on, 2012, pp. 520–527.

[71] N. Fairfield and D. Wettergreen, *Active localization on the ocean floor with multibeam sonar*, OCEANS 2008, Sep. 2008, pp. 1–10.

[72] Nathaniel Fairfield, George Kantor, Dominic Jonak, and David Wettergreen, *Autonomous exploration and mapping of flooded sinkholes*, The International Journal of Robotics Research **29** (2010), no. 6, 748–774.

[73] Nathaniel Fairfield, George Kantor, and David Wettergreen, *Real-time SLAM with octree evidence grids for exploration in underwater tunnels*, Journal of Field Robotics **24** (2007), no. 1-2, 03–21.

[74] Maurice F. Fallon, Michael Kaess, Hordur Johannsson, and John J. Leonard, *Efficient AUV navigation fusing acoustic ranging and side-scan sonar*, Robotics and Automation (ICRA), 2011 IEEE International Conference on, May 2011, pp. 2398–2405.

[75] Maurice F. Fallon, Georgos Papadopoulos, John J. Leonard, and Nicholas M. Patrikalakis, *Cooperative AUV navigation using a single maneuvering surface craft*, International Journal of Robotics Research **29** (2010), 1461–1474.

[76] M.F. Fallon, G. Papadopoulos, and J.J. Leonard, *A measurement distribution framework for cooperative navigation using multiple AUVs*, Robotics and Automation (ICRA), 2010 IEEE International Conference on, May 2010, pp. 4256–4263.

[77] Cheng Fang and S. Anstee, *Coverage path planning for harbour seabed surveys using an autonomous underwater vehicle*, OCEANS 2010 IEEE - Sydney, May 2010, pp. 1–8.

[78] J. Fawcett, V. Myers, D. Hopkin, A. Crawford, M. Couillard, and B. Zerr, *Multiaspect classification of sidescan sonar images: Four different approaches to fusing single-aspect information*, Oceanic Engineering, IEEE Journal of **35** (2010), no. 4, 863–876.

[79] B. Ferreira, A. Matos, and N. Cruz, *Single beacon navigation: Localization and control of the MARES AUV*, OCEANS 2010, Sep. 2010, pp. 1–9.

[80] A. Folk, B. Armstrong, E. Wolbrecht, H.F. Grip, M. Anderson, and D. Edwards, *Autonomous underwater vehicle navigation using moving baseline on a target ship*, OCEANS 2010, Sep. 2010, pp. 1–7.

[81] Dieter Fox, Wolfram Burgard, Hannes Kruppa, and Sebastian Thrun, *A probabilistic approach to collaborative multi-robot localization*, Auton. Robots **8** (2000), no. 3, 325–344.

[82] Yoav Gabriely and Rimon, *Spanning-tree based coverage of continuous areas by a mobile robot*, Annals of Mathematics and Artificial Intelligence **31** (2001), 77–98.

[83] E. Gallimore, J. Partan, I. Vaughn, S. Singh, J. Shusta, and L. Freitag, *The WHOI micromodem-2: A scalable system for acoustic communications and networking*, OCEANS 2010, Sep. 2010, pp. 1–7.

[84] B. Garau, A. Alvarez, and G. Oliver, *AUV navigation through turbulent ocean environments supported by onboard H-ADCP*, Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on, May 2006, pp. 3556–3561.

[85] A. Gasparri, B. Krishnamachari, and G. Sukhatme, *A framework for multi-robot node coverage in sensor networks*, Ann. Math Artif. Intell. **52** (2008), 281–305.

[86] S.K. Ghosh, J.W. Burdick, A. Bhattacharya, and S. Sarkar, *Online algorithms with discrete visibility - exploring unknown polygonal environments*, Robotics Automation Magazine, IEEE **15** (2008), no. 2, 67–76.

[87] B. Grocholsky, A. Makarenko, and H. Durrant-Whyte, *Information-theoretic coordinated control of multiple sensor platforms*, Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on, vol. 1, Sept. 2003, pp. 1521–1526.

[88] Ben Grocholsky, *Information-theoretic control of multiple sensor platforms*, Ph.D. thesis, Australian Centre for Field Robotics, University of Sydney, 2002.

[89] O. Hegrenaes and E. Berglund, *Doppler water-track aided inertial navigation for autonomous underwater vehicle*, OCEANS 2009 - EUROPE, May 2009, pp. 1–10.

[90] O. Hegrenaes and O. Hallingstad, *Model-aided INS with sea current estimation for robust underwater navigation*, Oceanic Engineering, IEEE Journal of **36** (2011), no. 2, 316–337.

[91] E. Hernandez, P. Ridao, D. Ribas, and A. Mallios, *Probabilistic sonar scan matching for an AUV*, Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, Oct. 2009, pp. 255–260.

[92] G.M. Hoffmann and C.J. Tomlin, *Mobile sensor network control using mutual information methods and particle filters*, Automatic Control, IEEE Transactions on **55** (2010), no. 1, 32–47.

[93] Geoffrey A. Hollinger, Brendan Englot, Franz S. Hover, Urbashi Mitra, and Gaurav S. Sukhatme, *Active planning for underwater inspection and the benefit of adaptivity*, I. J. Robotic Res. **32** (2013), no. 1, 3–18.

[94] Geoffrey A. Hollinger, Arvind A. Pereira, Valerio Ortenzi, and Gaurav S. Sukhatme, *Towards improved prediction of ocean processes using statistical machine learning*, Robotics: Science and Systems Workshop on Robotics for Environmental Monitoring (Sydney, Australia), Jul. 2012.

[95] Franz S Hover, Ryan M Eustice, Ayoung Kim, Brendan Englot, Hordur Johannsson, Michael Kaess, and John J Leonard, *Advanced perception, navigation and planning for autonomous in-water ship hull inspection*, The International Journal of Robotics Research **31** (2012), no. 12, 1445–1464.

[96] A. Howard, M.J. Matark, and G. Sukhatme, *Localization for mobile robot teams using maximum likelihood estimation*, Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on, vol. 1, 2002, pp. 434–439.

[97] Guoquan Huang, Nikolas Trawny, Anastasios Mourikis, and Stergios Roumeliotis, *Observability-based consistent EKF estimators for multi-robot cooperative localization*, Autonomous Robots **30** (2011), 99–122.

[98] Guoquan P. Huang, A.I. Mourikis, and S.I. Roumeliotis, *An observability-constrained sliding window filter for SLAM*, Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, 2011, pp. 65–72.

[99] Luyue Huang, Bo He, and Tao Zhang, *An autonomous navigation algorithm for underwater vehicles based on inertial measurement units and sonar*, Informatics in Control, Automation and Robotics (CAR), 2010 2nd International Asia Conference on, vol. 1, Mar. 2010, pp. 311–314.

[100] Yifeng Huang and K. Gupta, *Collision-probability constrained PRM for a manipulator with base pose uncertainty*, Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, oct. 2009, pp. 1426–1432.

[101] Baro Hyun, *Optimal information-based classification*, Ph.D. dissertation, University of Michigan, 2011.

[102] Baro Hyun, Pierre T. Kabamba, and Anouck R. Girard, *Optimally-informative path planning for dynamic Bayesian classification*, Optimization Letters (2012), no. 6, 1627–1642.

[103] E. Rimon I. Kamon and E. Rivlin, *Tangentbug: A range-sensor based navigation algorithm*, International Journal of Robotics Research **17** (1998), no. 9, 934–953.

[104] Michael V. Jakuba, Christopher N. Roman, Hanumant Singh, Christopher Murphy, Clayton Kunz, Claire Willis, Taichi Sato, and Robert A. Sohn, *Long-baseline acoustic navigation for under-ice autonomous underwater vehicle operations*, Journal of Field Robotics **25** (2008), no. 11-12, 861–879.

[105] L. Jaulin, *A nonlinear set membership approach for the localization and map building of underwater robots*, Robotics, IEEE Transactions on **25** (2009), no. 1, 88–98.

