Simultaneous Stochastic Mapping and Localization

by

Hans Jacob Sverdrup Feder

S.B., Mechanical Engineering (1995)
Massachusetts Institute of Technology

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

Doctor of Philosophy in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1999

Author .....
Department of Mechanical Engineering
May 15th, 1999

Certified by .....
John J. Leonard
Assistant Professor of Ocean Engineering
Thesis Supervisor

Accepted by .....
Anin A. Sonin
Chairman, Committee on Graduate Students

Simultaneous Stochastic Mapping and Localization

by

Hans Jacob Sverdrup Feder

Submitted to the Department of Mechanical Engineering
on May 15th, 1999, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Mechanical Engineering

Abstract

In order to create truly autonomous mobile robots, the task of building an accurate map of an *a priori* unknown environment and concurrently using that map to navigate is a central problem. This thesis focuses on methods for performing concurrent mapping and localization using a feature-based approach. The concurrent mapping and localization problem is cast as a stochastic estimation problem. Based on Kalman filtering techniques, augmented stochastic mapping is introduced as a method for performing concurrent mapping and localization in realistic scenario simulations and experiments. The role of data association ambiguity, track initiation and track deletion in the presence of uncertainty and non-linear system dynamics are addressed. A novel approach is introduced to overcome the computational complexity inherent in mapping large areas with many features. Adaptive concurrent mapping and localization based on choosing the robot's action so as to maximize the expected Fisher information is introduced in order to achieve improved performance. Results from simulations, land and underwater experiments, and post-processing of oceanic data are presented to demonstrate the validity of the proposed approaches. Once a region is mapped and localization information is available, planning collision free trajectories from the current position to the goal position is important for reliable mobile robot operations. In this context, a novel path-planning algorithm based on harmonic potentials is introduced for performing path-planning and obstacle avoidance in dynamic environments.

Thesis Supervisor: John J. Leonard
Title: Assistant Professor of Ocean Engineering

Acknowledgments

I dedicate this thesis to my parents, who never stopped believing in me and for guiding me through life while at the same time letting me be completely free. In addition I thank my father for introducing me to the world of science and my mother for introducing me to the world of art and humanities.

I would like to sincerely thank my advisor, Professor John Leonard, who kept me on track, who patiently took the time to explain things I did not understand, who listened to my ideas and took them seriously, who inspired me, challenged me and made graduate school a wonderful experience. His high ethical standard, genuine interest in the field, keen intellect, and dedication to students has made him a perfect advisor and a role model.

Claudia Rodriguez gave me the motivation, vision and courage to change research field and advisor as I felt disillusioned by the world of research and the life as a graduate student. Without her, I would not have regained my faith and love for science.

My most sincere thanks goes to my thesis committee members, Professor Derek Rowell and Professor Seth Lloyd. Derek's guidence in pursuing useful and novel problems, and helping define realistic goals has been an imense help. Seth's ability to grasp new challenges immediately and give truely original views and approaches to any and all problems was invaluable. It has been an honor having such truly generous scientists on my committe.

I would also like to thank the Norwegian Research Council for their generous support through grant 109338/410.

There are, of course, many others to whom I owe my thanks. No matter how many I list, there will always be more. To all of you my most heartfelt thanks.

Contents

List of Figures

List of Tables

List of Notations

Below is a list of the symbols and notation used. Some additional notes may be of use to the reader. Lowercase letters (Roman and Greek) are in general used to indicate variables, functions, or indices. Uppercase letters are in general used for sets and matrices, with the notable exception of $\mathbf{X}$ which is used to represent the *true* state (as opposed to the *estimated* state $\mathbf{x}$). Time indexes and any conditioning factors are set in subscripts. That is, the subscript $k|k-1$ in $\mathbf{x}_{k|k-1}$ indicates the estimated state $\mathbf{x}$ of the true state $\mathbf{X}$ at time k conditioned on all information up to and including time step $k-1$. Superscripts A and B are used to refer to estimates in a submap A and B respectively. Variables typeset in a sans serif font, for example, x, represents the variable in normal font, $\mathbf{x}$, at the same instance in time, but after a transition between maps has occurred.

A list of common abbreviations have been included here as well.

$\mathcal{F}$: A general dynamic model.

$\mathcal{H}$: A general observation model.

$\mathcal{N}$: A normal (Gaussian) distribution with the first argument being its mean and the second argument being the variance.

$\mathcal{M}$: Transformation relating the Fisher information between time steps recursively.

C: The covariance operator. For instance, $C(\mathbf{x}, \mathbf{x}) = E(\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T)$.

C: The cost function used in adaptive concurrent mapping and localization. It is defined by $C(\mathbf{P}) = \pi \prod_j \sqrt{\lambda_j(\mathbf{P}_{rr})} + \pi \sum_{i=1}^{N} \prod_j \sqrt{\lambda_j(\mathbf{P}_{ii})}$ and gives the total area of all the error ellipses defined by the matrix $\mathbf{P}$.

D: The event of something being detected (Appendix 2.B).

E: The expectation operator defined by $E(\mathbf{x}) = \int_{-\infty}^{\infty} p(\mathbf{x})\mathbf{x}d\mathbf{x}$, where $p(\mathbf{x})$ is the probability density of $\mathbf{x}$.

$\mathbf{F}$: Linear dynamic model matrix (Appendix 2.A.)

$\mathbf{F_x}$: Jacobian of the dynamic model $\mathbf{f}$ with respect to $\mathbf{X}$ evaluated at $\mathbf{x}_{k|k}$.

F: The set of possible extraneous returns (Appendix 2.B).

$\mathbf{G}$: Process noise scaling matrix. Dependent on $\mathbf{u}$.

$\mathbf{H}$: Linear observation model matrix (Appendix 2.A).

$\mathbf{H_x}$: Jacobian of the observation model $\mathbf{h}$ with respect to $\mathbf{X}$ evaluated at $\mathbf{x}_{k|k-1}$.

$\mathbf{I}$: Fisher information matrix or the identity matrix.

$\Im$: Function taking the imaginary part of its argument.

$\mathbf{K}$: Kalman filter gain matrix.

$\mathbf{L}$: Jacobian of $\mathbf{l}$ with respect to $\mathbf{x}_r$, evaluated at $\mathbf{x}_{r_{k|k}}$.

L_j: Half length of panel j (path planning).

N: Number of elements in the state vector. Number of features in the map.

P_D: Probability of detection.

$\mathbf{P}$: System (error) covariance matrix defined by $\mathbf{P} = E\left((\mathbf{x}-\mathbf{X})(\mathbf{x}-\mathbf{X})^T\right) = E(\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T)$.

$\mathbf{P}_{ii}$: Feature i's (error) covariance, defined by $\mathbf{P}_{ii} = E(\tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_i^T)$.

$\mathbf{P}_{ff}$: Feature-to-feature (error) covariance, defined by $\mathbf{P}_{ff} = E([\tilde{\mathbf{x}}_1^T \ldots \tilde{\mathbf{x}}_N^T]^T [\tilde{\mathbf{x}}_1^T \ldots \tilde{\mathbf{x}}_N^T])$.

$\mathbf{P}_{rf}$: Vehicle-to-feature (error) covariance, defined by $\mathbf{P}_{rf} = E\left([\tilde{\mathbf{x}}_r \tilde{x}_1 \ \tilde{\mathbf{x}}_r \tilde{y}_1 \ \ldots \mathbf{x}_r \tilde{x}_N \ \tilde{\mathbf{x}}_r \tilde{y}_N]\right)$.

$\mathbf{P}_{rr}$: Vehicle (error) covariance defined by $\mathbf{P}_{rr} = E(\tilde{\mathbf{x}}_r \tilde{\mathbf{x}}_r^T)$.

$\mathbf{P}_{ri}$: Vehicle-to-feature i (error) covariance, defined by $\mathbf{P}_{ri} = E([\tilde{\mathbf{x}}_r \tilde{x}_i \ \tilde{\mathbf{x}}_r \tilde{y}_i])$.

$\mathbf{P}_{fr}$: Feature-to-vehicle (error) covariance. $\mathbf{P}_{fr} = \mathbf{P}_{rf}^T$.

$\mathbf{Q}$: Dead reckoning observation covariance matrix, $\mathbf{Q} = E(\mathbf{d_m} \mathbf{d_m}^T)$.

$\mathbf{S}$: Innovation covariance matrix, $\mathbf{S} = C(\boldsymbol{\nu}, \boldsymbol{\nu}^T)$.

$\mathbf{R}$: Observation covariance matrix, $\mathbf{R} = E(\mathbf{d_z} \mathbf{d_z}^T)$.

$\Re$: Function taking the real part of its argument.

T: Time period between each measurement, $t = kT$.

T: Set of observable features (Appendix 2.B).

U_θ: Set of sonar scanning angles.

$\mathbf{U}$: Velocity field (path planner).

$\mathbf{V}$: Velocity of vehicle (path planner).

V: A volume.

V_c: Nominal velocity of vehicle.

$\mathbf{X}$: Actual (true) system state vector.

α: An angle.

c: The volume of the d dimensional unit sphere, i.e. 2, π, $4\pi/3$, etc.

d: Dimensionality number, i.e. a two dimensional space has $d = 2$ (Appendix 2.B).

$\mathbf{d_m}$: Dead-reckoning measurement noise process. Assumed to be white and Gaussian.

$\mathbf{d_x}$: Process noise (assumed to be white and Gaussian).

$\mathbf{d_z}$: Observation noise process (assumed to be white and Gaussian).

$\mathbf{f}$: Nonlinear dynamic model.

$\mathbf{h}$: Nonlinear observation model.

i: Subscript index identifying a feature number; the imaginary number, $i^2 = -1$.

j, k, n: Time indices (placed in subscripts).

$\mathbf{l}$: Transformation used for adding a new element to the state vector.

$\mathbf{m}$: Dead-reckoning measurement vector, defined by $\mathbf{m} = [\varphi \ \nu]^T$.

$p(\cdot)$: A probability density function.

r: A radius.

$\mathbf{u}$: Actions, or system control input.

u: Speed in x-direction for path planning.

δv: Change in velocity control input.

$\delta\phi$: Change in heading control input.

$\mathbf{n}$: Normal vector to a surface.

t: Continuous time variable.

v: Vehicle's speed; speed in y-direction for path planner.

v: A small volume (Appendix 2.B).

$\mathbf{x}$: Estimated system state vector.

$\mathbf{x}_{k|j}$: Estimated state at time step k given all the information up to time step j.

$\mathbf{x}_f$: Estimated state vector for all features, $\mathbf{x}_f = [\mathbf{x}_1^T \ldots \mathbf{x}_N^T]^T$.

$\mathbf{x}_i$: Estimated state vector for feature i given by $[x_i \; y_i]^T$.

$\mathbf{x}_r$: Vehicle state given by $[x_r \; y_r \; \phi \; v]^T$ representing the vehicle's east, north, heading and velocity respectively.

x_i: East position of feature i.

x_r: Vehicle's east position.

y_i: North position of feature i.

y_r: Vehicle's north position.

$\mathbf{z}$: Measurement vector (range and bearing measurements).

w: A complex potential (path planning).

λ: The mean of the Poisson distribution of false returns (clutter points).

λ_j: The j-th eigenvalue of its argument. Panel strength of the j-th panel (path planning).

λ_N: Mean of the Poisson distribution of true features (Appendix 2.B).

λ_F: Same as λ (Appendix 2.B).

γ: Gating parameter.

ρ: Signifies the standard deviation number. For instance, when integrating a normal distribution over $\rho = 2$, yields the 95.5% confidence volume (Appendix 2.B).

ϕ: Vehicles heading; a potential field (path planning).

ψ: Stream function.

φ: Heading measurement by on-board compass (used for dead-reckoning).

ν: Velocity measurement of vehicle relative to water column (used for dead-reckoning.)

$\boldsymbol{\nu}$: The innovation, $\boldsymbol{\nu} = \mathbf{z} - \mathbf{H}\mathbf{x}$.

σ: A standard deviation.

σ_i: The square root of the i'th eigenvalue of the innovation covariance matrix $\mathbf{S}$ (Appendix 2.B).

θ: Angle of a panel (path planning).

$\tilde{\cdot}$: The tilde character '~' over a variable indicates the error vector between the estimate and the true state. For instance, $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{X}$.

$\sim$: The tilde character placed in front of a variable signifying an event, yields the negative of the event. For example, $\sim D$, reads *not* detected, as D signifies detected (Appendix 2.B).

Common abbreviations

AUV: Autonomous underwater vehicle, see Figure 1-1.

ASM: Augmented stochastic mapping. An algorithm based on stochastic mapping with the incorporation of a track initiator, track deleter and a data association filter. See Section 2.4.

BLS: Bayes Least Square estimator.

CML: Concurrent mapping and localization. The problem for a robot of mapping and unknown environment and concurrently using that map to localize. In the context of a "CML algorithm" it is meant a stochastic mapping based algorithm for performing CML.

DNN: Delayed nearest neighbor. A logic based algorithm used to perform track initiation. See Section 2.4.

EKF: Extended Kalman filter, see Appendix 2.A.

FLS: Forward look sonar. A sonar that is placed on the ship so that it looks forwards.

IMAN: Integrated mapping and navigation. A multiple-hypothesis approach to CML (Chapter 1).

INS: Inertial navigations system. A system based on the use of gyros whose acceleration readings are integrated twice to obtain position information.

GPS: Global Positioning System. A constellation of satellites that by use of a GPS receiver allow for the determination of the receivers position to about 30 meters accuracy globally. A differential GPS (DGPS) uses a correction signal from a transmitter on land to bring the accuracy to a couple of meters globally.

HRA: An electronically scanned high resolution sonar.

MMSE: Minimum mean square error.

NNSF: Nearest neighbor standard filter. A gated nearest neighbor data association technique. In its original definition of Bar-Shalom [4], the vehicle uncertainty is not taken into account as is done in this thesis. However, we use the term NNSF to refer to a gated nearest neighbor data association technique where the vehicle uncertainty is taken into account.

RCD: Regions of constant depth. When using sonars, due to the width of a sonar beam, a surface will be observed over a range of angles, causing distance measurements to be close to horizontal over this range of angles in an angle

versus range plot. An RCD is also defined to be the range and angle of the mode of the returns determined to form regions of constant depth.

ROV: Remotely operated vehicle. As opposed to AUV's, ROV's are operated by humans typically through a cable link.

PDF: Probability distribution function.

SM: Stochastic mapping. An extended Kalman filter arranged in such a way as to perform CML (see Chapter 2).

TOF: Time-of-flight. Refers to the time it takes form a sonar signal is emitted until its reflection of some surface is detected. Using this, the distance to the reflecting surface can be determined (see Chapter 2).

rms: Root mean square, i.e., the rms of a set of values, $\mathbf{x} = [x_1 \dots x_N]$ is given by $\sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}$.

Chapter 1

Introduction

Navigating on the open seas was the gravest of scientific challenges in the 18th century[112]. Lacking the ability to determine their position, sailors were literally lost at sea as soon as they lost sight of land. Ships ran aground on rocky shores; those traveling well-known routes were easy prey to pirates. As a single ship would frequently carry merchandise across the oceans worth more than the English treasury, the navigation problem posed a major national concern. Today, obtaining position information is a simple task by use of the Global Positioning System. However, the methods used by humans for navigating in large, unknown environments are in many cases not available to the mobile robot, leaving them to navigate by the same basic methods of the 18th century – integrating velocity and heading measurements. This thesis is dedicated to the navigation problem of autonomous underwater vehicles, and mobile robots in general. The navigation problem is viewed in a more general sense, by requiring the concurrent mapping of the environment and localization using this map. The problems encountered by the mobile robot of deciding where to move and how to plan a path from the current position to the goal are viewed as part of the general navigation problem.

1.1 Mobile robot navigation

Navigation is a fundamental requirement for obtaining truly autonomous mobile robots. Due to its critical importance and theoretical challenges this problem has been studied extensively in the robotics research community, however, many fundamental questions remain unanswered. A key challenge for mobile robots is coping with uncertainty, a point stressed by Thomás Lozano-Pérez:

> "The key scientific and technological issue in robotics is that of coping with uncertainty ... In fact, the uncertainty is such that one of the most challenging activities for a mobile robot is simply going from point A to point B." [35]

The navigation problem, according to Leonard and Durrant-Whyte [69], can broadly be stated as the problem of answering these questions: "where am I?", "where am I going?" and "how do I get there?" However, in order to answer any of these questions, one needs to know what environment and coordinate system these questions refer to. A coordinate system along with information about the location of environmental features with respect to this coordinate system is called a *map* of the environment. Thus, there is one important question that needs to be answered before any of the prior question can be addressed, namely "what is my map?" Knowing that you are at position $(264, 424)$ or at point A gives no information unless you know which map you are referring these coordinates to. Likewise, answering the second and third questions by "I want to move to C through path P" is of little value unless you can relate this to a specific map. Because of the necessity of being able to answer the first and last question, that is, the question of localization and the question of defining a map, in answering the two other questions, the problem of navigation is often defined as that of localization and mapping. Most of the work in this thesis is dedicated to the problem of navigation, that is, mapping and localization.

The question "where do I want to go?", is a problem of defining a mission goal and is most often defined *a priori*. Thus, this problem has not explicitly been addressed in this thesis. However, a solution arises in the development of an algorithm that

adaptively decides where to go in order to increase the navigation performance in situations where the mission goal is not defined *a priori.*

Once a map of the environment is available from which accurate localization information can be derived, the problem of how to find a collision free path from the current position to the goal position becomes important, that is, the robot needs to find a solution to the question "how do I get there?" The problem of path-planning is a field that has been studied extensively in the literature. However, the problem of path-planning in dynamic environments still remains a difficult problem as conventional methods would require the planned path to be recomputed every time something in the environment changes. Thus, operating in environments with continuously moving objects might cause the algorithm to get stuck, indefinitely computing new paths. For this reason, the problem of path-planning in dynamic environments is addressed and a solution is proposed.

Implementation of a navigation system that uses artificial beacons or *a priori* known maps of the environment, in addition to senor systems that can provide accurate and reliable measurements of the beacons or map features, is straightforward for today's robots. Similarly, the task of building an accurate map of the environment given the robot's position is straightforward. However, the much harder problem of solving the complete problem simultaneously, that is, the problem of concurrently mapping an *a priori* unknown environment and using this map to precisely localize the robot, still remains an open issue. We will refer to this problem as the concurrent mapping and localization (CML) problem. The approaches to CML can roughly be categorized into three main approaches: (1) "grid-based", (2) "feature-based" and (3) "topological". The following section summarizes the state of the art of these approaches to mobile robot navigation.

1.2 Approaches to mobile robot navigation

The *grid-based* approach to the CML problem is the conceptually simplest approach. The approach was first introduced by Moravec and Elfes [86] in 1985 and extended and utilized by others [37, 85, 114, 127]. This approach is also referred to as a "metric" approach to CML [119, 120]. In grid-based approaches, the environment is divided into a grid of cell's of some fixed physical size. Each cell is assigned a probability for the existence of an object at the cells location. A cell that is completely filled with an object would be assigned the value 1, while a cell that for sure is free of any object will be assigned the value 0. This algorithm pixilates the environment and, thereby, generates a map of the environment. Such a map is referred to as a certainty grid. To localize a robot, the robot creates a new certainty grid of its local environment. A search through a set of previously acquired global certainty grids is performed to find the best match (correlation) between the new and old certainty grids. The location of the robot that results in the highest correlation is defined as the new estimate of the robot's location. Once the vehicle's location is determined, the measurements that define the new certainty grid are merged with the old certainty grid, thereby, presumably, increasing the accuracy of the old certainty grid. The major advantages of grid-based approaches to CML are that they are intuitively simple, easy to implement and naturally extend to higher dimensions [114]. The current state-of-the-art is exemplified in the work of Thrun *et al.*, who have successfully used a grid-based approach to perform navigation in a museum [23].

While grid-based approaches are conceptually simple and relatively successful for certain environments, differentiating similar environments is difficult, the computational cost of localization is high, and the high storage requirement all pose serious problems [120]. Grid-based approaches have a weak theoretical foundation as data are smeared, thereby at times removing critical information or compounding information inconsistently. Their quality and robustness are critically dependent on the sensor system used and how sensor information is interpreted and transfered into the

uncertainty grid. Further, false alarms, data association ambiguities and no returns are all difficult to incorporate.

Feature-based approaches to CML use easily identifiable attributes in the environment, such as planes, corners and edges in indoor environments, and build a metrically accurate map of the location of these landmarks [17, 29, 40, 89, 111]. The pioneering work in the field was done by Smith, Self and Cheeseman [111] with their introduction of stochastic mapping. In stochastic mapping, the state of the vehicle and all the features in the map are represented by a single estimated state vector and an estimated covariance matrix. The system is based on an extended Kalman filter (EKF) [4, 46] to build and update a feature map of the environment from the sensor measurements (typically a sonar or a laser range finder) as well as a dead-reckoning model for the vehicle. Localization is performed concurrently by updating of the vehicle's position through an EKF. That is, when features in the environment are re-observed, these measurements are used to improve the position estimate of the robot and all the features estimates (i.e. the map). This approach is based on the assumption that the uncertainties have a Gaussian distribution. The main advantage of stochastic mapping is that it can provide metrically accurate navigation information robustly and can incorporate false returns, drop-outs and data-association ambiguities. Stochastic mapping does not *per se* address the data association problem, that is, how to resolve uncertainties in the origins of measurements. Further, it does not incorporate changing environments. These are, naturally, critical for the implementation of any real applications of stochastic mapping. Thus, extensions to stochastic mapping that, to a varying degree of autonomy, incorporate track initiation, data association and track deletion, have been implemented [30, 29, 17, 51, 40, 42]. The successful utilization of stochastic mapping requires that features can be extracted from the environment, and further that data association errors are few. Further, the computational complexity of operating in large areas has been viewed as one of the most significant challenges to performing feature-based CML. In most feature-based approaches, the computational

complexity, also referred to as the *map scaling problem*, is due to the ever-growing number of vehicle-to-feature and feature-to-feature correlations that must be maintained as the size of the operating environment increases.

A notable implementation of a feature-based approach to CML is that of Smith [105] with the introduction of integrated mapping and navigation (IMAN). IMAN explicitly addresses the problem of data association. IMAN forms multiple hypotheses for each measurement (similar to what is done in multiple hypotheses tracking [95]) but in addition forms multiple hypotheses for the location of the robot. That is, if a measurement of a feature in the environment is obtained, but it is ambiguous whether feature A or feature B was observed, two hypotheses are generated. As more measurements are obtained, one of the hypotheses is likely to be eliminated as it becomes inconsistent with the measurements. Further, the estimated robot position would differ depending on which of these hypotheses are chosen, thereby forming multiple hypotheses for the vehicle's state as well. For each set of consistent hypotheses, stochastic mapping is performed. To eliminate tracking of an explosively increasing number of hypotheses, different pruning and gating techniques are employed. This method is the most general and theoretically consistent approach to the data association problem. So far, the implementation of IMAN, even when utilizing aggressive pruning strategies, has been computationally intractable for realistic scenarios [105].

Topological approaches to the CML problem do not attempt to build metrically accurate maps of the environment. Instead, graph-like descriptions of the environment are generated, where nodes correspond to "significant places" in the environment that are easy to distinguish, and arcs connecting these nodes correspond to actions or action-sequences that connect neighboring places. Such an approach has been proposed by Brooks [21], Mataric [76, 77, 78] and others [56, 101, 118]. It was believed that humans and animals do not produce accurate metric maps of the environment they operate in, thus motivating a topological approach in the artificial intelligence (AI) community. These approaches utilize reactive rules and are behavior-based,

thus the control and obstacle avoidance problems often become part of the navigation problem. For instance, in trying to navigate through a labyrinth, the solution might very simply be stated as follows: move forward until you are about to hit something, turn left and move forward. This algorithm would, actually, be successful in navigating in a labyrinth. These methods are appropriate for navigation and obstacle avoidance in simple environments and might be useful for creating models of human and animal cognition [118], however, they have not been applied successfully to large environments. Furthermore, for many operations, metrically accurate maps that are globally referenced are a necessity for achieving the goal of the mission.

In general, all approaches to concurrent mapping and localization exhibit rapidly increasing computational complexity with the size of the area to be mapped. Further, the problem of data association, that is, resolving ambiguities in the source of measurements, still remains an open issue. A third problem that is particularly critical for feature-based and topological approaches is the problem of identifying suitable features or "significant places" from the environment to be used for localization and map building purposes.

Having outlined the main approaches to mobile robot navigation, we now turn to the particular challenges faced by autonomous underwater vehicles. Motivated by the need for navigation systems for autonomous underwater vehicles, this study of mobile robot navigation is set in the context of autonomous underwater vehicles.

1.3 Autonomous underwater vehicles and CML

The concurrent mapping and localization (CML) problem is particularly critical in environments where no *a priori* map of the environment is readily available. Although the capability of a robot to perform CML rather than just mapping or localization alone is desirable, it is often not a critical requirement. Accurate external positioning information or *a priori* maps suitable for navigation purposes can, if needed, often

be supplied. For instance, prior maps of the floor plans of buildings can often be obtained relatively easily. The challenge for the mobile robot then becomes one of map maintenance, obstacle avoidance and path planning as the prior map might be incomplete or the environment might change. Further, if needed, easily identifiable artificial land-marks can be placed in the environment for accurate localization purposes. Similarly, the Global Positioning System is readily available for most outdoor mobile robots, or accurate maps of the environment can often be generated if necessary. However, readily available positioning systems or accurate maps are not, in general, available underwater for use by autonomous underwater vehicles (AUV's) or for autonomous mobile robot navigation underground. Thus, in our view, these are two of the areas of mobile robot navigation where CML is truly a critical requirement, and for this reason, the most important, useful and interesting context for the study of CML.

The problem of CML becomes particularly challenging for autonomous underwater vehicles: electro-magnetic signals penetrate only a few meters in water, thereby greatly limiting their applicability for sensing and communication. Acoustic signals, on the other hand, can penetrate long distances underwater. However, sensor information based on acoustics is generally low bandwidth and noisy. AUVs are often required to be in constant motion in order to maintain controllability. Further, AUVs operate in an "active" environment, with the presence of unknown currents, which makes navigation more difficult. The power limitation of AUVs is often more severe than for most land-based robots. Accurate navigation remains one of the primary obstacles for having truly autonomous underwater vehicles. We believe, however, that the potential for AUVs that perform concurrent mapping and localization with consistent, bounded errors for long-term, large-scale missions are truly immense, both for scientific, commercial and military applications.

Due to the frequent requirement of accurate, metric navigation during AUV missions, utilizing a topological approach to the CML problem of AUVs is not viable.

Further, the underwater environment does not have a simple topological structure, such as office environments, with doors and corridors. The primary sensor used for making observations of the environment by AUVs is sonar, which exhibits frequent drop-outs and false returns. Unlike grid-based approaches, feature-based approaches can address the physics of the sonar, data association, and ambiguities in the sensor readings explicitly. Further, feature-based approaches have a more solid theoretical foundation based on optimal estimation theory [111] and, we believe, have a greater potential for providing more accurate estimates as well as consistent, bounded errors. This, we feel, makes a feature-based approach to CML for AUVs more desirable than a grid-based approach. Thus, this thesis takes a feature-based approach to the CML problem faced by AUVs.

The use of AUVs for scientific purposes have been advocated for some time [7, 9, 15] and many successful missions have been accomplished [8, 10, 13]. The Odyssey IIB, depicted in Figure 1-1, is an example of an AUV primary designed for performing scientific missions. Most AUVs designed for scientific purposes are relatively inexpensive and small. They often can explore very deep oceans. For instance, the Odyssey IIB is rated to 6000 meters. Further, their operational cost can be much lower than that of remotely operated vehicles (ROVs) due to less requirements of the host vessel. AUVs also have a advantage over ROVs in that AUVs can operate in areas that are unreachable by ROVs, such as, under the polar ice cap. The potential capabilities of AUVs that navigate accurately for surveying large areas, measuring many quantities, more quickly, cheaply and accurately than any current technologies, will give us access to tremendous amounts of oceanic data not presently available. The impact this data can have on our understanding of the oceans, marine life and the interaction between the ocean and atmosphere is truly great.

Commercially, there are many operations that could be performed more effectively by accurately navigating AUVs. AUVs would also make viable commercial applications that otherwise would be infeasible. Many of the missions that are now

Figure 1-1: An Odyssey IIB class scientific autonomous underwater surface craft (AUV) developed at MIT Sea Grant [11].

performed by ROVs could potentially be performed at a much lower cost using AUVs. For example, inspections of underwater oil pipelines, moorings etc. could potentially be done more cost effective by utilizing specialized AUVs. A limited number of AUV's designed for commercial applications do exist. For example, the Hugin unmanned underwater vehicle [115], performs accurate, cost-effective, bathymetric mapping up to 600 meters depth. However, navigation information is provided through an acoustic modem link from a survey vessel.

The potential application of AUVs for military purposes are only limited by the imagination. The most obvious uses are for covert operations, surveillance and mine-countermeasures in shallow and very shallow water. The goal of mine-countermeasures is to detect, localize and neutralize mines, often for the purpose of providing a safe path for amphibious invasion [22, 92]. During mine-countermeasures complete cov-

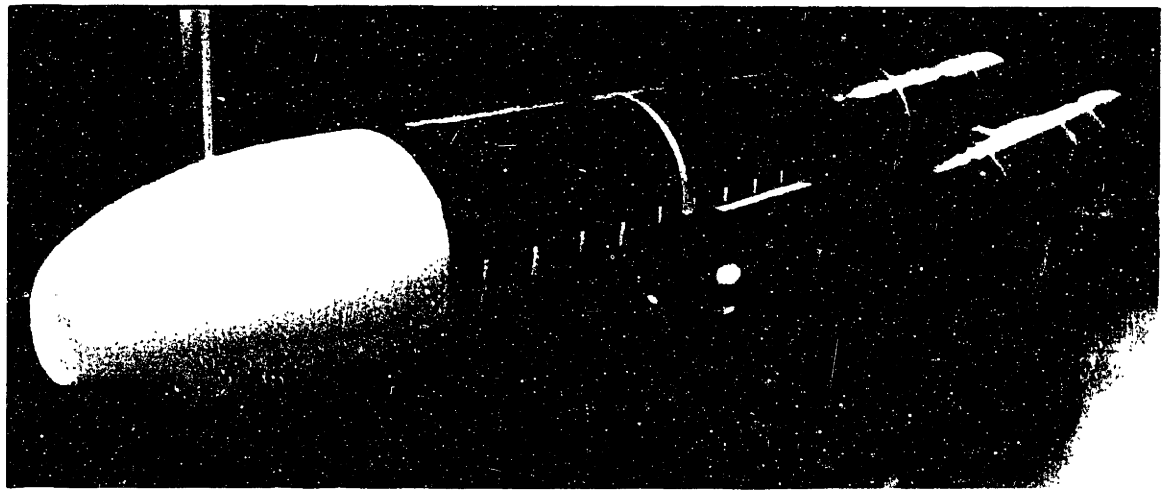

Figure 1-2: Two of the Naval Underwater Warfare Center's AUVs. The LDUUV is shown on the top left, while the 21UUV is shown its side to the right.

erage of the region is required, the region is likely to be *a priori* unknown. Further, it is often important that the mission is covert. Another complication is that the environmental forces in shallow water are often significant. Traditionally, divers or trained marine mammals have been used for identifying mine-like objects. However, these operations are dangerous and alternative approaches that would reduce or eliminate the need for direct involvement, while at the same time exhibiting a high level of performance in mine clearing are desirable. In such regions, the use of AUVs can be an attractive alternative. One or more AUVs, equipped with commercially available sonar transducers and instruments, can provide a solution to the mine-countermeasures problem in shallow water, provided that accurate navigation can be performed. Navigation is a key requirement as complete coverage must be accomplished and accurate localization of the mines must be achieved in order to neutralize the mines.

Figure 1-2 shows two US Navy AUVs developed at the Naval Underwater Warfare Center in Newport, Rode Island [27]. The Large Diameter Unmanned Underwater Vehicle (LDUUV) is a military research vehicle for evaluation of state-of-the-art AUV technologies. It is 300 inches in length and 26.5 inches in diameter. It is equipped with

a highly accurate laser gyro inertial navigation system (INS) and a high performance Doppler velocity sonar to achieve dead-reckoning errors of less than 1% of distance traveled. The 21UUV is a test-bed for performing advanced mine countermeasures. The 21UUV is 250 inches long, with a displacement of approximately 2800 pounds.

1.4 AUV navigation techniques

Navigation is one of the key challenges that limits our capability to use AUVs to address problems of critical importance to society. The three primary methods for navigation of AUVs are (1) dead-reckoning and inertial navigation systems, (2) acoustic beacon navigation, and (3) map-based navigation techniques. These methods are summarized in this section.

1.4.1 Dead reckoning and inertial navigation systems

The traditional and most obvious navigation technique is that of dead reckoning in which the vehicle velocity is is integrated in time to obtain new position estimates [74]. Measurement of the velocity components of the vehicle is usually accomplished with a compass and a log.[1] The principal problem is that the presence of an ocean current will add a velocity component to the vehicle that is not detected by the log. In the vicinity of the shore, ocean currents can exceed 2 knots. Consequently, dead reckoning for power-limited AUVs, operating at low speeds (3-6 knots), involving water-relative speed measurements can generate extremely poor position estimates.

In inertial navigation systems (INS), the accelerations of the vehicle are integrated twice in time to derive the updated position [64]. Cost and power consumption have historically made INS systems unattractive for small AUVs, however, this may change as systems get smaller and cheaper in the future. For operations near the seabed, Doppler Velocity Sonar (DVS) sensors can be used to measure the vehicle's velocity

[1] A log measures the vehicle's speed relative to the water column.

relative to the ground. The integration of this information in a Kalman filter can greatly improve performance of dead reckoning.

The problem with exclusive reliance on dead reckoning or inertial navigation is that position error increases without bound as the distance traveled by the vehicle increases. The rate of increase will be a function of ocean currents, the vehicle speed, and the quality of dead reckoning sensors.

1.4.2 Beacon-based navigation

Acoustic energy propagates well in the ocean, and hence acoustic transponders can be used as "beacons" to guide the motion of an AUV without the need for resurfacing. Primarily two types of system have been employed [52, 54, 81]: long baseline (LBL) and ultra-short baseline (USBL). Both systems employ external transducers or transducer arrays as aids to navigation. Position fixes are obtained by detecting the outgoing pulse from the transducers together with the *a priori* knowledge to the location of the beacons to infer the vehicles location.

Deploying a LBL or USBL system to provide accurate navigation for an AUV requires careful calibration of the positions of the individual beacons. This is costly and not viable for many situations, such as, under the polar cap or in deep ocean environments.

1.4.3 Map-based navigation

For some applications of AUVs, the use of acoustic beacons is undesirable or impractical. If an accurate *a priori* map of the environment is available, one approach to globally-referenced position estimation is to use measurements of geophysical parameters, such as bathymetry, magnetic field, or gravitational anomaly [47, 79, 121, 122]. These approaches are based on matching sensor data with an *a priori* environment map, under the assumption that there is sufficient spatial variation in the parameter(s) being measured to permit accurate localization.

The major disadvantage of using geophysical maps, is that accurate maps appropriate for navigation are simply not available for most areas. Further, accurate and reliable matching of sensor readings to *a priori* maps poses a significant challenge.

As up-to-date, high-quality map may be unavailable in the operating area of interest, thus motivating research into the problem of concurrent mapping and localization (CML).

A basic problem with CML based AUV navigation systems, is the identification and recognition of natural features in the environment. Sonar data can be notoriously difficult to interpret. Spurious measurements due to multiple reflections are common. Considered in isolation, an individual sonar return yields insufficient information to determine the shape of an object. The fundamental capability required is to combine the information provided by multiple sonar returns obtained from different sensing location. Approaches for obtaining reliable feature extraction in context of CML have been successful for simple man-made features, while reliable feature extraction for natural oceanic features have only recently been addressed [24]. However, motivated by the abilities of bats [6, 61, 102] and dolphins [93] to navigate in cluttered environment using sonar, new sonars and processing techniques have been developed to mimic the performance of bat and dolphin sonar systems and object detection capabilities. The use of these systems for CML presents an interesting topic for future research.

Another major challenge encountered by stochastic mapping is the fact that the technique scales (at best) quadratically with the number of features present. The drawback of increasing computational complexity with the number of features mapped have been addressed by several authors [29, 51, 124]. However, these approaches have either diverged, or failed to provide long-term, globally bounded errors. Chapter 3 introduces a promising new method for solving this problem.

AUVs have a limited range due to limited power consumption and tend to have dynamical constraints, such as loss of controllability at low speeds, that require fast

access to information about the environment for safe navigation. An interesting research area is to adaptively choose a sensing and motion strategy to obtain the most information about the environment. Such a strategy is introduced in Chapter 4. Similar adaptive strategies are seen in the way dolphins use sonar to localize and navigate by moving their heads from side to side. Bats adaptively increases the number of sound pulses emitted as they approach a target. Further, due to the low rate of information obtained from sonar, adaptive strategies are especially beneficial.

1.5 Thesis contributions

This thesis makes the following contributions:

- An analysis of the long-term performance of stochastic mapping for autonomous underwater vehicle navigation.
- A solution to the problem of map scaling.
- A method for performing adaptive concurrent mapping and localization.
- A technique for performing path planning in dynamic environments.

1.6 Thesis overview

The overall structure for this thesis is one in which each chapter can, to a large extent, be read independently of the other chapters. This structure will make reading easier for those who are mainly interested in particular aspects of the work.

The structure of this thesis is as follows: **Chapter 2** reviews stochastic mapping as a kernel for performing concurrent mapping and localization (CML). Methods for performing track initiation, track deletion and data association in combination with stochastic mapping are introduced. These methods are integrated with stochastic mapping, culminating in the Augmented Stochastic Mapping (ASM) algorithm for

performing CML. The long-term performance of ASM is analyzed by extensive simulations, experiments and post-processing of oceanic data.

Chapter 3 presents Decoupled Stochastic Mapping (DSM) as a solution to the map scaling problem of stochastic mapping. The map scaling problem refers to the problem of maintaining an ever increasing number of vehicle-to-feature and feature-to-feature correlations as the size of the operation environment increases. This, in turn, causes the computational resources to be exhausted.

Chapter 4 introduces Adaptive Stochastic Mapping, as a method for performing adaptive CML by maximizing the expected information from the next action of the vehicle, thereby obtaining increased CML performance.

Chapter 5 presents a method for performing path-planning in dynamic environments based on fluid analogies. This approach enable real-time computation of paths from local information among multiple dynamic objects.

Finally, **Chapter 6** presents a summary of the thesis, the main contributions and suggestions for future research. In addition, the nomenclature used in this thesis is included before Chapter 1.

1.7 Summary

The basic questions for a mobile robot addressed in this thesis are the questions of "what is my map?", "where am I?", "where am I going?", and "how do I get there?" The two first questions constitute the navigation problem. The third question is one of defining a goal, and the last question is one of path-planning. The current state-of-the-art in the three main approaches to mobile robot navigation, grid-based, feature-based and topological, were summarized. The critical need for accurate navigation of autonomous underwater vehicles (AUVs) was motivated and the particular constraints of AUVs were presented. Finally, the main contributions of this thesis were stated and an outline of the thesis was given.

Chapter 2

Navigating in the Unknown

The shortcomings of existing AUV navigation methods, outlined in Chapter 1, motivate the development of concurrent mapping and localization (CML) techniques. The goal of CML is for an AUV to build a map of an *a priori* unknown environment, while concurrently using that map to navigate. Simply stated, CML addresses the questions "what is my map?" and "where am I?" simultaneously. This chapter is devoted to the development of a feature-based approach to CML and its performance is analyzed via simulations, experiments and post-processing of oceanic data.

2.1 Introduction

While feature-based CML has been an important research topic in the robotics research community, there are many open issues that remain unanswered. One of the most important issues is to consider the long-term performance of feature-based CML. The duration of missions which have been considered in the existing literature have been surprisingly short — on the order of tens of meters in distance and/or a few tens of minutes in time. To enable true long-term autonomy, a key question that emerges is how does feature-based CML perform for missions over large areas with long time durations? What levels of performance are possible? What failures can occur and

what are the consequences of these failures?

To address these questions, the specific feature-based CML algorithm of augmented stochastic mapping (ASM), used throughout this thesis, and the associated modeling assumptions are presented.

It is highly unlikely that a general theoretical analysis of the asymptotic performance of CML can be derived that would be applicable over a wide range of situations and in the presence of nonlinearities and data association ambiguity. This is due to complexity that arises from non-linear estimation problems. Most characterizations of estimators only apply to linear cases, such as, the Cramér-Rao lower bound [5] and the notion of an efficient estimator. Further, in order to obtain a tractable estimator, simplifications are introduced (such as, linearization), thereby greatly reducing the certainty or the extent to which anything can be proven. As an alternative, this chapter presents a simulation study of ASM that analyzes the performance of the technique for long duration AUV missions for several different AUV operating scenarios. The positioning errors of the algorithm are compared against the uncertainty bounds for these estimates that are provided by the algorithm. A critical question for the usefulness of ASM is its ability to perform highly accurate CML in real world scenarios. For this reason, an experiment in a testing tank and post-processing of oceanic data provided by the US Navy is presented, confirming the viability of the approach. The types of failures that can occur during CML and their consequences are also examined.

The ASM algorithm for performing CML is based on stochastic mapping (SM), which was first introduced by Self, Smith and Cheeseman [111].

The structure of this chapter is as follows. Section 2.2 describes the AUV model used throughout the thesis. Section 2.3 derives the stochastic mapping approach to CML. Section 2.4 adds elements to stochastic mapping to make it viable for real life experiments and missions, cumulating in the technique of augmented stochastic mapping (ASM). Section 2.5 analyzes in simulation the performance of ASM for two

representative long duration missions. Section 2.6 presents a comparison between simulations and experiments performing ASM in a testing tank. Section 2.7 presents results for the performance of ASM during real world missions by post-processing oceanic data. Section 2.8 discusses some of the different failure modes that can occur in ASM due to state estimation divergence and data association errors. Section 2.9 draws conclusions and discusses future research and Section 2.10 gives a brief summary of this chapter. Appendix 2.A discusses the Kalman filter and Appendix 2.B analyses the performance of the gated nearest neighbor data association technique.