[106] Jian Jin and Lie Tang, *Coverage path planning on three-dimensional terrain for arable farming*, J. Field Robotics **28** (2011), no. 3, 424–440.

[107] H. Johannsson, M. Kaess, B. Englot, F. Hover, and J. Leonard, *Imaging sonar-aided navigation for autonomous underwater harbor surveillance*, Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, Oct. 2010, pp. 4396–4403.

[108] H. Johannsson, M. Kaess, M. F. Fallon, and J. J. Leonard, *Temporally scalable visual SLAM using a reduced pose graph*, Tech. Report MIT-CSAIL-TR-2012-013, Computer Science and Artificial Intelligence Laboratory, MIT, May 2012.

[109] Matthew Johnson-Roberson, Oscar Pizarro, Stefan B. Williams, and Ian Mahon, *Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys*, Journal of Field Robotics **27** (2010), no. 1, 21–51.

[110] Brian J. Julian, Michael Angermann, Mac Schwager, and Daniela Rus, *A scalable information theoretic approach to distributed robot coordination*, Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, Sept. 2011, pp. 5187–5194.

[111] M. Kaess, A. Ranganathan, and F. Dellaert, *iSAM: Incremental smoothing and mapping*, Robotics, IEEE Transactions on **24** (2008), no. 6, 1365–1378.

[112] G.C. Karras, S.G. Loizou, and K.J. Kyriakopoulos, *On-line state and parameter estimation of an under-actuated underwater vehicle using a modified dual unscented Kalman filter*, Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, Oct. 2010, pp. 4868–4873.

[113] R.R. Khan, T. Taher, and F.S. Hover, *Accurate geo-referencing method for AUVs for oceanographic sampling*, OCEANS 2010, Sep. 2010, pp. 1–5.

[114] O. Khatib, *Real-time obstacle avoidance for manipulators and mobile robots*, International Journal of Robotics Research **5** (1986), 90–98.

[115] A. Kim and R. M. Eustice, *Real-time visual SLAM for autonomous underwater hull inspection using visual saliency*, Robotics, IEEE Transactions on **PP** (2013), no. 99, 1–15.

[116] Ayoung Kim and R. Eustice, *Pose-graph visual SLAM with geometric model selection for autonomous underwater ship hull inspection*, Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, Oct. 2009, pp. 1559–1565.

[117] Ayoung Kim and R.M. Eustice, *Toward AUV survey design for optimal coverage and localization using the Cramer Rao Lower Bound*, OCEANS 2009, Oct. 2009, pp. 1–7.

[118] James C. Kinsey, Ryan M. Eustice, and Louis L. Whitcomb, *A survey of underwater vehicle navigation: Recent advances and new challenges*, Conference on Manoeuvering and Control of Marine Craft, 2006, pp. 1–12.

[119] J.C. Kinsey and L.L. Whitcomb, *In situ alignment calibration of attitude and Doppler sensors for precision underwater vehicle navigation: Theory and experiment*, Oceanic Engineering, IEEE Journal of **32** (2007), no. 2, 286–299.

[120] W. J. Kirkwood, *Development of the DORADO mapping vehicle for multibeam, subbottom, and sidescan science missions*, Journal of Field Robotics **24** (2007), no. 6, 487–495.

[121] A.T. Klesh, P.T. Kabamba, and A.R. Girard, *Path planning for cooperative time-optimal information collection*, American Control Conference, 2008, June 2008, pp. 1991–1996.

[122] ———, *Optimal path planning for uncertain exploration*, American Control Conference, 2009. ACC '09., 2009, pp. 2421–2426.

[123] D. Kruger, R. Stolkin, A. Blum, and J. Briganti, *Optimal AUV path planning for extended missions in complex, fast-flowing estuarine environments*, Robotics and Automation, 2007 IEEE International Conference on, april 2007, pp. 4265–4270.

[124] F.R. Kschischang, B.J. Frey, and H.-A. Loeliger, *Factor graphs and the sum-product algorithm*, Information Theory, IEEE Transactions on **47** (2001), no. 2, 498–519.

[125] J. J. Kuffner and S. M. LaValle, *RRT-connect: An efficient approach to single-query path planning*, IEEE International Conference on Robotics and Automation, 2000, pp. 995–1001.

[126] Clayton Kunz, Chris Murphy, Hanumant Singh, Claire Pontbriand, Robert A. Sohn, Sandipa Singh, Taichi Sato, Chris Roman, Ko-ichi Nakamura, Michael Jakuba, Ryan Eustice, Richard Camilli, and John Bailey, *Toward extraplanetary under-ice exploration: Robotic steps in the Arctic*, Journal of Field Robotics **26** (2009), no. 4, 411–429.

[127] R. Kurazume, S. Nagata, and S. Hirose, *Cooperative positioning with multiple robots*, Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on, May 1994, pp. 1250–1257.

[128] N.H. Kussat, C.D. Chadwell, and R. Zimmerman, *Absolute positioning of an autonomous underwater vehicle using GPS and acoustic measurements*, Oceanic Engineering, IEEE Journal of **30** (2005), no. 1, 153–164.

[129] J. C. Latombe L. E. Kavraki, P. Svestka and M. H. Overmars, *Probabilistic roadmaps for path planning in high-dimensional configuration spaces*, IEEE Transactions on Robotics and Automation **12** (1996), no. 4, 566–580.

[130] Alain Lambert and Nadine Le Fort-Piat, *Safe task planning integrating uncertainties and local maps federations*, I. J. Robotic Res. **19** (2000), no. 6, 597–611.

[131] Cara LaPointe, *Virtual long baseline (VLBL) autonomous underwater vehicle navigation using a single transponder*, Master's thesis, Massachusetts Institute of Technology, 2006.

[132] J.C. Latombe, *Robot motion planning*, Kluwer Academic Publishers, 1991.

[133] S.M. LaValle, *Motion planning*, Robotics Automation Magazine, IEEE **18** (2011), no. 1, 79–89.

[134] J. Le Ny and G.J. Pappas, *On trajectory optimization for active sensing in Gaussian process models*, Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on, Dec. 2009, pp. 6286–6292.

[135] Tae-Kyeong Lee, Sang-Hoon Baek, Young-Ho Choi, and Se-Young Oh, *Smooth coverage path planning and control of mobile robots based on high-resolution grid map representation*, Robotics and Autonomous Systems **59** (2011), no. 10, 801–812.

[136] S.G. Lemon, *Towed-array history, 1917-2003*, Oceanic Engineering, IEEE Journal of **29** (2004), no. 2, 365–373.

[137] John J. Leonard, Andrew A. Bennett, Christopher M. Smith, and Hans Jacob S. Feder, *Autonomous underwater vehicle navigation*, MIT Marine Robotics Laboratory Technical Memorandum (1998).

[138] K.Y.K. Leung, T.D. Barfoot, and H. Liu, *Decentralized localization of sparsely-communicating robot networks: A centralized-equivalent approach*, Robotics, IEEE Transactions on **26** (2010), no. 1, 62 –77.

[139] H. Li, Yi Fu, K. Elgazzar, and L. Paull, *Path planning for multiple unmanned aerial vehicles using genetic algorithms*, Electrical and Computer Engineering, 2009. CCECE '09. Canadian Conference on, 2009, pp. 1129–1132.

[140] H. Li, F. Karray, O. Basir, and Insop Song, *A framework for coordinated control of multiagent systems and its applications*, Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on **38** (2008), no. 3, 534–548.

[141] Yan Li, Hai Chen, Meng Joo Er, and Xin-Min Wang, *Coverage path planning for UAVs based on enhanced exact cellular decomposition method*, Mechatronics **21** (2011), 876–885.

[142] David G. Lowe, *Object recognition from local scale-invariant features*, Computer Vision, IEEE International Conference on **2** (1999), 1150–1157.

[143] T. Lozano-Perez and M.Wesley, *An algorithm for planning collision-free paths among polyhedral obstacles*, Communications of the ACM **22** (1979), no. 10, 560–570.