2.2 AUV modeling

In this section we develop a simplified model to capture the dynamics of an AUV, present the sonar model, and state our assumptions. These models and assumptions are, with some minor differences, used throughout this thesis.

In our work we have restricted our modeling to two dimensional space. This is done for simplicity of the models. Extending the modeling to three dimensions is a relatively straightforward task. While the challenges of representing complex vehicle models and three dimensional objects can be great, the structure of the basic uncertainty management problem encountered in CML remains the same.

2.2.1 Sonar model

In performing CML, observations of the environment useful for localization and mapping purposes must be available. In the marine environment, the seabed contains information useful for navigation purposes as it is relatively static. However, long-range remote observation of the seabed is only possible with sonar, as electro-magnetic sensing devices have a limited operation range in water.

A simple, off-the-shelf sonar operates by sending out a directed sound pulse and waiting for the return off a reflecting surface (feature) in the environment. By knowing

the speed of sound in the environment (approximately 1500 m/s), the sonar measures the time-of-flight (TOF) of the returning pulse to deduce the distance to the reflecting surface. The measurements obtained from sonar are generally ambiguous and noisy. First, due to the relatively slow speed of sound, measurements are obtained slowly in comparison to electro-magnetic sensors used on land (such as radar or optical sensors). For instance, getting a TOF reading in one direction of something 700 meters away takes almost a second. Second, the pulse has a beam-width, thus the pulse propagates out in a range of directions. When the TOF is measured, there is an uncertainty as to the exact direction to the reflecting surface. This makes the angle measurement uncertain. Third, the temperature, flow, and composition of the water column all affect the speed of sound, causing inaccurate TOF measurements. Forth, drop-outs, false-returns and multiple reflections are all common when using sonar, causing difficult data association problems, that is, determining the source of the reflection. Drop-outs occur as the reflecting surface from the object might be too poor to yield a strong enough return for the sonar to detect. False-returns occur as the acoustic background noise in the ocean environment triggers a TOF measurement. Multiple reflections occur when the pulse bounces off several surfaces, thereby giving rise to invalid TOF measurements.

The many ambiguities of sonar measurements are a result of the complex physics of sonar [20, 55], which are not fully understood. Increased knowledge of the physics of sonar can reduce many of these ambiguities and result in greater utilization of sonars for remote sensing.

There are two basic types of sonars, mechanically scanned and electronically scanned. When using a mechanically scanned sonar, the transducer is physically turned to the desired direction and a TOF measurement is obtained. When using a mechanically scanned sonar, a narrow beam is desired to achieve high accuracy for the direction to the reflecting surface. Mechanically scanned sonars are simple to operate, but slow. In the experiments in this thesis we use a mechanically scanned sonar. Due

to the width of the sonar beam, multiple TOF recordings occur from a single object during the scan. Such a set of TOF measurements is called a region of constant depth (RCD) [69]. An example of this is given in Figure 2-1 where a Polaroid ultrasonic

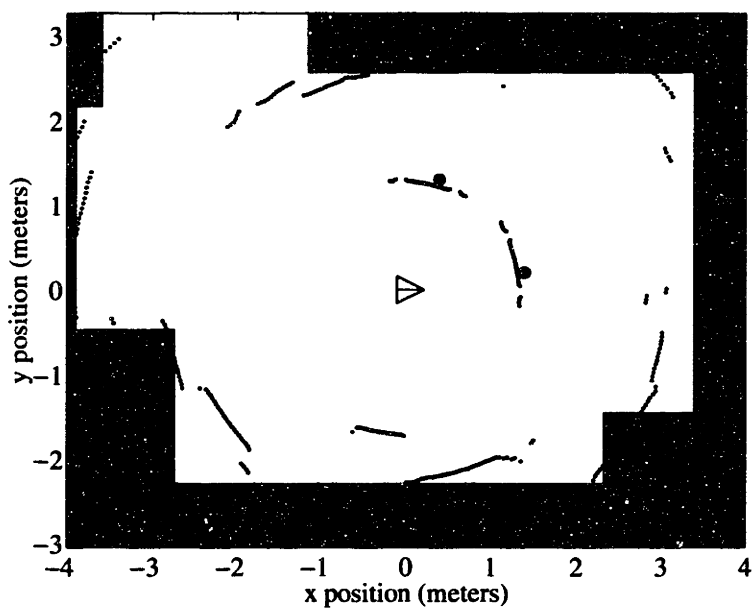

Figure 2-1: The returns from a Polaroid ultrasonic sensor (dots) with a rough shaded room model superimposed. The location of the senor is marked by a triangle.

sensor has taken a complete 360 degree scan of a room. In obtaining a single range and bearing measurement to a reflecting surface, the TOF at the bearing mode of the TOFs in a RCD is used as the range and bearing measurement. This method is called RCD extraction, and works well for specular environments.

An electronically scanned sonars emits an acoustic pulse with a known beam pattern covering a range of angles. The returning wave is detected by an array of hydrophones. An image is created by time-delay-and-sum beam-forming [55]. Electronically scanned sonars can detect reflecting surfaces over a whole area from a single emitted acoustic pulse. In our simulations, we assume that the vehicle is equipped with either a mechanically or electronically scanned sonar.

The attributes of a sonar sensor are taken into account by modeling the sonar

returns as noisy (Gaussian in range and bearing). Drop-outs are modeled by a probability of detection, P_D. False-returns are incorporated by a Poisson distribution of false returns with a mean of λ over the viewing area of the sonar. Dealing with data association ambiguities is discussed further in Section 2.4.1.

The viewing area of the sonar is the sector defined by the maximum range and angle coverage by the sonar and explicitly modeled.

In order to use sonar returns to estimate the location of features and of the vehicle, a mathematical model for the measurements is required. We assume the sonar emits an acoustic pulse (a "ping") every $t = kT$ seconds, where T is a constant period and k is a discrete time index. The estimated vehicle states and estimated feature states are modeled by a single state vector $\mathbf{x}_k$ and the true state vector by $\mathbf{X}_k$. The mathematical model for a RCD recording from an environmental feature (reflecting surface) is given by

$$\mathbf{z}_k = \mathbf{h}(\mathbf{X}_k) + \mathbf{d}_\mathbf{z}, \tag{2.1}$$

where $\mathbf{z}_k$ is the observation vector of range and bearing measurements relative to the vehicle. The observation model, $\mathbf{h}$, defines the nonlinear coordinate transformation from state to observation coordinates. The stochastic process $\mathbf{d}_\mathbf{z}$, is assumed to be white, Gaussian, and independent of $\mathbf{x}_0$. The covariance of $\mathbf{d}_\mathbf{z}$ is given by $\mathbf{R}$. The Gaussian assumption is a simplification, but not significantly different from the true stochastic process and thus a valid approximation.

2.2.2 Vehicle dynamic model

In our implementation, we denote the vehicle's state estimate by $\mathbf{x}_r = [x_r\ y_r\ \phi\ v]^T$ to represent the vehicle's east position, north position, heading and speed as shown in Figure 2-2. The dynamic model imitates an AUV equipped with control surfaces and a single aft thruster for propulsion, moving at a nominal forward speed of V_c. The

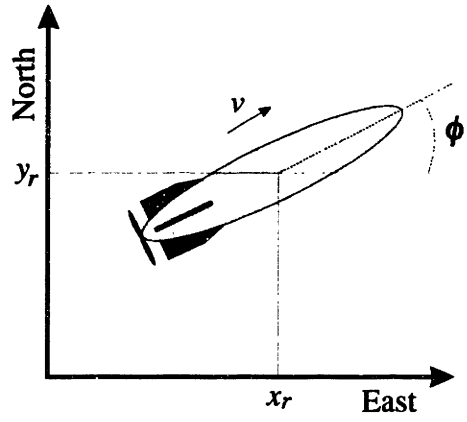

Figure 2-2: Definitions of the AUV's states used in the model.

control input $\mathbf{u}$ to the vehicle is given by a change in heading, $\delta\phi$, and speed, δv, of the vehicle to model changes in rudder angle and forward thrust, that is, $\mathbf{u} = [\delta\phi \ \delta v]^T$.

A general dynamic model of a vehicle can be defined as

$$\mathbf{X}_{r_{k+1}} = \mathbf{f}(\mathbf{X}_{r_k}, \mathbf{u}_k) + \mathbf{G}(\mathbf{u}_k)\mathbf{d}_\mathbf{x}, \tag{2.2}$$

where $\mathbf{f}$ is a nonlinear model taking the vehicle's current (true) state, $\mathbf{X}_{r_k}$, and control input, $\mathbf{u}_k$, as input variables. The noise process in heading and velocity, $\mathbf{d}_\mathbf{x} = [d_\phi \ d_v]^T$, is scaled as a function of the control input by $\mathbf{G}(\mathbf{u}_k)$ to account for unmodeled dynamics and noise. As for the sonar model, k signifies a discrete time index and the model is updated at times $t = kT$ for a constant period T.

In this thesis we have used the following definitions for $\mathbf{f}$ and $\mathbf{G}$

$$\mathbf{f}(\mathbf{X}_r, \mathbf{u}) = \mathbf{X}_r + \begin{bmatrix} 0 & T\cos(\phi) \\ 0 & T\sin(\phi) \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{u}, \tag{2.3}$$

$$\mathbf{G}(\mathbf{u}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \gamma_\phi & 0 \\ 0 & \gamma_\phi \delta v^2 \end{bmatrix}. \tag{2.4}$$

The scaling factor γ_ϕ is typically orders of degrees squared while the magnitude of γ_v is typically order of percent squared. The two first zero rows of $\mathbf{G}$ are due to the fact that there is no uncertainty as to the physical relationship between position and velocity. That is, the position of the vehicle is, per definition of position and velocity, the integral of velocity. The off-diagonal terms in the bottom two rows of $\mathbf{G}$ are by assumption set to zero. Although a minor point, if a known correlation between heading and speed existed it could easily be incorporated. However, we feel confident that for many AUV models, the assumption that the cross-correlation between the process noise in heading and speed is zero is reasonable. The stochastic process $\mathbf{d_x}$ is assumed to be Gaussian white noise and independent of $\mathbf{d_z}$. The covariance of $\mathbf{d_x}$ is for convenience set equal to the identity matrix, as any scaling is placed in $\mathbf{G}$. The Gaussian assumption is a simplification, however by appropriately choosing $\mathbf{G}$, a good approximation is achieved.

The vehicle dynamic model of Equation (2.3) does not take into account any of the vehicle's real dynamics. Thus the model is very general. Even if an accurate dynamic model for a specific vehicle was generated, a control input designed to give a specific response, such as, turn five degrees, would yield an inaccurate response by the vehicle (i.e. it might turn only four degrees). This is because there will always be unmodeled dynamics and uncertainties involved. That is, a control input of $\mathbf{u}$ applied to the real vehicle would have the same effect as a control input of $\mathbf{u} + \delta\mathbf{u}$ applied to the model of Equation (2.2). The unknown perturbation $\delta\mathbf{u}$ is modeled by the stochastic process $\mathbf{G}(\mathbf{u}_k)\mathbf{d_x}$.

We assume that the AUV is equipped with a log to measure its speed relative to the water column and an on board compass to yield a measurement of the heading.

Letting $\mathbf{m} = [\varphi \ \nu]^T$ be the vector of speed and heading measurement, the measurement model is given by

$$\mathbf{m} = \begin{bmatrix} \varphi \\ \nu \end{bmatrix} + \mathbf{d_m}, \tag{2.5}$$

where $\mathbf{d_m}$ is a Gaussian stochastic process with covariance given by $\mathbf{Q}$. The output, $\mathbf{m}$, from these instruments is integrated over time to yield a dead-reckoning model.

Having outlined the vehicle model, the next section describes stochastic mapping, which is a feature-based approach to navigation for mobile robots operating in unknown environments.

2.3 Stochastic mapping

The two seminal research efforts in feature-based concurrent mapping and localization were performed by Smith, Self, and Cheeseman [111], who first published the stochastic mapping algorithm, and Moutarlier and Chatila [88], who provided the first implementation of this type of approach with real data. Other notable implementations of variations of stochastic mapping include Rencken [96], Castellanos *et al.* [26], and Chong and Kleeman [30].

Stochastic mapping is simply a special way of organizing the states in an extended Kalman filter for the purpose of feature relative navigation. The much celebrated Kalman filter is an computationally efficient minimum mean square error estimator *with a dynamic system* describing variations in the unknown parameters. The extended Kalman filter (EKF) is the technique of linearizing a non-linear dynamic system for use in a Kalman filter and is the technique employed by stochastic mapping. A description of Kalman filtering is included in Appendix 2.A.

In the implementation of stochastic mapping, the observation and dynamic models for the AUV given in Section 2.2 are used. The measurements are used to create a

map of the environment, which, in turn, is used to localize the vehicle. We use

$$\mathbf{x}_{k|k} = \mathbf{X}_k + \tilde{\mathbf{x}}_k \tag{2.6}$$

to represent the estimated system state vector $\mathbf{x} = [\mathbf{x}_r^T \ \mathbf{x}_1^T \ldots \mathbf{x}_N^T]^T$, where $\mathbf{x}_r$ and $\mathbf{x}_1 \ldots \mathbf{x}_N$ are the estimated AUV and feature states, respectively, $\mathbf{X}$ is the true state vector, and $\tilde{\mathbf{x}}$ is the error of the estimate. k is a time index as defined in Section 2.2. The subscript in $\mathbf{x}_{k|k-1}$ will be used to signify the state estimate at time k given all information up to time step $k-1$. The estimated error covariance, $\mathbf{P}_{k|k} = E\{\tilde{\mathbf{x}}_k \tilde{\mathbf{x}}_k^T\}$, of the system takes the form

$$\mathbf{P}_{k|k} = \begin{bmatrix} \mathbf{P}_{rr} & \mathbf{P}_{r1} & \cdots & \mathbf{P}_{rN} \\ \mathbf{P}_{1r} & \mathbf{P}_{11} & \cdots & \mathbf{P}_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{Nr} & \mathbf{P}_{N1} & \cdots & \mathbf{P}_{NN} \end{bmatrix}_{k|k} . \tag{2.7}$$

The sub-matrices, $\mathbf{P}_{rr}$, $\mathbf{P}_{ri}$ and $\mathbf{P}_{ii}$ are the vehicle-to-vehicle, vehicle-to-feature, and feature-to-feature covariances, respectively.

Thus, the vehicle and the map are represented by a single state vector, $\mathbf{x}$, with an associated estimate error covariance $\mathbf{P}$ at each time step. The state estimate of feature i is represented by $\mathbf{x}_i = [x_i \ y_i]^T$. Given the definition of $\mathbf{x}$ and $\mathbf{P}$, an extended Kalman filter (EKF) is employed to estimate the state $\mathbf{x}$ and covariance $\mathbf{P}$ given the measurements $\mathbf{z}$. As with the Kalman filter, the estimation occurs through a Kalman update step which occurs when features are re-observed and a prediction step caused by the movement of the vehicle. However, as new features are observed new feature estimates, called *tracks*, must be added to the state vector and covariance. Thus, the evolution of stochastic mapping is divided into re-observation of features, new feature integration and vehicle movement. These steps are described next.

2.3.1 Re-observation of features

When a feature i is re-observed, we use the update step of the extended Kalman filter[1] to update the vehicle's state and the map. By introducing the following definitions:

$$\begin{aligned}\bar{x}_i &= x_{i_{k|k-1}} - x_{r_{k|k-1}},\\ \bar{y}_i &= y_{i_{k|k-1k}} - y_{r_{k|k-1}},\end{aligned}$$

the observation model $\mathbf{h}$ of Equation (2.1) for feature i takes the form

$$\begin{aligned}\mathbf{z}_{i_k} &= \begin{bmatrix} r_{i_k} \\ \theta_{i_k} \end{bmatrix} = \begin{bmatrix} \sqrt{\bar{x}_i^2 + \bar{y}_i^2} \\ \tan^{-1}(\bar{y}_i/\bar{x}_i) - \phi_k \end{bmatrix} + \mathbf{d}_{\mathbf{z}i} \\ &= \mathbf{h}_i(\mathbf{x}_k) + \mathbf{d}_{\mathbf{z}i}.\end{aligned}$$

The noise process $\mathbf{d}_{\mathbf{z}i}$ is assumed to be white and Gaussian with covariance $\mathbf{R}_i$ in accordance with the assumptions for Equation (2.1). If the N features $i_1 \ldots i_N$ are re-observed, the observation model becomes

$$\mathbf{z}_k = \begin{bmatrix} \mathbf{z}_{i_{1k}} \\ \vdots \\ \mathbf{z}_{i_{N_k}} \end{bmatrix}, \mathbf{h} = \begin{bmatrix} \mathbf{h}_{i_1} \\ \vdots \\ \mathbf{h}_{i_N} \end{bmatrix}, \mathbf{R} = \begin{bmatrix} \mathbf{R}_{i_N} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{R}_{i_n} \end{bmatrix},$$

with the Jacobian of $\mathbf{h}$ given by $\mathbf{H}_\mathbf{x} = \mathbf{H}_\mathbf{x}(\mathbf{x}_{k|k-1}) = d\mathbf{h}(\mathbf{X})/d\mathbf{X}|_{\mathbf{X}=\mathbf{x}_{k|k-1}}$. These matrices are used in the update step of the extended Kalman filter, to yield

$$\begin{aligned}\mathbf{x}_{k|k} &= \mathbf{x}_{k|k-1} + \mathbf{K}_k\left(\mathbf{z}_k - \mathbf{h}(\mathbf{x}_{k|k-1})\right),\\ \mathbf{P}_{k|k} &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1},\end{aligned}$$

where $\mathbf{K}_k$ is the extended Kalman filter gain given by

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_\mathbf{x}^T\left(\mathbf{H}_\mathbf{x}\mathbf{P}_{k|k-1}\mathbf{H}_\mathbf{x}^T + \mathbf{R}_k\right)^{-1}.$$

[1] See Appendix 2.A.

2.3.2 New feature integration

If the vehicle observes a new feature $\mathbf{z}_{\text{new}} = [r\ \theta]$ with respect to the vehicle's reference frame, a new feature state $\mathbf{x}_{N+1}$, called a *track*, is estimated and incorporated by

$$\mathbf{x}_{N+1} = \mathbf{l}(\mathbf{x}_{k|k}, \mathbf{z}_{\text{new}}) = \begin{bmatrix} x_r + r\cos(\phi + \theta) \\ y_r + r\sin(\phi + \theta) \end{bmatrix}.$$

The new feature is integrated into the map by adding this new state to $\mathbf{x}$ and $\mathbf{P}$. That is

$$\mathbf{x}_{k|k} \leftarrow \begin{bmatrix} \mathbf{x}_{k|k} \\ \mathbf{x}_{N+1} \end{bmatrix},$$

$$\mathbf{P}_{N+1N+1} = \mathbf{L}_{\mathbf{x}_r}\mathbf{P}_{rr}\mathbf{L}_{\mathbf{x}_r}^T + \mathbf{L}_{\mathbf{z}}\mathbf{R}\mathbf{L}_{\mathbf{z}}^T,$$

$$\mathbf{P}_{N+1\,i} = \mathbf{P}_{i\,N+1}^T = \mathbf{L}_{\mathbf{x}_r}\mathbf{P}_{ri}.$$

Where $\mathbf{L}_{\mathbf{x}_r}$ and $\mathbf{L}_{\mathbf{z}}$ are the Jacobian of $\mathbf{l}$ with respect to the robot state $\mathbf{x}_r$ evaluated at $\mathbf{x}_{r_{k|k}}$, and to $\mathbf{z}_{\text{new}}$ evaluated at $\mathbf{z}_{\text{new}}$. The covariance of the state of the new features is given *a priori* by $\mathbf{R}$ in the case that the new track is initiated from a single measurement. In practice, initiation of a new feature track is based on several measurements due to the imperfection of sensor measurements. In such a case, the covariance for each measurement, $\mathbf{R}$, should be compounded for all the measurements that are used to initiate the new feature track. This will result in a smaller covariance for the initiated track than $\mathbf{R}$.

2.3.3 Vehicle movement

When the vehicle moves, the resulting vehicle state estimate at time step $k+1$ is given by taking the expectation of the state transition model in Equation (2.2), that is,

$$\mathbf{x}_{r_{k+1|k}} = E[\mathbf{f}(\mathbf{X}_k, \mathbf{u}_k)] \approx \mathbf{f}(\mathbf{x}_{r_{k|k}}, \mathbf{u}_k).$$

Since the stochastic process $\mathbf{d_x}$ has zero mean, the term $\mathbf{G}(\mathbf{u}_k)\mathbf{d_x}$ becomes zero when computing the expected value over Equation (2.2) to determine $\mathbf{x}_{r_{k+1|k}}$.

The covariance, $\mathbf{P}_{k|k}$, is propagated through the linearized state transition model (see Appendix 2.A.2) to yield $\mathbf{P}_{k+1|k}$ given by

$$\mathbf{P}_{k+1|k} = \mathbf{F_x}\mathbf{P}_{k|k}\mathbf{F_x}^T + \mathbf{G}(\mathbf{u}_k)\mathbf{G}^T(\mathbf{u}_k).$$

Here $\mathbf{F_x}$ is the Jacobian of $\mathbf{f}$ with respect to $\mathbf{X}$ evaluated at $\mathbf{x}_{k|k}$.

Figure 2-3 illustrates the process of stochastic mapping. In Figure 2-3a, it is assumed that the initial state of the vehicle, $\mathbf{x}_r$, is known accurately, resulting in a small initial $\mathbf{P}_{rr}$. The initial system state vector is given by the vehicle's state and covariance, that is, $\mathbf{x} = \mathbf{x}_r$ and $\mathbf{P} = \mathbf{P}_{rr}$. The vehicle senses a feature to its left through a sonar reading and initializes the estimate $\mathbf{x}_1$ of this feature and adds it to the system state estimate as described in Section 2.3.2. This results in an expansion of the state vector to $\mathbf{x} = [\mathbf{x}_r^T \; \mathbf{x}_1^T]^T$ and the covariance to

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{rr} & \mathbf{P}_{r1} \\ \mathbf{P}_{1r} & \mathbf{P}_{11} \end{bmatrix}.$$

The covariance $\mathbf{P}_{11}$ defines how confident we are in our estimate of the feature's position and is represented by a solid ellipse in Figure 2-3a. This ellipse, which is centered around the estimated position of the feature, signifies that with some

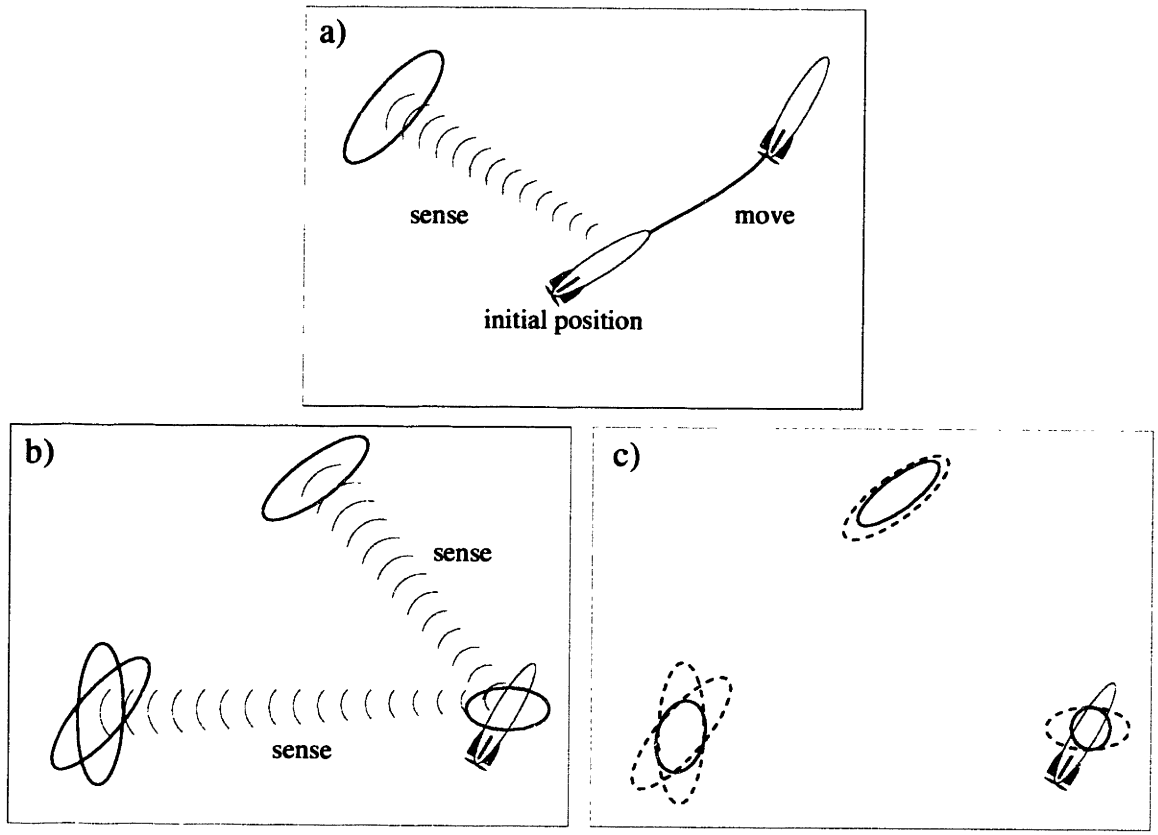

Figure 2-3: Illustration of stochastic mapping in the presence of two features. **a)** New feature integration and vehicle motion. The solid ellipse represents a confidence area for the observed feature. **b)** New feature integration and re-observation of feature. The vehicle's position becomes uncertain due to the movement of the vehicle, represented by a solid ellipse around the vehicle's position. Solid ellipses represent the measurement confidence regions from the current (b) and previous (a) time steps. **c)** The resulting feature and vehicle estimates and confidence regions (solid ellipse) along with the measurement confidence ellipses from the previous time step due to the re-observation of the first feature.

confidence (say 99%) the *true* feature position will be inside this area. Next, the vehicle moves to a new position, arriving at the vehicle position of Figure 2-3b. Since a vehicle movement is never precise, the position of the vehicle becomes more uncertain due to the movement, that is, $\mathbf{P}_{rr}$ grows as described in Section 2.3.3. This vehicle uncertainty is represented by the solid ellipse around the vehicle's position in Figure 2-3b.

In Figure 2-3b, the vehicle senses a second feature, feature 2. The state of feature 2, $\mathbf{x}_2$, is estimated from the measurement and integrated into the system state vector according to Section 2.3.2, that is

$$\mathbf{x} \leftarrow \begin{bmatrix} \mathbf{x}_r \\ \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}, \quad \mathbf{P} \leftarrow \begin{bmatrix} \mathbf{P}_{rr} & \mathbf{P}_{r1} & \mathbf{P}_{r2} \\ \mathbf{P}_{1r} & \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{2r} & \mathbf{P}_{12} & \mathbf{P}_{22} \end{bmatrix}.$$

The vehicle also re-observes feature 1. Since the vehicle is observing feature 1 from a different location than that of the first observation of feature 1 in Figure 2-3a, the confidence ellipse for this measurement is now tilted with respect to the confidence ellipse of this feature estimate in Figure 2-3a.

The feature re-observation step described in Section 2.3.1 is illustrated in Figure 2-3c. By combining the two observations of feature 1, the estimate of feature 1 is improved (i.e. $\mathbf{P}_{11}$ becomes smaller). This is represented by the smaller solid error ellipse for feature 1. This improved estimate of feature 1 is propagated to the other states, thereby improving the estimates of the vehicle and feature 2. This improvement is shown by the solid ellipses, which are smaller than the dotted ellipses that define the estimates before the re-observation step. Thus, through the feature re-observation step, the entire $\mathbf{P}$ becomes smaller, meaning that all feature estimates as well as the vehicle's estimate have improved.

Since the measurement of feature 1 was used to improve the estimate of feature 2 and the vehicle, the states become more correlated. When the vehicle moves, the vehicle and feature states become less correlated as the uncertainty introduced by the vehicle motion only affects the vehicle's estimate. As a result, in the limit as the number of re-observation of features goes to infinity, the feature-to-feature correlations approach unity, that is, perfect correlation. In the case of using point features as described here, perfect correlation means that $\mathbf{P}_{ii} = \mathbf{P}_{jj} = \mathbf{P}_{ij}$. However, the vehicle-to-feature correlations do not approach unity. This is a direct consequence of the

convergence of the Kalman filter. That is, each measurement made will bring some information to the system, causing the information of the system to increase. Thus, a measurement can only result in one becoming *more* confident in the estimate, that is, a smaller covariance. Since there is no process noise for features as they are assumed to be stationary, taking an infinite number of measurements of a feature and taking the average of these measurements, which is what the Kalman filter does, one will become more and more confident about this average, that is, the feature states. Similarly, as the feature states are correlated (that is, observing one feature gives some information about the location of the other feature, as illustrated in Figure 2-3c), one will become more confident in the location of all features. In particular, the *relative* position of *all* feature will in the limit of infinite re-observations become completely known. That is, the covariance of the relative position between all features will tend to zero. However, the *global* covariance of the features will tend to the covariance of the feature that is known most accurately globally, which will be the first feature observed, unless the vehicle's position gets reset at some time, say through a GPS reset. That is, since the vehicles position was not precisely known during the initiation of the first feature, this global uncertainty in all feature positions (the map) can never be recovered.

As a result of the map converging to a perfect *local* map, obtaining perfect *global* knowledge of a feature's location will result in that *all* features' locations (the map) will become known perfectly *globally*. More simply, you can create a perfect local map of a region, but if you do not know whether the region you mapped is in Africa or in Europe, you will never be able to reduce this initial global uncertainty without some additional external information. However, if accurate global information was provided stating that one of the streets you mapped is a particular street in Paris, you will know accurately where all the features in your map are *globally*. Dissanayake *et al.* have provided a more mathematical description of these properties in [36].

These properties are dependent on that the underlying assumptions of the Kalman filter hold. Due to the errors introduced by the linearization performed in stochastic

mapping, it is difficult to prove that the location of all the features in the map actually converge to the *true* location of features.

2.4 Augmented stochastic mapping

The previous section outlined the stochastic mapping approach to feature based navigation. In order to employ stochastic mapping in a real world scenario, we have to be able to extract features from the environment. As noted in Section 2.2, sonars are notorious for exhibiting drop-outs, false returns, no-returns and noise. Thus, addressing the problem of *data association* is critical for the validity of the observation model (Equation (2.1)) and in employing a stochastic mapping based approach to CML. For this purpose, data association strategies are explored next in the context of a mobile robot using sonar as a means for observing features in the environment. Combining these strategies with stochastic mapping culminates in the augmented stochastic mapping (ASM) algorithm.

2.4.1 Track initiation, association, and deletion

Due to the presence of clutter and dropouts, data association, track initiation and track deletion is performed. The objective of data association is to assign measurements to the features in the environment from which they originate, while rejecting spurious measurements. In this thesis, measurement to feature association is performed using a gated nearest neighbor approach in innovation space [4]. This approach incorporates sensor uncertainty as well as the uncertainty in the vehicle's state. In order to include the vehicle's uncertainty in performing data association using a nearest neighbor gating technique, we need to transform the vehicle's uncertainty into measurement space and add this uncertainty to the measurement uncertainty. We assume that the true measurement of feature i at time k conditioned upon all measurements up to time step $k-1$ is normally distributed in measurement space.

Further, it is assumed that the transformation from vehicle space to measurement space retains the Gaussianity of the estimated state. Under these assumptions, one may define the innovation matrix $\mathbf{S}_i$ for feature i as

$$\mathbf{S}_i = \mathbf{H}_{\mathbf{x}_i} \begin{bmatrix} \mathbf{P}_{rr} & \mathbf{P}_{ri} \\ \mathbf{P}_{ir} & \mathbf{P}_{ii} \end{bmatrix}_{k|k} \mathbf{H}_{\mathbf{x}_i}^T + \mathbf{R}, \quad \text{with } \mathbf{H}_{\mathbf{x}_i} = \left. \frac{d\mathbf{h}_i([\mathbf{X}_r\ \mathbf{X}_i])}{d[\mathbf{X}_r\ \mathbf{X}_i]} \right|_{[\mathbf{X}_r\ \mathbf{X}_i]=[\mathbf{x}_{r_{k|k-1}}\ \mathbf{x}_{i_{k|k-1}}]}, \tag{2.8}$$

where $\mathbf{h}_i$ is the observation model for feature i. $\mathbf{H}_{\mathbf{x}_i}$ is the linearized transformation from vehicle space to measurement space. The nearest neighbor gating is performed in innovation space. That is, defining the innovation $\boldsymbol{\nu}_i = \mathbf{z}_i - \mathbf{h}_i(\mathbf{X})$, the validation region, or gate, is given by

$$\boldsymbol{\nu}_i^T \mathbf{S}_i^{-1} \boldsymbol{\nu}_i \leq \gamma. \tag{2.9}$$

The value of the parameter γ is obtained from the χ^2 distribution. For a system with 2 degrees of freedom, a value of $\gamma = 9.0$ yields the region of minimum volume that contains the measurement with a probability of 98.9% [4]. This validation procedure defines where a measurement is expected to be found. If a measurement is outside this region, it is considered too unlikely to arise from feature i. If several measurements gate with the same feature i, the closest (i.e. most probable) one is chosen. An analysis of the performance of gated nearest neighbor data association is given in Appendix 2.B.

The initiation of new feature tracks is performed using a delayed nearest neighbor (DNN) initiator. The DNN initiator is similar in spirit to the logic-based multiple target track initiator described by Bar-Shalom and Fortmann [4]. One important difference in our method is that the vehicle's position is uncertain, and this uncertainty has to be included when performing gating and finding the nearest neighbor. It is assumed that a sonar return originates from not more than one feature. In performing

track initiation, all measurements that have not been matched with any feature over the last N time steps are stored. That is, any measurement that was not matched to a known feature is a potential new feature. At each time step a search for clusters of more than $M \leq N$ measurements over this set of unmatched measurements is performed. For each of these clusters, a new feature track is initiated. A cluster is defined as at most one measurement at each time step that gates according to Equation (2.9) with all other measurements in the cluster. For our systems, where the probability of false returns is relatively low (λ of Section 2.2.1 being less than 10) and the probability of detection is relatively high (say 80%-90%), values of $M = 3$ or 4 and $N = M + 1$ are sufficient.

A track deletion capability is also incorporated to provide a limited capability to operate in dynamic environments. When a map feature is predicted to be visible but is not observed for several time steps in a row, it is removed from the map [67]. This is motivated by assuming a probability of detection $P_D < 1$. Thus, if the feature has not been observed over the last r time steps during which an observation was expected of the feature, the probability of the feature being at the expected location is $(1 - P_D)^r$, assuming that the observations are independent. Thus, setting a threshold on r is equivalent to setting a threshold on the probability that the feature still exists at the predicted location. This track deleter is only implemented when operating in changing environments and should also be utilized in areas of high clutter.

We also utilized a simple track deletion strategy that checks for consistency of the estimated features. That is, if the 95% error bounds for a feature estimate overlap by more than a certain percentage, typically 30%, with another feature's 95% error bounds, the feature with the smallest error bound is deleted. This causes the system to remove tracks for features that are physically too close to each other for reliable data association. Track deletion must be performed with caution as good tracks may be erroneously discarded.

We will refer to the implementation of the stochastic mapping algorithm with

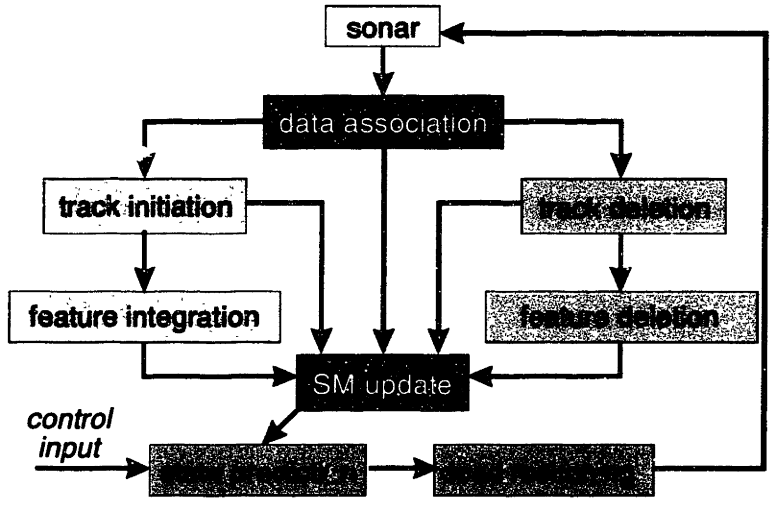

Figure 2-4: Structure of the augmented stochastic mapping algorithm.

the addition of logic-based track initiation and deletion and nearest-neighbor data association as augmented stochastic mapping (ASM). Figure 2-4 shows a flow-chart representation of the structure of the augmented stochastic mapping algorithm.

In the next section we will conduct a performance analysis of ASM in simulation. In Section 2.6 we will show results from an experiment in a testing tank using ASM. As a last performance evaluation, we will show results from the post-processing of sonar data taken from the Narragansett Bay area, Rhode Island, in Section 2.7.

2.5 Performance analysis: Simulations

To evaluate the long-term performance of ASM, a variety of AUV missions have been simulated. Two different scenarios are reported in this section. In the first scenario, the vehicle is required to repeatedly perform a "lawnmower" survey of a region with point features randomly distributed over a 1.2 km by 1.2 km area. In the second scenario, the vehicle is required to travel back-and-forth between two clusters of point features that are approximately 15 km apart. The parameters used for these simulations are summarized in Table 2.1. In each scenario, false measurements are generated by assuming that the number of spurious returns has a Poisson distribution with an expected value of λ. The range and angle of the spurious returns are uniformly distributed over the field of view of the sonar. The probability of detection of features is set to $P_D = 0.9$. Sonar measurements are obtained at 1 Hz. The simulations have been implemented using Matlab. The execution speed is faster than real time running on a Pentium 233MHz processor.

Table 2.1: Augmented SM simulation parameters.

sampling period, T	1 sec.
maximum sonar range	300 m
sonar coverage angle	$\pm 40°$
range measurement standard deviation	0.5 m
bearing measurement standard deviation	5°
feature probability of detection	0.90
vehicle cruise speed, V_c	2.5 m/s
speed process standard deviation	5% of δv
heading process standard deviation	2.0°
dead reckoning speed standard deviation	0.4 m/s
dead reckoning heading standard deviation	3.0°
initial position uncertainty std. dev.	1.5 m
initial heading uncertainty std. dev.	5.0°
initial speed uncertainty std. dev.	0.2 m/s
gate parameter γ	9
clutter parameter λ	2
track initiation parameters	$M = 5, N = 4$

2.5.1 Survey in a current

In this simulation, the goal was for the AUV to perform a survey of an unknown area of approximately 1.2 km by 1.2 km, in the presence of clutter, dropouts and an unknown current of 7 cm/s in the north-east direction utilizing augmented stochastic mapping (ASM). The area contained 50 features randomly distributed over the survey area. The desired path of the vehicle along with the true position of the features are shown in the top plot of Figure 2-5. The AUV starts at position $(0, 0)$ and follows a grid survey until it reaches $(0, 1000)$, when the AUV returns to its initial position and repeats the survey. This process is performed approximately 5 times, and the total mission duration is approximately 5.5 hours. In order to compensate for the presence of the current, a process noise was modeled for the vehicle's position with a standard deviation of 0.85 meters.

The bottom plot of Figure 2-5 shows the actual path of the vehicle, the estimated positions of the features, and all the sonar returns obtained during the mission. Figure 2-6 shows plots of the position, heading, and velocity errors of the vehicle versus time, along with 3σ (99% highest confidence region) bounds. As can be seen from these figures, all estimates remain bounded. The vehicle's mean absolute position error is 2.5 meters, the mean absolute heading error is 1.4° and the mean absolute speed error is 2 cm/s. The mean standard deviation in position is 4.3 meters, the mean heading standard deviation is 1.8°, and the mean speed standard deviation is 2.5 cm/s.

The global map of the features is also accurate. Figure 2-7 show a histogram of the distribution of the error in estimated feature locations. From this figure, we see that the average error in the estimated feature location is less than 1.5 meters, and the maximum error of any feature estimate is only 2.25 meters. Figure 2-8 shows the error of the estimated position for all 50 features in the map along with the 3σ error bounds. This plot suggests that the error bounds are overly conservative.

Figure 2-9 compares the actual path for an AUV performing the mission using

dead-reckoning alone with that of one using stochastic mapping. As can be seen from this figure, the estimated vehicle position is off by more than 1.4 km when using dead-reckoning.

Figure 2-10 compares the absolute position errors for dead-reckoning and augmented SM. As expected, the dead-reckoning error grows linearly with time, while the CML error estimate remains bounded.