[144] V. Lumelsky and A. Stepanov, *Path planning strategies for point mobile automaton moving amidst unknown obstacles of arbitrary shape*, Algorithmica **2** (1987), 403–430.

[145] T. Maki, H. Kondo, T. Ura, and T. Sakamaki, *Imaging vent fields: SLAM based navigation scheme for an AUV toward large-area seafloor imaging*, Autonomous Underwater Vehicles, 2008. AUV 2008. IEEE/OES, Oct. 2008, pp. 1–10.

[146] A. Mallios, P. Ridao, E. Hernandez, D. Ribas, F. Maurelli, and Y. Petillot, *Pose-based SLAM with probabilistic scan matching algorithm using a mechanical scanned imaging sonar*, OCEANS 2009 - EUROPE, May 2009, pp. 1–6.

[147] A. Mallios, P. Ridao, D. Ribas, and E. Hernandndez, *Probabilistic sonar scan matching SLAM for underwater environment*, OCEANS 2010 IEEE - Sydney, May 2010, pp. 1–8.

[148] A. Mallios, P. Ridao, D. Ribas, F. Maurelli, and Y. Petillot, *EKF-SLAM for AUV navigation under probabilistic sonar scan-matching*, Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, Oct. 2010, pp. 4404–4411.

[149] R. Mannadiar and I. Rekleitis, *Optimal coverage of a known arbitrary environment*, Robotics and Automation (ICRA), 2010 IEEE International Conference on, 2010, pp. 5525–5530.

[150] A. Matos and N. Cruz, *AUV navigation and guidance in a moving acoustic network*, Oceans 2005 - Europe, vol. 1, Jun. 2005, pp. 680–685.

[151] S.D. McPhail and M. Pebody, *Range-only positioning of a deep-diving autonomous underwater vehicle from a surface ship*, Oceanic Engineering, IEEE Journal of **34** (2009), no. 4, 669–677.

[152] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Gool, *A comparison of affine region detectors*, International Journal of Computer Vision **65** (2005), no. 1, 43–72.

[153] P.A. Miller, J.A. Farrell, Yuanyuan Zhao, and V. Djapic, *Autonomous underwater vehicle navigation*, Oceanic Engineering, IEEE Journal of **35** (2010), no. 3, 663–678.

[154] Marco Morgado, Paulo Oliveira, and Carlos Silvestre, *Tightly coupled ultrashort baseline and inertial navigation system for underwater vehicles: An experimental validation*, J. Field Robotics **30** (2013), no. 1, 142–170.

[155] A.I. Mourikis and S.I. Roumeliotis, *Performance analysis of multirobot cooperative localization*, Robotics, IEEE Transactions on **22** (2006), no. 4, 666–681.

[156] L. Murphy and P. Newman, *Risky planning: Path planning over costmaps with a probabilistically bounded speed-accuracy tradeoff*, Robotics and Automation (ICRA), 2011 IEEE International Conference on, May 2011, pp. 3727–3732.

[157] United States Navy, *The navy unmanned undersea vehicle (UUV) master plan (tech rep. A847115)*, Tech. report, U.S. Navy, 2004.

[158] E.D. Nerurkar and S.I. Roumeliotis, *Asynchronous multi-centralized cooperative localization*, Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, Oct. 2010, pp. 4352–4359.

[159] E.D. Nerurkar, S.I. Roumeliotis, and A. Martinelli, *Distributed maximum a posteriori estimation for multi-robot cooperative localization*, Robotics and Automation, 2009. ICRA '09. IEEE International Conference on, May 2009, pp. 1402–1409.

[160] Esha D. Nerurkar, Ke X. Zhou, and S.I. Roumeliotis, *A hybrid estimation framework for cooperative localization under communication constraints*, Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, 2011, pp. 502–509.

[161] P. Newman and J. Leonard, *Pure range-only sub-sea SLAM*, Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on, vol. 2, Sep. 2003, pp. 1921–1926.

[162] Paul Newman, www.robots.ox.ac.uk/ pnewman/TheMOOS/.

[163] Paul M. Newman, John J. Leonard, and Richard J. Rikoski, *Towards constant-time SLAM on an autonomous underwater vehicle using synthetic aperture sonar*, In Proceedings of the Eleventh International Symposium on Robotics Research, 2003, pp. 409–420.

[164] B. Nguyen, D. Hopkin, and H. Yip, *Autonomous underwater vehicles a transformation of mine counter-measure operations*, Defense & Security Analysis **24** (2008), no. 3, 247–266.

[165] Timo Oksanen and Arto Visala, *Coverage path planning algorithms for agricultural field machines*, J. Field Robotics **26** (2009), no. 8, 651–668.

[166] Edwin Olson, John J. Leonard, and Seth Teller, *Robust range-only beacon localization*, Oceanic Engineering, IEEE Journal of **31** (2006), no. 4, 949–958.

[167] A. Ortiz, J. Antich, and G. Oliver, *A Bayesian approach for tracking undersea narrow telecommunication cables*, OCEANS 2009 - EUROPE, May 2009, pp. 1–10.

[168] Athanasios Papoulis and S. Unnikrishna Pillai, *Probability, random variables and stochastic processes*, fourth ed., McGraw Hill, 2002.

[169] K. Pathak, A. Birk, and N. Vaskevicius, *Plane-based registration of sonar data for underwater 3D mapping*, Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, Oct. 2010, pp. 4880–4885.

[170] L. Paull, S. Saeedi, H. Li, and V. Myers, *An information gain based adaptive path planning method for an autonomous underwater vehicle using sidescan sonar*, Automation Science and Engineering (CASE), 2010 IEEE Conference on, 2010, pp. 835–840.

[171] L. Paull, S. Saeedi, M. Seto, and H. Li, *Sensor-driven online coverage planning for autonomous underwater vehicles*, Mechatronics, IEEE/ASME Transactions on **pp** (2012), no. 99, 1–12.

[172] L. Paull, S. Saeedi Gharahbolagh, M. Seto, and H. Li, *Sensor driven online coverage planning for autonomous underwater vehicles*, Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, 2012, pp. 2875–2880.

[173] Liam Paull, Sajad Saeedi, and Howard Li, *Path planning for autonomous underwater vehicles*, Autonomy for Marine Robots (Mae L. Seto, ed.), Springer, 2012, pp. 177–224.

[174] Liam Paull, Sajad Saeedi, Mae Seto, and Howard Li, *AUV navigation and localization - a review*, IEEE Journal of Oceanic Engineering (2013), In Press.

[175] Liam Paull, Carl Thibault, Amr Nagaty, Mae Seto, and Howard Li, *Sensor-driven area coverage for an autonomous fixed-wing unmanned aerial vehicle*, IEEE Transactions on Cybernetics (2013), Accepted.

[176] A.M. Percival and M.A. Stoddard, *PATHA: A planning aid for tasking heterogeneous assets for route survey or mine countermeasures operations*, OCEANS 2010, Sept. 2010, pp. 1–8.

[177] Y. Petillot, F. Maurelli, N. Valeyrie, A. Mallios, P. Ridao, J. Aulinas, and J. Salvi, *Acoustic-based techniques for autonomous underwater vehicle localization*, Journal of Engineering for the Maritime Environment **224** (2010), 293–307.

[178] Y. Petillot, I. Tena Ruiz, and D.M. Lane, *Underwater vehicle obstacle avoidance and path planning using a multi-beam forward looking sonar*, Oceanic Engineering, IEEE Journal of **26** (2001), no. 2, 240–251.

[179] C. Petres, Y. Pailhas, P. Patron, Y. Petillot, J. Evans, and D. Lane, *Path planning for autonomous underwater vehicles*, Robotics, IEEE Transactions on **23** (2007), no. 2, 331–341.

[180] M. Pfingsthorn, A. Birk, and H. Bulow, *Uncertainty estimation for a 6-DOF spectral registration method as basis for sonar-based underwater 3D SLAM*, Robotics and Automation (ICRA), 2012 IEEE International Conference on, 2012, pp. 3049–3054.