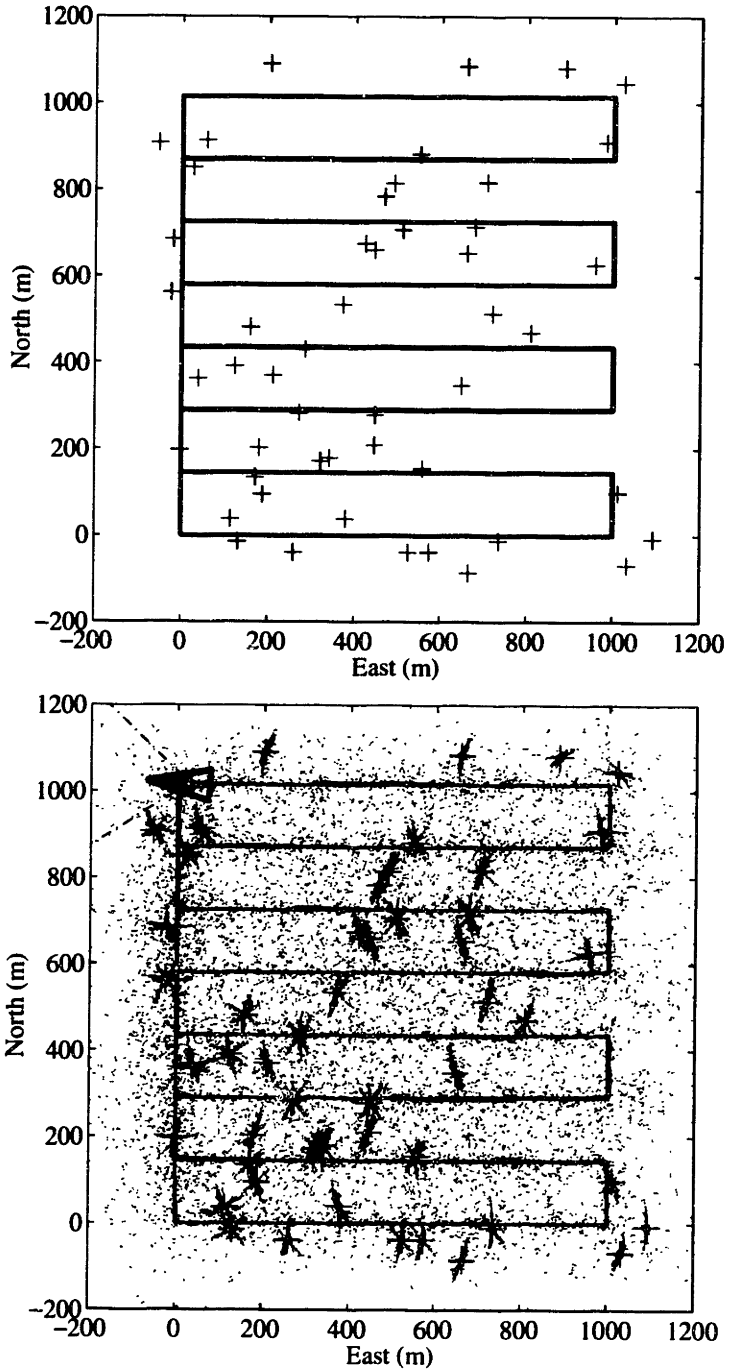

Figure 2-5: **Top:** The desired survey path of the vehicle (solid line) and the locations of the 50 randomly distributed point features (plus signs). **Bottom:** The actual and estimated paths of the vehicle (solid lines), the estimated feature locations (plus signs), and error ellipses for the feature location estimates (3σ confidence bounds). Small dots indicate all the sonar measurements that were obtained during the entire 5.5 hour mission. The estimated and actual final positions of the AUV are marked as triangles in the upper left portion of the figure.

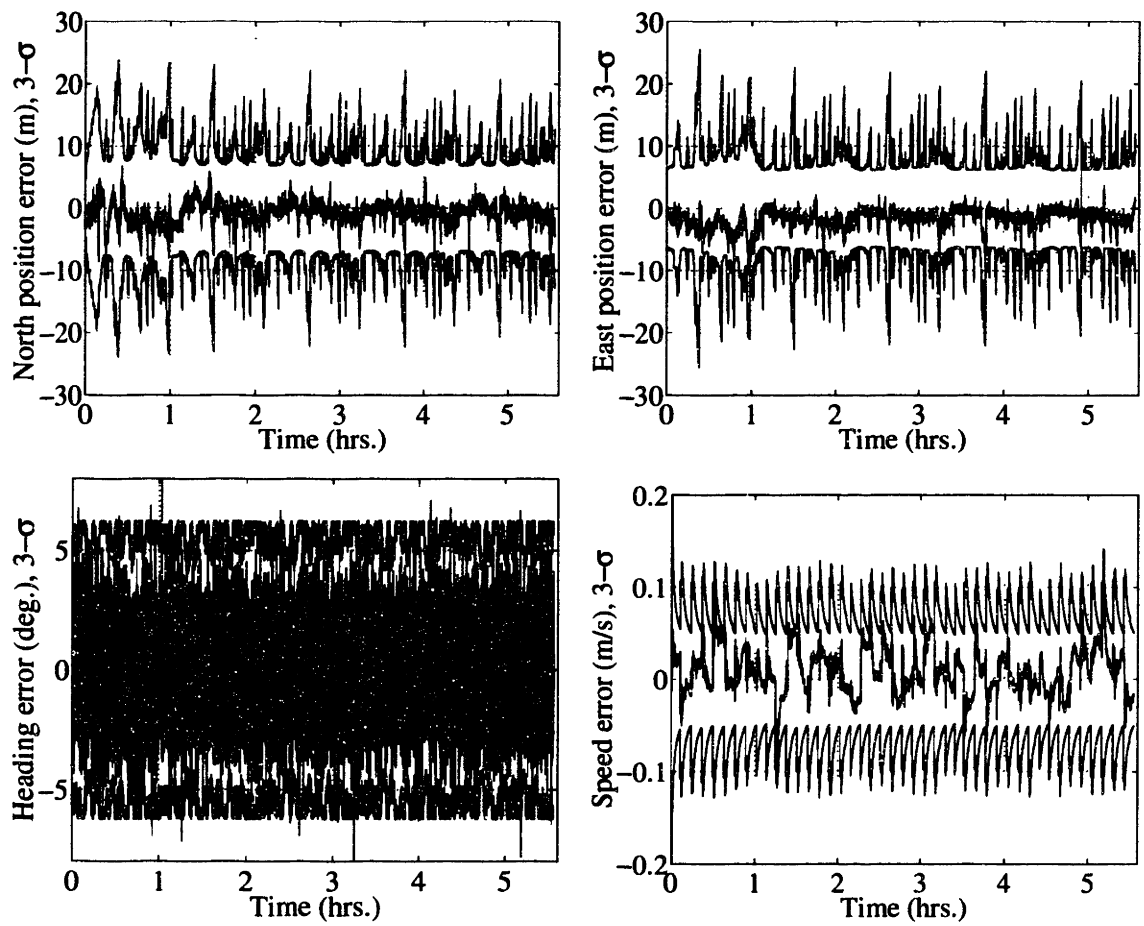

Figure 2-6: Errors and 3σ bounds (99% highest confidence bounds) for the position, heading and velocity estimates produced by the stochastic mapping algorithm.

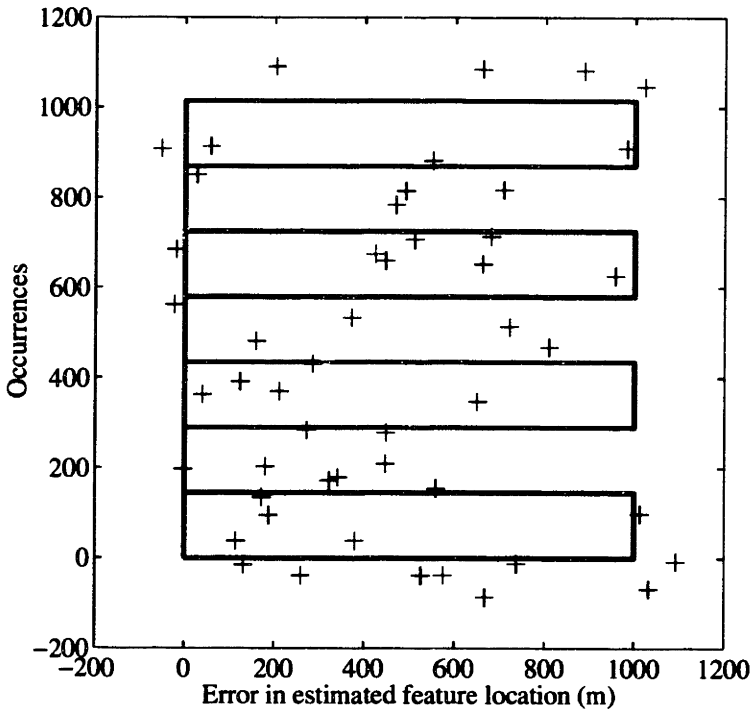

Figure 2-7: Histogram of errors in the location estimates for environmental features.

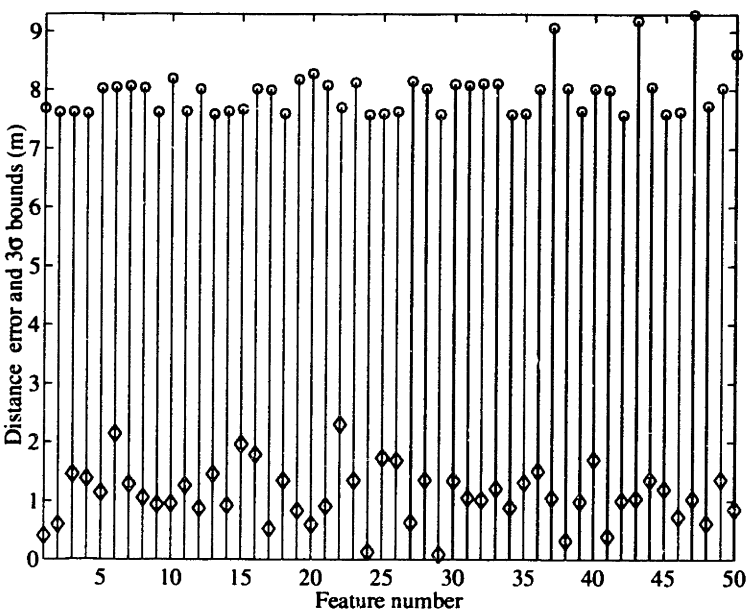

Figure 2-8: Distance error in the estimated feature locations for all 50 features (diamonds) and associated estimated 3σ error bounds calculated by the algorithm (circles).

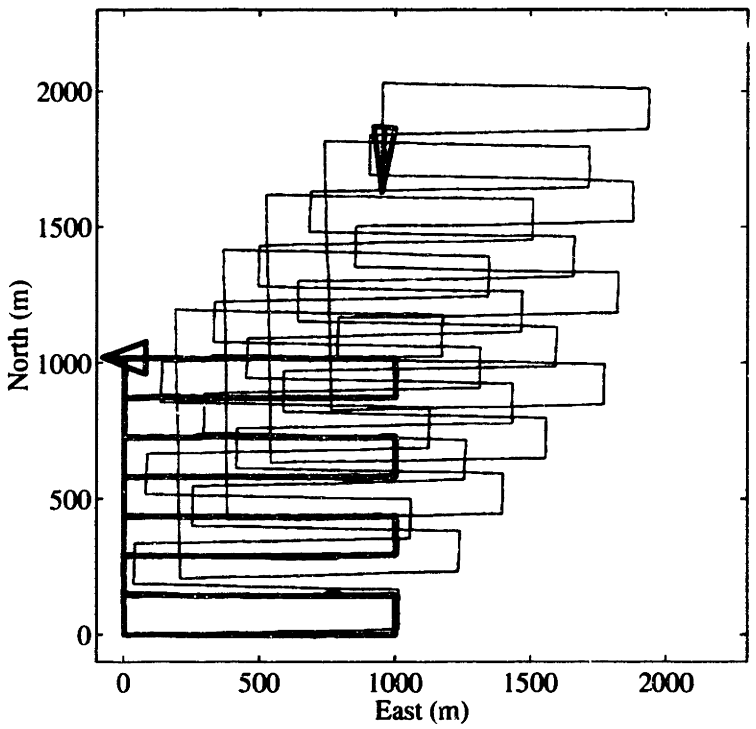

Figure 2-9: Comparison of dead-reckoning and stochastic mapping. In this figure, the actual path of the vehicle using only dead-reckoning (thin solid line) is superimposed on the vehicle's path when using ASM (thick solid line). At the end of the mission, the dead-reckoning estimate is more than 1400 meters off, while the error in the stochastic mapping estimate is less than 5 meters.

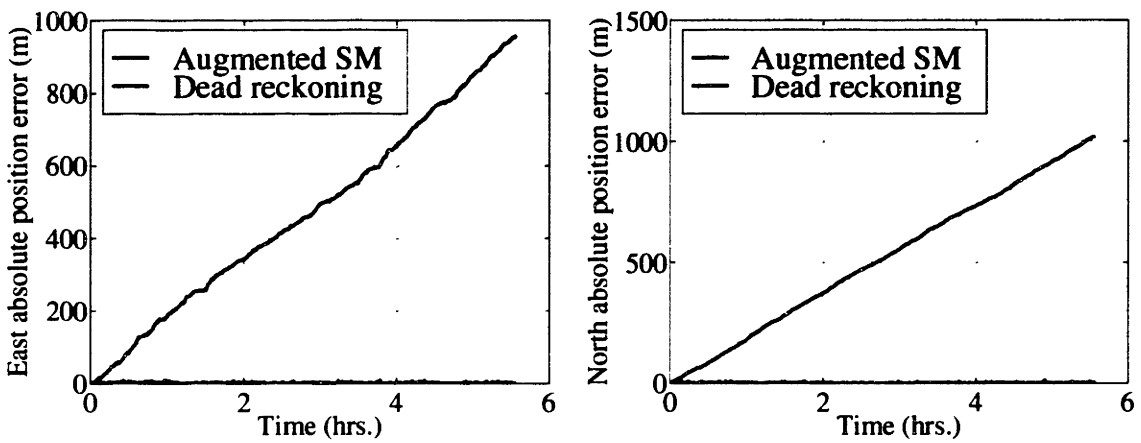

Figure 2-10: Absolute position error in north and east directions for dead-reckoning and stochastic mapping.

2.5.2 Back-and-forth transit mission

In this section, we examine the performance of the algorithm when there are few features in the environment and large distances between groups of features. The parameters used in this experiment are listed in Table 2.1. Unlike the previous section, no current is present in this scenario, and thus no vehicle position process noise is modeled. In these experiments, the vehicle is set to survey along a line about 15 km long, with only 5 features at each end. A plot of the final position of the vehicle, its path, all sonar returns and the actual feature locations is shown in Figure 2-11. In Figure 2-12, the position error for this run and the 3σ bounds are plotted. As expected, as soon as the vehicle leaves the area with features, the error bounds grow at the rate of dead-reckoning, and the actual error grows as well. However, when the vehicle reaches the new unmapped area in the upper right of Figure 2-11 with the second grouping of features, the AUV uses the information obtained from these features to localize itself more accurately. This occurs approximately 1.5 hours into the run. When the first grouping of features is re-observed about 3 hours into the run, the vehicle is able to re-localize to less than 5 meters accuracy with 99% confidence. As the vehicle moves back-and-forth between the group of features, more accurate localization and mapping is achieved.

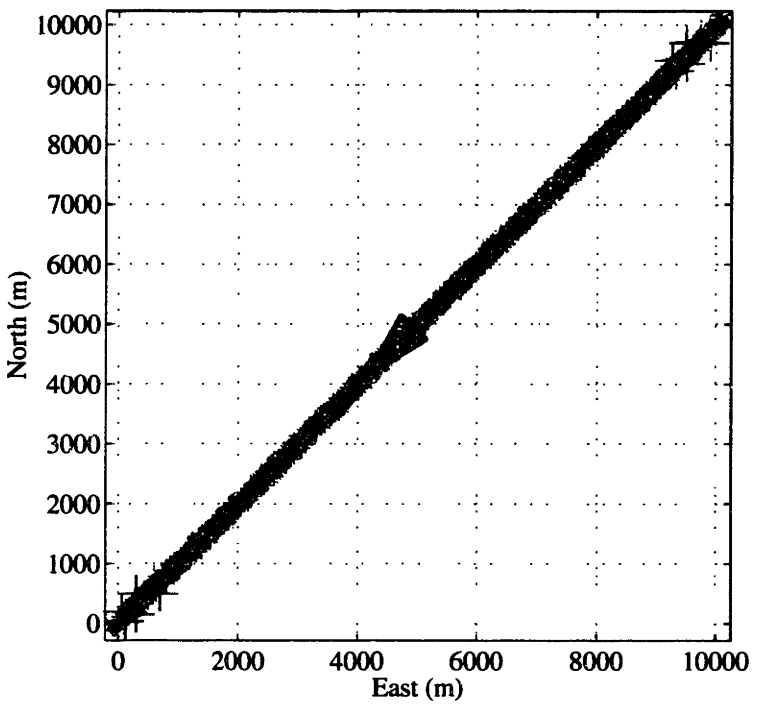

Figure 2-11: Path of the vehicle (solid line), locations of the 10 environmental features (crosses), and all the sonar returns (dots) obtained during the 5.5 hour back-and-forth transit mission.

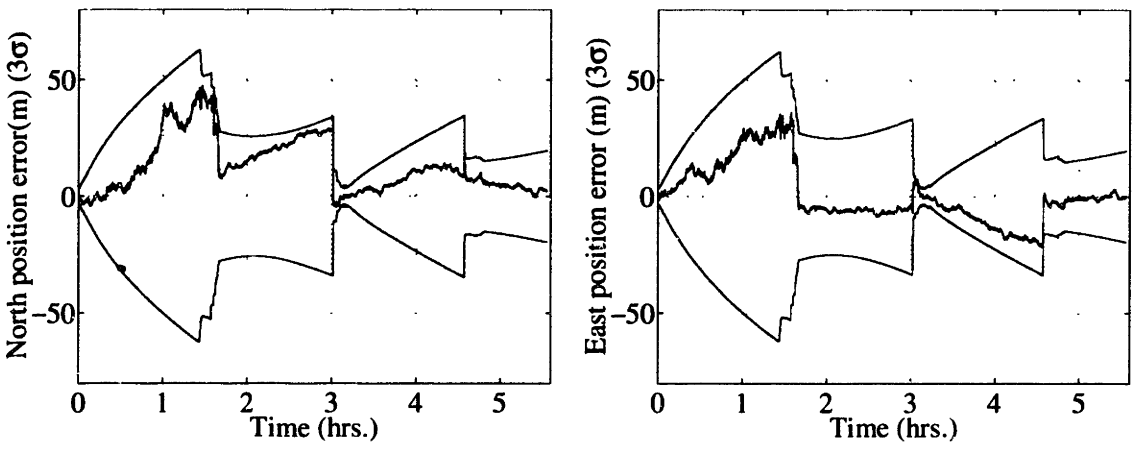

Figure 2-12: Position error of the AUV during the back-and-forth transit mission and the 3σ error bounds.

2.6 Performance analysis: Experiment

Simulation studies provide the flexibility of considering any event and scenario, however, the uncertainties and deviations between model and the real world are difficult to replicate precisely. Thus, performing controlled laboratory experiments often gives valuable insight to a method and its validity. In this section, the in-water experiment of a modeled AUV conducting a "lawnmower" survey performing ASM in a 9.4 by 3.5 by 1.5 meter testing tank is presented. Section 2.6.1 describes the setup of the experiment followed by the results in Section 2.6.2. For comparison purposes, the same mission was also performed in simulation.

2.6.1 Setup

A planar robotic positioning system mounted on top of the tank was used to position a sonar mounted on a vertical shaft submerged in the water. The system has three degrees of freedom, x, y and shaft rotation. The robotic system is equipped with encoders for accurate position reading and commanding. The positioning system was controlled by a Compumotor AT6450 controller card. A C++ interface was developed for commanding the system.

Twenty eight fishing bobbers were placed on the right side of the tank to act as environmental features and submerged to approximately the same plane as the sonar. The 1.25 cm radius of the Styrofoam bobbers was known to the ASM algorithm.

The sonar used in this experiment was a Panametrics V318-SU 500 kHz transducer with a $\pm 4.7°$ beam width. The sonar signal (waveform) was captured using GageScope 1012 analog to digital converter boards on a PC sampling at 2 MHz. The digital signal was passed into a Matlab script which performed match filtering of the signal to detect the returning pulse. Once the return pulse was identified, the TOF and the range to the target was computed. For each range measurement, five waveforms were captured with the sonar pointing in the same direction before match

filtering was performed. This was done as the electrical and acoustic noise levels were high in our laboratory. The method of RCD extraction [69] was used to determine the range and bearing to each feature. The transducer was fixed to the rotating shaft of the positioning system. The shaft was rotated 0.6 degrees, on average, between each TOF measurement. A 360 degree scan of the tank using this sonar is shown in Figure 2-13. The tank walls were removed from the RCD extraction.

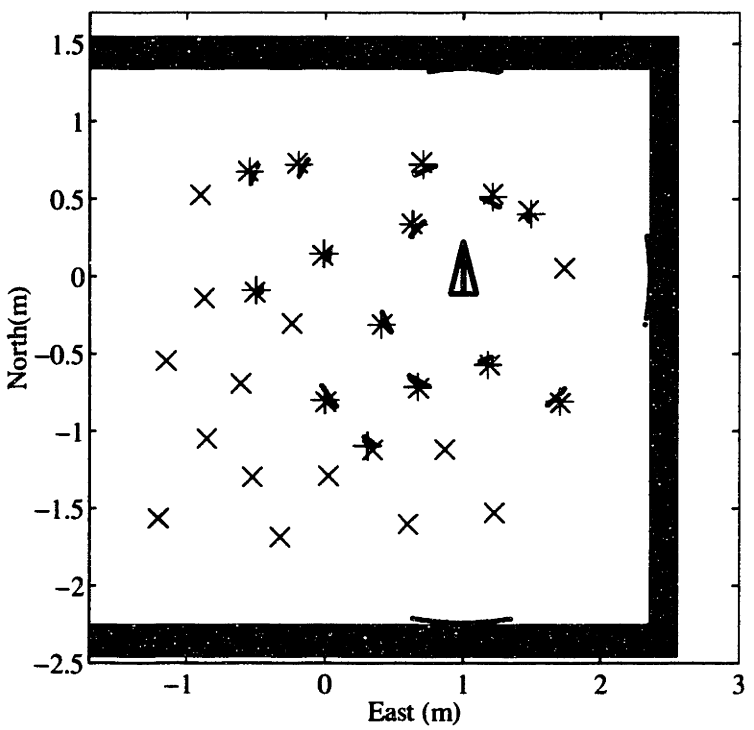

Figure 2-13: A 360° scan of the tank with the Panametrics sonar. The location of the sonar is marked by a triangle, individual sonar returns are marked by a dot, the true location of a feature (bobber) is marked by '×', and the result of the RCD extractor is marked by +. The tank walls are shaded.

The system operated on two computers, one dedicated to obtaining sonar measurements and the other for operating the robotic positioning system and performing ASM. The sonar measurements were passed to the C++ program controlling the positioning system for combining them with the current state of the positioning system. Once a scan was obtained, it was passed to a Matlab script executing ASM and ex-

tracting RCDs. As a time step was completed, the new (true) position of the vehicle was passed back to the positioning system for execution of the AUV's dynamics and the initiation of a new scan. The integration of these programs resulted in a closed loop system for performing CML in the testing tank.

The experiment was designed to simulate an AUV, however, due to the limited size of the testing tank, the AUV's parameters were scaled down by a factor of 100. However, not all parameters could be scaled in such a way as to replicate the AUV of Section 2.5. Thus, a simulation using the parameters of the experiment was conducted for comparison with the experiment.

2.6.2 Experimental results

The parameters used in the experiment and simulation are shown in Table 2.2. Scaling everything by 100 would yield the parameters of the AUV we were modeling. The main difference between these parameters and those of Table 2.1, is the nominal speed of the vehicle, $V_c = 0.1$ m/s (i.e., 10 m/s for the vehicle modeled), the range uncertainty in the sonar, and the survey path. Further, due to the time required to make a scan with the mechanical sonar we employed, the time period T is during the experiment longer than the 1 sec. that is assumed for the vehicle we are modeling.

Figure 2-14 shows the survey path of the vehicle along with the position of all the bobbers. The initial position of the sonar is marked by an arrow. The end of the path is marked by a square. Once one pass through the path was completed, another one was initiated. This process was repeated until the termination of the experiment.

Figure 2-15 displays the actual and estimated path of the AUV along with the estimated feature locations and their 3σ bonds (99% highest confidence regions) for the experiment (left) and in simulation (right). The overshooting of the corners relative to the desired path of Figure 2-14 is due to the faster vehicle speed compared to that of Section 2.5, causing the AUV to overshoot its mark. As can be seen from these plots, the resulting path and feature estimates are similar for the experiment

Table 2.2: Augmented SM experiment parameters.

sampling period, T	1 sec.
maximum sonar range	200 cm
sonar coverage angle	$\pm 40°$
range measurement standard deviation	2 cm
bearing measurement standard deviation	5°
feature probability of detection, P_D	0.90
vehicle cruise speed, V_c	10 cm/T
speed process standard deviation	5% of δv
heading process standard deviation	2.0°
dead reckoning speed standard deviation	4.5 cm/T
dead reckoning heading standard deviation	3.0°
initial position uncertainty std. dev.	1.0 cm
initial heading uncertainty std. dev.	5.0°
initial speed uncertainty std. dev.	0.1 cm/T
gate parameter γ	9
track initiation parameters	$M = 3, N = 2$

and the simulation. A couple of extra features were mapped during the experiment. These were caused by unevenness in the tank bottom detected by the sonar and mapped.

Figure 2-16 and Figure 2-17 shows the position, heading and speed error along with the 3σ error bounds (99% highest confidence region) for the experiment and simulation. Comparing these plots, we notice an excellent agreement between experiment and simulation. During the experiment, it took a slightly longer time before a feature was initiated, thus causing a slightly higher error bound initially than in the simulation. This might have been caused by a lower probability of detection in the experiment than the $P_D = 0.9$ modeled in the simulation. As a consequence, the simulated results are a little better than the experimental results. The mean distance error for the experiment was 1.3 cm with a standard deviation of 2.2 cm. For the simulation, the numbers were 1.2 cm and 2.0 cm, respectively. As can be observed from comparing Figure 2-16 and Figure 2-17, it was during the first 500 seconds, which is equivalent to one pass through the mission path, the simulations did a little

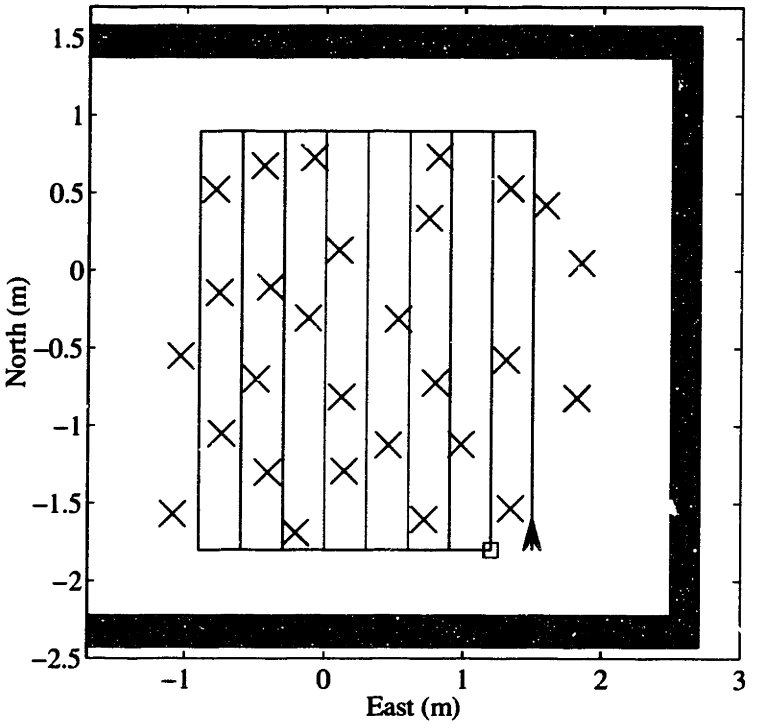

Figure 2-14: The desired path of the AUV (solid line), along with the true feature positions, marked by crosses. The initial position and direction of the AUV is marked by an arrow. The end of the path before it is repeated is marked by a square.

better, while the steady state after 2000 seconds is very similar for the two plots.

Turning to Figure 2-18 we see that not only was accurate localization of the vehicle achieved, but also the feature location estimates were accurate. The mean feature distance error was 1 cm for the experiment and 0.7 cm for the simulation. The mean distance standard deviation was 1.7 cm for both the experiment and the simulation.

Figure 2-19 shows the distance error of the first feature initiated for the experiment (left) and the simulation (right) through time. As expected, the error bound is a monotonically decreasing function of time.

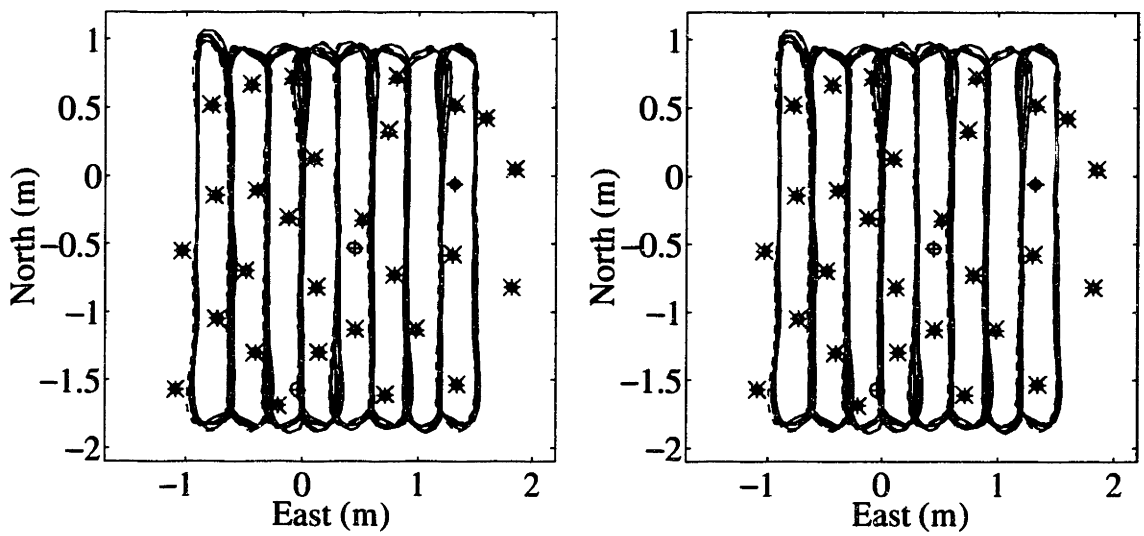

Figure 2-15: The estimated path (solid line), the actual path (dashed line), along with the true feature location, '×', the estimated feature location, '+', and the 3σ error ellipse for the feature estimates. **Left:** Experimental result. **Right:** Simulated result.

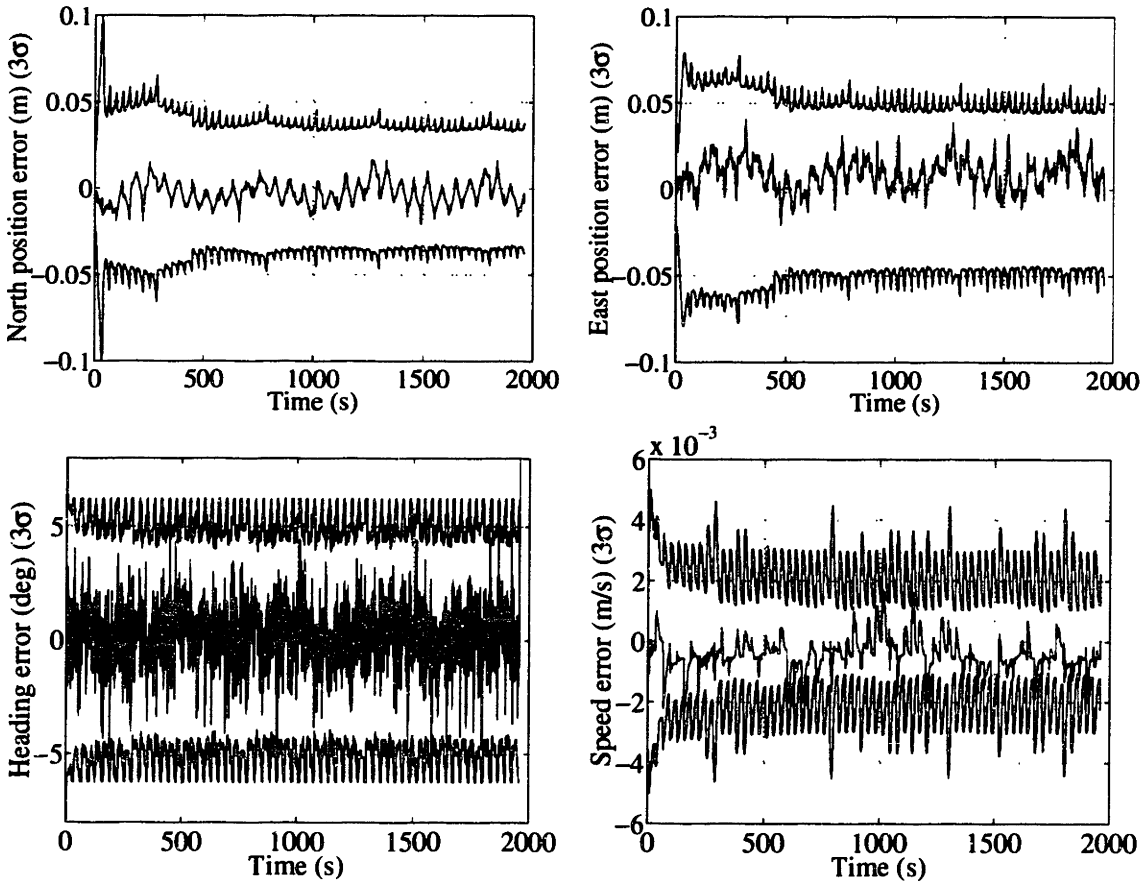

Figure 2-16: Errors and 3σ bounds (99% highest confidence bounds) for the position, heading and velocity estimates produced by the ASM *experiment* in the testing tank.

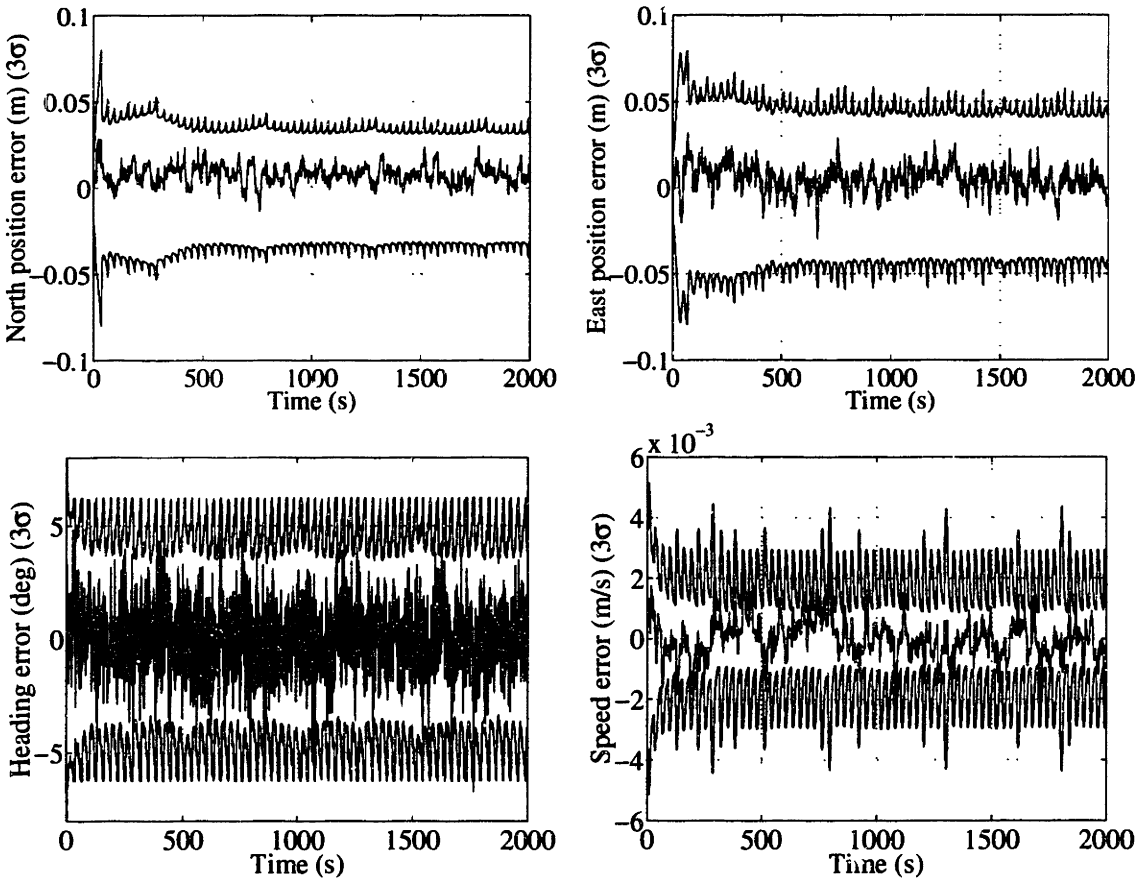

Figure 2-17: Errors and 3σ bounds (99% highest confidence bounds) for the position, heading and velocity estimates produced by *simulation* of ASM in the testing tank.

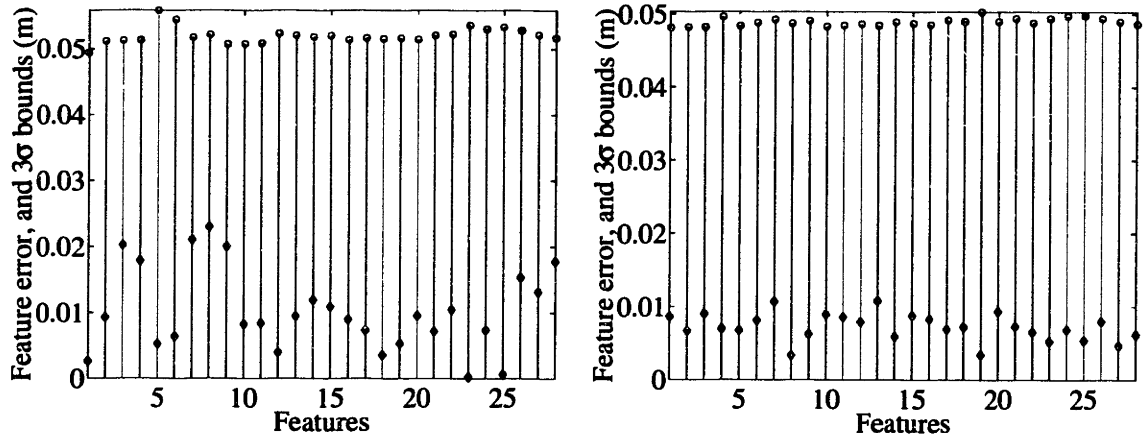

Figure 2-18: Estimated feature error and 3σ bounds. **Left:** Experimental result. **Right:** Simulation result.

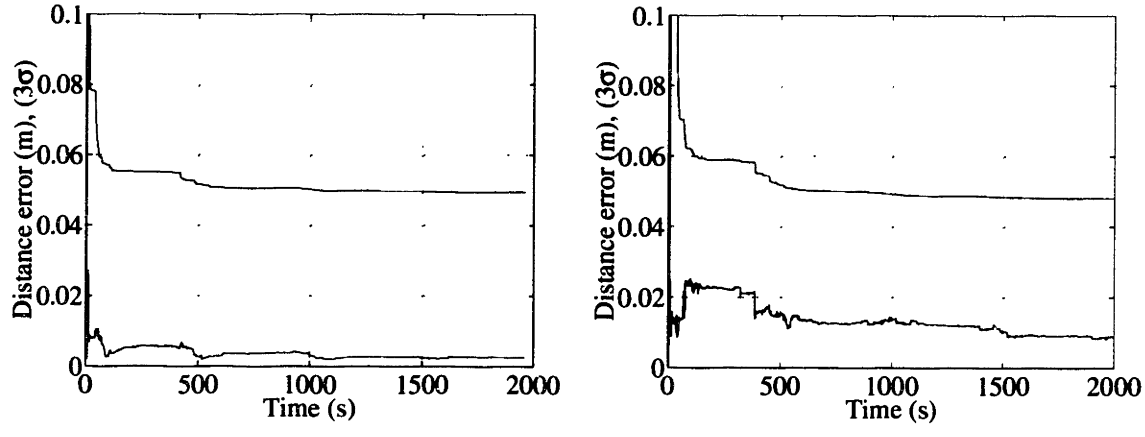

Figure 2-19: Evolution of feature estimation error and 3σ bounds versus time for the first feature initialized (i.e., not the same physical feature). **Left:** Experimental result. **Right:** Simulation result.

2.7 Post-processing of oceanic data

The viability of ASM for performing concurrent mapping and localization was shown by simulations and tank experiments in the previous sections. Although the results of the previous sections are convincing, a real AUV mission performing ASM would be the most convincing result. We have not had access to an AUV, but to the next best alternative, that is, data collected from an ocean vessel. This data can be post-processed in order to verify the validity of the algorithm. The data available to us was taken using an electronically scanned, high resolution array (HRA) forward look sonar (FLS) mounted on the side of a surface ship equipped with an INS system as well as differential GPS performing a survey in Narragansett Bay, Rhode Island, for the explicit purpose of assessing the navigation capabilities of AUVs. This data was provided to us by Mr. Robert N. Carpenter of the Naval Undersea Warfare Center (NUWC) [24].

As the dynamics and control input to the ship were not provided to us, a slightly revised dynamic model and stochastic mapping algorithm was used and is described next, followed by the results, in Section 2.7.2.

2.7.1 Stochastic mapping

The modeling assumption used in the post-processing of the data from NUWC is outline in Section 2.2. Point features were extracted from the sonar returns based on signal to noise ratios by NUWC as described in [25]. As no information was given about the dynamic model for the vessel collecting data using the FLS, the following

simple dynamic model was used

$$\mathbf{X}_{k+1} = \mathbf{f}(\mathbf{X}_k, \mathbf{u}_k) + \mathbf{d}_\mathbf{x} = \begin{bmatrix} x_k + \delta v_k T \cos(\phi_k) \\ y_k + \delta v_k T \sin(\phi_k) \\ \phi_k + \delta\phi_k \\ v_k + \delta v_k \end{bmatrix} + \mathbf{d}_\mathbf{x}, \tag{2.10}$$

where $\mathbf{d}_\mathbf{x}$ is a white Gaussian noise process. The control input at time k, $\mathbf{u}_k = [\delta\phi \ \ \delta v]_k^T$, is represented by a change in heading, $\delta\phi$, and a change in speed, δv, and was backed out from the change in heading and velocity that had to occur in order for a vehicle with the dynamic model $\mathbf{f}$ of Equation (2.10) to get from the integrated INS position estimate at time $k-1$ to the integrated INS estimated position at time step k. The INS position estimates were used to model the unrecorded control input to the vessel, rather than the more natural choice of using the INS readings as observations in a dead reckoning model. This was done as no vessel dynamic model or control input was recorded. In order to perform stochastic mapping on-board in real-time, a dynamic model should be developed, the control input would be available and the INS measurements would be used directly in a dead reckoning model. Such a system would be more accurate and robust than the results obtained when post-processing without the vehicle model nor control input available. We thus believe our result to be a lower limit on the performance of the system.

Figure 2-20 shows a flow-chart representation of the structure of the stochastic mapping algorithm utilized to perform concurrent mapping and localization given the data from the on-board INS and FLS.

2.7.2 Post-processing results

The parameters used in the augmented stochastic mapping algorithm are shown in Table 2.3.

The data set was collected from Narragansett Bay in Rhode Island in an area

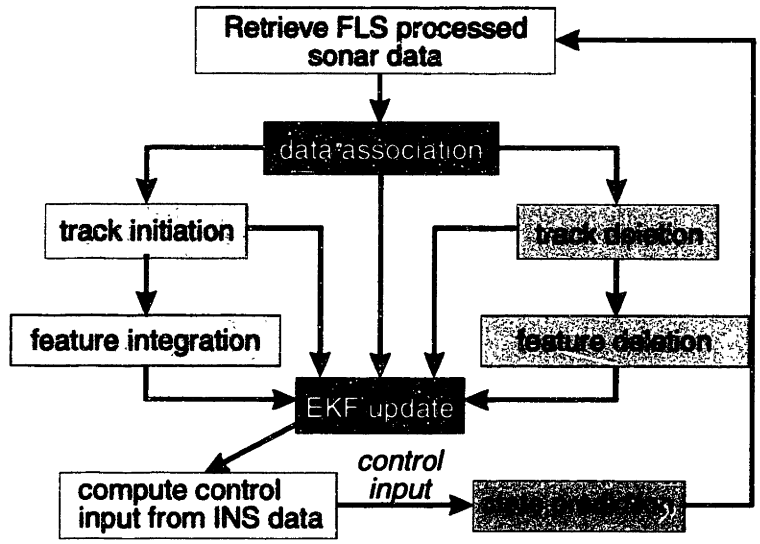

Figure 2-20: Structure of stochastic mapping algorithm utilized in the post-processing of the INS and FLS data in order to perform concurrent mapping and localization.

Table 2.3: Parameters used for ASM post-processing.

sampling period, T	20 sec.
range measurement standard deviation	1 m
bearing measurement standard deviation	0.6°
speed process standard deviation	4 cm/T
combined heading and INS process std. dev.	0.9°
INS speed measurements std. dev.	4.4 cm/T
gate parameter γ	10
track initiation parameters	$M = 5, N = 4$

known as Halfway Rock. Figure 2-21 depicts the Narragansett Bay area around NUWC. An enlarged view of the Halfway Rock area is shown on the right in Figure 2-21. The steep slope along the east of the Halfway-Fiske shoal gives way to a shipping channel. The bathymetry in the test area varies from 15 to 0 meters deep (Halfway Rock broaches the surface). Also, since the area is well inside Narragansett Bay it is shielded from much of the ocean swell, thus allowing the ship's dynamics to be similar to that of an AUV.

Figure 2-22 shows all the returns obtained from the HRA sonar during the data

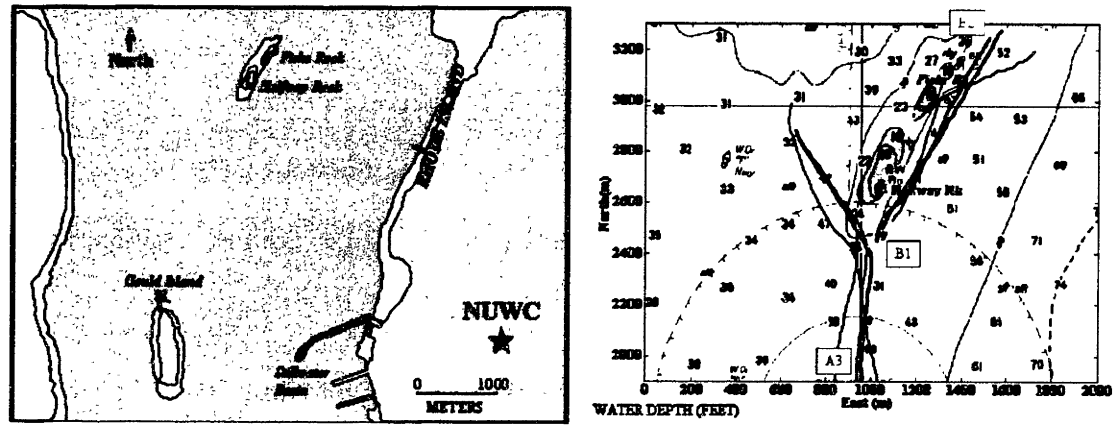

Figure 2-21: Narragansett Bay area around NUWC (left), and the run area (right). Courtesy of Carpenter [24]

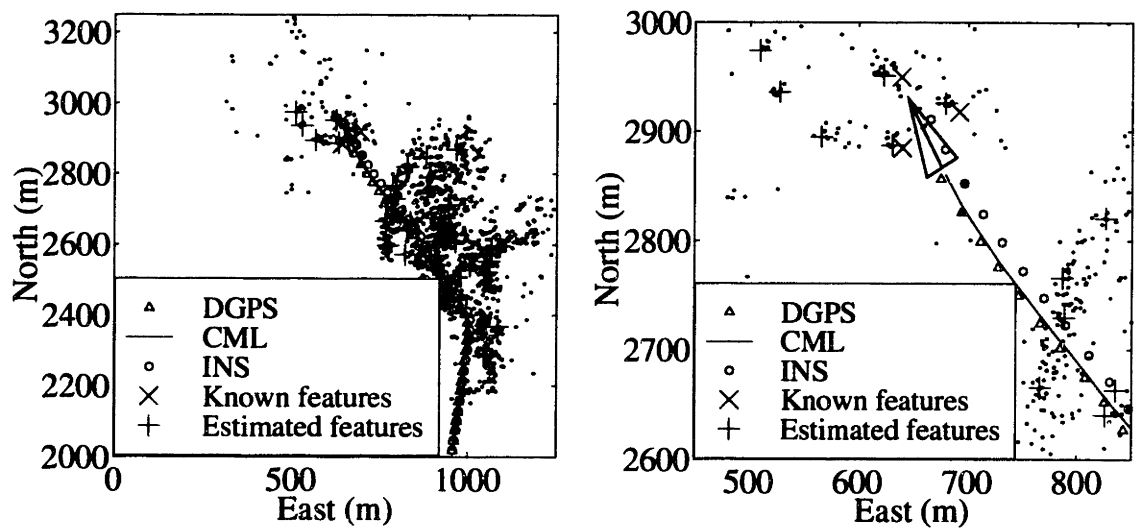

Figure 2-22: Result of post-processing of augmented stochastic mapping performed from the data collected at Narragansett Bay leg A3 marked on the right figure of Figure 2-21. Returns from the HRA sonar are shown by small dots, the DGPS position fixes at each measurement point are shown by triangles, the CML result is drawn as a solid line, the INS result is shown by small circles, the three known features are represented by '×' signs, and while estimated feature locations are shown by '+' signs. **Left:** The result for the entire mission. **Right:** A magnification of the last 12 measurement points and estimates for the mission.

collection at Narragansett Bay of leg marked A3 in Figure 2-21. There are three known features, marked by '×', located near the final position of the vehicle in addition to environmental features. Turning to the left plot of Figure 2-22 we see that the sonar returns are disperse and it is hard to pick out single objects by eye. Further, many environmental features have been initiated. Turning to the right of Figure 2-22 we see that quite good estimates of the know features were obtained. Also, the estimated position of the vehicle performing CML is closer than that of INS to the "true" DGPS position. Figure 2-23 displays the vehicle's position as estimated from DGPS, CML and INS, along with the 3σ error ellipses for all the modeled environmental features as well as two if the known features. The error ellipses are, however quite large, and thus provide relatively poor positioning information. The reason for the relatively large uncertainty in feature estimates is due to the very low re-observation rate of features that occurred during this mission, as can be seen from Figure 2-24.

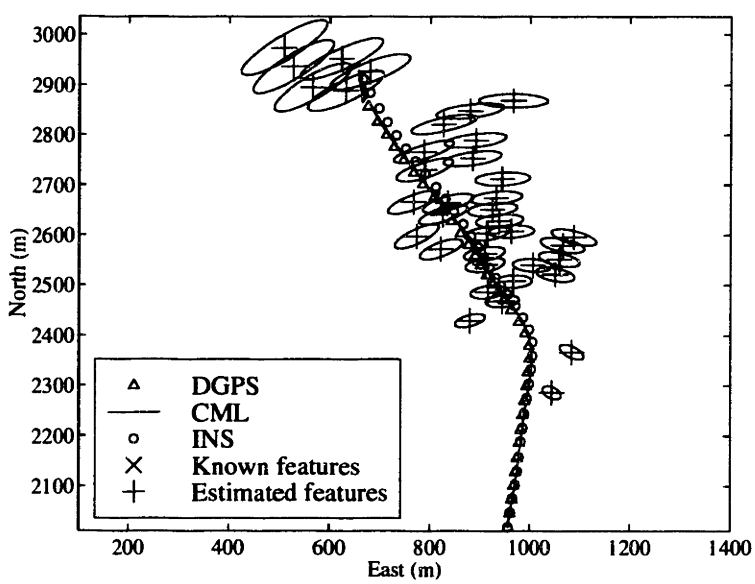

Figure 2-23: The estimated features locations, marked by '+', along with the 3σ (99% highest confidence region) ellipses for each of these estimates for post-processing of the data from leg A3, shown in Figure 2-22.

The superiority of ASM over dead-reckoning is illustrated by Figure 2-25, where the INS error grows linearly without bound, and the CML (that is, ASM) error

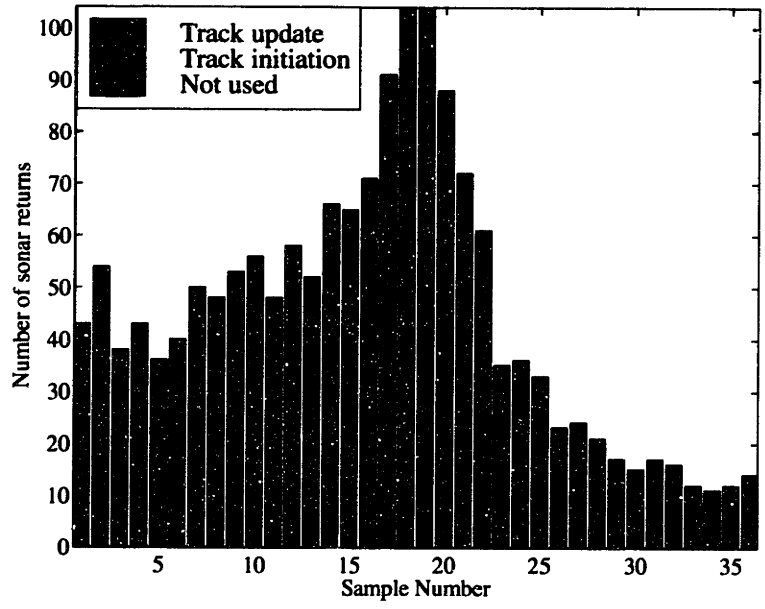

Figure 2-24: The distribution of returns from the HRA used for track updating, track initiation and the returns that were rejected as a function of sample number, that is, 20 seconds time increments at which sonar returns were obtained. Notice how few of the returns that were used for track updating and thus for improving mapping and localization in the environment.

remains bounded. At $t = 120$ seconds, CML yields a worse estimate than the INS due to a data association error. Figure 2-26 shows the errors and 3σ errors bounds for the position, heading and velocity estimates produced by post-processing of the data through the ASM algorithm. The increase in uncertainty in position towards the end of the run is due to the fact that no new features were observed during the last part of the data collection.

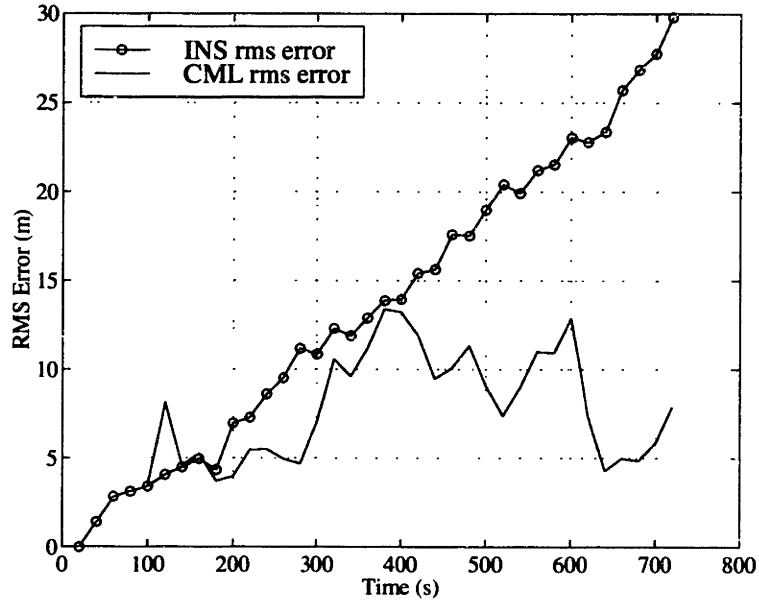

Figure 2-25: A comparison of root mean square error when relying on the inertial navigation system (INS) versus the CML result produced by augmented stochastic mapping as a function of time.

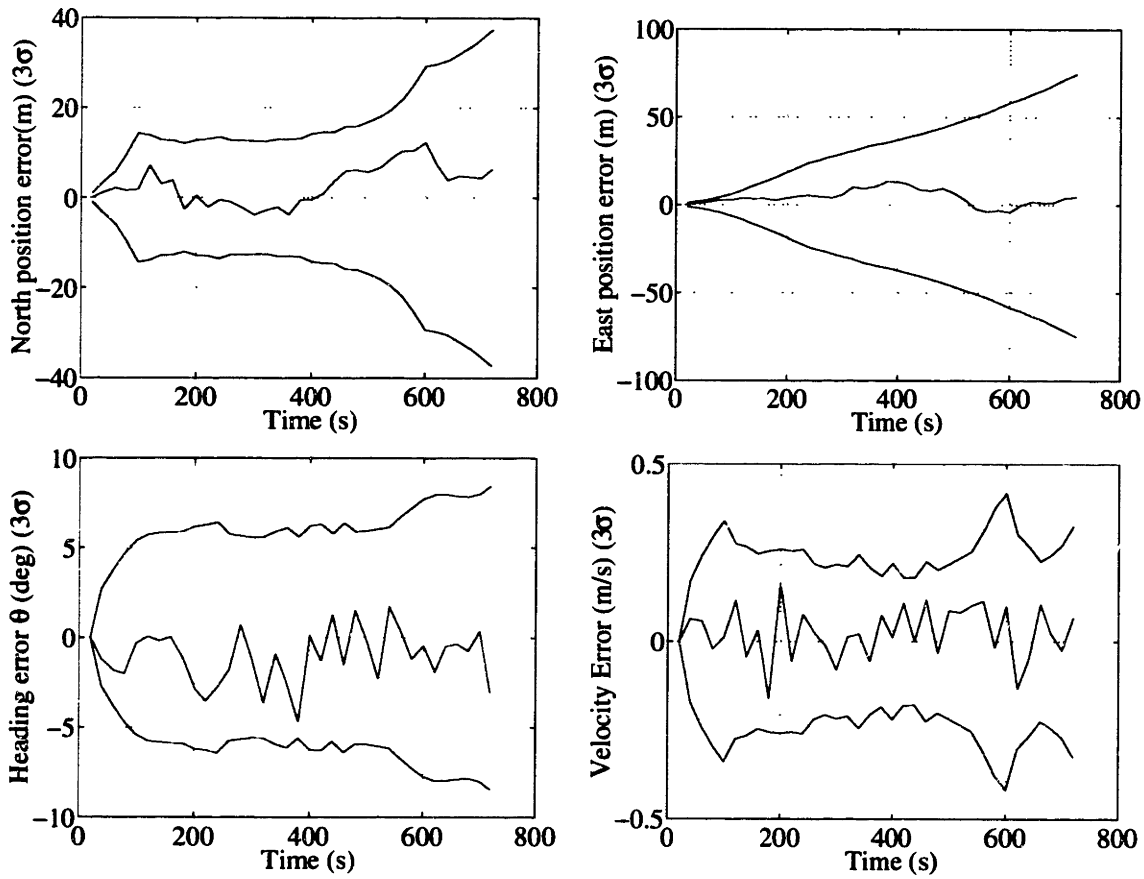

Figure 2-26: Errors and 3σ bounds (99% highest confidence bounds) for the position, heading and velocity estimates produced by post-processing the data from leg A3 with the augmented stochastic mapping algorithm for performing CML.

2.8 Failure modes

In this section, describe several different types of failures that have been encountered in different CML missions that were performed under similar conditions. There are three basic modes of failure for the system: divergence due to data association errors, map slip (re-mapping), and mapping of ghost features (excess mapping). All three of these failure modes are a consequence of giving a measurement the wrong interpretation.

The most catastrophic failures of CML occur when measurements from one feature are associated with the wrong feature for several consecutive time steps. The result is that the system state vector is updated with erroneous data and the actual system errors will move outside the error bounds specified by the system covariance matrix. The system becomes overconfident and the estimate diverges. This type of failure is most troublesome because the algorithm believes it is still providing accurate navigation information. This type of error will occur more frequently when multiple features are close together and data association is more ambiguous. In addition, unmodeled dynamics, currents and high clutter densities may lead the system to make this type of error. As seen in Figure 2-27, when a data association error occurs, the vehicle's state estimate moves far outside the estimated error bounds. Thus, one promising way of resolving data association errors on-the-fly is to compare the estimated vehicle's state after the incorporation of the last sonar measurements in the stochastic mapping (SM) update step, to the estimate produced with dead-reckoning alone. If the SM update predicts the state of the vehicle to be outside the dead-reckoning error bounds, a data association fault may have occurred. This is the case since a dead-reckoning estimate will *always* bound the true error and the SM update *should* always produce an improvement over dead-reckoning, if there were no data association errors. Thus, once a data association fault occurs, the track that was incorrectly associated with the measurement can be identified and errors with association to this track can be eliminated in the future. Knowing this, the SM update step can be

redone without the incorrect data association, thereby maintaining consistency.

A second mode of failure that is encountered is "map-slip" as shown on the left of Figure 2-28. This occurs if the vehicle's position is close to the error bounds, and due to the linearization of the non-linear measurement-to-vehicle coordinate transformation, *all* the features are re-mapped in new locations that are slightly shifted with respect to the original map. A higher-level reasoning process may be able to detect this type of failure during the mission and correct for it. Alternatively, this failure could be corrected during post-processing. The danger of not resolving the failure on-the-fly is that the algorithm is more prone to making incorrect data association decisions which could lead to divergence.

The third type of failure is excess mapping illustrated on the right of Figure 2-28. Due to multiple spurious measurements that by chance mutually support one another, features that does not exist in reality are added to the map. This might occur if track initiation is too aggressive, track deletion is too conservative, or if there is a very high density of clutter. This type of failure does not usually have a catastrophic effect on the accuracy of the AUV position estimate. The false features tend to be randomly distributed and infrequently updated. Their presence, however, increases the computational burden and increases the likelihood that measurements originating from a nearby true feature will be incorrectly associated.

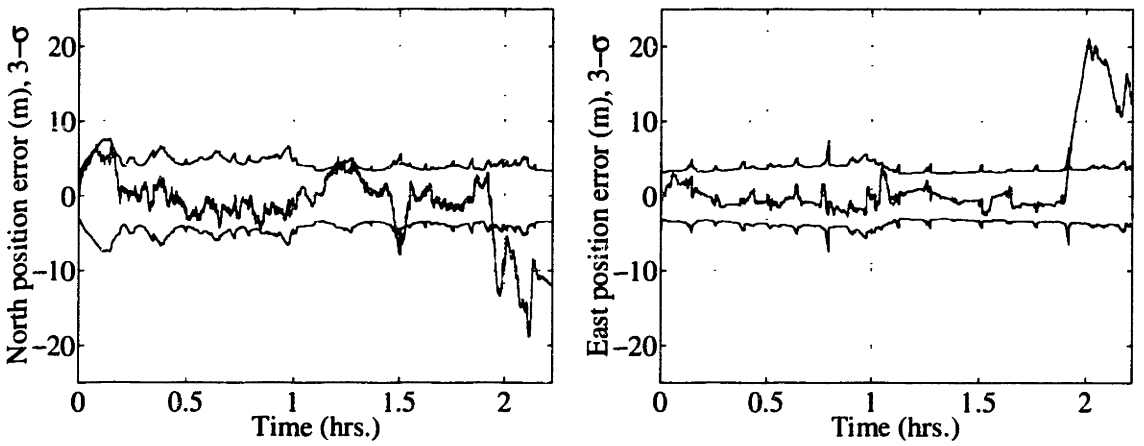

Figure 2-27: Example of divergence due to data association error.

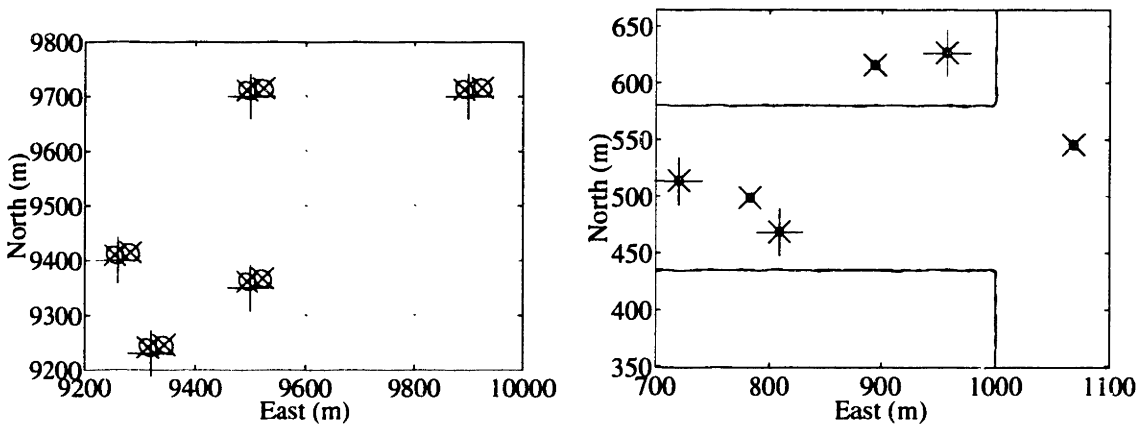

Figure 2-28: **Left:** Example of map slip/re-mapping. **Right:** Example of excess mapping.

2.9 Conclusions

In this chapter, we have presented an AUV model and developed augmented stochastic mapping (ASM) for performing CML by an AUV. The issues of long-term performance of ASM for an AUV equipped with a forward look sonar was investigated. Sensor uncertainty, dropouts, false returns, vehicle dynamic model uncertainty, and external current disturbances were incorporated. The results indicate that CML can provide position estimation with long-term bounded error for navigation in unknown, unmapped environments. With reasonable assumptions about sensor and vehicle performance, positioning errors of a few meters were achieved for missions over five hours in duration. The consistency of the error estimates produced by the algorithm is maintained.

To further validate the method of ASM, an experiment in a testing tank modeling an AUV and post-processing of oceanic data was performed. The experiment showed excellent agreement with the simulation, thereby confirming the validity of the simulations. By post-processing oceanic data we obtained a significant improvement in vehicle localization by performing CML in comparison to the localization achieved by the inertial navigation system (INS).

Several of the failure modes for CML have been illustrated. These include estimator divergence due to data association ambiguity, map slip, and initialization of false targets that arise from spurious measurements. Surprisingly, acceptable CML performance is maintained in many situations even when these errors occur. In general, however, such errors increase the likelihood of data association mistakes which may cause the CML process to fail catastrophically.

Future research is necessary in four important areas: 1.) mitigation of computation complexity, 2.) adaptive sensing and control, 3.) feature extraction, and 4.) management of data association uncertainty. Computational complexity emerges as a serious issue in the extension of CML to missions over very large areas with thousands of features. As the size of the system covariance matrix increases, the computational

burden of stochastic mapping increases as $\mathcal{O}(n^3)$.

Adaptive sensing and motion strategies have the potential to improve state estimation accuracy and robustness of CML [43].

The approach assumes that the environment contains salient features, and that the vehicle's sensors are capable of detecting them. There are many open issues for research in feature extraction, such as the detection and representation of complex natural terrain features. However, we find the feature extraction techniques utilized by Carpenter *et al.* for the processing of data from the US Navy's HRA forward look imaging sonar [80, 24] and used in Section 2.7 promising.

Improved data association methods will be critical to enable operation in more highly cluttered environments and to improve the robustness and reliability of CML. Data association provides the means to initialize new features into the map, to associate measurements with map features, and to detect changes in the environment. Improved decision making helps prevent failures and enables detection and recovery when failures do occur. Hybrid estimation offers a theoretical framework for investigation of more advanced data association decision making techniques [108, 105].

2.10 Summary

Extended missions in unknown regions present a significant navigational challenge for autonomous underwater vehicles (AUVs). CML technology can provide a significant improvement in the navigational capabilities of AUVs and can enable new missions in unmapped regions without reliance on acoustic beacons or surfacing for GPS resets. In this chapter, an AUV model was presented and augmented stochastic mapping (ASM) was developed as a technique for performing concurrent mapping and localization for mobile robots in general and for AUVs in particular. The validity of ASM as a technique for performing long-term concurrent mapping and localization (CML) was investigated for the scenario of an AUV making observations of point features

in the environment with a forward look sonar. Simulation results demonstrated that position estimates with long-term bounded errors of a few meters can be achieved under realistic assumptions about the vehicle, its sensors, and the environment. Further validation of ASM was demonstrated by the excellent agreement between simulation and experiment in a testing tank, and by post-processing of oceanic data. Potential failure modes of the algorithm, such as divergence and map slip, were discussed.

The necessity of future research in four important areas of 1.) computational complexity, 2.) adaptive sensing and control, 3.) feature extraction, and 4.) data association uncertainty was identified. In the next chapter, we will address the first of these issues and present a new solution to the computational complexity encountered by ASM. Chapter 4 will address the issue of adaptive sensing and control in CML.

2.A Kalman filtering

In this section we will outline the Kalman filter and the extended Kalman filter. For a full derivation we refer the reader to [5, 46, 93].

A Kalman filter is an computationally efficient linear minimum mean square error (MMSE) estimator for a dynamic system. We will limit ourselves to discrete time systems, as the implementation is assumed to be performed using a digital computer.[2]

Next we will define a simple Kalman filter as a least squares estimator and subsequently extend this filter to a nonlinear system, known as the extended Kalman filter (EKF). EKF is the core method used in stochastic mapping.

2.A.1 The Kalman filter

The Kalman filter is a linear MMSE estimator for a dynamic system. If the dynamic model is linear and the noise processes are Gaussian, then the Kalman filter is an optimal estimator. If the system is linear but the noise are non-Gaussian, the Kalman filter is the best linear estimator. In this derivation we will assume a linear dynamic system with Gaussian noise processes. A state-space representation of a discrete-time stochastic system can be written in vector discrete form as

$$\mathbf{X}_{k+1} = \mathbf{F}\mathbf{X}_k + \mathbf{d}_{\mathbf{x}_k}. \tag{2.11}$$

Here $\mathbf{X}_k$ is the system's (true) state vector at time step k. The matrix $\mathbf{F}$ is the discrete time dynamic system model, and $\mathbf{d}_{\mathbf{x}_k}$ is the discrete time noise process.

In addition some physical quantities, $\mathbf{z}_k$, of the system are observed and modeled

[2] Continuous time Kalman filters also exist, but will not be discussed here. See [46] for a derivation.

by a linear observation model, $\mathbf{H}$, given by

$$\mathbf{z}_k = \mathbf{H}\mathbf{X}_k + \mathbf{d}_{\mathbf{z}_k}, \tag{2.12}$$

where $\mathbf{d}_{\mathbf{z}_k}$ is the measurement noise process. The time index k in $\mathbf{d}_{\mathbf{x}_k}$ and $\mathbf{d}_{\mathbf{z}_k}$ will often be dropped to simplify notation.

The stochastic noise processes $\mathbf{d}_\mathbf{x}$ and $\mathbf{d}_\mathbf{z}$ are assumed to be white, uncorrelated Gauss-Markov processes [93] with covariance $\mathbf{G}$ and $\mathbf{R}$, respectively. That is, the system covariance $\mathbf{G}$ and the observation covariance $\mathbf{R}$ are given by

$$\begin{aligned} E(\mathbf{d}_{\mathbf{x}_k}\mathbf{d}_{\mathbf{x}_n}^T) &= \delta_{kn}\mathbf{G} \\ E(\mathbf{d}_{\mathbf{z}_k}\mathbf{d}_{\mathbf{z}_n}^T) &= \delta_{kn}\mathbf{R} \end{aligned}$$

and

$$E(\mathbf{d}_{\mathbf{x}_k}\mathbf{d}_{\mathbf{z}_n}^T) = \mathbf{0}. \tag{2.13}$$

where

$$\delta_{kn} = \begin{cases} 1 \text{ if } k = n, \\ 0 \text{ if } k \neq n. \end{cases}$$

Further, without loss of generality it can be assumed that the noise processes are zero-mean [46, 3].

To clarify notation, let $\mathbf{X}_k$ denote the *true* state at time step k and $\mathbf{x}_{k|k-1}$ be the *estimated* state at time step k given all information up to time step $k-1$. The error state is defined by

$$\tilde{\mathbf{x}}_{k|k} = \mathbf{x}_{k|k} - \mathbf{X}_k. \tag{2.14}$$

Using these definitions the Kalman filter can be derived as a linear MMSE estimator

for the dynamic system given by Equations (2.11) through 2.13. The condition of linearity and MMSE can be separated and stated as

$$\begin{aligned} \mathbf{x}_{k|k} &= \mathbf{x}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\mathbf{x}_{k|k-1}) \quad ① \\ \mathbf{x}_{k|k} &= \min_{\tilde{\mathbf{x}}_{k|k}} \left\{ E(\tilde{\mathbf{x}}_{k|k}\tilde{\mathbf{x}}_{k|k}^T) \stackrel{\text{def}}{=} \mathbf{P}_{k|k} \right\} \quad ② \end{aligned} \tag{2.15}$$

where the condition ① imposes the linear constraint for some gain $\mathbf{K}$ to be determined. The expectation operator is defined by E and the condition ② is the MMSE problem, which is solved for the optimal gain $\mathbf{K}$ given ①. Combining Equations (2.12) and (2.15) ① we have

$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}_k\mathbf{H}(\mathbf{X}_k - \mathbf{x}_{k|k-1}) + \mathbf{K}_k\mathbf{d}_{\mathbf{z}_k}.$$

Subtracting $\mathbf{X}_k$ from both sides of the above equation and utilizing the definition of $\tilde{\mathbf{x}}_{k|k-1} = \mathbf{x}_{k|k-1} - \mathbf{X}_k$, we get

$$\begin{aligned} \tilde{\mathbf{x}}_{k|k} &= \tilde{\mathbf{x}}_{k|k-1} - \mathbf{K}_k\mathbf{H}\tilde{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{d}_{\mathbf{z}_k} \\ &= (\mathbf{I} - \mathbf{K}_k\mathbf{H})\tilde{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{d}_{\mathbf{z}_k}, \end{aligned} \tag{2.16}$$

using this result in Equation (2.15) ② results in

$$\begin{aligned} \mathbf{P}_{k|k} &= (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_{k|k-1}(\mathbf{I} - \mathbf{K}_k\mathbf{H})^T + \mathbf{K}_k\mathbf{R}\mathbf{K}_k^T \\ &\quad +(\mathbf{I} - \mathbf{K}_k\mathbf{H})E(\tilde{\mathbf{x}}_{k|k-1}{\mathbf{d}_{\mathbf{z}_k}}^T)\mathbf{K}_k^T \\ &\quad +\mathbf{K}_k E(\mathbf{d}_{\mathbf{z}_k}\tilde{\mathbf{x}}_{k|k-1}^T)(\mathbf{I} - \mathbf{K}_k\mathbf{H})^T \\ &= (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_{k|k-1}(\mathbf{I} - \mathbf{K}_k\mathbf{H})^T + \mathbf{K}_k\mathbf{R}\mathbf{K}_k^T \end{aligned} \tag{2.17}$$

where the two last terms are dropped out since $E(\tilde{\mathbf{x}}_{k|k-1}\mathbf{d}_{\mathbf{z}_k}) = \mathbf{0}$ by Equation (2.13), that is, measurement errors are uncorrelated. The optimal gain $\mathbf{K}_k$ is chosen by minimizing the *length* of the estimation error vector. This is equivalent to minimizing

the trace of $\mathbf{P}_{k|k}$, thus[3]

$$\frac{\partial}{\partial \mathbf{K}_k}\text{trace}(\mathbf{P}_{k|k}) = \mathbf{0} = -2(\mathbf{I} - \mathbf{K}_k)\mathbf{P}_{k|k-1}\mathbf{H}^T + 2\mathbf{K}_k\mathbf{R},$$

and solving for $\mathbf{K}_k$ yields

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}^T(\mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R})^{-1}. \tag{2.18}$$

Substituting Equation (2.18) into Equation (2.17) and (2.15) ① yields the optimal estimate $\mathbf{x}_{k|k}$ and associated covariance of the estimate $\mathbf{P}_{k|k}$ given the previous estimates $\mathbf{x}_{k|k-1}$, $\mathbf{P}_{k|k-1}$ and the measurement $\mathbf{z}_k$. This is summarized in the *Kalman update* as

$$\boxed{\begin{aligned} \mathbf{x}_{k|k} &= \mathbf{x}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\mathbf{x}_{k|k-1}) \\ \mathbf{P}_{k|k} &= (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_{k|k-1}(\mathbf{I} - \mathbf{K}_k\mathbf{H})^T + \mathbf{K}_k\mathbf{R}\mathbf{K}_k^T \end{aligned}} \tag{2.19}$$

Now that the optimal estimate given a measurement is obtained, we need to propagate the estimate according to the dynamic model of Equation (2.11). That is,

$$\mathbf{x}_{k+1|k} \stackrel{def}{=} E(\mathbf{X}_{k+1}) = \mathbf{F}\mathbf{x}_{k|k},$$

since the noise process $\mathbf{d}_\mathbf{x}$ is assumed to be zero mean. The propagated covariance is obtained by first noting the relation

$$\begin{aligned} \tilde{\mathbf{x}}_{k+1|k} &= \mathbf{x}_{k+1|k} - \mathbf{X}_k \\ &= \mathbf{F}\tilde{\mathbf{x}}_{k|k} - \mathbf{d}_{\mathbf{x}_k}, \end{aligned}$$

[3]We here make use of the relation for the partial derivative of the trace of the product of two matrices A and B with B symmetric, $\frac{\partial}{\partial A}\text{trace}(ABA^T) = 2AB$ [46].

and then using the definition of $\mathbf{P}_{k+1|k} = E(\tilde{\mathbf{x}}_{k+1|k}\tilde{\mathbf{x}}_{k+1|k}^T)$ to achieve

$$\mathbf{P}_{k+1|k} = \mathbf{F}\mathbf{P}_{k|k}\mathbf{F}^T + \mathbf{G}.$$

The two prediction steps are summarized in the *Kalman prediction* step as

$$\boxed{\begin{aligned}\mathbf{x}_{k+1|k} &= \mathbf{F}\mathbf{x}_{k|k} \\ \mathbf{P}_{k+1|k} &= \mathbf{F}\mathbf{P}_{k|k}\mathbf{F}^T + \mathbf{G}.\end{aligned}} \tag{2.20}$$

Given an initial state estimate $\mathbf{x}_{0|-1}$ and covariance $\mathbf{P}_{0|-1}$ the Kalman filter algorithm is summarized in Figure 2-29.

1. Initialization:	$\mathbf{x}_{0\|-1}$	$=$	$E(\mathbf{X}_0)$
	$\mathbf{P}_{0\|-1}$	$=$	$E\left((\mathbf{x}_{0\|-1} - \mathbf{X}_0)(\mathbf{x}_{0\|-1} - \mathbf{X}_0)^T\right)$
2. Compute gain:	$\mathbf{K}_k$	$=$	$\mathbf{P}_{k\|k-1}\mathbf{H}^T\left(\mathbf{H}\mathbf{P}_{k\|k-1}\mathbf{H}^T + \mathbf{R}\right)^{-1}$
3. Update:	$\mathbf{x}_{k\|k}$	$=$	$\mathbf{x}_{k\|k-1} + \mathbf{K}_k\left(\mathbf{z}_k - \mathbf{H}\mathbf{x}_{k\|k-1}\right)$
	$\mathbf{P}_{k\|k}$	$=$	$(\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_{k\|k-1}(\mathbf{I} - \mathbf{K}_k\mathbf{H})^T + \mathbf{K}_k\mathbf{R}\mathbf{K}_k^T$
4. Prediction:	$\mathbf{x}_{k+1\|k}$	$=$	$\mathbf{F}\mathbf{x}_{k\|k}$
	$\mathbf{P}_{k+1\|k}$	$=$	$\mathbf{F}\mathbf{P}_{k\|k}\mathbf{F}^T + \mathbf{G}$
5. Increment k.	Goto step 2.		

Figure 2-29: The Kalman filter algorithm for the system of Equation (2.11) and (2.12).

In brief, the Kalman filter estimates the state $\mathbf{x}_{k|k-1}$ at time step k and summarizes the information up to time step $k-1$. From this it produces an updated state estimate $\mathbf{x}_{k|k}$ by linearly combining $\mathbf{x}_{k|k-1}$ with the observation $\mathbf{z}_k$. A predicted state is obtained from the system model to produce $\mathbf{x}_{k+1|k}$, thus completing the recursion. A known control input, $\mathbf{B}\mathbf{u}_k$ can be added to the system model of Equation (2.11), which will result in an addition of this term to the prediction step. Further, the

Kalman filter can be modified to handle colored noise and time dependent **F**, **H**, **G** and **R**.

Next we will derive the extended Kalman filter from the Kalman filter.

2.A.2 The extended Kalman filter

The key feature of the Kalman filter is that it uses a system and observation model, Equations (2.11) and (2.12), to transform means and covariances. In the Kalman filter approach, the system and observation model are linear and the transforms become linear. However, in general the models are not linear but take on the more general nonlinear form

$$\begin{aligned} \mathbf{X}_{k+1} &= \mathbf{f}(\mathbf{X}_k, \mathbf{u}_k) + \mathbf{d}_{\mathbf{x}_k} \\ \mathbf{z}_k &= \mathbf{h}(\mathbf{X}_k) + \mathbf{d}_{\mathbf{z}_k}, \end{aligned} \tag{2.21}$$

where **f** and **h** are the nonlinear system and observation models and $\mathbf{u}_k$ is a known control input. The other variables are the same as in Equations (2.11) and (2.12). However, propagating a covariance function through a nonlinear system does not guarantee that the statistics are conserved. Thus, in order to circumvent the problem of transformation of nonlinearities, the nonlinear models of Equation (2.21) are approximated through a Taylor series expansion, keeping only the first two terms.[4] This approximation is equivalent to the assumption that the estimated mean at the previous time step, $\mathbf{x}_{k|k}$, is approximately equal to the *true* system state, $\mathbf{X}_k$, at the previous time step. Once these linearizations have been performed and assuming that the approximation error is small, the filter is derived in the same manner as for the Kalman filter. That is, using the MMSE problem formulation of Equation (2.15), the

[4] Although not utilized in this thesis, the second order term may be included in the Taylor series expansion to improve the estimate. However, this comes at a computational cost [46].

linearity condition becomes

$$\begin{aligned} \mathbf{x}_{k|k} &= \mathbf{x}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - E[\mathbf{h}(\mathbf{X}_k)]) \\ &= \mathbf{x}_{k|k-1} + \mathbf{K}_k(\mathbf{h}(\mathbf{X}) - E[\mathbf{h}(\mathbf{X}_k)]) + \mathbf{K}_k\mathbf{d}_\mathbf{z}, \end{aligned} \tag{2.22}$$

where the second step was obtained by substituting in from Equation (2.21). However, computing the expectation in Equation (2.22) requires the knowledge of the underlying probability distribution of $\mathbf{X}_k$ which is not known. Thus, the nonlinear observation model is linearized through a Taylor series expansion. Keeping the first two terms of the Taylor series expansion of $\mathbf{h}$ about the mean $\mathbf{x}_{k|k-1}$ yields

$$\mathbf{h}(\mathbf{X}_k) \approx \mathbf{h}(\mathbf{x}_{k|k-1}) + \mathbf{H}_\mathbf{x}\tilde{\mathbf{x}}_{k|k-1}, \tag{2.23}$$

where $\mathbf{H}_\mathbf{x} = d\mathbf{h}(\mathbf{X})/d\mathbf{X}|_{\mathbf{X}=\mathbf{x}_{k|k-1}}$ is the Jacobian of $\mathbf{h}$ with respect to $\mathbf{X}$ evaluated at $\mathbf{x}_{k|k-1}$. Combining Equation (2.23) with Equation (2.21), the expectation in Equation (2.22) can be computed as

$$E[\mathbf{h}(\mathbf{X}_k)] = \mathbf{h}(\mathbf{x}_{k|k-1}) + \mathbf{H}_\mathbf{x}\mathbf{0} + \mathbf{0}. \tag{2.24}$$

The second term is zero as the estimator is required to be non-biased. The third term is zero by the assumption that the observation noise process, $\mathbf{d}_\mathbf{z}$, is zero mean.