[181] A.J. Plueddemann, A.L. Kukulya, R. Stokey, and L. Freitag, *Autonomous underwater vehicle operations beneath coastal sea ice*, Mechatronics, IEEE/ASME Transactions on **17** (2012), no. 1, 54–64.

[182] A. Prorok and A. Martinoli, *A reciprocal sampling algorithm for lightweight distributed multi-robot localization*, Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, 2011, pp. 3241–3247.

[183] A. Rajala and D. Edwards, *Allocating AUVs for mine map development in MCM*, OCEANS 2006 - Asia Pacific, May 2006, pp. 1–8.

[184] Carl Edward Rasmussen, *Gaussian processes for machine learning*, MIT Press, 2006.

[185] I.M. Rekleitis, G. Dudek, and E.E. Milios, *Multi-robot collaboration for robust exploration*, Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on, vol. 4, 2000, pp. 3164–3169.

[186] D. Ribas, P. Ridao, A. Mallios, and N. Palomeras, *Delayed state information filter for USBL-aided AUV navigation*, Robotics and Automation (ICRA), 2012 IEEE International Conference on, May 2012, pp. 4898–4903.

[187] D. Ribas, P. Ridao, J. Neira, and J.D. Tardos, *SLAM using an imaging sonar for partially structured underwater environments*, Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, Oct. 2006, pp. 5040–5045.

[188] D. Ribas, P. Ridao, J.D. Tardos, and J. Neira, *Underwater SLAM in a marina environment*, Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on, Nov. 2007, pp. 1455–1460.

[189] David Ribas, Pere Ridao, Juan Domingo Tards, and Jos Neira, *Underwater SLAM in man-made structured environments*, Journal of Field Robotics **25** (2008), no. 11-12, 898–921.

[190] Pere Ridao, Marc Carreras, David Ribas, and Rafael Garcia, *Visual inspection of hydroelectric dams using an autonomous underwater vehicle*, Journal of Field Robotics **27** (2010), no. 6, 759–778.

[191] S.I. Roumeliotis and G.A. Bekey, *Distributed multi-robot localization*, Tech. report, University of Minnesota, 2002.

[192] ———, *Distributed multirobot localization*, Robotics and Automation, IEEE Transactions on **18** (2002), no. 5, 781–795.

[193] Stergios I. Roumeliotis and Ioannis M. Rekleitis, *Propagation of uncertainty in cooperative multirobot localization: Analysis and experimental results*, Autonomous Robots **17** (2004), 41–54.

[194] I. Tena Ruiz, S. de Raucourt, Y. Petillot, and D.M. Lane, *Concurrent mapping and localization using sidescan sonar*, Oceanic Engineering, IEEE Journal of **29** (2004), no. 2, 442–456.

[195] Allison Ryan and J. Karl Hedrick, *Particle filter based information-theoretic sensing*, Robotics and Automation Systems **58** (2010), 574–584.

[196] S. Saeedi, L. Paull, M. Trentini, and H. Li, *Neural network-based multiple robot simultaneous localization and mapping*, Neural Networks, IEEE Transactions on **22** (2011), no. 12, 2376–2387.

[197] J. Salvi, Y. Petillot, and E. Batlle, *Visual SLAM for 3D large-scale seabed acquisition employing underwater vehicles*, Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, Sep. 2008, pp. 1011–1016.

[198] J. Salvi, Y. Petillot, S. Thomas, and J. Aulinas, *Visual SLAM for underwater vehicles using video velocity log and natural landmarks*, OCEANS 2008, Sep. 2008, pp. 1–6.

[199] Toby Schneider and Henrik Schmidt, *Unified command and control for heterogeneous marine sensing networks*, J. Field Robotics **27** (2010), no. 6, 876–889.

[200] Claude E. Shannon, *A mathematical theory of communication*, Bell System Technical Journal **27** (1948), no. 3, 379–423.

[201] R. Sharma, R.W. Beard, C.N. Taylor, and S. Quebe, *Graph-based observability analysis of bearing-only cooperative localization*, Robotics, IEEE Transactions on **28** (2012), no. 2, 522–529.

[202] A. F. Shchepetkin and J. C. McWilliams, *The regional oceanic modeling system (ROMS): a split-explicit, free-surface, topography-following-coordinate oceanic model*, OCEAN MODELLING **9** (2005), no. 4, 347–404.

[203] G. Shippey, M. Jonsson, and J.N.B. Pihl, *Position correction using echoes from a navigation fix for synthetic aperture sonar imaging*, Oceanic Engineering, IEEE Journal of **34** (2009), no. 3, 294–306.

[204] Gabe Sibley, Larry Matthies, and Gaurav Sukhatme, *Sliding window filter with application to planetary landing*, Journal of Field Robotics **27** (2010), no. 5, 587–608.

[205] Hanumant Singh, Chris Roman, Oscar Pizarro, Ryan Eustice, and Ali Can, *Towards high-resolution imaging from underwater vehicles*, The International Journal of Robotics Research **26** (2007), no. 1, 55–74.

[206] P. Skoglar, J. Nygards, and M. Ulvklo, *Concurrent path and sensor planning for a UAV - towards an information based approach incorporating models of environment and sensor*, Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, Oct. 2006, pp. 2436–2442.

[207] Ryan N. Smith, Yi Chao, Peggy P. Li, David A. Caron, Burton H. Jones, and Gaurav S. Sukhatme, *Planning and implementing trajectories for autonomous underwater vehicles to track evolving ocean processes based on predictions from a regional ocean model*, International Journal of Robotics Research **29** (2010), no. 12, 14751497.

[208] M. Soulignac, *Feasible and optimal path planning in strong current fields*, Robotics, IEEE Transactions on **27** (2011), no. 1, 89–98.

[209] J.R. Stack and C.M. Smith, *Combining random and data-driven coverage planning for underwater mine detection*, OCEANS 2003. Proceedings, vol. 5, Sep. 2003, pp. 2463–2468.

[210] B. Stenning and T.D. Barfoot, *Path planning with variable-fidelity terrain assessment*, Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, Oct. 2010, pp. 300–306.

[211] L. Stutters, Honghai Liu, C. Tiltman, and D.J. Brown, *Navigation technologies for autonomous underwater vehicles*, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on **38** (2008), no. 4, 581–589.

[212] Feng Sun, Wen Xu, and Jianlong Li, *Enhancement of the aided inertial navigation system for an AUV via micronavigation*, OCEANS 2010, Sep. 2010, pp. 1–4.

[213] David R. Thompson, David Wettergreen, and Francisco J. Calderóon Peralta, *Autonomous science during large-scale robotic survey*, J. Field Robotics **28** (2011), no. 4, 542–564.

[214] D.R. Thompson, S. Chien, Yi Chao, P. Li, B. Cahill, J. Levin, O. Schofield, A. Balasuriya, S. Petillo, M. Arrott, and M. Meisinger, *Spatiotemporal path planning in strong, dynamic, uncertain currents*, Robotics and Automation (ICRA), 2010 IEEE International Conference on, May 2010, pp. 4778–4783.

[215] Sebastian Thrun, Wolfram Burgard, and Dieter Fox, *Probabilistic robotics*, The MIT press, Cambridge, Massachusetts, USA, 2005.

[216] J. Tisdale, Zuwhan Kim, and J. Hedrick, *Autonomous UAV path planning and estimation*, Robotics Automation Magazine, IEEE **16** (2009), no. 2, 35–42.

[217] N. Trawny, S.I. Roumeliotis, and G.B. Giannakis, *Cooperative multi-robot localization under communication constraints*, Robotics and Automation, 2009. ICRA '09. IEEE International Conference on, May 2009, pp. 4394–4400.

[218] J. Vaganay, J.J. Leonard, J.A. Curcio, and J.S. Willcox, *Experimental validation of the moving long base-line navigation concept*, Autonomous Underwater Vehicles, 2004 IEEE/OES, Jun. 2004, pp. 59–65.

[219] Jur van den Berg, Pieter Abbeel, and Kenneth Y. Goldberg, *LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information*, I. J. Robotic Res. **30** (2011), no. 7, 895–913.

[220] Iuliu Vasilescu, Carrick Detweiler, Marek Doniec, Daniel Gurdan, Stefan Sosnowski, Jan Stumpf, and Daniela Rus, *AMOUR V: A hovering energy efficient underwater robot capable of dynamic payloads*, The International Journal of Robotics Research **29** (2010), no. 5, 547–570.

[221] D. Viegas, P. Batista, P. Oliveira, and C. Silvestre, *Position and velocity filters for intervention AUVs based on single range and depth measurements*, Robotics and Automation (ICRA), 2012 IEEE International Conference on, May 2012, pp. 4878–4883.

[222] M. Walter, F. Hover, and J. Leonard, *SLAM for ship hull inspection using exactly sparse extended information filters*, Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on, May 2008, pp. 1463–1470.

[223] C.W. Warren, *A technique for autonomous underwater vehicle route planning*, Oceanic Engineering, IEEE Journal of **15** (1990), no. 3, 199–204.

[224] Sarah E Webster, Ryan M Eustice, Hanumant Singh, and Louis L Whitcomb, *Advances in single-beacon one-way-travel-time acoustic navigation for underwater vehicles*, The International Journal of Robotics Research **31** (2012), no. 8, 935–950.

[225] S.E. Webster, R.M. Eustice, H. Singh, and L.L. Whitcomb, *Preliminary deep water results in single-beacon one-way-travel-time acoustic navigation for underwater vehicles*, Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, Oct. 2009, pp. 2053–2060.

[226] Ron Wein, Jur van den Berg, and Dan Halperin, *Planning high-quality paths and corridors amidst obstacles*, The International Journal of Robotics Research **27** (2008), 1213–31.

[227] 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*, Robotics and Automation (ICRA), 2010 IEEE International Conference on, May 2010, pp. 594–600.

[228] D.P. Williams, *On optimal AUV track-spacing for underwater mine detection*, Robotics and Automation (ICRA), 2010 IEEE International Conference on, May 2010, pp. 4755–4762.

[229] D.P. Williams, J. Groen, and W.L.J. Fox, *A fast detection algorithm for autonomous mine countermeasures*, NURC-FR-2011-006, NATO Underwater Research Center, October 2011.

[230] P. Woock, *Survey on suitable 3D features for sonar-based underwater navigation*, OCEANS, 2012 - Yeosu, May 2012, pp. 1–6.

[231] N.K. Yilmaz, C. Evangelinos, P. Lermusiaux, and N.M. Patrikalakis, *Path planning of autonomous underwater vehicles for adaptive sampling using mixed integer linear programming*, Oceanic Engineering, IEEE Journal of **33** (2008), no. 4, 522–537.

[232] Dana R. Yoerger, Michael Jakuba, Albert M. Bradley, and Brian Bingham, *Techniques for deep sea near bottom survey using an autonomous underwater vehicle*, The International Journal of Robotics Research **26** (2007), no. 1, 41–54.

[233] Benoit Zerr, John Fawcett, and Dave Hopkin, *Adaptive algorithm for sea mine classification*, 3rd International Conference & Exhibition on Underwater Acoustic Measurements: Technologies & Results, 2009, pp. 319–326.

[234] G. Zhang, S. Ferrari, and M. Qian, *An information roadmap method for robotic sensor path planning*, Journal of Intelligent and Robotic Systems **56** (2009), no. 1-2, 69–98.

[235] X.S. Zhou and S.I. Roumeliotis, *Robot-to-robot relative pose estimation from range measurements*, Robotics, IEEE Transactions on **24** (2008), no. 6, 1379–1393.

Appendix A

Mission Oriented Operating Suite with Interval Programming

Mission oriented operating suite (MOOS) is a middleware used by the U.S. Navy to implement multi-agent systems. The Interval Programming (IvP) functionality was later built so that higher level behaviours can be defined.

MOOS-IvP utilizes three main philosophical frameworks: the backseat driver, publish-subscribe, and behaviour-based control.

A.0.0.1 Backseat Driver:

In the backseat driver paradigm, the vehicle control and autonomy are decoupled. Essentially, MOOS does not worry about how the vehicle's navigation and control system function, as long as it receives accurate information from the sensors, and the vehicle can act upon the autonomy decisions that are sent back.

A.0.0.2 Publish-Subscribe:

All communication goes through the *MOOSDB*. MOOS applications publish events, which are variable-value pairs and applications also subscribe to different variables. Most of the MOOS applications will normally be sensors, actuators or the visualizers etc., but one is special: the *pHelmIvP* application. The *pHelmIvP* is responsible for reconciling all the desired behaviours.

A.0.0.3 Behaviour-Based Control:

Different desired behaviours are defined in the mission behaviour file. Behaviours can be selected from amongst predefined behaviours, or can be defined from scratch. On each iteration, the *pHelmIvP* generates an objective function from each behaviour, and generates a 'decision' consisting of variable-value pairs that define the action that will optimize the multiple objectives. In addition, attributes of the behaviours can be assigned dynamically using mode variables and hierarchical mode declarations.

More details are provided in [27].

Appendix B

Proof of Proposition 3.2.1

For simplicity of formulation, assume that cell $\mathbf{c}^i$ gets covered twice, once at time t_1 from pose $\mathbf{X}_{t_1}$ and once at time t_2 from pose $\mathbf{X}_{t_2}$, generating $\breve{T}^i_{t_1}$ and $\breve{T}^i_{t_2}$ respectively and without loss of generality assume that the look at time t_1 results in higher confidence: $p(\breve{T}^i_{t_1} = 1) > p(\breve{T}^i_{t_1} = 1)$. If either look indicates the presence of a mine, then we consider a mine to be present:

$$\begin{aligned} p(T^i_t = 1|\breve{T}^i_{t_1} = 1) = p(T^i_t = 1|\breve{T}^i_{t_2} = 1) = 1 \\ p(T^i_t = 1|\breve{T}^i_{t_1} = 0, \breve{T}^i_{t_2} = 0) = 0 \end{aligned} \tag{B.1}$$

Without loss of generality, assume that $p(\breve{T}^i_{t_1} = 1) \geq p(\breve{T}^i_{t_2} = 1)$. α is calculated as the acute angle of the intersection of two lines with directions ψ_1 and ψ_2 as shown in Fig. 3.4.

The combination of the two looks can be represented by the Bayes' network (BN) shown in Fig. 3.3. From the BN we can write the joint probability as:

$$p(T^i_t, \breve{T}^i_{t_1}, \breve{T}^i_{t_2}) = p(T^i_t|\breve{T}^i_{t_1}, \breve{T}^i_{t_2})p(\breve{T}^i_{t_1}, \breve{T}^i_{t_2}) \tag{B.2}$$

and we can recover the desired marginal distribution of T^i_t by marginalizing $\breve{T}^i_{t_1}$ and $\breve{T}^i_{t_2}$ out of (B.2):

$$\begin{aligned} p(T^i_t = 1) &= p(T^i_t = 1|\breve{T}^i_{t_1} = 1, \breve{T}^i_{t_2} = 1)p(\breve{T}^i_{t_1} = 1, \breve{T}^i_{t_2} = 1) \\ &\quad + p(T^i_t = 1|\breve{T}^i_{t_1} = 0, \breve{T}^i_{t_2} = 1)p(\breve{T}^i_{t_1} = 0, \breve{T}^i_{t_2} = 1) \\ &\quad + p(T^i_t = 1|\breve{T}^i_{t_1} = 1, \breve{T}^i_{t_2} = 0)p(\breve{T}^i_{t_1} = 1, \breve{T}^i_{t_2} = 0) \\ &\quad + p(T^i_t = 1|\breve{T}^i_{t_1} = 0, \breve{T}^i_{t_2} = 0)p(\breve{T}^i_{t_1} = 0, \breve{T}^i_{t_2} = 0) \\ &= p(\breve{T}^i_{t_1} = 1, \breve{T}^i_{t_2} = 1) + p(\breve{T}^i_{t_1} = 0, \breve{T}^i_{t_2} = 1) + p(\breve{T}^i_{t_1} = 1, \breve{T}^i_{t_2} = 0) \end{aligned} \tag{B.3}$$

and conversely:

$$p(T^i_t = 0) = 1 - p(T^i_t = 1) = p(\breve{T}^i_{t_1} = 0, \breve{T}^i_{t_2} = 0) \tag{B.4}$$