Combining equations (2.22), (2.23) and (2.24) and subtracting the true state $\mathbf{X}_k$ we arrive at the equivalent form of Equation (2.16)

$$\tilde{\mathbf{x}}_{k|k} = (\mathbf{I} + \mathbf{K}_k\mathbf{H}_\mathbf{x})\tilde{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{d}_\mathbf{z},$$

and the extended Kalman filter gain $\mathbf{K}_k$ and update step become identical to Equation (2.18) and Equation (2.19), respectively, with $\mathbf{H}_\mathbf{x}$ substituted for $\mathbf{H}$.

In order to arrive at the EKF prediction step, the nonlinear dynamic model, $\mathbf{f}$, is linearized through a Taylor series expansion about $\mathbf{x}_{k|k}$ as the probability distribution

of $\mathbf{X}_k$ is not available. Keeping only the first two terms of the expansion yields

$$\mathbf{f}(\mathbf{X}_k) \approx \mathbf{f}(\mathbf{x}_{k|k}, \mathbf{u}_k) + \mathbf{F}_\mathbf{x}\tilde{\mathbf{x}}_{k|k}, \tag{2.25}$$

where $\mathbf{F}_\mathbf{x} = d\mathbf{f}(\mathbf{X}, \mathbf{u}_k)/d\mathbf{X}|_{\mathbf{X}=\mathbf{x}_{k|k}}$ is the Jacobian of $\mathbf{f}$ with respect to $\mathbf{X}$, evaluated at $\mathbf{x}_{k|k}$.

Taking the expectation of the dynamic model in Equation (2.21) and utilizing Equation (2.25), the predicted state in the EKF becomes

$$\mathbf{x}_{k+1|k} = \mathbf{f}(\mathbf{x}_{k|k}, \mathbf{u}_k). \tag{2.26}$$

The predicted covariance is found by subtracting $\mathbf{X}_{k+1}$ from the left side and $\mathbf{f}(\mathbf{X}_k, \mathbf{u}_k) + \mathbf{d}_\mathbf{x}$ from the right side of Equation (2.26), and using Equation (2.25) to approximate $\mathbf{f}$ and computing the error covariance. That is,

$$\begin{aligned}
\mathbf{x}_{k+1|k} - \mathbf{X}_{k+1} &= \mathbf{f}(\mathbf{x}_{k|k}, \mathbf{u}_k) - \mathbf{f}(\mathbf{X}_k, \mathbf{u}_k) - \mathbf{d}_\mathbf{z} \\
\tilde{\mathbf{x}}_{k+1|k} &\approx \mathbf{f}(\mathbf{x}_{k|k}, \mathbf{u}_k) - \mathbf{f}(\mathbf{x}_{k|k}, \mathbf{u}_k) - \mathbf{F}_\mathbf{x}\tilde{\mathbf{x}}_{k|k} - \mathbf{d}_{\mathbf{z}_k} \\
&= -\mathbf{F}_\mathbf{x}\tilde{\mathbf{x}}_{k|k} - \mathbf{d}_{\mathbf{x}_k}. \\
\mathbf{P}_{k+1|k} \equiv E[\tilde{\mathbf{x}}_{k+1|k}\tilde{\mathbf{x}}_{k+1|k}^T] &= \mathbf{F}_\mathbf{x}\mathbf{P}_{k|k}\mathbf{F}_\mathbf{x}^T + \mathbf{G}_k.
\end{aligned}$$

Combining these results, the extended Kalman filter (EKF) is summarized in Figure 2-30. The form of the EKF is chosen to resemble that of the conventional Kalman filter, which is optimal for linear systems. However, the EKF is in general suboptimal due to the approximations of Equation (2.23) and Equation (2.25). The state estimate and the covariances are all based on the assumption that the approximation error is small. If this is not the case, the filter might diverge.

1. Initialization:	$\mathbf{x}_{0\|-1}$	$=$	$E(\mathbf{X}_0)$
	$\mathbf{P}_{0\|-1}$	$=$	$E\left((\mathbf{x}_{0\|-1}-\mathbf{X}_0)(\mathbf{x}_{0\|-1}-\mathbf{X}_0)^T\right)$
2. Compute gain:	$\mathbf{K}_k$	$=$	$\mathbf{P}_{k\|k-1}\mathbf{H}_\mathbf{x}^T\left(\mathbf{H}_\mathbf{x}\mathbf{P}_{k\|k-1}\mathbf{H}_\mathbf{x}^T+\mathbf{R}_k\right)^{-1}$
3. Update:	$\mathbf{x}_{k\|k}$	$=$	$\mathbf{x}_{k\|k-1}+\mathbf{K}_k\left(\mathbf{z}_k-\mathbf{h}(\mathbf{x}_{k\|k-1})\right)$
	$\mathbf{P}_{k\|k}$	$=$	$(\mathbf{I}-\mathbf{K}_k\mathbf{H}_\mathbf{x})\mathbf{P}_{k\|k-1}(\mathbf{I}-\mathbf{K}_k\mathbf{H}_\mathbf{x})^T+\mathbf{K}_k\mathbf{R}\mathbf{K}_k^T$
4. Prediction:	$\mathbf{x}_{k+1\|k}$	$=$	$\mathbf{f}(\mathbf{x}_{k\|k},\mathbf{u}_k)$
	$\mathbf{P}_{k+1\|k}$	$=$	$\mathbf{F}_\mathbf{x}\mathbf{P}_{k\|k}\mathbf{F}_\mathbf{x}^T+\mathbf{G}_k$
5. Increment k:	Goto step 2.		

Figure 2-30: The extended Kalman filter (EKF) for the nonlinear system of Equation (2.21).

2.A.3 Computational issues

Kalman filters are recursive estimators and thus computationally efficient. As can be noted from Figure 2-29 the covariance $\mathbf{P}$ and the Kalman gain $\mathbf{K}$ are independent of the observation, and thus can be pre-computed off-line. Thus, for linear systems a very large state vector can be estimated in real time. In the extended Kalman filter, the covariance and Kalman gain are *dependent* on the observations, thus $\mathbf{P}$, $\mathbf{K}$ and the Jacobians of the system and observation models must be computed on-line for each new observation. Thus, the non-linear filter is computationally expensive for large state vectors $\mathbf{x}$, and is proportional to the cube of the number of states. Thus, real-time computations are very challenging for systems with hundreds of states and more.

In much of the literature on Kalman filtering, the covariance update step is simplified further to

$$\mathbf{P}_{k|k} = (\mathbf{I}-\mathbf{K}\mathbf{H}_\mathbf{x})\mathbf{P}_{k|k-1}, \tag{2.27}$$

which is computationally more efficient than the *Joseph form covariance update* of

Equation (2.19). However, to prevent numerical instability in the computation of the Kalman filter due to round off errors, it is advised that the square-root filter or the Joseph form covariance update [5] be employed. These methods guarantee the positive definiteness of the covariance. Equation (2.27) on the other hand, is susceptible to round-off errors and will not guarantee positive definiteness. In this thesis the Joseph form is used.

2.B Performance of gated nearest-neighbor data association

The method in which *data association* is performed, that is, how the origins of measurements are determined, is one of the major factors affecting the performance of CML. Several methods for resolving the data association problem have been proposed in the literature, such as, the nearest neighbor standard filter (NNSF) [4], the probabilistic data association filter (PDAF) [4], and integrated mapping and navigation (IMAN) [108]. The NNSF is the simplest method to implement and is a gated nearest neighbor data association technique similar to the one used in this thesis. That is, the "nearest" return (measurement) to an estimated feature falling inside a validation gate is used for updating the state of the feature. The problem with choosing a nearest neighbor approach is that, with some probability, the associated measurement is not the correct one. This probability will be derived next.

The factors that affect the performance of the data association filter used in this thesis are environmental variables, such as the density λ_N of new (actual) features, the density λ_F of false features (clutter), and the validation gate size, γ. From the point of view of a given feature, extraneous measurements arise either from clutter or from new true features and have a spatial density at any given sampling time of

$$\lambda = \lambda_F + P_D \lambda_N, \tag{2.28}$$

where P_D is defined as the probability of detection. Assuming a uniform distribution of the extraneous features with density λ and a Gaussian distribution for the measurement of the actual feature, the probability that a return (measurement) from the extraneous features will be closer to the estimated state than one from the actual feature will give the probability for an incorrect data association when using NNSF. This probability will now be derived in the general d dimensional space.

Consider an d dimensional volume V where N extraneous features are placed randomly and independently. The density of points is given by

$$\lambda = \frac{N}{V}. \tag{2.29}$$

The probability that an extraneous feature lands in a small volume v is v/V. Therefore, the probability for *not* having an extraneous feature in v is $(1 - v/V)$. The probability, P_0, for having *no* extraneous feature within the volume v after having placed N extraneous features independently into the container volume V is given by $[(V - v)/V]^N = [1 - (\lambda v)/N]^N$. As N and V are increased, holding λ and v fixed, we find

$$P_0 = \lim_{N\to\infty}(1 - \frac{\lambda v}{N})^N = e^{-\lambda v}. \tag{2.30}$$

Now, the probability distribution for the observation of an actual feature, T, given that it was detected, D, is assumed from the Kalman filter to be normally distributed over the innovation, $\boldsymbol{\nu} = \mathbf{z} - \mathbf{h}(\mathbf{x})$, with the associated covariance matrix $\mathbf{S}$. That is,

$$p(T|D) = \mathcal{N}(\boldsymbol{\nu}, \mathbf{S}) = \frac{1}{(2\pi)^{d/2}\prod_i^d \sigma_i} e^{-\frac{1}{2}\boldsymbol{\nu}^T\mathbf{S}^{-1}\boldsymbol{\nu}}, \tag{2.31}$$

where σ_i are the square roots of the i'th eigenvalue of $\mathbf{S}$. The probability that an observation of an actual feature is in the d dimensional shell, dv, surrounding the

volume v, given that the feature was detected is given by

$$P(T|D) = \mathcal{N}(\boldsymbol{\nu}, \mathbf{S})dv \tag{2.32}$$

CPF where $\mathcal{N}(\boldsymbol{\nu}, \mathbf{S})$ is a normal distribution with mean $\boldsymbol{\nu}$ and covariance $\mathbf{S}$.

Assuming independence of the actual features observations and the extraneous features, the probability distribution of there being an extraneous return, F, originating inside the volume v, and an actual return in the shell surrounding v is given by

$$p(F|D) = \mathcal{N}(\boldsymbol{\nu}, \mathbf{S}) \cdot (1 - e^{-\lambda v}) \cdot dv. \tag{2.33}$$

From this equation we can find the probability that a return from an extraneous feature was closer to the estimated feature than the true return, inside the volume v by integrating Equation (2.33) over the volume v, i.e.

$$P(F|D, v) = \int_v \frac{1}{(2\pi)^{d/2} \prod_i^d \sigma_i} e^{-\frac{1}{2}\boldsymbol{\nu}^T \mathbf{S}^{-1} \boldsymbol{\nu}} (1 - e^{-\lambda v}) dv. \tag{2.34}$$

In the NNSF, gating is performed on the measurements before the nearest neighbor is chosen. Typically γ standard deviations are chosen, thus defining the volume v. More precisely, a ρ–standard deviation volume v in d dimensional space is given by

$$v = c\rho^d \prod_i^d \sigma_i, \tag{2.35}$$

where the constant c is the volume factor of the d dimensional unit sphere (2, π, $4\pi/3$, etc.) Therefore, the $d-1$ dimensional shell surrounding this volume is given by

$$dv = dc\rho^{d-1} \prod_i^d \sigma_i d\rho\,. \tag{2.36}$$

Further, notice that $\boldsymbol{\nu}^T\mathbf{S}^{-1}\boldsymbol{\nu} \leq \rho^2$ defines the volume v. Thus by changing the coordinate system and combining equations (2.34), (2.35) and (2.36) we arrive at the general form for the probability of a false association given that the actual feature was detected and a validation gate of γ standard deviations was used as

$$P(F|D,\gamma) = \int_0^\gamma \frac{cd}{(2\pi)^{d/2}} \rho^{d-1} e^{-\frac{1}{2}\rho^2} (1 - e^{-\lambda c \rho^d \prod_i^d \sigma_i}) d\rho, \quad \lambda > 0. \tag{2.37}$$

In order to find the probability of false association given the gate γ we need to include the probability of detection, P_D, of an actual feature. This is done by use of Bayes rule,

$$P(F|\gamma) = P(F|D,\gamma)P_D + P(F|\sim D,\gamma)(1 - P_D). \tag{2.38}$$

We notice that the probability of making a false association given that no actual feature was observed, $\sim D$, within the validation gate γ is given by

$$P(F|\sim D,\gamma) = 1 - e^{-\lambda c \gamma^d \prod_i^d \sigma_i}. \tag{2.39}$$

That is, the probability that an extraneous return falls within the validation gate γ. Combining equations (2.37), (2.38) and (2.39) we arrive at the general form in d dimensions for an incorrect association to take place when using the NNSF:

$$P(F|D) = P_D \int_0^\gamma \frac{dc}{(2\pi)^{d/2}} \rho^{d-1} e^{-\frac{1}{2}\rho^2} (1 - e^{-\lambda c \rho^d \prod_i^d \sigma_i}) d\rho + (1 - e^{-\lambda c \gamma^d \prod_i^d \sigma_i})(1 - P_D). \tag{2.40}$$

This integral can be solved analytically with the inclusion of the error function for certain dimensions, in particular $d = 1, 2$ and $d = 3$. For the NNSF in two dimensions

integrating out Equation (2.40) yields

$$P(F|\gamma) = (P_D/\beta)\left[(1-e^{-\frac{\gamma^2}{2}})\beta - (1-e^{-\frac{\gamma^2}{2}\beta})\right] + (1-e^{-\lambda c\gamma^d \prod_i^d \sigma_i})(1-P_D), \quad (2.41)$$

with $\beta = 1 + 2\pi\lambda\sigma_1\sigma_2$, and $\lambda > 0$.

The left figure in Figure 2-31 shows the probability of incorrect data association for different values of P_D, with $\gamma = \infty$. The right figure shows the change in the portability of incorrect data association for different gate values, γ. In these plots, the probability of detection is set to $P_D = 0.9$.

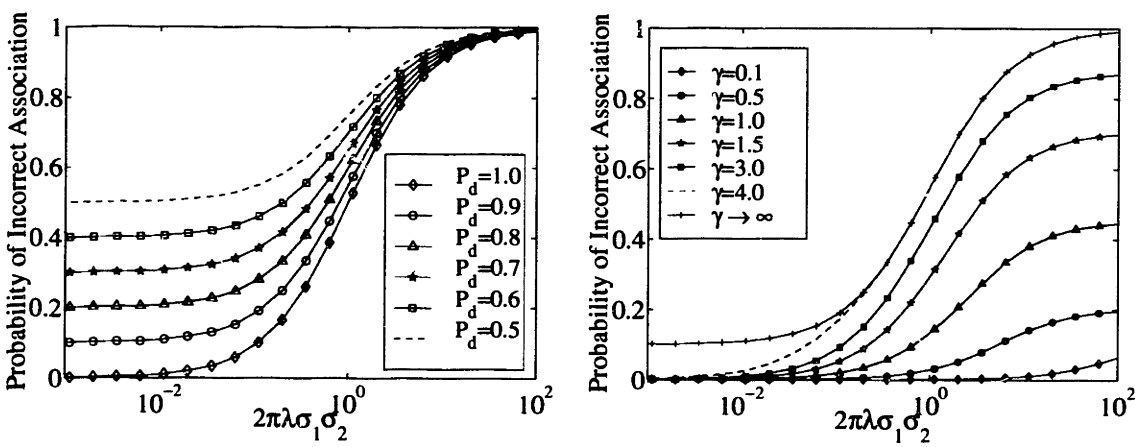

Figure 2-31: **Left:** Probability of incorrect data association for $\gamma = \infty$ for different probability of detections, P_D. **Right:** Probability of incorrect data association for $P_D = 0.9$ for different gate parameters γ.

Figure 2-31 makes explicit the performance of the data association technique employed in this thesis as a function of gate size, γ, probability of detection, P_D, and the expected number of extraneous feature measurements falling within a standard deviation ellipse of the measurement uncertainty in two dimensions, $2\pi\lambda\sigma_1\sigma_2$. Clearly, choosing a small gate size, decreases the probability of incorrect associations. This, however, comes at the expense of decreasing the probability of making *correct* associations as well. Thus, one should choose γ as large as possible in order to increase the

probability of correct associations. However, if $2\pi\lambda\sigma_1\sigma_2 \approx 1$, a gate size much greater than 1 would for most systems cause too many incorrect associations to take place. Further, the advantage of higher probability of detection is most prominent for small values of $2\pi\lambda\sigma_1\sigma_2$. This suggesting that efforts concentrated on reducing $2\pi\lambda\sigma_1\sigma_2$ is more fruitful than one focused on increasing P_D.

Chapter 3

Decoupled Stochastic Mapping

This chapter focuses on the map scaling problem, presenting new, computationally efficient solutions to the map-scaling problem that combine a decoupled map representation with consistent methods for transferring vehicle location information from one submap to another.

3.1 Introduction

A key stumbling block in the development and implementation of new methods for CML has been the issue of computational complexity, both in CPU time and memory requirements. Current state-of-the-art approaches for CML incur a high computational penalty as the size or complexity of the operating environment is increased. In the case of a stochastic mapping approach to CML, map scaling is the primary cause of complexity. Map scaling refers to the problems encountered in maintaining an ever-growing number of vehicle-to-feature and feature-to-feature correlations as the size of the operating environment increases [68].

The optimal algorithm that retains all correlations [110] encounters an $\mathcal{O}(n^3)$ computational burden, where n is proportional[1] to the number of features, which

[1] If n_s is the number of states used to describe a feature and N is the number of features, then

becomes intractable as the size of the operating environment is increased. The correlations arise because the locations of neither the vehicle nor the features are ever known precisely. If a mobile robot uses an observation of an imprecisely known target to update its position, the resulting vehicle position estimate becomes correlated with the feature location estimate. Likewise, correlations are introduced if an observation taken from an imprecisely known position is used to update the location estimate of a feature in the map. In fact, it can be shown that the correlation coefficient between features will approach unity, that is, perfect correlation, in the limit as the number of observations goes to infinity. Thus, methods which neglect these correlations miss one of the central properties of the map, and will result in failure of the algorithm in the long-term. The failure of simple strategies that ignore the correlations (such as [68]) has been demonstrated in simulations by Uhlmann *et al.* [124] and in experiments by Castellanos *et al.* [26].

An alternative is to attempt to bound the correlations. Uhlmann and colleagues introduced a tool for CML state estimation called covariance intersection (CI) that produces consistent error estimates in situations where correlation information between states is unknown [124]. The methods for CML implemented with CI maintain conservative estimates, but are too conservative because they *always* ignore the correlations, thus causing an unbounded growth of uncertainty for large-scale long-duration missions [123]. The idea behind our work is to develop a strategy that can maintain correlations in submap regions, but ignore correlations across submaps in a "safe" way such that consistent error bounds are maintained.

The objective of decoupled stochastic mapping (DSM) is to circumvent the $\mathcal{O}(n^3)$ computational burden and the $\mathcal{O}(n^2)$ memory requirement of conventional stochastic mapping [111] while providing consistent, globally-referenced error bounds and achieving performance that is comparable to full covariance stochastic mapping. The key ideas behind DSM are to use a decoupled map representation, consisting of multi-

---

$n = n_s N$.

ple globally-referenced submaps, and to employ *consistent* methods for transitioning vehicle position information between submaps. The DSM approach results in a constant upper bound for the computational requirements, both in CPU time and in memory requirements, while the storage requirements grow with the number of features mapped.

Several previous researchers have used multiple local maps in a CML algorithm. Betgé-Brezetz *et al.* [17] used multiple local maps to isolate odometry errors. This was an extension of "relocation-fusion", which is a suboptimal stochastic mapping method introduced by Moutarlier and Chatila to reduce susceptibility to odometric errors for land robots [90]. Chong and Kleeman used multiple local maps to address the issue of map scaling for indoor mobile robots using a novel sonar array that can identify plane and corner features from a single vantage point [29]. When the vehicle enters a new map region, relocation of the vehicle between maps is accomplished by building a new map and matching that map to a previously stored sub-map, using a matching technique developed exclusively for the advanced sonar sensor employed [29]. Our work is different because all the submaps in our DSM approach are *globally-referenced.* Consistent, globally-referenced error bounds are maintained by carefully managing the transitions between submaps. This enables *global* reduction of uncertainty in a submap upon re-entry to the map, which is fundamentally different from the work of Chong and Kleeman in which the global error of a submap can never decrease below its initial global uncertainty. Further comments regarding what we believe are the important differences between our work and Chong and Kleeman's work are provided in Section 3.6.

As discussed in Chapter 2, the stochastic mapping approach assumes that distinctive features of the environment can be extracted from sensor data and represented as points in an appropriate parameter space. Other types of representations are possible and have been employed with success in land and marine robot systems. For example, Thrun [119] has demonstrated highly successful navigation of indoor mobile robots

using a combination of grid-based [37] and topological [62] modeling. Stewart [113] and Singh [103, 104] have successfully employed grid-based modeling techniques for marine sensor data fusion. Our hypothesis is that in many marine environments, salient features can be found and reliably extracted using state-of-the-art processing techniques [24, 80], enabling a feature-based approach to CML [109]. Feature extraction is not addressed in this chapter; our simulations assume that measurements of point features are provided and that clutter and drop-outs are present.

The structure of this chapter is as follows. Section 3.2 summarizes the theory of augmented stochastic mapping (ASM) developed in Chapter 2. ASM is a full covariance stochastic mapping algorithm which incorporates nearest-neighbor data association and logic-based initiation and deletion of features and is performed in each submap. Section 3.3 describes decoupled stochastic mapping, in which the environment is represented in terms of multiple overlapping submaps. Section 3.3.1 describes the cross-map vehicle relocation strategy for transitioning between submaps, and Section 3.3.2 describes the cross-map vehicle updating map transition technique. These two methods transfer vehicle position estimate information across submaps while maintaining consistent error bounds. Section 3.3.3 uses the cross-map vehicle relocation method to describe single-pass DSM, which is a constant-time algorithm for large-scale CML. Single-pass DSM exhibits small, linearly growing errors with distance from the starting point. Section 3.3.4 describes multi-pass DSM, which uses a combination of the two map transition strategies to enable robust and consistent reductions of global errors between submaps to achieve a performance bound that is comparable to the result obtained using full covariance stochastic mapping. Section 3.4 presents simulation results of full covariance stochastic mapping, single-pass DSM and multi-pass DSM, as well as a comparison of the two new approaches for large-scale, long-duration missions. Section 3.6 summarizes our results, and discusses open issues for future research. Section 3.7 gives a brief chapter summary. Appendix 3.A provides a mathematical justification of the consistency of the cross-map

updating submap transition technique.

3.2 Augmented stochastic mapping

The kernel used for performing CML is the technique of augmented stochastic mapping (ASM) developed in Chapter 2. ASM is a feature-based approach to CML based on stochastic mapping. Stochastic mapping is a particular way of organizing the states in an extended Kalman filter in order to perform CML (see Appendix 2.A) and is described in detail in Section 2.3.

In ASM, as detailed in Chapter 2, a single estimated state vector $\mathbf{x}_{k|j} = [\mathbf{x}_r^T \; \mathbf{x}_1^T \ldots \mathbf{x}_N^T]_{k|j}^T$ is use to represent the robot's state, $\mathbf{x}_r$, and all the feature states, $\mathbf{x}_1 \ldots \mathbf{x}_N$ at time index k, given all information up to time index j. The estimated error covariance takes the form

$$\mathbf{P}_{k|k} = \begin{bmatrix} \mathbf{P}_{rr} & \mathbf{P}_{r1} & \cdots & \mathbf{P}_{rN} \\ \mathbf{P}_{1r} & \mathbf{P}_{11} & \cdots & \mathbf{P}_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{Nr} & \mathbf{P}_{N1} & \cdots & \mathbf{P}_{NN} \end{bmatrix}_{k|k} = \begin{bmatrix} \mathbf{P}_{rr} & \mathbf{P}_{rf} \\ \mathbf{P}_{fr} & \mathbf{P}_{ff} \end{bmatrix}_{k|k}. \tag{3.1}$$

The vehicle's estimated state $\mathbf{x}_r = [x_r \; y_r \; \phi \; v]^T$ represents the vehicle's east position, north position, heading, and speed, respectively. The estimated state of feature i is represented by $\mathbf{x}_i = [x_i \; y_i]^T$. The dynamic model used in the algorithm for modeling an AUV is described in Section 2.2. In short, the dynamic model for the AUV is given by

$$\mathbf{X}_{k+1} = \mathbf{f}(\mathbf{X}_k, \mathbf{u}_k) + \mathbf{G}(\mathbf{u}_k)\mathbf{d}_\mathbf{x}, \tag{3.2}$$

where, $\mathbf{f}$ and $\mathbf{G}$ are defined by Equation (2.3) and (2.4), respectively, and $\mathbf{X}$ is the *true* state. The control input $\mathbf{u}$ to the vehicle is given by a change in heading, $\delta\phi$, and

speed, δv, of the vehicle to model changes in rudder angle and forward thrust, that is, $\mathbf{u}_k = [\delta\phi\ \delta v]^T$. The stochastic process $\mathbf{d_x}$ is white and Gaussian and independent of $\mathbf{x}_0$.

Observations are made by range and bearing measurements of features relative to the vehicle's state described by the observation model $\mathbf{h}$ (see Section 2.2), that is

$$\mathbf{z}_k = \mathbf{h}(\mathbf{X}_k) + \mathbf{d_z}, \tag{3.3}$$

where $\mathbf{d_z}$ is a white, Gaussian, and independent of $\mathbf{x}_0$ and $\mathbf{d_x}$.

Given the dynamic model, $\mathbf{f}$, and observation model, $\mathbf{h}$, stochastic mapping is employed to estimate the state $\mathbf{x}$ and covariance $\mathbf{P}$ given the measurements. Track[2] initiation and data association are performed by use of a delayed nearest neighbor track initiator and a gated nearest neighbor data association filter as described in Section 2.4. In this chapter, we only utilize a track deleter that checks for consistency among tracks, as described in Section 2.4.

In Chapter 2 we showed that the ASM algorithm produces consistent, long-term bounded error for navigation in unknown, unmapped environments. With reasonable modeling assumptions about sensor and vehicle performance, including, false returns, drop-outs and the presence of an unknown current, positioning errors of a few meters were continuously achieved during missions extending over five hours in duration. However, in Chapter 2, only 50 features were present in the 1.2 km by 1.2 km survey area, so that the computations could be performed with Matlab on a 233 MHz Pentium processor faster than real time. If the number of features is increased to more than about 100 features, the real-time performance criteria cannot be met by any currently off-the-shelf personal computer. This computational difficulty is the primary motivation for the development of a decoupled approach to CML.

[2]A *track* is a a feature estimate.

3.3 Decoupled mapping

In decoupled stochastic mapping, we break the global map into multiple, globally-referenced submaps. Each submap has its own vehicle position estimate, a set of feature estimates and a corresponding estimated covariance matrix produced by ASM. In this chapter, the physical size of all submaps are the same, and their physical location is implicitly determined *a priori*. More advanced, adaptive strategies for map location and sizing could be utilized. In the strategy employed here, a new submap is created and initialized when the vehicle moves out of the current submap. The submaps overlap slightly in order to not cause excessive map switching. It should be noted that a submap will only be dependent on the maps created before it and not on the ones created after it.

The idea of creating a new, globally referenced submap when the computational cost of the current map has become sufficiently high is a simple one. The first challenge is how to be able to reuse old submaps and thus be able to move between maps in a consistent manner. The greater challenge is how to used the information gained from mapping one submap when re-entering an old submap in a consistent manner in which some of the new information is passed on and exploited. This is a subtle but extremely important point. If submap A has been generated and it is decided that one needs to generate a new submap B, the states of the vehicle as well as all the features being mapped in B will be *dependent* on the state of the vehicle when submap B was created, and thus, in turn, dependent on submap A and the vehicle state in submap A. Thus, submaps A and B become *dependent*. Thus, any attempt to move between the two maps and use the information gained by being in one submap to increase the certainty of the other submap must be made with caution so that already existing (dependent) information is not reused, thus making the system over confident. We have developed two methods for moving between maps. Cross-map vehicle relocation is used for consistently utilizing an old map, and bears some similarity to the submap relocation strategy of Chong and Kleeman [29]. The second

method, cross-map vehicle updating, can use the information gained in one submap to improve the state estimates in the second submap.

Next, the ideas and development of cross-map vehicle relocation and cross map vehicle updating are presented. Then, the use of these techniques in order to perform single-pass and multi-pass DSM is described.

3.3.1 Cross-map vehicle relocation

The purpose of cross-map vehicle relocation is to enable a vehicle to retrieve a submap of an already mapped region. In this step, there is only the requirement that this is done conservatively, and not that information for improving the accuracy of the vehicle estimate is passed between the submaps. To make the discussion more concrete, let us assume that the vehicle is currently in submap A and is entering submap B, which already exists. The state estimate of the vehicle and all the features of submap A are described by $\mathbf{x}^A$ and covariance by $\mathbf{P}^A$, in analogy with Equation (3.1):

$$\mathbf{P}^A_{k|k} = \begin{bmatrix} \mathbf{P}^A_{rr} & \mathbf{P}^A_{r1} & \cdots & \mathbf{P}^A_{rN} \\ \mathbf{P}^A_{1r} & \mathbf{P}^A_{11} & \cdots & \mathbf{P}^A_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}^A_{Nr} & \mathbf{P}^A_{N1} & \cdots & \mathbf{P}^A_{NN} \end{bmatrix}_{k|k} = \begin{bmatrix} \mathbf{P}^A_{rr} & \mathbf{P}^A_{rf} \\ \mathbf{P}^A_{fr} & \mathbf{P}^A_{ff} \end{bmatrix}_{k|k}. \tag{3.4}$$

Similarly, let $\mathbf{x}^B$ and $\mathbf{P}^B$ define submap B. In performing relocation upon entry into submap B, the vehicle state estimate from submap B, $\mathbf{x}^B_r$, is the estimate at the time the vehicle left submap B during the last visit to this submap. Thus, this vehicle state estimate is not of interest as it no longer estimates the physical location of the vehicle. Further, the feature states of submap A, $\mathbf{x}^A_f$, are not of interest when operating in submap B as these estimates are per definition not maintained in submap B. The estimates that *do* describe the physical quantities of interest upon re-entry into submap B are the vehicle's state estimate of submap A, $\mathbf{x}^A_r$, and the feature

estimates of submap B, $\mathbf{x}_r^B$. Thus, our new state estimate upon re-entering submap B from submap A is given by

$$\mathsf{x}^B = \begin{bmatrix} \mathbf{x}_r^A \\ \mathbf{x}_f^B \end{bmatrix}. \tag{3.5}$$

The san serif font, x, is deliberately used to distinguish the state of submap B *before* relocation, $\mathbf{x}^B$, and the state of submap B, x^B, *after* the relocation. The same font selection will be used to distinguish the covariances, $\mathbf{P}$, before a relocation or update step and the covariance, P, after a relocation or update step. This notation is chosen as the estimates both exist at the same point in mission time.

In order to ensure the consistency upon re-entry into submap B, our estimate of the covariance after the relocation step, P^B, must be greater or equal to the *true* covariance of the state, x^B, of submap B after relocation. That is[3] ,

$$\mathsf{P}^B \geq C(\mathsf{x}^B, \mathsf{x}^B) \Rightarrow \begin{cases} \mathsf{P}_{rr}^B \geq C(\mathsf{x}_r^B, \mathsf{x}_r^B) & ① \\ \mathsf{P}_{ff}^B \geq C(\mathsf{x}_f^B, \mathsf{x}_f^B) & ② \\ \mathsf{P}_{ff}^B - \mathsf{P}_{fr}^B {\mathsf{P}_{rr}^B}^{-1} \mathsf{P}_{rf}^B \geq \mathbf{0} & ③ \end{cases} \tag{3.6}$$

where C defines the covariance operator defined by

$$C(\mathbf{x}, \mathbf{x}) = E[(\mathbf{x} - \mathbf{X})(\mathbf{x} - \mathbf{X})^T], \tag{3.7}$$

where E is the expectation operator. The (i, j) element of this matrix is defined by

$$E[(x_i - X_i)(x_j - X_j)] = \int_{-\infty}^{\infty} p(X_i, X_j)(x_i - X_i)(x_j - X_j) dX_i dX_j, \tag{3.8}$$

where $p(X_i, X_j)$ is the joint probability of states X_i and X_j and where, as in Equa-

[3] By $>$ for matrixes we mean in the positive definite sense. That is, for any matrixes $\mathbf{A}$ and $\mathbf{B}$, if $\mathbf{A} > \mathbf{B}$ then $\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{x}^T\mathbf{B}\mathbf{x} > 0, \forall\, \mathbf{x}$.

tion (2.6), capital letters define true states and small letters represents estimated states.

Returning to Equation (3.6), from the properties of stochastic mapping and submap A we know that constraint ① is met if

$$\mathsf{P}^B_{rr} \geq \mathbf{P}^A_{rr} \text{ since } \mathbf{P}^A_{rr} \geq C(\mathbf{x}^A_r, \mathbf{x}^A_r) = C(\mathbf{x}^B_r, \mathbf{x}^B_r). \tag{3.9}$$

From submap B we know that constraint ② is met if

$$\mathsf{P}^B_{ff} \geq \mathbf{P}^B_{ff} \text{ since } \mathbf{P}^B_{ff} \geq C(\mathbf{x}^B_f, \mathbf{x}^B_f) = C(\mathbf{x}^B_f, \mathbf{x}^B_f). \tag{3.10}$$

However, we have not gained or lossed any information about the features, thus the inequality in Equation (3.10) must be an equality, that is

$$\mathsf{P}^B_{ff} = \mathbf{P}^B_{ff}. \tag{3.11}$$

The last constraint, ③, from Equation (3.6) places a constraint on the relative values of P^B_{rr}, P^B_{ff} and P^B_{rf}. We know from the properties of Kalman filtering, and from stochastic mapping in particular, that

$$\mathbf{P}^B_{ff} - \mathbf{P}^B_{fr}\mathbf{P}^{B^{-1}}_{rr}\mathbf{P}^B_{rf} \geq \mathbf{0}, \tag{3.12}$$

is assured as required by ③. From a geometric point of view, we do not have the freedom to change $\mathbf{P}^B_{rf}$ in Equation (3.12) as this will "squeeze" the total error ellipse formed by $\mathbf{P}^B$ along the hyper planes formed by the vectors $\mathbf{x}^B_r$ and $\mathbf{x}^B_f$, which will cause non-conservative estimates. Thus, we do not have the freedom to change $\mathbf{P}^B_{rf}$ and

$$\mathsf{P}^B_{rf} \leftarrow \mathbf{P}^B_{rf}. \tag{3.13}$$

Further, given Equation (3.11), the only freedom we are left with in constraint ③ of Equation (3.6) is in the choice of P^B_{rr}. We note that the matrix on the left side of constraint ③,

$$\mathsf{P}^B_{ff} - \mathsf{P}^B_{fr}\mathsf{P}^{B\,-1}_{rr}\mathsf{P}^B_{rf}, \tag{3.14}$$

has a physical interpretation — it is the residual error in the location of the features if a perfect measurement of the vehicle location became available.[4] As the vehicle and feature locations become more correlated, the size of this matrix decreases. In the limit as the vehicle and feature locations become completely correlated, the matrix of Equation (3.14) decreases to zero.[5] Since the time when the vehicle left submap B, the degree of correlation between the vehicle and the features of submap B cannot increase. New process noise affecting the vehicle will decrease the extent to which improved knowledge of the vehicle location can imply improved knowledge of the feature locations. Hence, when the vehicle re-enters submap B, the size of the matrix in Equation (3.14) must be increased, that is

$$\mathsf{P}^B_{ff} - \mathsf{P}^B_{fr}\mathsf{P}^{B\,-1}_{rr}\mathsf{P}^B_{rf} > \mathbf{P}^B_{ff} - \mathbf{P}^B_{fr}\mathbf{P}^{B\,-1}_{rr}\mathbf{P}^B_{rf} \tag{3.15}$$

Using Equation (3.11) and Equation (3.13), Equation (3.15) becomes

$$\begin{aligned} \mathsf{P}^B_{fr}\mathsf{P}^{B\,-1}_{rr}\mathsf{P}^B_{rf} &< \mathbf{P}^B_{fr}\mathbf{P}^{B\,-1}_{rr}\mathbf{P}^B_{rf} \\ \mathsf{P}^B_{rr} &> \mathbf{P}^B_{rr}. \end{aligned} \tag{3.16}$$

We note that if we add any positive definite matrix to $\mathbf{P}^B_{rr}$ in Equation (3.12), the greater or equal sign will become a greater than sign. Thus, all constraints are met

[4] This can be seen by applying the Kalman filter equations with a measurement of vehicle location with $\mathbf{R}_{k+1} = 0$ from the notation of Appendix 3.A.

[5] Perfect knowledge of the vehicle location would imply perfect knowledge of the feature locations.

if we let

$$\mathbf{P}^B_{rr} \leftarrow \mathbf{P}^B_{rr} + \mathbf{P}^A_{rr}. \tag{3.17}$$

The cross-map relocation step for moving from submap A to submap B can be summarized as

$$\mathbf{x}^B \leftarrow \begin{bmatrix} \mathbf{x}^A_r \\ \mathbf{x}^B_f \end{bmatrix}, \quad \mathbf{P}^B \leftarrow \begin{bmatrix} \mathbf{P}^A_{rr} + \mathbf{P}^B_{rr} & \mathbf{P}^B_{rf} \\ \mathbf{P}^B_{fr} & \mathbf{P}^B_{ff} \end{bmatrix}.$$

3.3.2 Cross-map vehicle updating

The goal of cross-map updating is to obtain an improved vehicle state estimate using vehicle state information from a neighboring submap, while maintaining a conservative error covariance for the submap. Given a more accurate vehicle location estimate, $\mathbf{x}^A_r$, from a neighboring submap that was created earlier than submap B, the question is, how can we safely transfer the vehicle location information into submap B? To explain our strategy, it is useful to first consider several strategies which are unsuccessful for one reason or another.

The first idea one might consider is to simply replace the vehicle state estimate for submap B with the state estimate from submap A, $\mathbf{x}^B_r \leftarrow \mathbf{x}^A_r$, and to replace the corresponding vehicle covariance submatrix, $\mathbf{P}^B_{rr} \leftarrow \mathbf{P}^A_{rr}$. The problem with this approach is that the submap B covariance matrix, $\mathbf{P}^B$, can no longer be guaranteed to be positive definite; such a replacement violates the physical meaning of a covariance matrix and thus violates the consistency of the submap.

A second idea that overcomes this would be to use the machinery of the Kalman filter to perform the update — that is, to use the vehicle location estimate from submap A as a measurement of the vehicle location in a Kalman filter that uses

the vehicle covariance in submap A, $\mathbf{P}_{rr}^{A}$, as the assumed measurement noise in the Kalman filter update equations for submap B. While this strategy maintains positive semi-definiteness, it violates the independence assumptions of the Kalman filter and produces overconfident estimates of both the vehicle locations and the features. Because the vehicle location estimates $\mathbf{x}_r^A$ and $\mathbf{x}_r^B$ are not independent, using $\mathbf{x}_r^A$ as a measurement of the vehicle position results in some information being used twice. In addition, due to the correlations between the vehicle and the features in submap B, performing a vehicle update measurement in this way also updates the locations of the feature states in submap B, and greatly reduces their error covariances, even though no new information about the locations of these features has been obtained. In effect, a "mistaken update" is made to the feature locations. Hence consistency is lost for both the vehicle location estimate and the feature location estimates.

To amend this, two additional steps are taken. We can negate the effect of using vehicle location information twice by "throwing away" the old vehicle estimate before the update. This is done by setting $\mathbf{x}_r^B$ to a completely random position within the submap, and inflating the vehicle covariance $\mathbf{P}_{rr}^{B}$ by a vast amount, equivalent to the vehicle being anywhere in the entire global map. After the Kalman update, the resulting vehicle state estimate is effectively replaced by $\mathbf{x}_r^B$, its covariance becomes $\mathbf{P}_{rr}^{B}$ and the correlation terms $\mathbf{P}_{rf}^{B}$ between the vehicle and features are greatly reduced. In effect, the great inflation of the vehicle state estimate before the update serves to "de-correlate" the vehicle estimate in submap B from its feature location estimates.

We are still left, however, with the fact that the map B feature states have been updated and the corresponding features covariances greatly reduced, despite the addition of no new information about their locations. To rectify this, we simply add the *a priori* map B feature covariance $\mathbf{P}_{ff}^{B}$ to the *a posteriori* feature covariance after the Kalman update. We know that the resulting covariance must be positive semi-definite, because the Kalman filter always yields a positive definite matrix, and adding a positive semi-definite matrix to any submatrix of a positive semi-definite

matrix always yields a positive semi-definite matrix. In addition, we know that the resulting feature covariance matrix must be consistent, that is, it must bound the actual feature location estimates. We know this because the residual covariance after the cross-map Kalman update bounds the amount of the "mistaken update" of the feature locations that occurs during the Kalman update, and the *a priori* feature covariance bounded the feature location estimates before the update. Appendix 3.A provides a more detailed mathematical justification. Because the Kalman gain is independent of $\mathbf{P}^B_{ff}$, it is equivalent to multiply $\mathbf{P}^B_{ff}$ by two before the Kalman update.

This approach is similar to what occurs during cross-map vehicle relocation: optimality is sacrificed, but consistency is maintained. In this way, improved vehicle location information is transferred from submap A to submap B, and subsequent measurements of the features in submap B allow the accuracy of submap B to be improved. The cross-map vehicle update step is summarized in Figure 3-1.

Having two tools for passing between submaps, cross-map vehicle relocation and cross-map vehicle updating, we now move on to define two concurrent mapping and localization algorithms that utilize these methods. The first one is single-pass decoupled stochastic mapping, which is suitable for missions in which maximum accuracy is desired after only a single transit through the environment. The second method is labeled multi-pass decoupled stochastic mapping. Multi-pass DSM is more suitable when the vehicle will be able to make repeated passes through its environment, improving the accuracy of its map with each pass.