Proposition B.0.1. *In the case that the two looks are independent, then the total confidence is given by:*

$$p(T^i_t = 1) = 1 - [(1 - p(\breve{T}^i_{t_1} = 1))(1 - p(\breve{T}^i_{t_2} = 1))] \tag{B.5}$$

Proof. In the case that the looks are independent, then $p(\breve{T}^i_{t_1} = 1, \breve{T}^i_{t_2} = 1) = p(\breve{T}^i_{t_1} =$

$1)p(\breve{T}^i_{t_2} = 1)$ so from (B.4):

$$\begin{aligned} p(T^i_t = 1) &= 1 - p(T^i_t = 0) = 1 - p(\breve{T}^i_{t_1} = 0, \breve{T}^i_{t_2} = 0) = 1 - p(\breve{T}^i_{t_1} = 0)p(\breve{T}^i_{t_2} = 0) \\ &= 1 - [(1 - p(\breve{T}^i_{t_1} = 1))(1 - p(\breve{T}^i_{t_2} = 1)) \end{aligned} \tag{B.6}$$

□

This extends to K looks as:

$$p(T^i_t) = 1 - \prod_{k=1}^{K}(1 - p(\breve{T}^i_{t_k})) \tag{B.7}$$

Proposition B.0.2. *If the looks are not independent then the following inequality holds:*

$$p(T^i_t = 1) \geq \max\{p(\breve{T}^i_{t_1} = 1), p(\breve{T}^i_{t_2} = 1)\} \tag{B.8}$$

Proof. From (B.3) we have:

$$\begin{aligned} p(T^i_t = 1) &= p(\breve{T}^i_{t_1} = 1, \breve{T}^i_{t_2} = 1) + p(\breve{T}^i_{t_1} = 0, \breve{T}^i_{t_2} = 1) + p(\breve{T}^i_{t_1} = 1, \breve{T}^i_{t_2} = 0) \\ &= p(\breve{T}^i_{t_1} = 1|\breve{T}^i_{t_2} = 1)p(\breve{T}^i_{t_2} = 1) + p(\breve{T}^i_{t_1} = 0|\breve{T}^i_{t_2} = 1)p(\breve{T}^i_{t_2} = 1) \\ &\quad + p(\breve{T}^i_{t_1} = 1, \breve{T}^i_{t_2} = 0) \\ &= p(\breve{T}^i_{t_2} = 1)[p(\breve{T}^i_{t_1} = 1|\breve{T}^i_{t_2} = 1) + p(\breve{T}^i_{t_1} = 0|\breve{T}^i_{t_2} = 1)] \\ &\quad + p(\breve{T}^i_{t_1} = 1, \breve{T}^i_{t_2} = 0) \\ &= p(\breve{T}^i_{t_2} = 1) + p(\breve{T}^i_{t_1} = 1, \breve{T}^i_{t_2} = 0) \\ &\geq p(\breve{T}^i_{t_2} = 1) \end{aligned} \tag{B.9}$$

and a similar factorization yields $p(T^i_t = 1) \geq p(\breve{T}^i_{t_1} = 1)$. If $p(T^i_t = 1) \geq p(\breve{T}^i_{t_1} = 1)$ and $p(T^i_t = 1) \geq p(\breve{T}^i_{t_2} = 1)$ hold then $p(T^i_t = 1) \geq \max\{p(\breve{T}^i_{t_1} = 1), p(\breve{T}^i_{t_2} = 1)\}$ must hold. □

In the case of K looks, (B.8) extends to:

$$p(T^i_t = 1) \geq \max\{p(\breve{T}^i_{t_1} = 1), p(\breve{T}^i_{t_2} = 1), ..., p(\breve{T}^i_{t_K} = 1)\} \tag{B.10}$$

Appendix C

Derivation of Equation (4.29) and (4.30)

In order to transform the certain location of the cell i in the global frame ${}^{g}c^{i} = [{}^{g}x^{i}, {}^{g}y^{i}]^{T}$ into the uncertain location of the cell in the sensor frame, ${}^{s}C_{t}^{i}$, it is mapped through the uncertain homogeneous transformation ${}^{s}C_{t}^{i} = {}_{g}^{s}T_{t}\ {}^{g}c^{i}$ defined in (4.1), where ${}_{g}^{s}T_{t}$ is given by:

$$ {}_{g}^{s}T_{t} = \begin{bmatrix} \cos\psi_t & -\sin\psi_t & 0 \\ \sin\psi_t & \cos\psi_t & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -X_t \\ 0 & 1 & -Y_t \\ 0 & 0 & 1 \end{bmatrix}. \tag{C.1}$$

If we define the 2×2 rotation matrix:

$$\mathcal{R}(\psi_t) \triangleq \begin{bmatrix} \cos\psi_t & -\sin\psi_t \\ \sin\psi_t & \cos\psi_t \end{bmatrix} \tag{C.2}$$

then we can express the cell location in the uncertain sensor frame as:

$${}^{s}C_{t}^{i} = \mathcal{R}(\psi_t){}^{g}c^{i} - \mathcal{R}(\psi_t)[X_t, Y_t]^{T} \tag{C.3}$$

which has has a Gaussian distribution with mean $\mu_c = [\mu_{x_c}, \mu_{y_c}]$ and covariance Σ_c where:

$$\begin{aligned} \mu_c &= \mathcal{J}(\psi_t){}^{g}c^{i} - \mathcal{J}(\psi_t)[\mu_{X_t}, \mu_{Y_t}]^{T}, \\ \Sigma_c &= \mathcal{J}(\psi_t)\Sigma_t\mathcal{J}(\psi_t)^{T}. \end{aligned} \tag{C.4}$$

The expression for Σ_c can be expanded with (4.30) to yield:

$$\Sigma_c = \begin{bmatrix} \sigma_x^2 c^2\psi - \rho_t\sigma_x\sigma_y s2\psi + \sigma_y^2 s^{\psi} & \sigma_x^2 c\psi s\psi + \rho_t\sigma_x\sigma_y c2\psi - \sigma_y^2 c\psi s\psi \\ \sigma_x^2 c\psi s\psi + \rho_t\sigma_x\sigma_y c2\psi - \sigma_y^2 c\psi s\psi & \sigma_x^2 s^2\psi + \rho_t\sigma_x\sigma_y s2\psi + \sigma_y^2 c^2\psi \end{bmatrix} \tag{C.5}$$

where $s\psi$ and $c\psi$ are shorthand for $\sin\psi$ and $\cos\psi$ respectively, $s2\psi$ and $c2\psi$ are shorthand for $\sin 2\psi$ and $\cos 2\psi$ respectively, and $s^2\psi$ and $c^2\psi$ are shorthand for $\sin^2\psi$ and $\cos^2\psi$ respectively.