① de-correlate

$$\mathbf{x}^B \leftarrow \begin{bmatrix} \boldsymbol{\phi}^B \\ \mathbf{x}_f^B \end{bmatrix}, \quad \mathbf{P}^B \leftarrow \begin{bmatrix} \mathbf{P}_{rr}^B + \Phi^B & \mathbf{P}_{rf}^B \\ \mathbf{P}_{fr}^B & 2\mathbf{P}_{ff}^B \end{bmatrix}$$

where $\boldsymbol{\phi}^B = \text{random}(\text{map}_\text{B})$, $\Phi^B >> \text{size}(\text{map}_\text{B}) >> \mathbf{P}_{rr}^B$

② update

$$\mathbf{z} = \mathbf{H}\mathbf{x}^B + \mathbf{v}, \quad \mathbf{H} = [\mathbf{I}\ \mathbf{0}]$$
$$\mathbf{z} \leftarrow \mathbf{x}_{rr}^A, \quad C(\mathbf{v},\mathbf{v}) \leftarrow \mathbf{P}_{rr}^A$$

$$\begin{aligned} \mathbf{K} &= \mathbf{P}^B\mathbf{H}^T(\mathbf{H}\mathbf{P}^B\mathbf{H}^T + \mathbf{P}_{rr}^A)^{-1} \\ \mathbf{x}^B &\leftarrow \mathbf{x}^B + \mathbf{K}(\mathbf{z} - \mathbf{H}\mathbf{x}^B) \\ \mathbf{P}^B &\leftarrow (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}^B(\mathbf{I} - \mathbf{K}\mathbf{H})^T + \mathbf{K}C(\mathbf{v},\mathbf{v})\mathbf{K}^T \end{aligned}$$

Figure 3-1: Summary of cross-map updating for a vehicle moving from submap A to submap B. Submap A was created before submap B. An EKF update of the vehicle position in submap B using the vehicle location from submap A as a measurement is performed in the last three equations. By "random(map_B)", we mean a random value from all possible values the vehicle can take in submap B. By "size(map_B)" we mean the square of the maximum value that any of the vehicle states can have in submap B.

3.3.3 Single-pass DSM algorithm

The purpose of single-pass decoupled stochastic mapping (DSM) is to perform concurrent mapping and localization as accurately as possible the *first* pass through the region. For this reason, we utilize and create dependencies between features in adjacent maps. This causes a slower growth in the uncertainties, but prevents us from doing a cross-map vehicle update and *all* traverses between submaps are done through cross-map vehicle relocation. Thus, we are unable to reduce the uncertainties of a submap below the uncertainty that it had when it was created.

The important difference between single-pass DSM and multiple-pass DSM lies in the way submaps are created. In both methods, a new map is created when the vehicle leaves the region of a submap and moves into unexplored territory. However, in single-pass DSM, the vehicle state as well as a set of features, called *correspondence features*, are used to initialize the new map. These features are chosen because they have a very high likelihood of being re-observed at some time during the vehicle's time spent in the new map. Thus, because these features are known more accurately, the vehicle is able to produce a more accurate map than without the existence of these features. As a result, the uncertainty grows more slowly than if these features were not included when creating a submap.

To be more precise, let us assume that we are moving out of submap A and are about to create submap B. From submap A we have $\mathbf{P}^A$ and $\mathbf{x}^A$ which, for clarity, can be separated into the states and estimate for the vehicle, denoted by subscript r, correspondence features, denoted by subscript F, and the remaining features, denoted by subscript f. That is,

$$\mathbf{P}^A = \begin{bmatrix} \mathbf{P}^A_{rr} & \mathbf{P}^A_{rF} & \mathbf{P}^A_{rf} \\ \mathbf{P}^A_{Fr} & \mathbf{P}^A_{FF} & \mathbf{P}^A_{Ff} \\ \mathbf{P}^A_{fr} & \mathbf{P}^A_{fF} & \mathbf{P}^A_{ff} \end{bmatrix}, \quad \mathbf{x}^A = \begin{bmatrix} \mathbf{x}^A_r \\ \mathbf{x}^A_F \\ \mathbf{x}^A_f \end{bmatrix}. \tag{3.18}$$

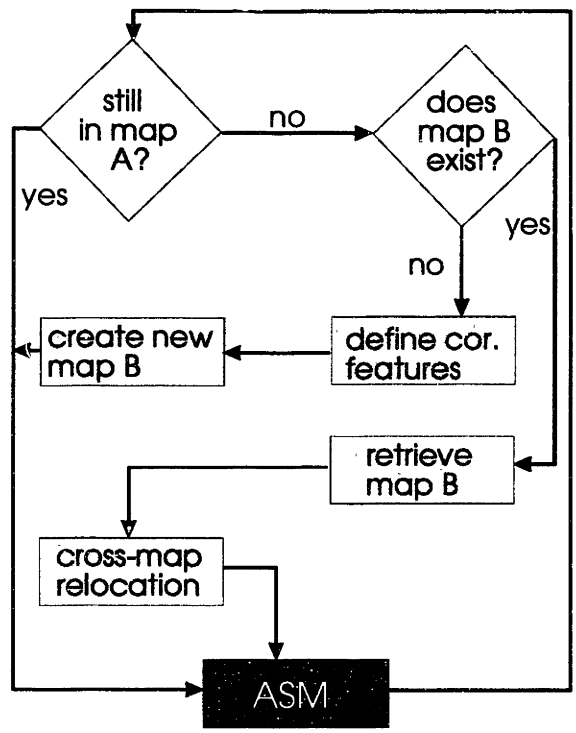

Figure 3-2: Structure of the single-pass decoupled stochastic mapping algorithm. The augmented stochastic mapping (ASM) algorithm structure is outlined in Figure 2-4.

Then submap B is created from $\mathbf{P}^A$ and $\mathbf{x}^A$ in the following way:

$$\mathbf{P}^B \leftarrow \begin{bmatrix} \mathbf{P}^A_{rr} & \mathbf{P}^A_{rF} \\ \mathbf{P}^A_{Fr} & \mathbf{P}^A_{FF} \end{bmatrix}, \quad \mathbf{x}^B \leftarrow \begin{bmatrix} \mathbf{x}^A_r \\ \mathbf{x}^A_F \end{bmatrix} \tag{3.19}$$

A block diagram of the single-pass decoupled stochastic mapping algorithm is shown in Figure 3-2.

Notice that cross-map vehicle updating is not utilized in this algorithm. If it was, the algorithm would become overconfident because of the dependent information between maps caused by the correspondence feature estimates, that is, $\mathbf{x}_F$, $\mathbf{P}_{FF}$, $\mathbf{P}_{Fr}$ and $\mathbf{P}_{rF}$. Notice, that there might be features that are mapped in adjacent submaps that are *not* correspondence features, and there might well be correspondence features that are *never* observed from the new submaps. However, the more *a priori* information one has about the environment and the properties of the sensor being

employed, the more these inefficiencies can be reduced.

3.3.4 Multi-pass DSM algorithm

The purpose of multi-pass decoupled stochastic mapping is to have a method where the global error of a submap can be reduced by revisiting the submap. For this reason, multi-pass DSM is suitable for areas that will be revisited. The reduction of global uncertainty of a submap is obtained by utilizing cross-map vehicle updating. Further, it should be noted that this can *only* be performed to a submap that was created *later* in time due to the dependence between the submaps. For the instances when the vehicle travels back to a submap that was created earlier in time, cross-map vehicle relocation is employed just as in single-pass DSM.

In order for cross-map vehicle updating to be applicable, correspondence features are not initialized. Thus, if the vehicle is moving out of submap A to create submap B, submap B is only created from the vehicle state estimate of submap A. That is, using the same notation as in Equation (3.18),

$$\mathbf{P}_{rr}^{B} \leftarrow \mathbf{P}_{rr}^{A}, \quad \mathbf{x}_{r}^{B} \leftarrow \mathbf{x}_{r}^{A}. \tag{3.20}$$

As there are no correspondence features used in multi-pass DSM, the feature states between submaps are independent in the sense that their estimates do not share any measurements. However, there will likely be features in adjacent maps that are represented in both submaps. It should be noticed that the feature states of a map will be (weakly) dependent on all earlier maps because a map is always initiated with the vehicle estimate and state from the earlier map upon creation of the later map. For this reason, cross-map vehicle updating can only be performed to a later map, and not the reverse.

It should also be noted that the multi-pass and single-pass algorithms can be used interchangeably. That is, if during a mission, one expect to revisit some areas but

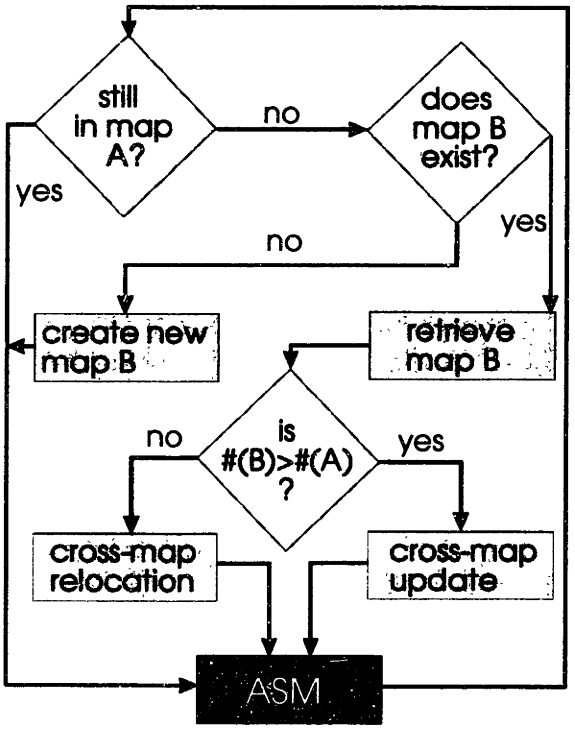

Figure 3-3: Algorithm structure for the multi-pass DSM method. In contrast with single-pass DSM, two different map transition strategies are used. Cross-map vehicle relocation is used when transitioning from higher to lower maps, and cross-map vehicle updating is used when transitioning from lower to higher maps. Also, in multi-pass DSM, the submap regions do not overlap, although different submaps can possess estimates of the same environmental features that are independent (in the sense that they do not share any measurements.) The augmented stochastic mapping (ASM) algorithm structure is outlined in Figure 2-4.

not others, the multi-pass algorithm should be used in the areas one expect to revisit while the single-pass algorithm for the areas one will not revisit.

3.4 Simulation results

To evaluate the performance of decoupled stochastic mapping for performing concurrent mapping and localization, a number of long-term, large-scale AUV missions have been simulated. In this chapter we first present a detailed analysis and comparison of single-pass and multi-pass DSM with the optimal (single map) augmented stochastic mapping (ASM) algorithm[6]. The survey area and number of features were chosen so that an ASM result could be obtained. Secondly, results are presented for the performance of decoupled stochastic mapping for very long-term missions with a large number of features. The comparison to ASM can not be made, as the computational complexity would become intractable for any of the computers we have available.

The parameters used for these simulations are summarized in Table 3.4 and are based on the characterization of an AUV such as the Odyssey II [12] equipped with a forward looking electronically scanned sonar [97]. The AUV is assumed to be given a differential GPS reset at the start of the mission, thus providing an initial position uncertainty of less than one meter. The power consumption of the AUV is not taken into consideration.

In these simulations, false measurements are generated by assuming that the number of spurious returns has a Poisson distribution with an expected value of λ. The range and angle of the spurious returns are uniformly distributed over the field of view of the sonar. The probability of detection of features is set to $P_D = 0.9$. Sonar measurements and dead reckoning measurements are obtained at 1 Hz. In these simulations, the physical boundaries of the submaps were set to be 525 by 525 meter squares. This size was chosen so that, given the feature density, the number of features in each submap would generally be less than 50. Further, they are placed so that the boundaries overlap by 25 meters. This was done to prevent excess transitions

[6]The reader is reminded that using DSM with only one map for the entire region is identical to using the ASM algorithm alone. Utilizing ASM for the entire area is the optimal estimation solution as all cross-correlations are maintained.

Table 3.1: Decoupled SM simulation parameters.

sampling period, T	1 sec.
maximum sonar range	200 m
sonar coverage angle	$\pm 40°$
range measurement standard deviation	0.5 m
bearing measurement standard deviation	5°
feature probability of detection	0.90
vehicle cruise speed	2.5 m/s
speed process standard deviation	5% of δv
heading process standard deviation	2.0°
dead reckoning speed standard deviation	0.4 m/s
dead reckoning heading standard deviation	3.0°
initial position uncertainty std. dev.	0.7 m
initial heading uncertainty std. dev.	5.0°
initial speed uncertainty std. dev.	0.2 m/s
gate parameter γ	9
clutter parameter λ	1
track initiation parameters	$M = 5, N = 4$

between submaps when the vehicle moves along the boundary between two submaps. This also prevents the generation of excess submaps for missions in which the vehicle travels close to the boundaries of the survey area.

The performance improvement of these algorithms in comparison to navigation based on dead-reckoning information alone is not included in this chapter. The significant superiority of stochastic mapping in the presence of unknown currents was documented in Chapter 2.

These simulations have been implemented using Matlab. The execution speed is about three times faster than real time for the two DSM methods and about ten times slower than real time for the optimal ASM simulation running on a Pentium 333 MHz processor.

3.4.1 DSM versus ASM

First, we consider the comparison of the two DSM algorithms with the full covariance "optimal" ASM algorithm. The goal was for the AUV to perform a survey of an unknown area of approximately 1 km by 1 km, in the presence of clutter and dropouts.

The area contained 110 features, randomly distributed over the survey area. The number of features in this survey was limited in order to obtain results from the ASM algorithm for comparison. The placement of the features and the desired path of the vehicle were the same for all three scenarios to facilitate easy comparison of the results. The desired path of the AUV and the true feature locations are shown in Figure 3-4.

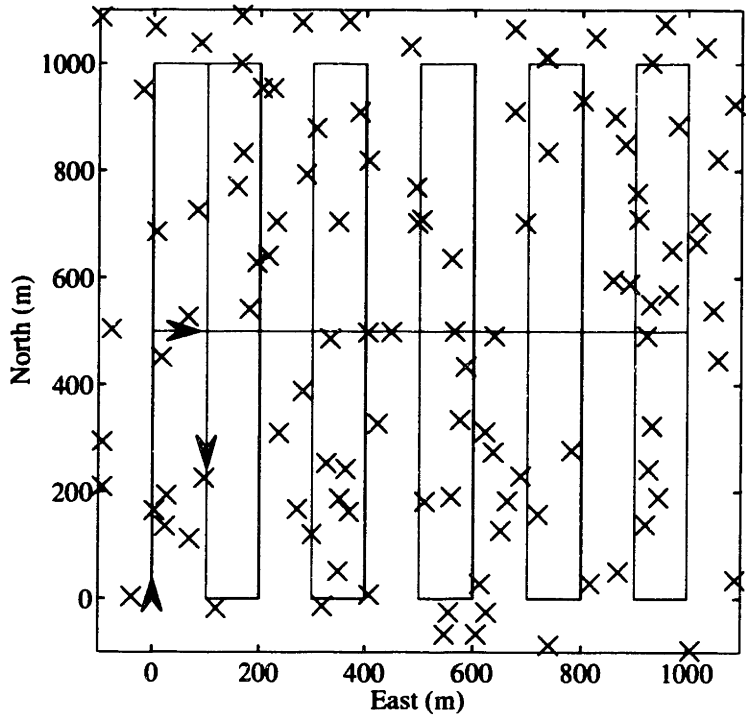

Figure 3-4: The desired survey path of the vehicle (solid line) and the location of the 110 randomly distributed point features (crosses). The vehicle starts at (0,0) meters and follows the path of the arrows.

The vehicle starts at position (0,0) meters and follows the desired path in the direction of the arrows. Thus it will first map the lower half of the area and then map the upper half. Upon completion, it returns to (0,0) and repeats the survey. A little more than four complete cycles are performed, resulting in a total mission duration of just over eight hours.

We will first present the result of this survey using the full-covariance ASM algo-

rithm, followed by the results for single-pass DSM, and finally the results for multi-pass DSM will be presented.

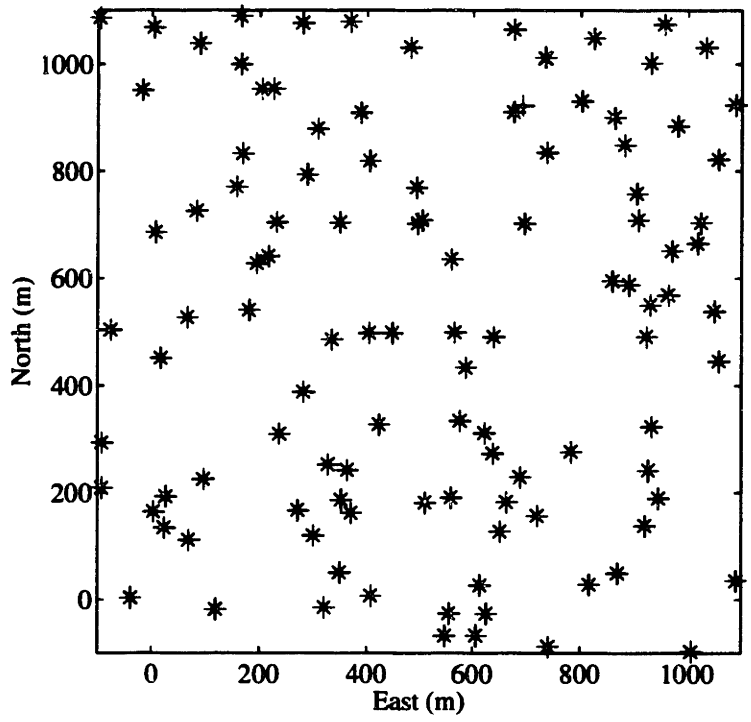

Figure 3-5: The true feature positions (marked by '×') along with the estimated feature positions (marked by '+') at the termination of the 8 hour mission using ASM.

Figure 3-5 shows the true feature positions along with the estimated feature positions at the termination of the survey using ASM. The 3σ (99% highest confidence region) for the feature initial estimates are also plotted, but are too small to show up on the plot. Figure 3-6 shows plots of the position, heading, and velocity errors of the vehicle versus time, along with the 3σ (99% highest confidence region) bounds. As the vehicle travels further from its start position, the uncertainty grows and comes to a peak at a time when just under one hour of the mission time has elapsed. This is because very few features are re-observed during this period in reaching the upper right corner of the survey area. However, as the vehicle moves from right to left in the upper half of the survey area, its uncertainty decreases as features are re-observed.

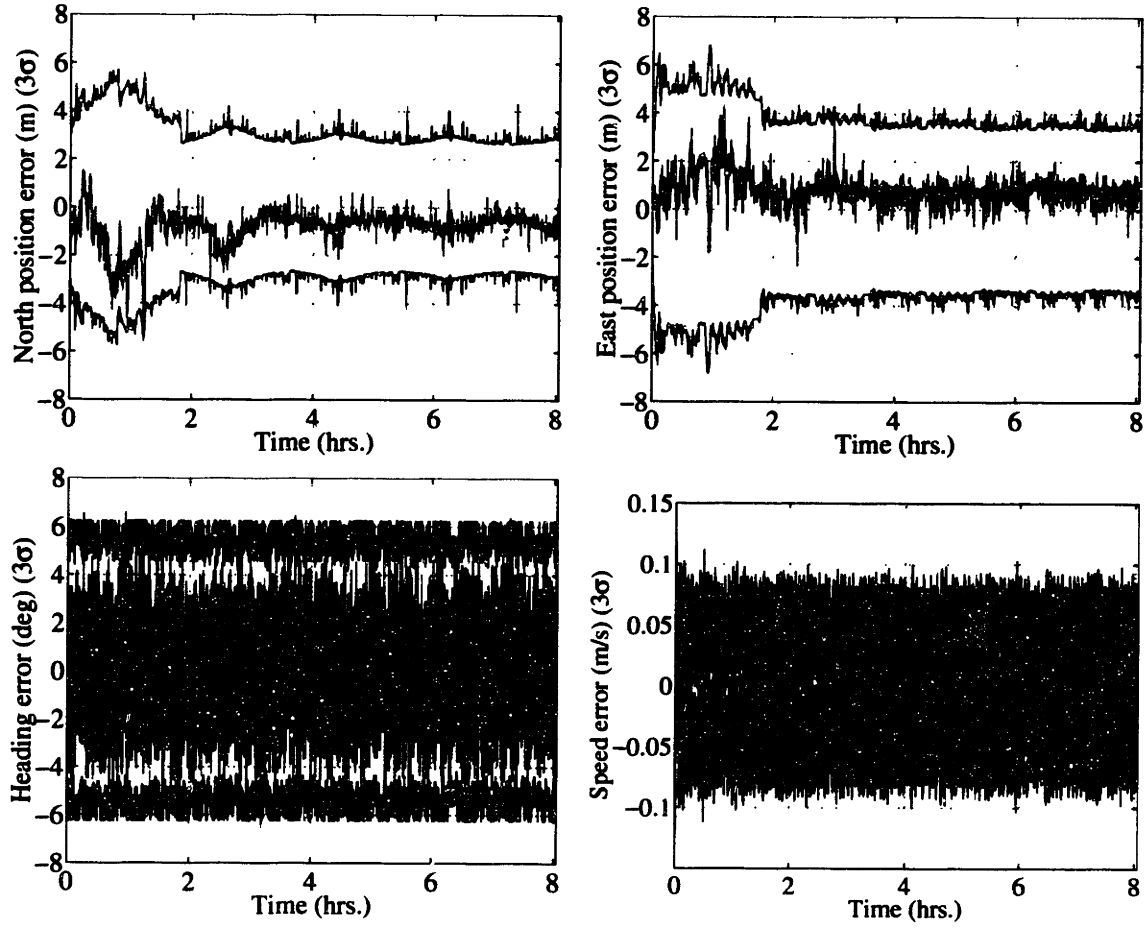

Figure 3-6: Errors and 3σ bounds (99% highest confidence bounds) for the position, heading and velocity estimates produced by performing the survey using augmented stochastic mapping (ASM) algorithm.

By the time the vehicle has completed one lap of the survey area, its uncertainty has been reduced to a "steady state" level. This occurs a little before two hours into the mission. In continuing the mission, the vehicle is, by utilizing the cross-correlations, able to obtain 3σ bounds for both its north and east coordinates of less than four meters for all later times. As the mission is continued, the slightly higher uncertainties in the north position when far from the origin will be reduced.

The 3σ bounds for the heading and speed of the AUV are bounded by 6 degrees and about 10 cm/s for all times. This results from the fact that these quantities

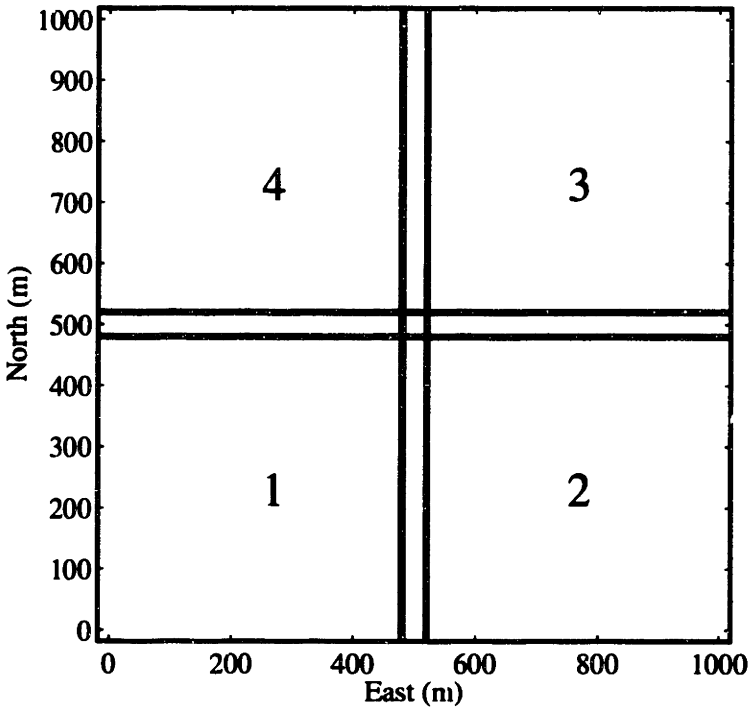

Figure 3-7: The submap partition of the survey area (shown in Figure 3-4) as generated by the two DSM algorithms. The numbers signify the order in which the maps were created.

are measured directly (and used in the dead-reckoning update), and they can be determined independently of the uncertainty of the vehicle's position.

In performing DSM, the survey area is automatically partitioned into submaps. As mentioned before, each submap bounds a 525 m by 525 m square region. After one pass of the survey path, four maps have been initialized.

Figure 3-7 shows the location of the submaps as generated by the DSM algorithm. The numbers signify the order in which submaps were created. That is, submap 1 was created before submap 2, and so on.

Figure 3-8 shows a snapshot at which the AUV is in the process of mapping submap 3. The submap boundaries, true feature locations, estimated feature locations, estimated vehicle position, and the estimated position of the correspondence features are marked. In addition, the vehicle state estimates corresponding to the

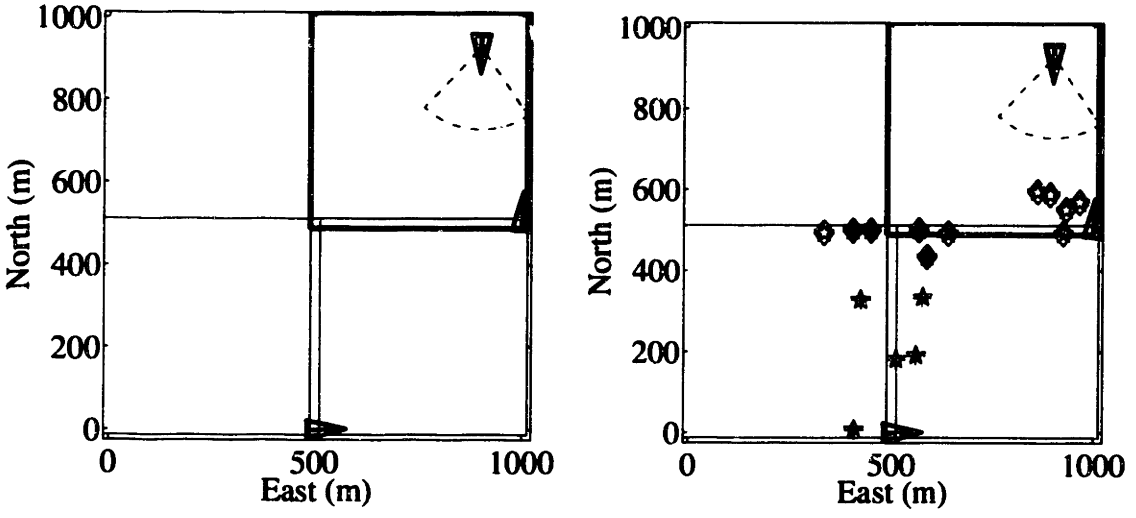

Figure 3-8: **Left:** The plot shows the true position of all features (marked by '×') along with all feature estimates (marked by '+') and 3σ error ellipses that exist at the current time (too small to be seen). The vehicle is marked as a triangle and the sonar coverage is shown by a dashed dotted line. Three submaps have been created, and the AUV is in the process of mapping submap 3. **Right:** This plot shows the correspondence features of the submaps used in the creation of submaps in single-pass DSM. Correspondence features are marked in both the original submap and in the submap that was generated from them. The correspondence features are marked by plus signs, stars and diamonds for submaps 1, 2 and 3, respectively. The triangles in submaps 1 and 2 show the estimated state of the vehicle when it exited these submaps.

locations where the vehicle exited submaps 1 and 2 are shown. Note that since no information is passed between submaps regarding correspondence features after the creation of a submap, each submap will get slightly different estimates for the correspondence features. As the correspondence features between two submaps are known to model the *same physical objects*, they can potentially be used to reduce the *global* uncertainty of submaps. At this time, we have not attempted to utilize this information.

Figure 3-9 shows plots of the position, heading, and velocity errors of the vehicle versus time, along with the 3σ (99% highest confidence region) bounds for single-

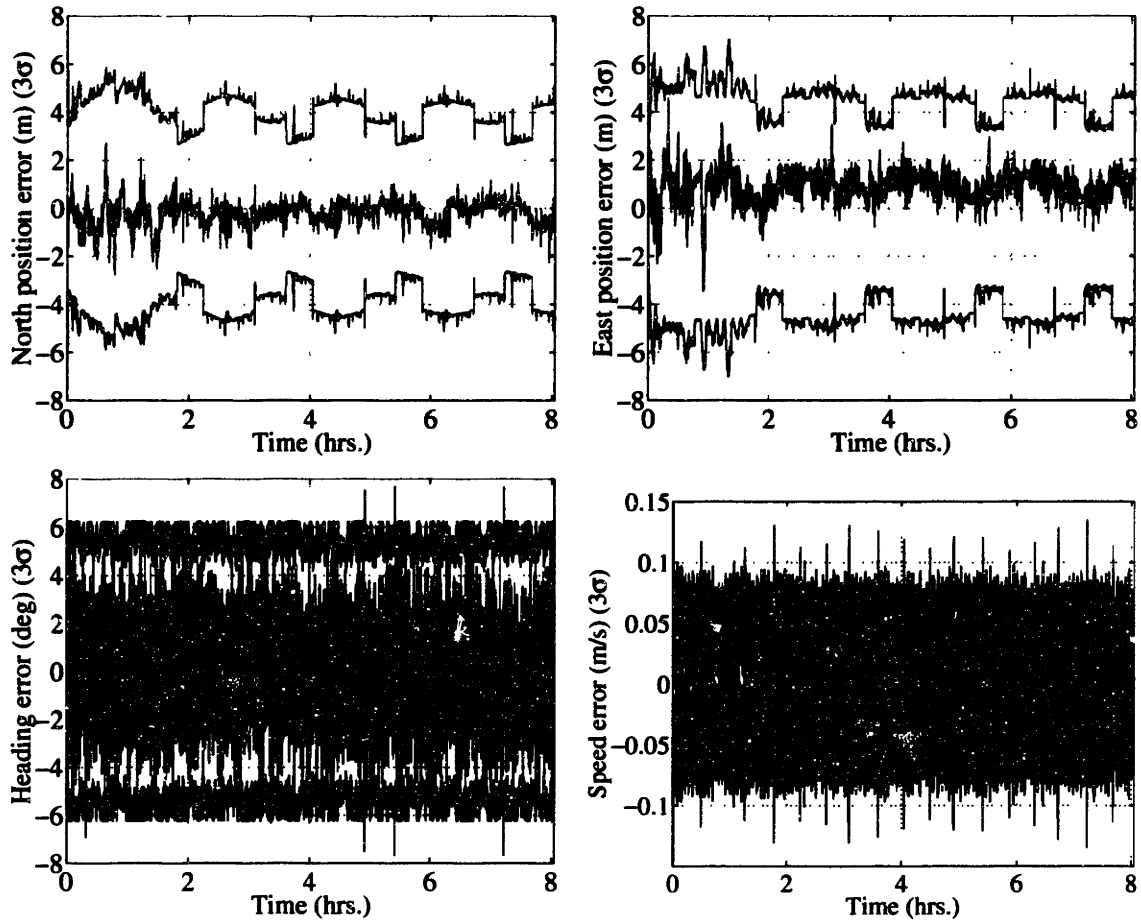

Figure 3-9: Errors and 3σ bounds (99% highest confidence bounds) for the position, heading and velocity estimates produced by single-pass DSM.

pass DSM. These plots should be compared to the equivalent plots of Figure 3-6. As in Figure 3-6, the position uncertainty of the AUV grows as the distance from the starting point increases. Further, after the first pass through the survey path, the ASM and the single-pass DSM results look very similar and achieve close to the same error bounds. The crucial difference between the methods is that ASM estimates the correlations between *all* features, while single-pass DSM only estimates the correlations within submaps. ASM is able to exploit all the correlations and thus reduce the global error at all locations. Single-pass DSM is unable to reduce the global

uncertainty of submaps below the uncertainty upon creation of the submap. This can be seen from the "steps" in the north and east 3σ bounds after the completion of the first pass through the survey area (that is, after the first 2 hours of the mission).

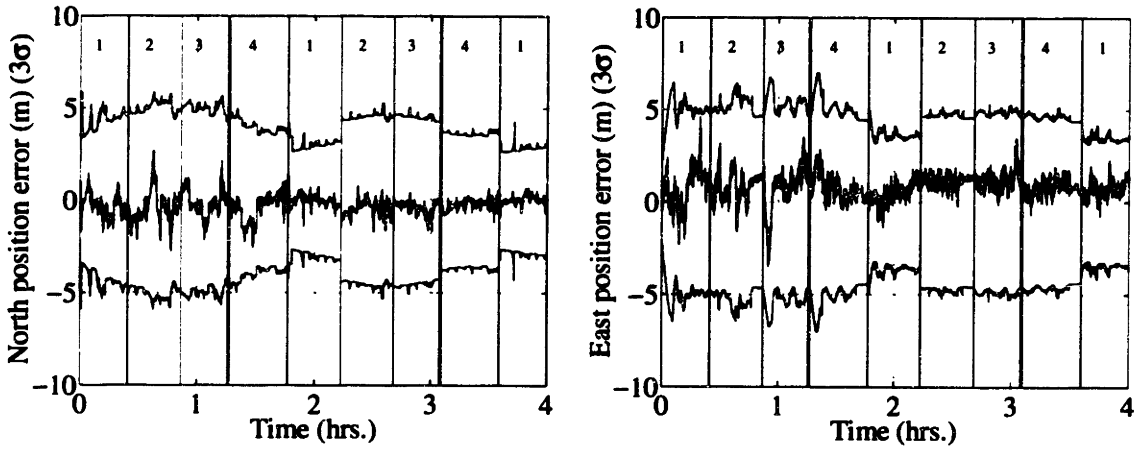

Figure 3-10: Expanded view of the north and east AUV position errors and the 3σ bounds for single-pass DSM, for the same run as shown in Figure 3-9. The vertical lines mark the transition times at which the AUV moved from one submap to another. The numbers identify the active submap during the period between the submap transitions.

Figure 3-10 shows a magnified version of the first two plots of Figure 3-9. In addition, vertical lines are drawn at the times when transitions between the submaps occur, and between the vertical lines, a number is shown to designate which submap is currently active. This clarifies the independence of the uncertainties in the submaps and emphasize the "steps" of the bounds. In the limit as time goes to infinity, each submap's error bounds converge to the smallest value within each submap. However, this minimum error cannot be reduced from the value that is obtained from the first pass through the current submap. For example, the 3σ bounds of submap 2 will always be about 0.5 meters larger than that of submap 1.

When the vehicle exits submap 3, it moves into an area that is shared by submaps 1 and 4. Since submap 1 was created before submap 4, the estimates in this submap are more accurate. Thus, the single-pass DSM algorithm performs a cross-map vehicle

relocation from submap 3 to submap 1, then spends a small time in submap 1 before creation or relocation to submap 4. This is the reason why submap 4 has smaller error bounds than submaps 2 and 3.

Having shown the results for ASM and single-pass DSM for the mission depicted in Figure 3-4, we now turn to the results for multi-pass DSM. The location and order of submap creation for multi-pass DSM are identical to that of single-pass DSM, as shown in Figure 3-7.

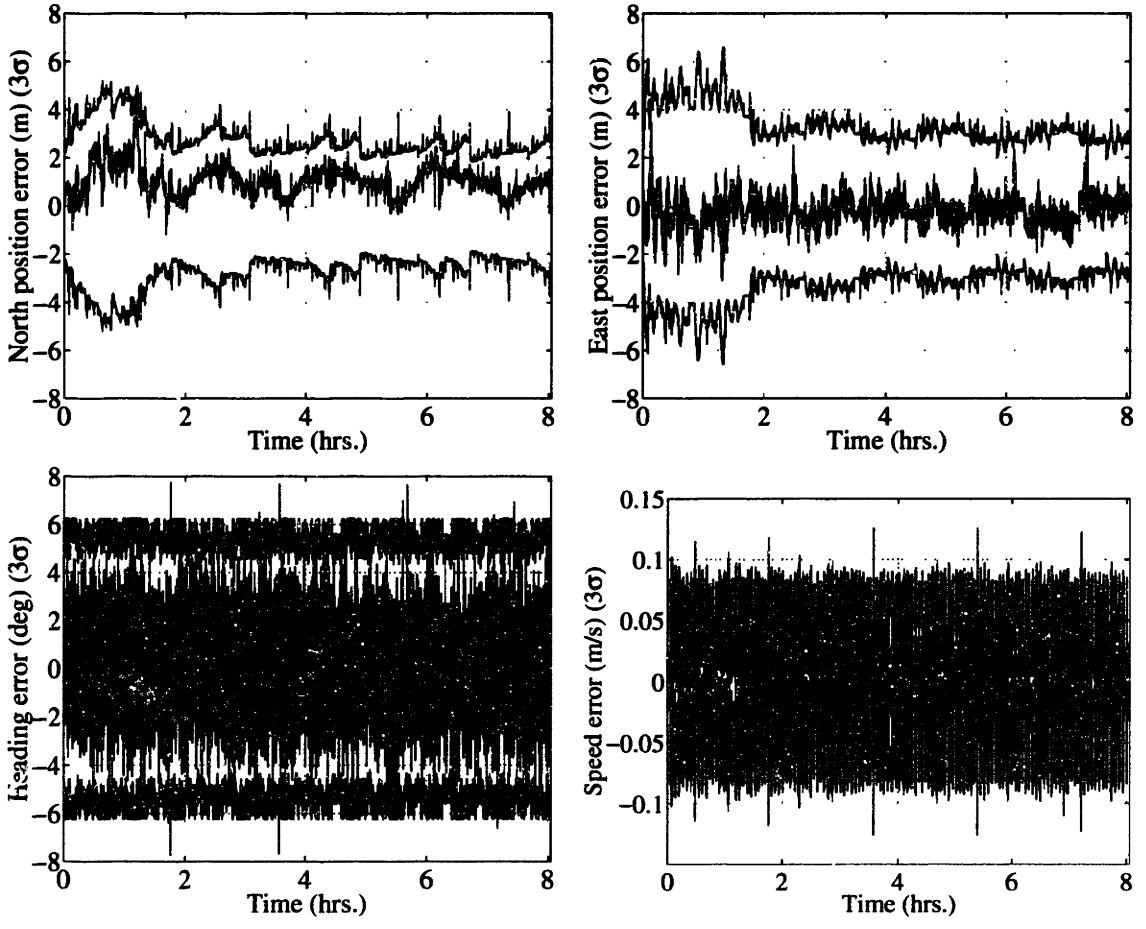

Figure 3-11: Errors and 3σ bounds (99% highest confidence bounds) for the position, heading and velocity estimates produced by multi-pass DSM.

Figure 3-11 shows the position, heading and velocity errors of the vehicle versus

time along with the 3σ bounds for the survey performed by multi-pass DSM. These results should be compared to the results from utilizing ASM (Figure 3-6) and single-pass DSM (Figure 3-9). By considering the top two plots of Figure 3-11, we see that they resemble the optimal result of ASM (Figure 3-6) more than the results from single-pass DSM. Clearly, the vehicle does better after the first pass through the survey area (that is, after about 2 hours) than before. Thus, the algorithm is capable of reducing the *global* error *everywhere* and not only locally in the submaps, as for single-pass DSM. However, one can see that the error bounds are a little more uneven than that of ASM, and reducing the uncertainties takes a little more time. However, as time goes to infinity, the two solutions should be identical. Another thing to note from these plots is that the position error bounds are actually *smaller* for this particular multi-pass DSM survey than for the particular ASM run shown in Figure 3-6. This is due to the fact that, by coincidence, multi-pass DSM was more "lucky" in the sonar returns it got at the start of the mission. Thereby, the vehicle was able to localize itself more accurately in the initial phase of multi-pass DSM. This better localization early on propagates through the whole system for the rest of the mission. This highlights a very critical point for all stochastic mapping approaches: the initial phase of mapping is crucial as it defines a lower limit for the error bounds. As Figure 3-11 and Figure 3-9 show, the initial phase is more crucial to the error bounds achieved than whether or not the ASM or the multi-pass DSM approach to CML is utilized. Thus we believe that the results from multi-pass DSM are close to the optimal solution.

Figure 3-12 shows a magnified version of the first two plots of Figure 3-11. It is useful to compare this to the equivalent plot for single-pass DSM shown in Figure 3-10. Again, the numbers between vertical lines signify the currently active submap at different times during the mission. As in the results for single-pass DSM, the AUV passes through submap 1 before entering submap 4. Thus, the error bounds in submap 4 are decreased. It should be noticed that the error bounds do not increase

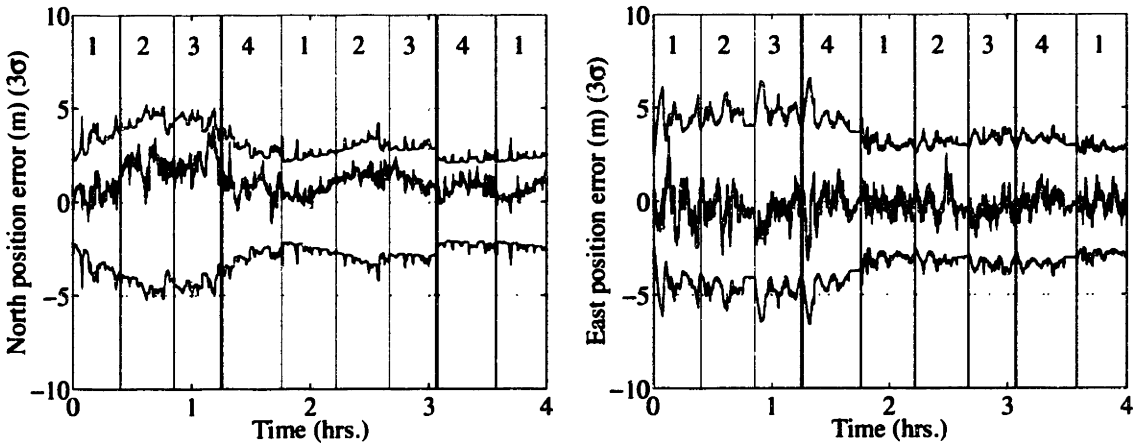

Figure 3-12: Expanded view of north and east AUV position errors and the 3σ bounds for multi-pass DSM, for the same run as shown in Figure 3-11.

in "steps" as a function of submap number, as is the result when using single-pass DSM.