As a result of Assumption 4.4.1 the distribution of the cell location in the ${}^{s}x_{c^i}$ direction can be marginalized out:

$$p({}^{s}y_{c^i}) = \int_{-\infty}^{\infty} p({}^{s}c_i)d^{s}x_{c^i} \tag{C.6}$$

Appendix D

Models and Jacobians

D.1 Process and Measurement Models

As is common in robotics [215], the control inputs are replaced some measure of odometry, or change in pose. In this case, the odometry is derived from the DVL velocity data. To fit the standard kinematic model:

$$\mathbf{X}_t^i = f(\mathbf{X}_{t-1}^i, \mathbf{U}_t^i) + \boldsymbol{\zeta}_t^i, \tag{D.1}$$

we require to transform the velocity in the body frame (V_t^u, V_t^v) into a change in position in the global frame. Additionally the noise associated with the DVL velocity sensor measurements must be transformed and scaled to obtain a distribution for the additive odometry noise $\boldsymbol{\zeta}_t^i$:

$$\mathbf{U}_t^i = \mathcal{R}(\Psi_t^i) \begin{bmatrix} V_t^u \\ V_t^v \\ 0 \end{bmatrix} \Delta t \tag{D.2}$$

where

$$\mathcal{R}(\Psi_{t-1}^i) = \begin{bmatrix} \cos\Psi_{t-1}^i & \sin\Psi_{t-1}^i & 0 \\ -\sin\Psi_{t-1}^i & \cos\Psi_{t-1}^i & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{D.3}$$

and Δt is the reciprocal of the frequency of the DVL. The additive noise term, $\boldsymbol{\zeta}_t^i$, is assumed to be Gaussian with mean 0 and covariance $\mathbf{Q}_t^i$. The covariance $\mathbf{Q}_t^i$ calculated by:

$$\mathbf{Q}_t^i = \Delta t^2 \mathcal{R}(\Psi_{t-1}) \begin{bmatrix} \sigma_u^2 & 0 \\ 0 & \sigma_v^2 \end{bmatrix} \mathcal{R}(\Psi_{t-1})^T \tag{D.4}$$

where σ_u and σ_v are the root mean square error values of the DVL sensor obtained from the manufacturer, and ^{T} indicates the transpose.

The proprioceptive sensors (GPS and compass) observe the state through the GPS measurement model:

$$\begin{aligned} {}^G\mathbf{Z}_t^i =& h^G(\mathbf{X}_t^i, {}^G\boldsymbol{\delta}_t^i) \\ =& \begin{bmatrix} X_t^i \\ Y_t^i \end{bmatrix} + {}^G\boldsymbol{\delta}_t^i \end{aligned} \tag{D.5}$$

where ${}^G\boldsymbol{\delta}_t^i \sim \mathcal{N}(0, \boldsymbol{\Lambda}^G)$ is zero mean Gaussian noise associated with the GPS sensor reading (covariance assumed constant with time and across all robots), and the compass measurement model:

$$\begin{aligned} {}^{C}\mathbf{Z}_t^i =& h^C(\mathbf{X}_t^i, {}^{C}\boldsymbol{\delta}_t^i) \\ =& \Psi_t^i + {}^{C}\boldsymbol{\delta}_t^i \end{aligned} \tag{D.6}$$

where ${}^{C}\boldsymbol{\delta}_t^i \sim \mathcal{N}(0, \mathbf{\Lambda}^C)$ is zero mean Gaussian noise associated with the compass measurement ${}^{C}Z_t^i$ (covariance assumed constant with time and across all robots).

The inter-robot range measurements relate the states of robots through:

$$\begin{aligned} \mathbf{R}_{t_k}^{i,j} =& h^R(\mathbf{X}_{t_k^i}^i, \mathbf{X}_{t_k^j}^j) + {}^{R}\boldsymbol{\delta}_{t_k}^{i,j} \\ =& \left|\left| \begin{bmatrix} X_{t_k^i}^i \\ Y_{t_k^i}^i \end{bmatrix} - \begin{bmatrix} X_{t_k^j}^j \\ Y_{t_k^j}^j \end{bmatrix} \right|\right| + {}^{R}\boldsymbol{\delta}_{t_k}^{i,j} \end{aligned} \tag{D.7}$$

where ${}^{R}\boldsymbol{\delta}_{t_k}^{i,j} \sim \mathcal{N}(0, \mathbf{\Lambda}^R)$ is zero mean Gaussian noise associated with the inter-robot range measurements, whose covariance is assumed to be constant with time and independent of range, a claim experimentally validated in [224]. The operator $||\cdot||$ is the Euclidean norm.

D.2 Process and Measurement Model Jacobians

In order to linearize the process and measurement models we require the Jacobians. Here we assume that we are linearizing around the point $\mathbf{x}_{1:t}^{1:N}$

Process factor:

$$\begin{aligned} f(\mathbf{x}_{t-1}^i, \mathbf{u}_t^i) - \mathbf{x}_t^i \approx& f({}^{m}\mathbf{x}_{t-1}^i, \mathbf{u}_t^i) - {}^{m}\mathbf{x}_{t-1}^i + {}^{m}F_{t-1}^i(\mathbf{x}_{t-1}^i - {}^{m}\mathbf{x}_{t-1}^i) - (\mathbf{x}_t^i - {}^{m}\mathbf{x}_t^i) \\ =& F_{t-1}^i(\mathbf{x}_{t-1}^i - {}^{m}\mathbf{x}_{t-1}^i) - I_n(\mathbf{x}_t^i - {}^{m}\mathbf{x}_t^i) - a_t^i \end{aligned} \tag{D.8}$$

where

$$\begin{aligned} {}^{m}F_{t-1}^i \triangleq& \frac{\partial f(\mathbf{x}_{t-1}^i, \mathbf{u}_t^i)}{\partial \mathbf{x}_{t-1}^i}|_{\mathbf{x}_{t-1}^i = {}^{m}\mathbf{x}_{t-1}^i} \\ =& \begin{bmatrix} 1 & 0 & \Delta t[-V_u \sin {}^{m}\psi_{t-1} + V_v \cos {}^{m}\psi_{t-1}] \\ 0 & 1 & -\Delta t[V_u \cos {}^{m}\psi_{t-1} + V_v \sin {}^{m}\psi_{t-1}] \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{D.9}$$

is the Jacobian of f with respect to $\mathbf{x}_{t-1}^i$ evaluated at the linearization point ${}^{m}\mathbf{x}_{t-1}^i$,

$$a_t^i \triangleq f({}^{m}\mathbf{x}_{t-1}^i, u_t^i) - {}^{m}\mathbf{x}_{t-1}^i \tag{D.10}$$

is the residual , and I_n is the $n \times n$ identity matrix where n is the dimension of the instantaneous state, in this case 3.

GPS factor:

$$\begin{aligned} h^G(\mathbf{x}_t^i) - {}^{G}\mathbf{z}_t^i \approx& h^G({}^{m}\mathbf{x}_t^i) - {}^{G}\mathbf{z}_t^i + {}^{m}H_t^G(\mathbf{x}_t^i - {}^{m}\mathbf{x}_t^i) \\ =& {}^{m}H_t^G(\mathbf{x}_t^i - {}^{m}\mathbf{x}_t^i) - b_t^i \end{aligned} \tag{D.11}$$

where

$$
\begin{aligned}
{}^{m}H_t^G = & \frac{\partial h^G(\mathbf{x}_t^i)}{\partial \mathbf{x}_t^i}|_{\mathbf{x}_t^i = {}^m\mathbf{x}_t^i} \\
= & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}
\end{aligned}
\tag{D.12}
$$

and

$$
b_t^i \triangleq h^G({}^m\mathbf{x}_t^i) - {}^G\mathbf{z}_t^i \tag{D.13}
$$

is the residual.

Compass factor:

$$
\begin{aligned}
h^C(\mathbf{x}_t^i) - {}^C\mathbf{z}_t^i \approx & \, h^C({}^m\mathbf{x}_t^i) - {}^C z_t^i + {}^mH_t^C(\mathbf{x}_t^i - {}^m\mathbf{x}_t^i) \\
= & \, {}^mH_t^C(\mathbf{x}_t^i - {}^m\mathbf{x}_t^i) - c_t^i
\end{aligned}
\tag{D.14}
$$

where

$$
\begin{aligned}
{}^{m}H_t^C = & \frac{\partial h^C(\mathbf{x}_t^i)}{\partial \mathbf{x}_t^i} \; |_{\mathbf{x}_t^i = {}^m\mathbf{x}_t^i} \\
= & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\tag{D.15}
$$

and

$$
c_t^i \triangleq h^C({}^m\mathbf{x}_t^i) - {}^C\mathbf{z}_t^i \tag{D.16}
$$

Range factor:

$$
\begin{aligned}
h^R(\mathbf{x}^i_{t^i_k}, \mathbf{x}^j_{t^j_k}) - \mathbf{r}^{i,j}_{t_k} \approx & \, h^R({}^m\mathbf{x}^i_{t^i_k}, {}^m\mathbf{x}^j_{t^j_k}) - \mathbf{r}^{i,j}_{t_k} + {}^mH^R_{t_k}(\mathbf{x}_{1:t} - {}^m\mathbf{x}^{1:N}_{1:t}) \\
= & {}^mH^R_t(\mathbf{x}^{1:N}_{1:t} - {}^m\mathbf{x}^{1:N}_{1:t}) - d^i_t
\end{aligned}
\tag{D.17}
$$

where

$$
\begin{aligned}
{}^mH^R_{t_k} = & \, \frac{\partial h^R(\mathbf{x}^i_{t^i_k}, \mathbf{x}^j_{t^j_k}) - r^{i,j}_{t_k}}{\partial \mathbf{x}^{1:N}_{1:t}|_{\mathbf{x}^{1:N}_{1:t} = {}^m\mathbf{x}^{1:N}_{1:t}}} \\
= & \frac{1}{((x^i_{t^i_k} - x^j_{t^j_k})^2 + (y^i_{t^i_k} - y^j_{t^j_k})^2)^{1/2}} * \\
& \begin{bmatrix} 0 & \cdots & x^i_{t^i_k} - x^j_{t^j_k} & y^i_{t^i_k} - y^j_{t^j_k} & 0 & \cdots & 0 & x^j_{t^j_k} - x^i_{t^i_k} & y^j_{t^j_k} - y^i_{t^i_k} & \cdots & 0 \end{bmatrix}
\end{aligned}
\tag{D.18}
$$

and

$$
d^i_t \triangleq h^R({}^m\mathbf{x}^i_{t^i_k}, {}^m\mathbf{x}^j_{t^j_k}) - \mathbf{r}^{i,j}_{t_k} \tag{D.19}
$$

Vita

Candidate's full name: Liam Paull

2008 - Present Ph.D., Electrical and Computer Engineering
University of New Brunswick
Advisors: Dr. Howard Li, Dr. Mae Seto and Dr. Liuchen Chang
Thesis Title: "Robust Online Survey Planning for Single and Multiple Autonomous Underwater Vehicles"

2007 - 2008 M.Sc., Electrical and Computer Engineering (Not Completed)
University of New Brunswick
Advisor: Dr. Liuchen Chang
Topic: "Advanced Control of Domestic Water Heaters for Demand Side Management"
Note: Fast-tracked to Ph.D. Results were published in [J4].

2001 - 2004 B.Sc.
Computer Engineering
McGill University

Journal Articles Accepted or Published

[J1] **Liam Paull**, Sajad Saeedi, Mae Seto, Howard Li. "Sensor-Driven Online Coverage Planning for Autonomous Underwater Vehicles." *IEEE/ASME Transactions on Mechatronics*. 2012. Early Access Online.

[J2] **Liam Paull**, Sajad Saeedi, Mae Seto, Howard Li. "AUV Navigation and Localization - A Review." *IEEE Journal of Oceanic Engineering*. In Press.

[J3] **Liam Paull**, Carl Thibault, Amr Nagaty, Howard Li. "Sensor-Driven Area Coverage for an Autonomous Fixed-Wing Unmanned Aerial Vehicle." *IEEE Transactions on Systems, Man, and Cybernetics - Part B*. Accepted May 2013.

[J4] **Liam Paull**, Howard Li, Liuchen Chang. "A Novel Domestic Electric Water Heater Model for a Multi-Objective Demand Side Management Program." *Electric Power Systems Research*. 80(12), pp1446-1451. 2010.

[J5] Sajad Saeedi, **Liam Paull**, Michael Trentini, Mae Seto and Howard Li. "Map Merging for Multiple Robots Using Hough Peak Matching." *Robots and Autonomous Systems*. Conditionally accepted.

[J6] Sajad Saeedi, **Liam Paull**, Michael Trentini, and Howard Li. “Efficient Map Merging Using a Probabilistic Generalized Voronoi Diagram.” *Robotics and Automation Magazine.* Conditionally accepted.

[J7] Sajad Saeedi, **Liam Paull**, Michael Trentini, and Howard Li. “Map Merging for Multiple Robot Simultaneous Localization and Mapping.” *International Journal of Robotics and Automation.* In Press.

[J8] Sajad Saeedi, **Liam Paull**, Mike Trentini, Howard Li. “Neural Network-based Multiple Robot Simultaneous Localization and Mapping”. *IEEE Transactions on Neural Networks.* 22(12), p2376 - 2387. 2012.

Book Chapters

[B1] **Liam Paull**, Sajad Saeedi, Howard Li. “Path Planning for Autonomous Underwater Vehicles.” in *Autonomy for Marine Robots.* Springer 2012. Editor: Dr. Mae Seto. p177-224.

[B2] M.L. Seto, **L. Paull**, S. Saeedi. “Introduction to Autonomy for Marine Robots.” in *Autonomy for Marine Robots.* Springer 2012. Editor:: Dr. Mae Seto.

Refereed Conference Publications

[C1] **Liam Paull**, Sajad Saeedi, Mae Seto, Howard Li. “Sensor Driven Online Coverage Planning for Autonomous Underwater Vehicles.” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* 2012.

[C2] **Liam Paull**, Gaetan Severac, Guilherme V. Raffo, Julian M. Angel, Harold Boley, Maki K. Habib, Bao Nguyen, Veera R. S. Kumar, Sajad Saeedi G., Ricardo Sanz, Mae Seto, Aleksandar Stefanovski, Michael Trentini, Howard Li. “Towards An Ontology for Autonomous Robots.” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* 2012.

[C3] **Liam Paull**, Sajad Saeedi G., Mae Seto, Howard Li. “A Multi-Agent Framework with MOOS-IvP for Autonomous Underwater Vehicles with Sidescan Sonar Sensors.” *International Conference on Autonomous and Intelligent Systems.* p. 41-50. 2011.

[C4] **Liam Paull**, Sajad Saeedi, Howard Li, Vincent Myers. “An Information Gain Based Adaptive Path Planning Method for an Autonomous Underwater Vehicle Using Sidescan Sonar.” *IEEE Conference on Automation Science and Engineering (CASE).* pp.835-840. 2010.

[C5] Sajad Saeedi Gharahbolagh, **Liam Paull**, Michael Trentini, Mae Seto, Howard Li. “ Map Merging Using Hough Peak Matching.” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* 2012.

[C6] Sajad Saeedi Gharahbolagh, **Liam Paull**, Michael Trentini, Mae Seto, Howard Li. “Efficient Map Merging Using a Probabilistic Generalized Voronoi Diagram.” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* 2012.

[C7] Sajad Saeedi Gharahbolagh, **Liam Paull**, Michael Trentini, Howard Li. “Neural Network-based Multiple Robot Simultaneous Localization and Mapping.” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* p.880-885. 2011.

[C8] Sajad Saeedi Gharahbolagh, **Liam Paull**, Michael Trentini, Howard Li. “Multiple Robot Simultaneous Localization and Mapping.” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* p. 853-888. 2011.

[C9] Arnaldo Sepulveda, **Liam Paull**, Walid G. Morsi, Howard Li, Chris P. Diduch, Liuchen Chang. “A Novel Demand Side Management Program Using Water Heaters and Particle Swarm Optimization.” *Electric Power and Energy Conference (EPEC).* pp.1-5. 2010.

[C10] **Liam Paull**, Derek MacKay, Howard Li, Liuchen Chang. “A Water Heater Model for Increased Power System Efficiency.” *Canadian Conference on Electrical and Computer Engineering (CCECE).* pp.731-734. 2009.

[C11] **Liam Paull**, Howard Li, Liuchen Chang. “The development of a fuzzy neural system for load forecasting.” *Canadian Conference on Electrical and Computer Engineering (CCECE).* pp.923-926. 2008.