As a final comparison between single-pass and multi-pass DSM, we focus our attention on a single feature at the termination of the mission that was estimated in both methods in all submaps. The true feature location along with the estimates and corresponding 3σ error ellipses are shown in Figure 3-13. As can be seen from these plots, the error ellipses are smaller in multi-pass DSM than in single-pass DSM. This is expected, as we know that the multi-pass DSM algorithm reduces the global errors and thus, when performing a mission where regions are revisited, is superior to single-pass DSM. In fact, the feature that is known most accurately (that is, it has the smallest error ellipse) in single-pass DSM (generated from submap 1) is approximately the same as the largest error ellipse when using multi-pass DSM (which was generated by submap 3.)

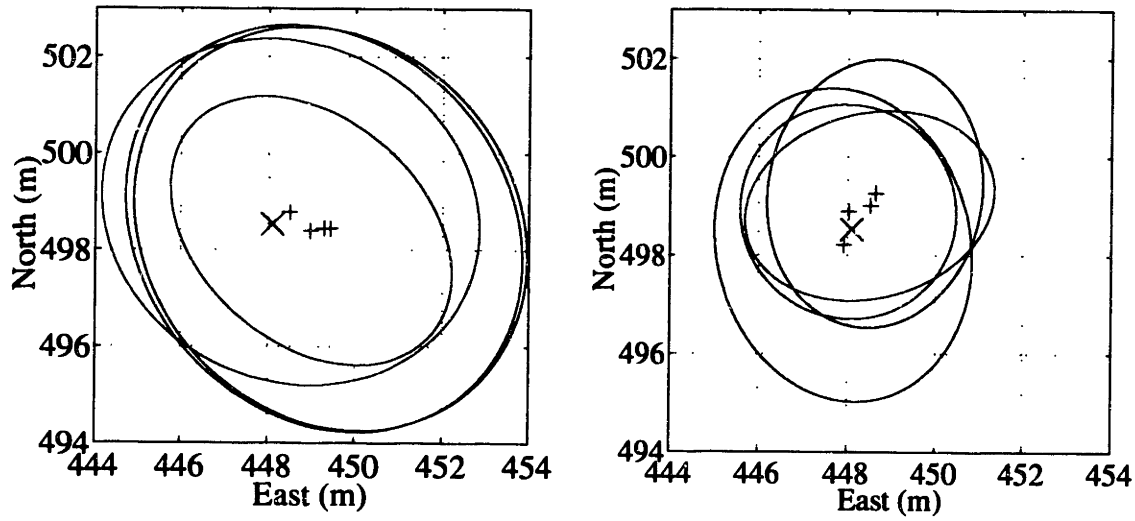

Figure 3-13: The true feature location (marked by '×') along with the estimated feature positions (marked by '+') and associated 3σ error ellipses as estimated in the four different submaps. The left figure is the result from single-pass DSM and the right figure is the result from multi-pass DSM. Moving from the true feature position to the right, the error ellipses correspond to submaps 1, 4, 3, and 2 for single-pass DSM, and to submaps 1, 2, 4, and 3 for multi-pass DSM.

3.4.2 Large-scale, long-term DSM

In this section we will demonstrate the usefulness of single-pass DSM and multi-pass DSM in surveying large-scale environments with 1200 environmental features for a mission duration of over 120 hours sampling at a rate of 1 Hz. The parameters for these mission are given in Table 1. As the spatial density of features is about 20% higher in these runs than in the previous runs, each submap will contain approximately 20% more features than in the previous runs. This causes the computation time to increase to be just a little faster than real time on a 333MHz Pentium processor. It is impossible to obtain results for these large size systems without using DSM on a conventional computer, thus no results for the the full covariance ASM algorithm for these runs can be obtained for comparison.

Figure 3-14 shows the desired path of the AUV through the 3 km by 3 km survey

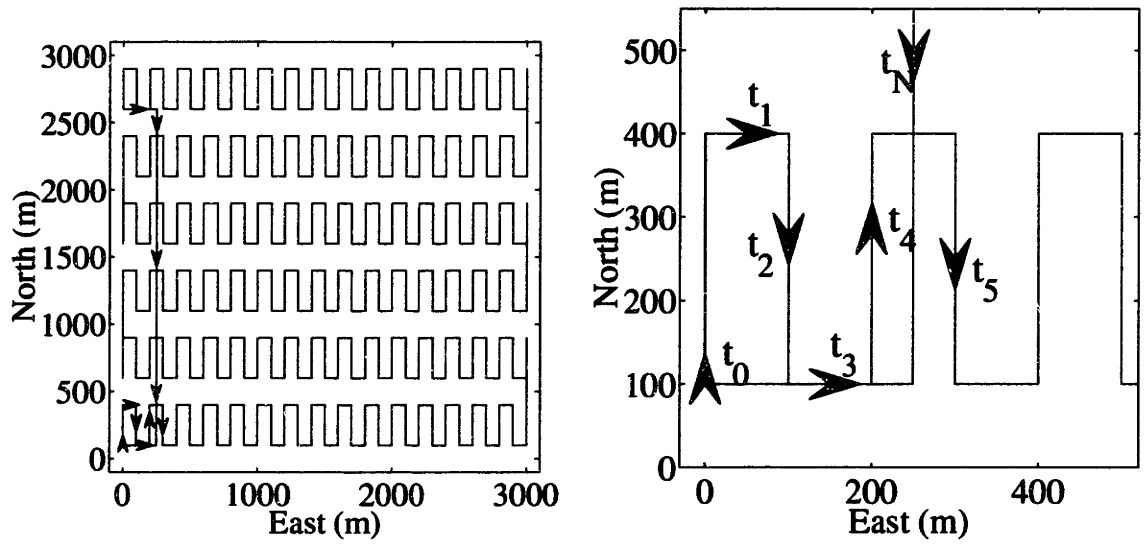

Figure 3-14: The desired survey path of the vehicle (solid line) in the 3 km by 3 km survey area with 1200 features. The arrows indicate the direction of movement of the robot initially and upon a single completion of the mission path. The right plot shows a magnification of the starting area for the mission. The time indices signify the starting point, t_0, and points on the path later on, t_1 to t_5. t_N signifies a point towards the end of completion of the mission path.

area. The vehicle starts at $(0, 100)$ meters and follows the desired path in the direction of the arrows. Only arrows for the initial stage and the final stage of the path have been included for clarity. Figure 3-15 shows the partition of the survey area into submaps as performed by the two DSM methods. The numbers signify the order in which the submaps were created. To illustrate the complexity of the problem, Figure 3-16 shows the returns of the sonar for the first 30 hours of the mission as well as the true position and estimated position of the 1200 environmental features in the survey area during multi-pass DSM.

Figure 3-17 shows plots of the position, heading, and velocity errors of the vehicle versus time, along with the 3σ bounds when using single-pass DSM for the survey area of Figure 3-14. In this simulation, the vehicle completed 11 laps of the survey path. The characteristic of single-pass DSM of a position uncertainty growing with

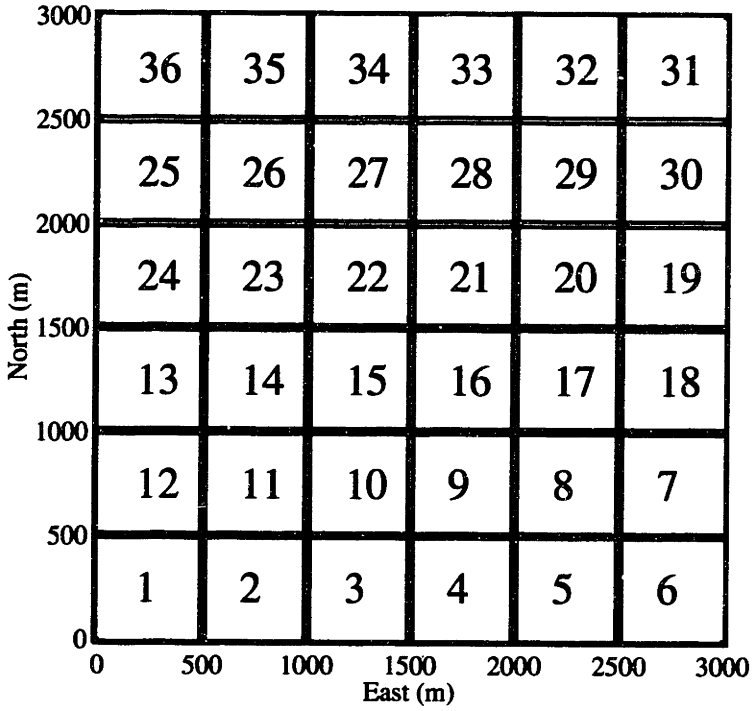

Figure 3-15: The partition of the survey area into 36 submaps performed by the two DSM methods for the long simulation run with 1200 features.

the vehicle's distance from its initial position is clear from these plots. It is apparent that information is *not* passed between submaps to enable the global reduction of uncertainties across maps, as occurs in multi-pass DSM, but as will be shown, the estimates during the first pass through the survey are more certain for single-pass DSM than for multi-pass DSM. The sharp spikes on the 3σ error bounds for the position estimates are artifacts of the cross-map vehicle relocation of Equation (3.17), where the covariance of the vehicle is increased upon re-entry into a submap.

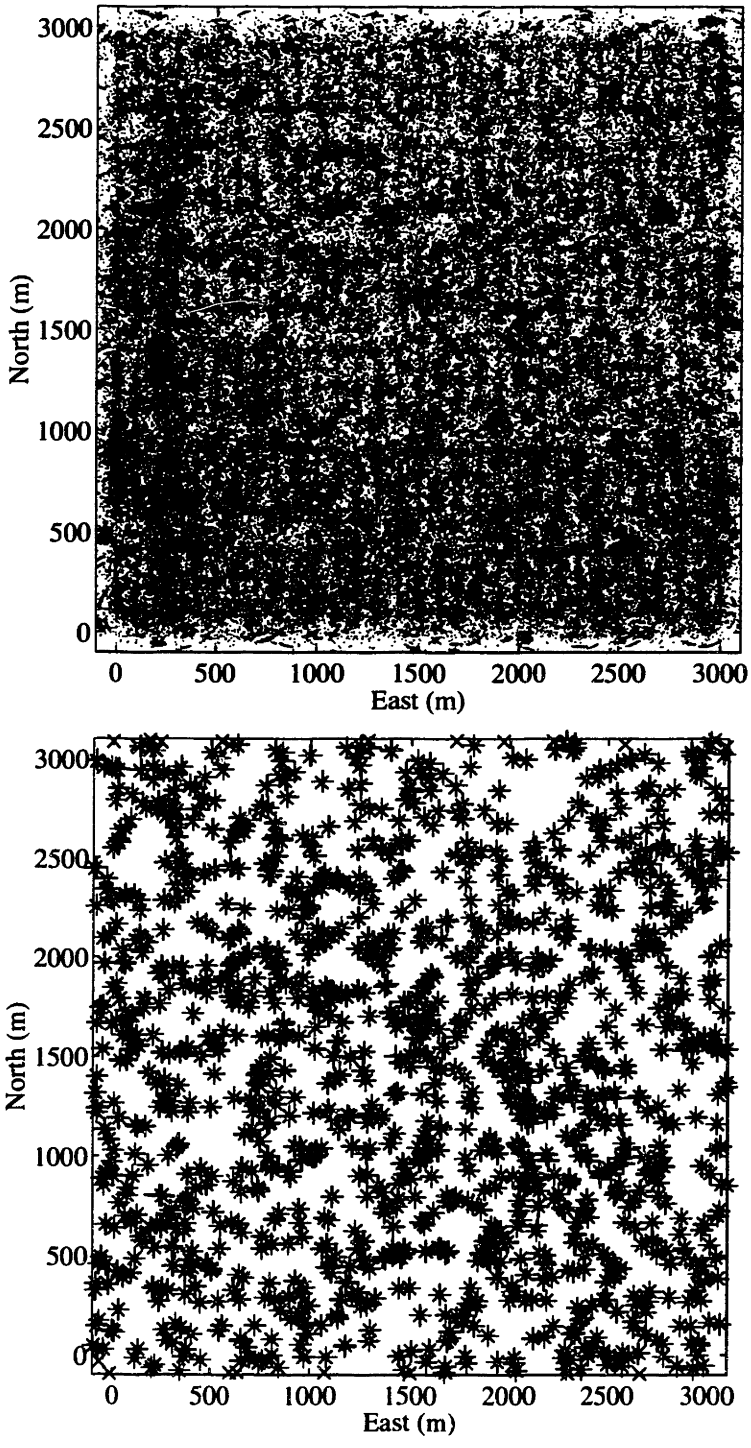

Figure 3-16: **Left:** The sonar returns for the first 30 hours of the mission (marked by small dots). **Right:** The true feature positions (marked by '×') and the estimated feature positions (marked by '+') and 3σ error ellipses for the long duration multi-pass DSM run.

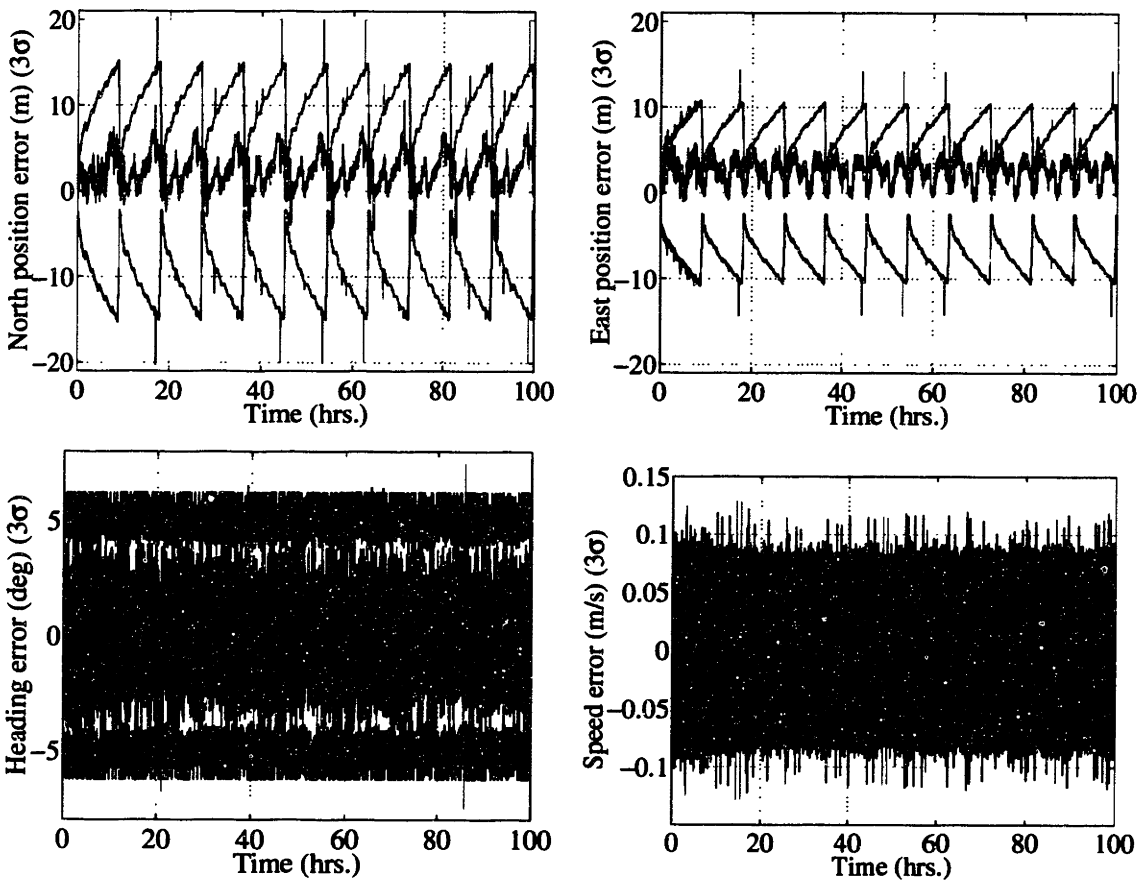

Figure 3-17: Errors and 3σ bounds (99% highest confidence bounds) for the position, heading and velocity estimates produced by single-pass DSM in surveying the area given in Figure 3-14.

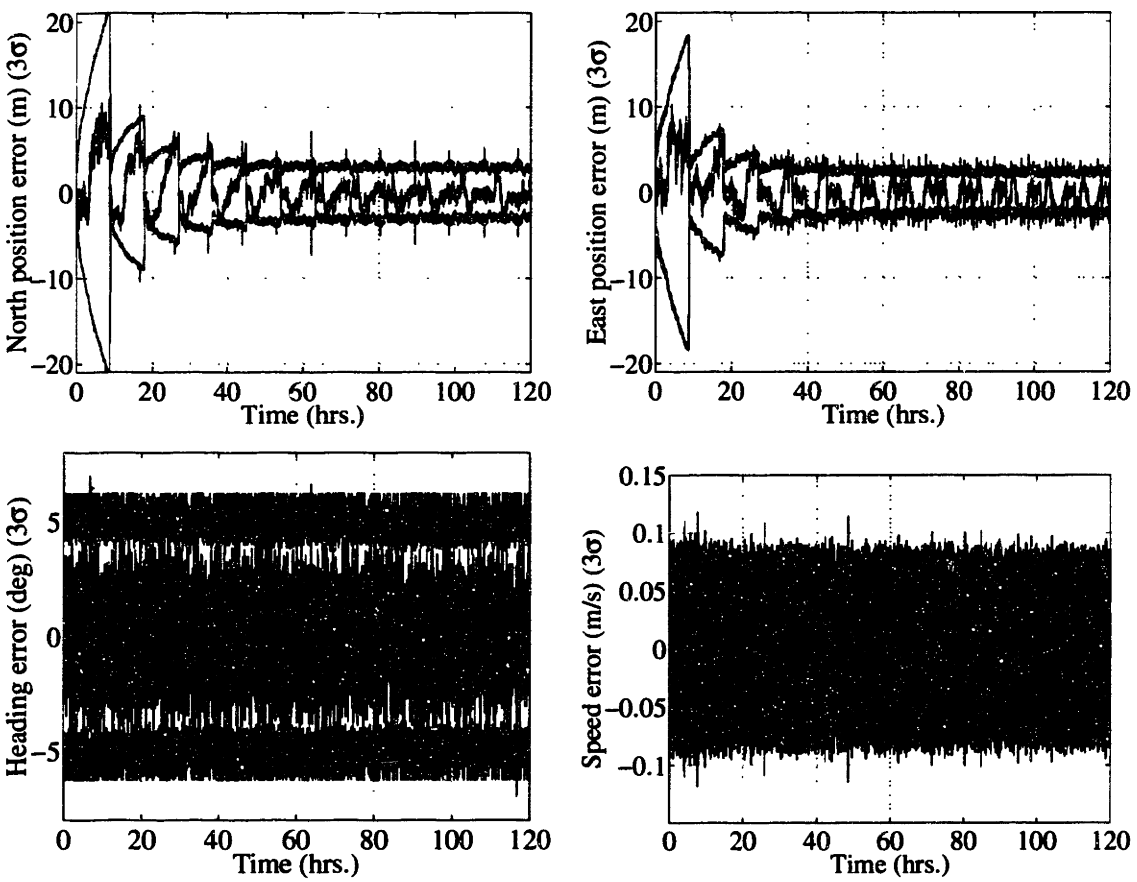

Figure 3-18: Errors and 3σ bounds (99% highest confidence bounds) for the position, heading and velocity estimates produced by multi-pass DSM.

Figure 3-18 shows the position, heading and velocity errors of the vehicle versus time, along with the 3σ error bounds when using multi-pass DSM for the survey area of Figure 3-15. In this 120 hour mission time simulation, the vehicle performed a little more than 13 laps of the mission path. As with Figure 3-17, occasional spikes in the position error bounds show up during cross-map vehicle relocation, which is performed when moving to a submap that was created earlier than the current submap. As can be seen from Figure 3-14 and Figure 3-15, this occurs when the vehicle travels from submap 36 to submap 1 at the completion of each cycle of the survey. From Figure 3-18 we can clearly see that information *is* passed between submaps in such a way that the *global* uncertainty of submaps is reduced. In the second pass through the desired path, the maximum position uncertainty is more than halved. Already, after three or four passes through the survey area, the system has reached a certainty close to the steady state, with a 99% position uncertainty for the vehicle of less than four meters. This implies that with relatively off the shelf equipment of moderate cost, and the ability to relatively reliably extract features from the environment, accuracy comparable to GPS for navigation of AUVs can be achieved. We see this as a considerable contribution.

Figure 3-19 and 3-20 show the north and east position error and bounds for mission times 98.4 to 100 hours for the single-pass DSM and the multi-pass DSM simulations of Figure 3-17 and 3-18, respectively. Figure 3-21 shows the errors in estimated feature locations along with the 3σ bounds for these estimates for the single-pass DSM and multi-pass DSM missions, respectively. The mean error achieved is 4.3 meters for single-pass DSM and 2.6 meters for multi-pass DSM, respectively.

Multi-pass DSM shows a considerable improvement over single-pass DSM in the long run as the survey area is revisited. However, during the first pass through the survey area, the maximum uncertainty when using multi-pass DSM is more than 30% higher than the result when using single-pass DSM. This provides the justification for utilizing single-pass DSM when the survey area is assumed to be traversed only

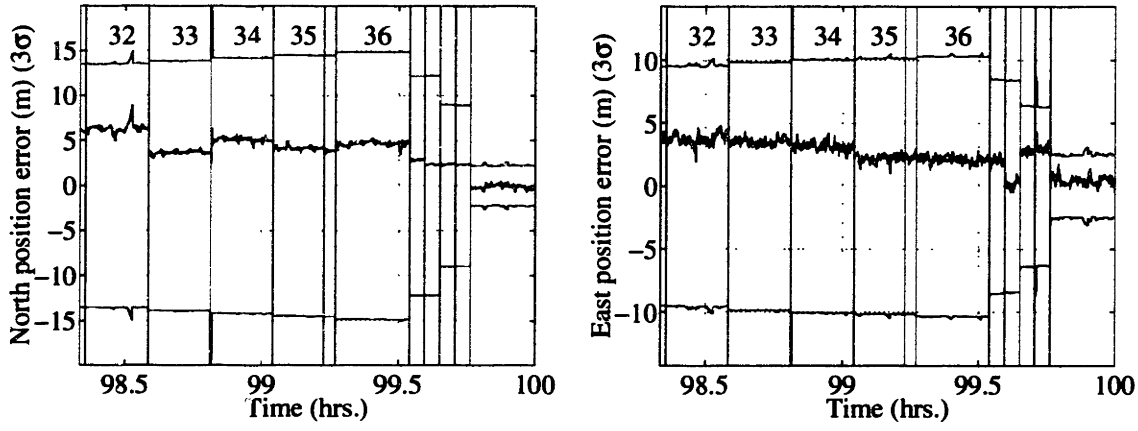

Figure 3-19: Magnification of the north and east AUV position estimates along with the 3σ bounds for the last 1.6 hours of the single-pass DSM survey shown in Figure 3-17. The vertical lines mark the transition times at which the AUV moved from one submap to another. The numbers identify the active submap during the period between the submap transitions. As can be seen, the error bounds are nearly constant within each submap, suggesting that the system has reached a steady state.

once. Multi-pass DSM, however, should be used if one anticipates multiple traversals of the environment.

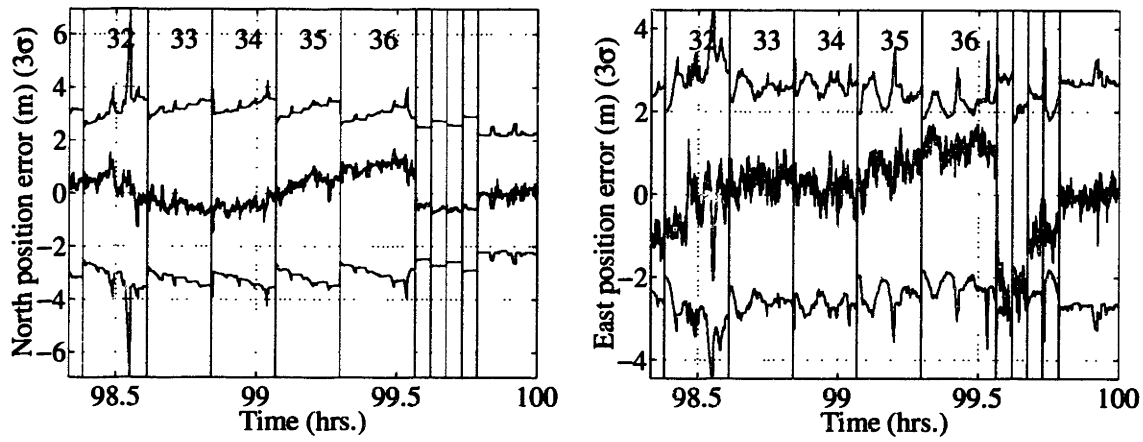

Figure 3-20: Magnification of the north and east AUV position estimates along with the 3σ bounds for a 1.6 hours segment near the end of the multi-pass DSM survey shown in Figure 3-18. The vertical lines mark the transition times at which the AUV moved from one submap to another. The numbers identify the submap number during the period between the submap transitions. The error bounds drop when cross-map vehicle updating is performed because submaps that are created later can draw upon more information than submaps that were created earlier.

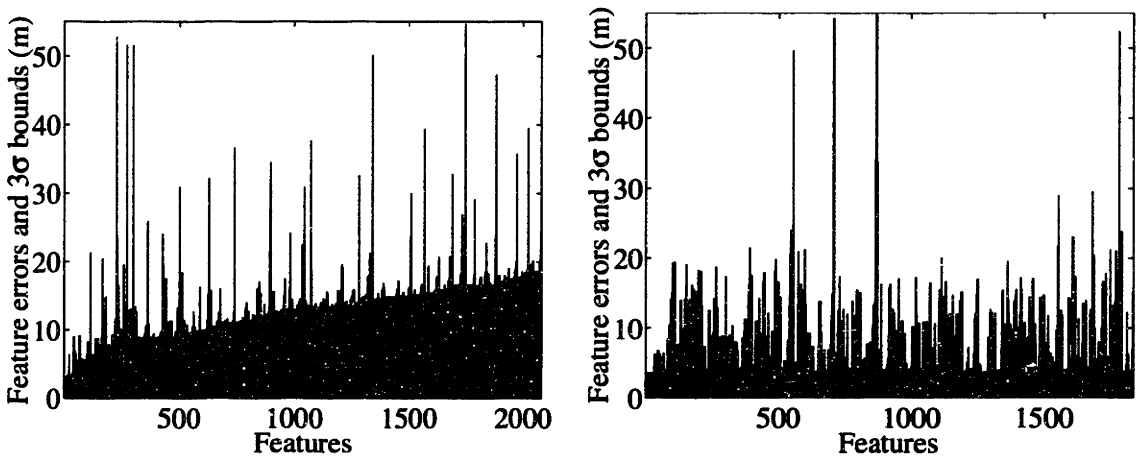

Figure 3-21: The distance in meters between the actual feature location (solid black) and the estimated feature location along with the 3σ bound (shaded red) for this distance. The errors and bounds have been plotted for each feature estimated, starting with the first estimated feature in submap 1 (which is given a feature number of 1), to the last feature estimated in submap 36 (which has a feature number close to 2000). **Left:** Single-pass DSM – the error bound grows slowly with the submap number, that is, with distance traveled from the vehicle's initial position when the submap was created. **Right:** Multi-pass DSM — the feature uncertainty and accuracy is relatively independent of the submap in which it exists.

Table 3.2: Decoupled SM experiment parameters.

sampling period, T	1 sec.
maximum sonar range	250 cm
sonar coverage angle	$\pm 40°$
range measurement standard deviation	2 cm
bearing measurement standard deviation	5°
feature probability of detection	0.90
vehicle cruise speed	10 cm/s
speed process standard deviation	5% of δv
heading process standard deviation	2.0°
dead reckoning speed standard deviation	0.45 cm/s
dead reckoning heading standard deviation	3.0°
initial position uncertainty std. dev.	0.7 cm
initial heading uncertainty std. dev.	5.0°
initial speed uncertainty std. dev.	0.2 cm/s
gate parameter γ	9
clutter parameter λ	1
track initiation parameters	$M = 5, N = 4$

3.5 Experimental results

In the previous section, extensive simulations were performed in order to validate the decoupled stochastic mapping approach to the map scaling problem. In this section, we present a simple multi-pass DSM experiment for further validation of the approach. The experiment was performed in the testing tank at MIT using the same setup as for the augmented stochastic mapping experiment of Section 2.6. As in Section 2.6, the parameters for the mission was chosen so as to simulate an AUV scaled down by a factor of 100 and it was assumed that the AUV would be able to obtain sonar readings over the $\pm 40°$ viewing area of the sonar every $T = 1$ second using an electronically scanned sonar. In the experiment, however, we used the Panametrics sonar that was mechanically scanned using the robotic positioning system. This caused the scan over the $\pm 40°$ viewing are of the to take approximately 2 minutes. As in Section 2.6 the experiment was also simulated in order to determine the quality of the simulations. Table 3.5 summarizes the parameters used in the experiment.

In the experiment, 90 fishing bobbers was used as features and was randomly placed in the testing tank as shown by the crosses in Figure 3-22. The sonar returns

from the tank walls were discarded. The sonar trajectory was set to perform a lawn-mower path starting at the lower right corner of the tank and moving towards the left (see Figure 3-22). The submap size was set to 4 by 4 meters and two submaps

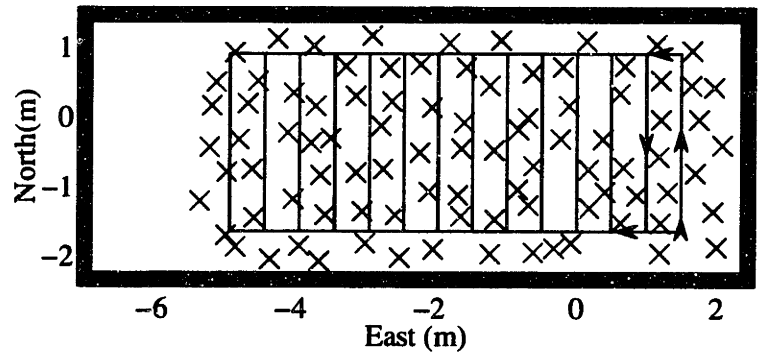

Figure 3-22: The desired path (solid line) along with the 90 fishing bobbers used as "environmental features" in the experiment. The small arrows indicate the starting path of the lawn-mower survey over the tank.

were used to perform concurrent mapping and localization during the experiment. Estimated result from the DSM algorithm was compared to the true position of the sonar as obtained from positioning decoders on the robotic positioning system. The entire mission lasted about 1250 time steps, allowing for two visits to both submaps.

The superposition of all sonar returns during the mission is shown in Figure 3-23. As can be seen from comparing this figure to Figure 3-16, the experiment actually

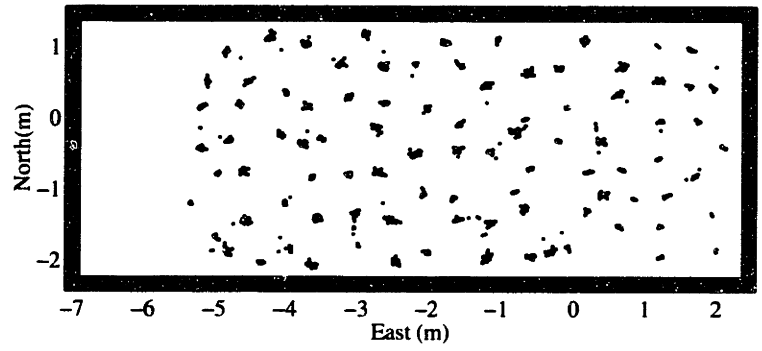

Figure 3-23: All the sonar returns obtained during the tank experiment.

have less false returns than was used in the simulations in the previous section. The

resulting position estimate error and 99% confidence bound for the experiment is shown in Figure 3-24 and for the simulation in Figure 3-25. The vertical lines identifies times when the vehicle estimate moved between submaps and the numbers in the plot identifies the submap number. As can be seen from these plots, there is excellent agreement between the simulation and the experiment. The bounds are virtually the same. The actual estimation error is, naturally, not identical as the estimates are stochastic processes. However, we see that they have the same structure.

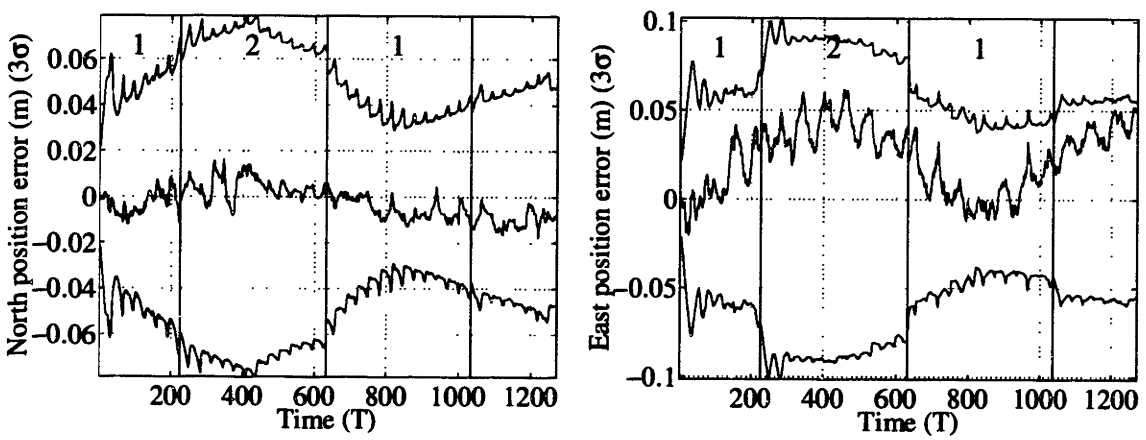

Figure 3-24: Errors and 3σ bounds (99% highest confidence bounds) for the position estimates from the *experiment* performing multi-pass DSM in the testing tank.

3.6 Discussion and conclusions

A computationally efficient method for large-scale CML has been an important but elusive goal in mobile robotics research. The methods presented here offer attractive solutions to the map scaling issue. The CPU and memory requirements are independent of the size of the map. The hard disk storage requirements of the method scale as $\mathcal{O}(mn)$ where m is the number of features states per submap and n is the total number of features.

The key innovation of our work is to maintain multiple globally-referenced submaps,

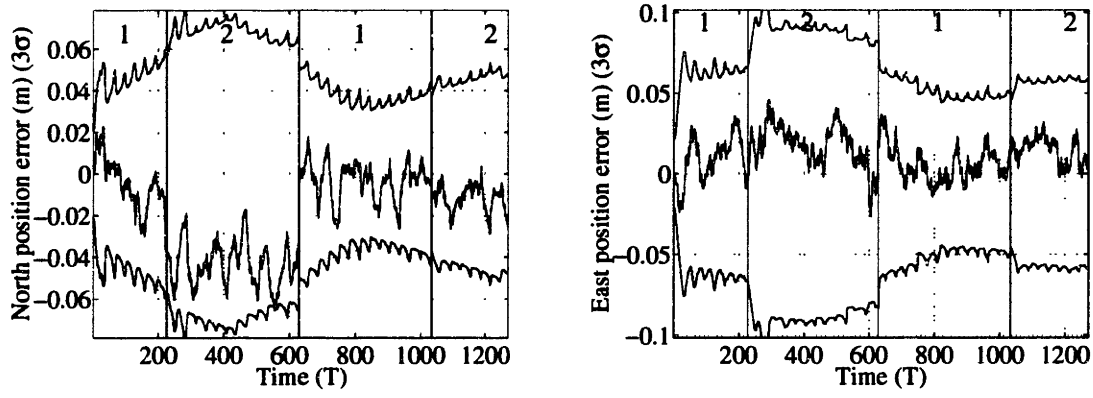

Figure 3-25: Errors and 3σ bounds (99% highest confidence bounds) for the position estimates from the *simulation* of the multi-pass DSM experiment of Figure 3-24.

each associated with a given map region, and to employ safe, consistent update methods for transferring vehicle location information between submaps as the vehicle crosses between regions. Extensive simulation results demonstrate successful CML for missions that are several orders of magnitude longer in duration and larger in scale than any work previously published in the literature.

Single-pass DSM, in which cross-map vehicle relocation is used to transition between submaps, is most suitable for a mission in which a vehicle is to traverse a region only once, building the best map it can while concurrently localizing itself. The method incurs a slowly growing error bound that is a linear function of the distance traveled from the origin of the mission. Subsequent traversals of the environment do not result in an improvement in map accuracy, but consistency of error estimates is maintained.

Multi-pass DSM combines two different methods for transitioning between map regions. When transitioning from later to earlier maps, the cross-map vehicle relocation strategy used in single-pass DSM is employed. However, when transitioning from a submap that was created before the submap that is being re-entered, the improved vehicle state estimate from the earlier submap is used as a state measurement

in the later submap. The old inaccurate vehicle state estimate from the later submap is discounted by greatly inflating its covariance before the update occurs, thereby de-correlating it from the earlier submap's vehicle state estimate. To counteract the violation of the independence assumptions of the Kalman filter that this process entails, the feature covariance matrix of the higher map is multiplied by two before the update. The resulting feature map covariance is demonstrated to bound the original feature covariance, thus providing a consistent update strategy.

The simulation results presented in this chapter demonstrate that both methods provide accurate vehicle and feature location estimates, maintain consistent error bounds, and do not diverge. In Section 3.5 we show excellent agreement between a multi-pass DSM simulations and real data experiments for the scenario of AUV navigation using a forward looking sonar in *a priori* unknown environments. In the experiment, 90 fishing bobbers and two submaps where used to perform concurrent mapping and localization.

Our method is perhaps closest in spirit to Chong and Kleeman's strategy for large-scale mapping in indoor environments using multiple local maps [29]. There are important differences, however. Chong and Kleeman wrote:

> "As a simple example, consider the scenario in which the robot creates a local map B after leaving a local map A. If the robot happens to re-enter local map A again and performs more sensing to obtain a better estimate of its position relative to local map A, this information is completely discarded if the robot enters local map B again. In other words, the improved estimate of position is not used to update the relative spatial position of local map B relative to local map A." [29]

With multi-pass DSM, we have realized an effective strategy for safely transferring vehicle position estimates between submaps, leading to incremental reduction of errors in feature location estimates throughout the environment, in a manner that is comparable to full covariance stochastic mapping. By revisiting one submap (say,

submap B) with an improved position estimate for the vehicle that was obtained in another submap (submap A), subsequent observations of the features in submap B, obtained from more accurately known vehicle positions, yield improved location estimates for submap B's features, with consistent error bounds.

An additional distinction of single-pass DSM from the work of Chong and Kleeman [29] is that, by using multiple, globally-referenced submaps, we eliminate the potential problem of the proliferation of duplicate local maps that cover the same area of the environment. By using cross-map vehicle relocation, we eliminate the need for local map construction and matching to relocate the vehicle when it enters a new submap. Thus continuous localization can be performed without interruption as the vehicle travels across map regions.

The methods presented here are attractive because they provide powerful tools to address more complex issues of the concurrent mapping and localization problem, such as map maintenance in dynamic environments [67], improved resolution of data association ambiguity [34], relocation, recovery from errors, and cooperative mapping and navigation by teams of mobile robots. For example, the hybrid estimation approach to CML of Integrated Mapping And Navigation (IMAN), described in Chapter 1, encountered extreme computational issues, because of the interactions between different vehicle and feature tracks [105, 106, 107]. The new methods presented here suggest a strategy for decoupling the interaction between data association hypotheses for different parts of an environment. Cluster partitioning may become a possibility despite the presence of navigation error [63, 34].

The problem of reliably extracting features from sensor data remains an important outstanding issue for the realization of CML in more complex and natural environments. As pointed out by Lozano-Pérez, "the performance of any sensing system is largely determined after the feature detectors have finished their work [72]". The implementation of CML for underwater vehicles operating in dynamic undersea environments will require advances in our ability to extract features from natural terrain

and to represent and manipulate them appropriately in the map representation. For point-like features, initial results in the post-processing of real data from a forward looking sonar presented in Section 2.7 are encouraging.

Another important issue for further research is the use of adaptive motion and sensing strategies for AUVs in order to perform CML faster, more accurately and with less energy than the conventional non-adaptive CML strategy.

3.7 Summary

This chapter presented decoupled stochastic mapping (DSM), a new computationally efficient method for large-scale concurrent mapping and localization (CML) that resolves the map scaling problem. DSM circumvents the $\mathcal{O}(n^3)$ computational burden of conventional stochastic mapping (where n is proportional the number of features), while providing consistent, globally-referenced error bounds.

In DSM, the global map is split into multiple globally-referenced submaps, each with its own vehicle track. Two new methods, referred to as (1) cross-map vehicle relocation and (2) cross-map vehicle updating, were developed for safely transferring vehicle state estimate information from one submap to another as the vehicle transitions between map regions.

Two variations of DSM were presented. Single-pass DSM uses cross-map vehicle relocation for all submap-to-submap transitions, achieving a small linear error growth. Multi-pass DSM uses a combination of both new vehicle transition strategies to achieve navigation and feature localization errors that improve with time, yielding results comparable to full covariance stochastic mapping. Experiment using real data showed excellent agreement between experiment and simulation of DSM.

The approach of DSM provides a starting point for solutions of other long-standing challenges in CML, such as incorporation of more powerful data association strategies, maintenance of maps in dynamic environments, and cooperative mapping and

navigation by multiple vehicles.

Another issue for future research is to improve CML performance through adaptive CML strategies. This interesting field of study is the focus of Chapter 4.

3.A Consistency of cross-map vehicle updating

The goal of this appendix is to demonstrate that consistent estimates of feature locations are maintained by the cross-map vehicle updating procedure described in Section 3.3.2. As described above in Section 3.2, in each submap four states are used to represent the vehicle and two states are used to represent each of the n features. Hence, the state vector for a submap is $(2n + 4) \times 1$ and the covariance matrix is $(2n + 4) \times (2n + 4)$. The predicted state vector for the vehicle and map features for submap B is:

$$\mathbf{x}^B_{k+1|k} = \begin{bmatrix} \mathbf{x}^B_r \\ \mathbf{x}^B_f \end{bmatrix},$$

and the predicted state covariance is

$$\mathbf{P}^B_{k+1|k} = \begin{bmatrix} \mathbf{P}^B_{rr} & \mathbf{P}^B_{rf} \\ \mathbf{P}^B_{fr} & \mathbf{P}^B_{ff} \end{bmatrix},$$

where $\mathbf{P}^B_{fr} = [\mathbf{P}^B_{rf}]^T$. The cross-map updating procedure applies $\mathbf{x}^A_r$, the vehicle estimate from submap A, as a measurement of vehicle position in submap B using a Kalman filter, after first randomizing $\mathbf{x}^B_r$ and greatly inflating $\mathbf{P}^B_{rr}$.

To apply this measurement of the vehicle location, the $\mathbf{H}$ matrix is:

$$\mathbf{H}_{k+1} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix},$$

where $\mathbf{I}$ is the 4×4 identity matrix and $\mathbf{0}$ is a $4 \times 2N$ matrix of zeros. The vehicle position covariance in map A, $\mathbf{P}^A_{rr}$, is used as the measurement noise covariance $\mathbf{R}_{k+1}$ associated with this measurement.

$$\mathbf{R}_{k+1} = \mathbf{P}^A_{rr}.$$

In the sequel, the superscript to designate submap B will be dropped for visual clarity. The innovation covariance $\mathbf{S}_{k+1}$ is:

$$\begin{aligned}\mathbf{S}_{k+1} &= \mathbf{H}_{k+1}\mathbf{P}_{k+1|k}\mathbf{H}_{k+1}^T + \mathbf{R}_{k+1} \\ &= \begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{P}_{rr} & \mathbf{P}_{rf} \\ \mathbf{P}_{fr} & \mathbf{P}_{ff}\end{bmatrix}\begin{bmatrix}\mathbf{I} \\ \mathbf{0}\end{bmatrix} + \mathbf{R}_{k+1} \\ &= \mathbf{P}_{rr} + \mathbf{R}_{k+1}.\end{aligned}$$

Next, we calculate the Kalman gain $\mathbf{K}_{k+1}$:

$$\begin{aligned}\mathbf{K}_{k+1} &= \mathbf{P}_{k+1|k}\mathbf{H}_{k+1}^T\mathbf{S}_{k+1}^{-1} \\ &= \begin{bmatrix}\mathbf{P}_{rr} & \mathbf{P}_{rf} \\ \mathbf{P}_{fr} & \mathbf{P}_{ff}\end{bmatrix}\begin{bmatrix}\mathbf{I} \\ \mathbf{0}\end{bmatrix}(\mathbf{P}_{rr} + \mathbf{R}_{k+1})^{-1} \\ &= \begin{bmatrix}\mathbf{P}_{rr}(\mathbf{P}_{rr} + \mathbf{R}_{k+1})^{-1} \\ \mathbf{P}_{fr}(\mathbf{P}_{rr} + \mathbf{R}_{k+1})^{-1}\end{bmatrix}.\end{aligned}$$

The updated state covariance is:

$$\begin{aligned}\mathbf{P}_{k+1|k+1} &= \mathbf{P}_{k+1|k} - \mathbf{K}_{k+1}\mathbf{S}_{k+1}\mathbf{K}_{k+1}^T \\ &= \begin{bmatrix}\mathbf{P}_{rr} & \mathbf{P}_{rf} \\ \mathbf{P}_{fr} & \mathbf{P}_{ff}\end{bmatrix} - \begin{bmatrix}\mathbf{P}_{rr}(\mathbf{P}_{rr} + \mathbf{R}_{k+1})^{-1} \\ \mathbf{P}_{fr}(\mathbf{P}_{rr} + \mathbf{R}_{k+1})^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{P}_{rr} + \mathbf{R}_{k+1}\end{bmatrix} \times \\ &\quad \begin{bmatrix}[(\mathbf{P}_{rr} + \mathbf{R}_{k+1})^{-1}]^T\mathbf{P}_{rr} & [(\mathbf{P}_{rr} + \mathbf{R}_{k+1})^{-1}]^T\mathbf{P}_{rf}\end{bmatrix} \\ &= \begin{bmatrix}\mathbf{P}_{rr} & \mathbf{P}_{rf} \\ \mathbf{P}_{fr} & \mathbf{P}_{ff}\end{bmatrix} - \begin{bmatrix}\mathbf{P}_{rr}[(\mathbf{P}_{rr} + \mathbf{R}_{k+1})^{-1}]^T\mathbf{P}_{rr} & \mathbf{P}_{rr}[(\mathbf{P}_{rr} + \mathbf{R}_{k+1})^{-1}]^T\mathbf{P}_{rf} \\ \mathbf{P}_{fr}[(\mathbf{P}_{rr} + \mathbf{R}_{k+1})^{-1}]^T\mathbf{P}_{rr} & \mathbf{P}_{fr}[(\mathbf{P}_{rr} + \mathbf{R}_{k+1})^{-1}]^T\mathbf{P}_{rf}\end{bmatrix} \\ &= \begin{bmatrix}\mathbf{P}_{rr} - \mathbf{P}_{rr}[(\mathbf{P}_{rr} + \mathbf{R}_{k+1})^{-1}]^T\mathbf{P}_{rr} & \mathbf{P}_{rf} - \mathbf{P}_{rr}[(\mathbf{P}_{rr} + \mathbf{R}_{k+1})^{-1}]^T\mathbf{P}_{rf} \\ \mathbf{P}_{fr} - \mathbf{P}_{fr}[(\mathbf{P}_{rr} + \mathbf{R}_{k+1})^{-1}]^T\mathbf{P}_{rr} & \mathbf{P}_{ff} - \mathbf{P}_{fr}[(\mathbf{P}_{rr} + \mathbf{R}_{k+1})^{-1}]^T\mathbf{P}_{rf}\end{bmatrix}\end{aligned}$$

The result of the Kalman update is that the new feature covariance submatrix is

reduced to become $\mathbf{P}_{ff} - \mathbf{P}_{fr}[(\mathbf{P}_{rr} + \mathbf{R}_{k+1})^{-1}]^T\mathbf{P}_{rf}$, even though we have obtained no new information about the features. We know from the behavior of the Kalman filter that

$$\mathbf{P}_{ff} - \mathbf{P}_{fr}[(\mathbf{P}_{rr} + \mathbf{R}_{k+1})^{-1}]^T\mathbf{P}_{rf}$$

must be positive definite. In the limit as the feature locations and vehicle location become completely correlated and the measurement noise covariance, $\mathbf{R}_{k+1}$, goes to zero, the *a posteriori* feature covariance goes to zero, e.g. $\mathbf{P}_{fr}[(\mathbf{P}_{rr} + \mathbf{R}_{k+1})^{-1}]^T\mathbf{P}_{rf}$ becomes $\mathbf{P}_{ff}$.

No information is gained for the features during this update, and via the Kalman gain the feature locations are slightly changed. The quantity

$$\mathbf{P}_{ff} - \mathbf{P}_{fr}[(\mathbf{P}_{rr} + \mathbf{R}_{k+1})^{-1}]^T\mathbf{P}_{rf}$$

bounds the *change* in the feature locations caused by the update. We can obtain a conservative, globally-referenced bound for the features simply by adding the *a priori* feature covariance $\mathbf{P}_{ff}$ to the *a posteriori* feature covariance:

$$\mathbf{P}_{k+1|k+1} = \begin{bmatrix} \mathbf{P}_{rr} - \mathbf{P}_{rr}\mathbf{P}_{rr}^T(\mathbf{P}_{rr} + \mathbf{R}_{k+1})^{-1} & \mathbf{P}_{rf} - \mathbf{P}_{rr}\mathbf{P}_{fr}(\mathbf{P}_{rr} + \mathbf{R}_{k+1})^{-1} \\ \mathbf{P}_{fr} - \mathbf{P}_{rr}^T\mathbf{P}_{rf}(\mathbf{P}_{rr} + \mathbf{R}_{k+1})^{-1} & \mathbf{P}_{ff} - \mathbf{P}_{fr}[(\mathbf{P}_{rr} + \mathbf{R}_{k+1})^{-1}]^T\mathbf{P}_{rf} + \mathbf{P}_{ff} \end{bmatrix}$$

Exactly the same result is obtained if, instead of adding $\mathbf{P}_{ff}$ after the update step, we add $\mathbf{P}_{ff}$ to the feature covariance before the update, or equivalently, $\mathbf{P}_{ff}$ is replaced by $2\mathbf{P}_{ff}$ as illustrated in Section 3.3.2. This is the case because the Kalman gain depends only on $\mathbf{P}_{rr}$ and $\mathbf{P}_{fr}$, and does not depend on $\mathbf{P}_{ff}$.

Chapter 4

Adaptive stochastic mapping

The previous chapters have assessed the potential of stochastic mapping and presented a new method for performing stochastic mapping in large-scale environments. This chapter presents a technique for adaptive concurrent mapping and localization based on optimizing the expected Fisher information due to an action. The validity of the approach is demonstrated in simulations and experiments.

4.1 Introduction

Adaptive sensing strategies have the potential to save time and maximize the efficiency of the concurrent mapping and localization process for an AUV and for mobile robots in general. Energy efficiency is one of the most challenging issues in the design of underwater vehicle systems [14]. Techniques for building a map of sufficient resolution as quickly as possible would be highly beneficial. Survey class AUVs must maintain forward motion for controllability [12], hence the ability to adaptively choose a sensing and motion strategy that obtains the most information about the environment is especially important.

Sonar is the principal sensor for AUV navigation. Possible sonar systems include mechanically and electronically scanned sonars, side-scan sonar and multi-beam

bathymetric mapping sonars [125]. The rate of information obtained from a mechanically scanned sonar is low, making adaptive strategies especially beneficial. Electronically scanned sonars can provide information at very high data rates, but enormous processing loads make real-time implementations difficult. Adaptive techniques can be used to limit sensing to selected regions of interest, dramatically reducing computational requirements.

In this chapter, adaptive sensing is formulated as the evaluation of different actions that the robot can take and the selection of the action that maximizes the amount of information acquired. This general problem has been considered in a variety of contexts [75, 99] but not specifically for concurrent mapping and localization. CML provides an interesting context within which to address adaptive sensing because of the trade-off between dead-reckoning error and sensor data acquisition. The information gained by observing an environmental feature from multiple vantage points must counteract the rise in uncertainty that results from the motion of the vehicle.

Our experiments use two different robot systems. One is an inexpensive wheeled land robot equipped with a single rotating sonar. Observations are made of several cylindrical targets whose location is initially unknown to the robot. Although this is a simplified scenario, these experiments provide a useful illustration of the adaptive CML process and confirm behavior seen in simulation. The second system is a planar robotic positioning system which moves a sonar within a 9.4 meter by 3.5 meter by 1.5 meter testing tank. The sonar is a mechanically scanned 675 kHz Imagenex model 881 sonar with a 2 degree beam. These characteristics are similar to alternative models used in the marine industry. Testing tank experimentation provides a bridge between simulation and field AUV systems. Repeatable experiments can be performed under identical conditions; ground truth can be determined to high accuracy.

Section 4.2 reviews previous research in concurrent mapping and localization and adaptive sensing. Section 4.3 develops the theory of adaptive stochastic mapping. Sections 4.4, 4.5.1, and 4.5.2 describe testing of the method using simulations, air

sonar experiments, and underwater sonar experiments. Section 4.6 provides concluding remarks and suggestions for future research. Finally, Section 4.7 provides a short summary of the chapter.

4.2 Previous work

Adaptive sensing has been a popular research topic in many different areas of robotics. Synonymous terms that have been used for these investigations include active perception [2], active vision [18], directed sensing [69], active information gathering [50], adaptive sampling [14], sensor management [75] and limited rationality [100]. A common theme that emerges is that adaptive sensing should be formulated as the process of evaluating different sensing actions that the robot can take and choosing the action that maximizes the information acquired by the robot. The challenge in implementing this concept in practice is to develop a methodology for quantifying the expected information for different sensing actions and evaluating them in a computationally feasible manner given limited *a priori* information.

Our approach is closest to Manyika [75], who formulated a normative approach to multi-sensor data fusion management based on Bayesian decision theory [16]. A utility structure for different sensing actions was defined using entropy (Shannon information) as a metric for maximizing the information in decentralized multi-sensor data fusion. The method was implemented for model-based localization of a mobile robot operating indoors using multiple scanning sonars and an *a priori* map. Feature location uncertainty and the loss of information due to vehicle motion error, which are encountered in CML, were not explicitly addressed.

Examples of the application of adaptive sensing in marine robotics include Singh [104, 103] and Bellingham and Willcox [14, 126]. Singh formulated an entropic measure of adaptive sensing and implemented it on the Autonomous Benthic Explorer. The implementation was performed using stochastic back-projection, a grid-based model-

ing technique developed by Stewart for marine sensor data fusion [113]. Bellingham and Willcox have investigated optimal survey design for AUVs making observations of dynamic oceanographic phenomena [14, 126]. Decreased vehicle speeds save power due to the quadratic dependence of drag on velocity, but susceptibility to space-time aliasing is increased.

An interesting motivation for adaptive sonar sensing is provided by the behavior of bats and dolphins performing echo-location. For example, dolphins are observed to move their heads from side to side as they discriminate objects [1]. Our hypothesis for this behavior is that sonar is more like touch than vision. A useful analogy may be the manner in which a person navigates through an unknown room in the dark. By reaching out for and establishing contact with walls, tables, and chairs, transitions from one object to the another can be managed as one moves across the room. Whereas man-made sonars tend to use narrow-band waveforms and narrow beam patterns, bat and dolphin sonar systems use broad-band waveforms and relatively wide beam patterns. A broad beam pattern can be beneficial because it provides a greater range of angles over which a sonar can establish and maintain "contact" by receiving echoes from an environmental feature. The task for concurrent mapping and localization is to integrate the information obtained from sonar returns obtained from different features as the sensor moves through the environment to estimate both the geometry of the scene and the trajectory of the sensor.

4.3 Adaptive augmented stochastic mapping

This section reviews the theory of augmented stochastic mapping and derives a metric for performing augmented stochastic mapping adaptively.

4.3.1 Augmented stochastic mapping

Stochastic mapping is a technique for performing CML that was introduced by Smith, Self, and Cheeseman [111] and is described in detail in Chapter 2. As mentioned in Chapter 2, stochastic mapping is simply a special way of organizing the states in an extended Kalman filter (EKF) for the purpose of feature relative navigation. An EKF (see Section 2.A) is a computationally efficient estimator for the states of a *given nonlinear dynamic system.* The assumptions are made that the noise processes are well modeled by Gaussian noise and the errors due to linearizations of the nonlinear system are small. The EKF for a system provides an *estimate*, $\mathbf{x}$, of both the (true) state of the system, $\mathbf{X}$, as well as an *estimate*, $\mathbf{P}$, of the (error) covariance. The covariance provides an estimate of the confidence in the estimate $\mathbf{x}$. A dynamic system is described by a dynamic model, $\mathcal{F}$, which defines the evolution of the system state $\mathbf{X}$ through time, and an observation model, $\mathcal{H}$, which relates observations (measurements) to the system's state $\mathbf{X}$.

By adding track[1] initiation, data association and track deletion to stochastic mapping, the technique of augmented stochastic (ASM) is obtained. We will here give a brief outline of ASM and the interested reader is referred to Chapter 2 for details. A somewhat different view on stochastic mapping than was presented in Chapter 2 is included to emphasize the information content of stochastic mapping for a more natural development of adaptive stochastic mapping. The vehicle model used in this chapter is slightly different from the one presented in Chapter 2 and is thus included below.

In stochastic mapping, the estimate of the robot's state, $\mathbf{x}_r$, and the features in the environment, $\mathbf{x}_1$ to $\mathbf{x}_N$, are represented by a single state vector, $\mathbf{x}_{k|k} = [\mathbf{x}_r^T \ \mathbf{x}_1^T \ldots \mathbf{x}_N^T]_{k|k}$, with an associated estimated error covariance $\mathbf{P}_{k|k}$ at each time step k. A subscript $k|j$ signifies an estimate at time step k given all information up to and including time step j. Time indexes are omitted where not crucial for clarity.

[1] A *track* is a feature estimate

As new features are added, $\mathbf{x}$ and $\mathbf{P}$ increase in size. The estimated error covariance, $\mathbf{P}_{k|k} = E\{(\mathbf{x}_{k|k} - \mathbf{X}_k)(\mathbf{x}_{k|k} - \mathbf{X}_k)^T\}$, of the system, where $\mathbf{X}_k$ is the *true* state at time step k, takes the form

$$\mathbf{P}_{k|k} = \begin{bmatrix} \mathbf{P}_{rr} & \mathbf{P}_{r1} & \cdots & \mathbf{P}_{rN} \\ \mathbf{P}_{1r} & \mathbf{P}_{11} & \cdots & \mathbf{P}_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{Nr} & \mathbf{P}_{N1} & \cdots & \mathbf{P}_{NN} \end{bmatrix}_{k|k} . \tag{4.1}$$

The sub-matrices, $\mathbf{P}_{rr}$, $\mathbf{P}_{ri}$ and $\mathbf{P}_{ii}$ are the vehicle-to-vehicle, vehicle-to-feature, and feature-to-feature covariances respectively. This form is significant as it allows us to separate the uncertainty associated with the robot, $\mathbf{P}_{rr}$, as well as the individual features, $\mathbf{P}_{ii}$, and this separation will be used in obtaining a measure of the information in our system.

Observations of features in the environment are related to the state by a state-to-observation transform $\mathbf{z}_k = \mathcal{H}(\mathbf{X}_k, \mathbf{d}_\mathbf{z})$, where $\mathbf{d}_\mathbf{z}$ is the measurement noise process. The *a posteriori* PDF of $\mathbf{X}_k$, given a set of measurements $\mathbf{Z}_k \equiv \{\mathbf{z}_k, \ldots, \mathbf{z}_1\}$, can be found from Bayes rule as

$$p(\mathbf{X}_k|\mathbf{Z}_k) = \frac{p(\mathbf{z}_k|\mathbf{X}_k)p(\mathbf{X}_k|\mathbf{Z}_{k-1})}{p(\mathbf{z}_k|\mathbf{Z}_{k-1})}. \tag{4.2}$$

The distribution $p(\mathbf{z}_k|\mathbf{X}_k)$ is defined as the likelihood function using the Likelihood Principle [16]. By knowing $p(\mathbf{X}_k|\mathbf{Z}_k)$ we can form an estimate $\mathbf{x}_k$ of the state $\mathbf{X}_k$.

In order to perform CML, the state transition (dynamic model)

$$\mathbf{X}_{k+1} = \mathcal{F}(\mathbf{X}_k, \mathbf{u}_k, \mathbf{d}_\mathbf{x}),$$

in addition to the observation transformation $\mathcal{H}$, must be known. Here $\mathbf{d}_\mathbf{x}$ is the process noise. If the stochastic processes $\mathbf{d}_\mathbf{z}$ and $\mathbf{d}_\mathbf{x}$ are assumed to be independent, white and Gaussian, and the state dynamic model $\mathcal{F}$ and observation model $\mathcal{H}$ are

both linear, the Bayes Least Square (BLS) estimator, $\mathbf{x}_{k+1|k} = \mathcal{F}(E\{\mathbf{X}_k|\mathbf{Z}_k\}, \mathbf{u})$, will be an *efficient* estimator of $\mathbf{X}$. However, in the general problem of CML, neither the dynamic model nor the observation model will be linear. Thus, an efficient estimator *cannot* be obtained. Further, propagating the system's covariance through nonlinear equations does not guarantee that the statistics will be conserved. Thus, in order to circumvent the problem of transformation of nonlinearities, the nonlinear models of $\mathcal{F}$ and $\mathcal{H}$ are approximated through a Taylor series expansion, keeping only the first two terms. That is,

$$\begin{aligned} \mathcal{F}(\mathbf{x}_k, u_k, \mathbf{d}_\mathbf{x}) &\approx \mathcal{F}(\mathbf{x}_{k|k}, \mathbf{u}_k, \mathbf{d}_\mathbf{x}) + \mathbf{F}_x \tilde{\mathbf{x}}_{k|k} \\ \Rightarrow \mathbf{x}_{k+1|k} &\approx E\{\mathcal{F}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_\mathbf{x})|\mathbf{Z}_k\}, \end{aligned} \tag{4.3}$$

where $\mathbf{F}_\mathbf{x} = d\mathcal{F}(\mathbf{x}, \mathbf{u}_k)/d\mathbf{X}|_{\mathbf{X}=\mathbf{x}_{k|k}}$ is the Jacobian of dynamic model $\mathcal{F}$ with respect to $\mathbf{x}$, evaluated at $\mathbf{x}_{k|k}$. Similarly, the observation model is approximated by

$$\mathcal{H}(\mathbf{x}_k, \mathbf{d}_\mathbf{z}) \approx \mathcal{H}(\mathbf{x}_{k|k-1}, \mathbf{d}_\mathbf{z}) + \mathbf{H}_x \tilde{\mathbf{x}}_{k|k-1}, \tag{4.4}$$

where $\mathbf{H}_\mathbf{x} = d\mathcal{H}(\mathbf{x})/d\mathbf{x}|_{\mathbf{X}=\mathbf{x}_{k|k-1}}$ is the Jacobian of the observation model $\mathcal{H}$ with respect to $\mathbf{x}$ evaluated at $\mathbf{x}_{k|k-1}$. These approximations are equivalent to the assumption that the estimated state at the previous time step, $\mathbf{x}_{k|k}$, is approximately equal to the *true* state, $\mathbf{X}_k$, at the previous time step. Once these linearizations have been performed and assuming that the approximation errors are small, we can now find the BLS estimator for this linearized system. The resulting procedure is that of the extended Kalman filter, described in detail in Appendix 2.A.

In the implementation in this chapter, we denote the vehicle's estimated state by $\mathbf{x}_r = [x_r \; y_r \; \phi]^T$ and the control input to the vehicle is given by $\mathbf{u} = [u_x \; u_y \; u_\phi]$. For

the dynamic model, $\mathcal{F}$, we use

$$\begin{aligned} \mathbf{X}_{r_{k+1}} =& \mathbf{f}(\mathbf{X}_k, \mathbf{u}_k) + \mathbf{G}(\mathbf{u}_k)\mathbf{d}_\mathbf{x} \\ =& \mathbf{X}_{r_k} + \mathbf{T}_{\phi_k}\mathbf{u}_k + \mathbf{G}(\mathbf{u}_k)\mathbf{d}_\mathbf{x}, \end{aligned} \tag{4.5}$$

where $\mathbf{G}(\mathbf{u}_k)$ scales the noise process $\mathbf{d}_\mathbf{x}$ as a function of the distance traveled, that is

$$\mathbf{G}(\mathbf{u}_k) = \begin{bmatrix} \gamma_x\sqrt{u_{x_k}^2 + u_{y_k}^2} & 0 & 0 \\ 0 & \gamma_y\sqrt{u_{x_k}^2 + u_{y_k}^2} & 0 \\ 0 & 0 & \gamma_\phi \end{bmatrix} \tag{4.6}$$

and

$$\mathbf{T}_{\phi_k} = \begin{bmatrix} \cos(\phi_k) & -\sin(\phi_k) & 0 \\ \sin(\phi_k) & \cos(\phi_k) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.7}$$

The ϕ in this expression comes from the robot's estimated state $\mathbf{x}_{r_k}$. The covariance of $\mathbf{d}_\mathbf{x}$ is for convenience set equal to the identity matrix, as any scaling is placed in $\mathbf{G}$. The matrix $\mathbf{G}$ is diagonal by assumption. If the correlations between the noise processes in the x, y, and ϕ direction were known, it should be included. However, this is a minor point and this form has has shown to work well for our systems.

Equation (4.5) does not take into account any of the vehicle's real dynamics. Thus the model is very general. However, if the vehicle's dynamics were known, they could be used to ensure that the vehicle movements remain realistic. In our experiments and simulations, we will constrain the robot to move only a certain distance each time step, thus making u_x and u_y dependent.

For the observation model we use

$$\mathbf{z}_k = \mathcal{H}(\mathbf{X}_k, \mathbf{d}_z) = \mathbf{h}(\mathbf{X}_k) + \mathbf{d}_z, \tag{4.8}$$

where $\mathbf{z}_k$ is the observation vector of range and bearing measurements. The observation model, $\mathbf{h}$, defines the nonlinear coordinate transformation from state to observation coordinates. The stochastic process $\mathbf{d}_z$, is assumed to be white, Gaussian, and independent of $\mathbf{x}_0$ and $\mathbf{d}_x$, and has covariance $\mathbf{R}$.

In our experiments, a sector including an object is scanned over multiple adjacent angles to yield multiple sonar returns that originate from the object. That is, isolated features and smooth surfaces appear as circular arcs (regions of constant depth [69]) in sonar scans. For the specular returns, an improved estimate of the range and bearing to the object is obtaining by grouping sets of adjacent returns with nearly the same range and taking the mode of this set of angles as the bearing measurement, and the range associated with this bearing as the range measurement to the object [83]. Rough surfaces [19] yield additional returns at high angles of incidence. These occur frequently in the data from our underwater sonar, but are not processed in the experiments reported in this chapter.

Implementing stochastic mapping with real data requires methods for data association and track initiation. For this reason, a logic based delayed nearest neighbor track initiator, a gated nearest neighbor data association filter and a logic based and consistency based track deleter were included. These are described in detail in Chapter 2 (Section 2.4).

4.3.2 Adaptation step

The goal of adaptive CML is to determine the optimal action given the current knowledge of the environment, sensors, and robot dynamics in a CML framework. To provide an intuitive understanding of this goal, imagine an underwater vehicles with

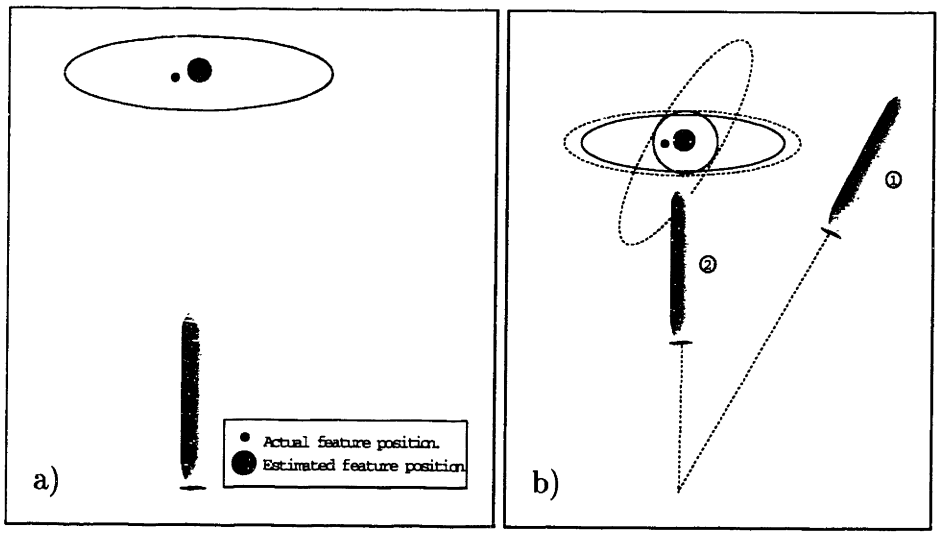

Figure 4-1: An autonomous underwater vehicle with no navigational uncertainty estimating the position of an environmental feature. The ellipses denotes the certainty (error ellipse) to which the feature position is known. **a)** The initial estimate of the feature. **b)** Given two possible new locations, ① and ②, to make the next observation from, position ① allows for a more accurate estimate of the feature's position, as the measurement of the feature is taken from a new angle, resulting in the tilted error ellipse associated with this measurement. Combining this (tilted error ellipse) with the error ellipse from a), the small dotted circular error ellipse is achieved. Taking the next observation from location ② only yields a slightly smaller error ellipse than that of the previous time step in a), shown by the dotted ellipse.

no navigational uncertainty estimating the position of a feature as depicted in Figure 4-1. As can be seen from this simple example of figure, it is clearly advantageous for the vehicle to take the next measurement from a new direction. By doing so, more information about the feature is extracted, and thus a better estimate can be obtained.

The essence of our model is to determine the action that maximizes the total knowledge (that is, the information) about the system in the presence of measurement *and* navigational uncertainty. By *adaptively* choosing actions, we mean that the next action of the robot is chosen so as to maximize the robot's information about its location and all the features' locations (the map).

The "amount" of information contained in Equation (4.2) can be quantified in various ways. Fisher information is of particular interest and is related to the estimate of the state $\mathbf{X}$ given the observations. The Fisher information for a *random* parameter[2] is defined [4] as the covariance of the gradient of the total log-probability, that is,

$$\begin{aligned} \mathbf{I}_{k|k} &\equiv E\{(\nabla_{\mathbf{x}} \ln p(\mathbf{X}, \mathbf{Z}_k))(\nabla_{\mathbf{x}} \ln p(\mathbf{X}, \mathbf{Z}_k))^T\} \\ &= -E\{\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T \ln p(\mathbf{X}_k, \mathbf{Z}_k)\}. \end{aligned} \tag{4.9}$$

Where $\nabla_{\mathbf{x}} = [\frac{d}{dx_1} \cdots \frac{d}{dx_N}]^T$ is the gradient operator with respect to $\mathbf{X} = [x_1 \cdots x_N]$, thus $\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T$ is the Hessian matrix. Applying Bayes rule to Equation (4.9), that is, $p(\mathbf{X}_k, \mathbf{Z}_k) = p(\mathbf{X}_k, \mathbf{z}, \mathbf{Z}_{k-1}) = p(\mathbf{X}_k, \mathbf{Z}_{k-1}) \cdot p(\mathbf{z}|\mathbf{X}_k)$, and noting the linearity of the gradient and expectation operators we obtain a Fisher information update:

$$\begin{aligned} E\{\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T \ln p(\mathbf{X}_k, \mathbf{Z}_k)\} =& E\{\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T \ln p(\mathbf{X}_k, \mathbf{Z}_{k-1})\} + E\left\{\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T \ln p(\mathbf{z}_k|\mathbf{X}_k)\right\} \\ \Leftrightarrow \quad \mathbf{I}_{k|k} =& \mathbf{I}_{k|k-1} - E\left\{\nabla_{\mathbf{x}} \nabla_{\mathbf{x}}^T \ln p(\mathbf{z}_k|\mathbf{X}_k)\right\}. \end{aligned} \tag{4.10}$$

The first term on the right represents the Fisher information, $\mathbf{I}_{k|k-1}$, before the last measurement, while the second term corresponds to the additional information gained by measurement $\mathbf{z}_k$.

If we obtain an efficient estimator for $\mathbf{X}$, the Fisher information will simply be given by the inverse of the error covariance of the state. Thus, under the assumption that Equations (4.3) and (4.4) hold, the inverse of the error covariance $\mathbf{P}$ will be an estimate for the Fisher information of the system. Under this assumption, and using Equation (4.5), the transformation $\mathcal{M}$ relating $\mathbf{I}_{k|k}$ to $\mathbf{I}_{k+1|k}$ can be found. By

[2]Strictly speaking, the Fisher information is only defined in the non-Bayesian view of estimation [4], that is, in the estimation of a *non-random* parameter. In the Bayesian approach to estimation, the parameter is *random* with a (prior) probability density function. However, as in the non-Bayesian definition of the Fisher information, the inverse of the the Fisher information for random parameter estimation is the Cramér-Rao lower bound for the mean square error.

combining this with Equation (4.10) we have a recursive Fisher information update, which depends on the actions $\mathbf{u}$ (inputs). $\mathcal{M}$ will generally depend on the state $\mathbf{X}_{k+1}$ as well, which is not available. By invoking the assumption of Equation (4.3), $\mathbf{X}_{k+1}$ is replaced by the estimate $\mathbf{x}_{k+1|k}$. Thus, $\mathcal{M}$ can be used to give us the optimal[3] action $\mathbf{u}_k$ to take given our model and assumptions.

At each time step the algorithm seeks to determine the transformation $\mathcal{M}$ and, from this, infer the optimal action $\mathbf{u}_k$. Combining the vehicle prediction and EKF update step of stochastic mapping, $\mathcal{M}$ is

$$\mathbf{I}_{k+1|k+1} = (\mathbf{F}_{\mathbf{x}}\mathbf{I}_{k|k}^{-1}\mathbf{F}_{\mathbf{x}}^{T} + \mathbf{G}(\mathbf{u}_k)\mathbf{G}(\mathbf{u}_k)^{T})^{-1} + \mathbf{H}_{\mathbf{x}}^{T}\mathbf{R}^{-1}\mathbf{H}_{\mathbf{x}}, \tag{4.11}$$

where $\mathbf{F}_{\mathbf{x}}$ and $\mathbf{H}_{\mathbf{x}}$ are the Jacobians of $\mathbf{f}$ and $\mathbf{h}$ with respect to $\mathbf{X}$ evaluated at $\mathbf{x}_{k|k}$ and $\mathbf{x}_{k+1|k}$ respectively. The first term on the right of Equation (4.11) represents the previous information of the system, as well as the loss of information that occurs due to the action $\mathbf{u}_k$. The second term represents the additional information gained by the system due to observations after the action $\mathbf{u}_k$. As this quantity is a function of $\mathbf{X}_{k+1}$, which is unknown, we approximate $\mathbf{X}_{k+1}$ by $\mathbf{x}_{k+1|k}$ by the assumption of Equation (4.3). The action that maximizes the information can be expressed as

$$\mathbf{u}_k = \max_{\mathbf{u}} \arg \mathbf{I}_{k+1|k+1} = \min_{\mathbf{u}} \arg \mathbf{P}_{k+1|k+1}. \tag{4.12}$$

The information is a matrix and we require a metric to quantify the information. Further, it is desired that this metric have a simple physical interpretation and that the metrics make intuitive sense. For example, using one of the standard measures of matrices, such as the determinant, trace or a standard matrix norm (1-norm, 2-norm, ∞-norm), makes little intuitive sense, as such a measure is a measure in the space of

[3]By optimal, we here mean that, under the assumption that the EKF is the best estimator for our state, the action $\mathbf{u}$ that maximizes the knowledge (information) of the system given the *current* knowledge of the system can be determined. This is *not* necessarily the optimal action for the *actual* system.

the dimension of the information matrix, while the actual CML problem in our case is a two dimensional problem. For instance, as a result of using the determinant as a measure, obtaining absolute knowledge of any state will result in the notation that the system has infinite information (exact knowledge), even though we might be very uncertain about the location of all of the features and the vehicle.

For concurrent mapping and localization, it is desirable to use a metric that makes explicit the tradeoff between uncertainty in feature locations and uncertainty in the vehicle position estimate. To accomplish this, we define the metric by a cost function $C(\mathbf{P})$, which gives the total area of all the error ellipses, (i.e., highest probability density regions) and is thus a measure of our confidence in our map and robot position. That is,

$$\begin{aligned} C(\mathbf{P}) &= \pi \prod_j \sqrt{\lambda_j(\mathbf{P}_{rr})} + \pi \sum_{i=1}^{N} \prod_j \sqrt{\lambda_j(\mathbf{P}_{ii})} \\ &= \pi \sqrt{\det(\mathbf{P}_{rr})} + \pi \sum_{i=1}^{N} \sqrt{\det(\mathbf{P}_{ii})} \ , \end{aligned} \tag{4.13}$$

where $\lambda_j(\cdot)$ is the j-th eigenvalue of its argument. Even though Equation (4.13) is necessarily the "best" measure, it is a measure that is simple and fulfills all the requirements given above.

The action to take is obtained by evaluating Equation (4.12) over the action space of the robot using the metric in Equation (4.13). This yields an adaptive stochastic mapping algorithm. This procedure optimizes the information locally at each time step, thus the adaptation step performs a local optimization. Notice that the action space of the robot is not limited to motion control inputs. Other actions and constraints can be readily included in the control input $\mathbf{u}$, such as, what measurements should be taken by the sonar. For this, we control the set of angles U_θ which are scanned by the sonar.

In principle Equation (4.12) can be solved in symbolic form using Lagrange multipliers. However, the symbolic matrix inversion required in Equation (4.11) is very tedious and results in a very large number of terms. Further, as more features are added, the inversion scales exponentially in the number of calculations. Numerical methods, however, can be used to evaluate Equation (4.12) quite efficiently. The experiments below do not use a numerical technique such as the simplex method to perform this optimization, however this could be readily incorporated in the future.

With the incorporation of the adaptive step, the adaptive augmented stochastic mapping algorithm is summarized as follows:

1. **state projection**: the system state (vehicle and features) is projected to the next time step using the state transition model $\mathcal{F}$, along with the control input $\mathbf{u}_k$;
2. **gating**: the closest feature to each new measurement is determined and gated with the feature and non-matching measurements are stored (Section 2.4);
3. **state update**: the EKF is used to update the estimated state vector, $\mathbf{x}$, when features are re-observed (Section 2.3.1);
4. **new feature generation**: new features are initialized using a delayed nearest neighbor data association strategy (Section 2.4);
5. **old feature removal**: out-of-date features are deleted by a logic based track deleter (Section 2.4); and
6. **adaptation step:** the next action, $\mathbf{u}_k$, to take is determined by optimizing Equation (4.13).

An outline of the algorithm for performing adaptive augmented stochastic mapping is shown in Figure 4-2.

4.4 Simulation results

The algorithm described above has been extensively tested in simulation. In these simulations, as well as in the experiments to follow, a numerical approximation was performed by evaluating Equation (4.13) at a fixed number of points in the action

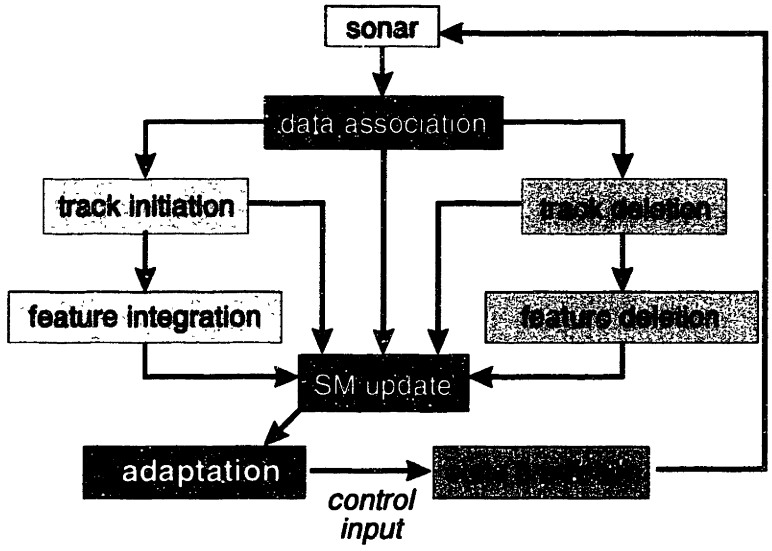

Figure 4-2: Structure of the adaptive augmented stochastic mapping (adaptive ASM) algorithm.

space. The robot was constrained to move a distance of 0, 10 or 20 cm at each time step, and could only turn in increments of 22.5°. Further, the vehicle was constrained to not get closer than 40 cm to the features (PVC tubes) as the sonar signal in this range becomes unreliable. In all these simulations, range error was assumed to have a standard deviation of 2 cm while the standard deviation for the bearing was 10°. Further, the sonar could move in increments of 0.9° between each measurement. Thus, a complete scan of 360° consisted of 400 sonar returns. The standard deviation for the vehicle odometry was set to 5% of the distance traveled. The angle uncertainty was set to 1°. These parameters were chosen as they resemble the situation for the air sonar experiments in Section 4.5.1. Two different types of simulation are described.

4.4.1 Adaptive vehicle motion

In these simulations, it was assumed that the robot stopped and took a *complete* 360° scan of the environment before continuing. The algorithm chose where to move adaptively. The algorithm took advantage of the fact that the measurements have different certainty in range and bearing, thus forming an error ellipse. Notice, however, that

if the observations of a target have nearly equal standard deviation in all directions, the robot will not move, as the loss of information from odometric error is larger than the information gained by taking a measurement from a different location.

Figure 4-3 shows the three typical paths of the robot in the presence of two features (8.4 cm radius PVC tubes) as a result of adaptation. The robot started at (0,0) and the paths (solid, dashed, and dashed-dotted lines) occurred with approximately equal frequency. The dotted lines around the PVC tubes denote the constraints placed on the robot for how close it could come to the PVC tubes while still obtaining valid sonar returns. (This is a limitation of the standard Polaroid sonar driver circuit). The resulting 95% confidence ellipses are drawn for the middle path, for the estimated position of the center of the features and for the robot's position. The robot's true position is indicated by a '+' while the estimated position is shown by a '×'. The robot stopped moving after about 15 time steps, as more information would be lost than gained by moving. Also notice that the robot moved larger distances in the beginning before taking the next scan than at the end of the run; more information was gained by moving and taking a measurement from a different location in the beginning than towards the end of a run.

Figure 4-4 shows the average, over 2000 runs, of the total cost (as defined by Equation (4.13)) of the system under adaptation as a percentage of the total cost of the system when moving randomly (solid line), or moving along the negative x-axis (dashed line) without adaptation. Moving along the negative x-axis is a worst case scenario and provides a lower bound on the performance of straight line movement. Moving in a straight line from the initial position between the two features would practically be identical to the adaptive solution, thus presenting an upper bound for straight line motion. In all the runs without adaptation, the robot moved a distance of 10 cm at each time step. As can be seen, the adaptation procedure obtains a cost of about 60% of the non-adaptive strategies after about 8 time steps. The lower cost signifies that the adaptive strategy has obtained more information about the

environment and have thus produced a more accurate estimate of the robot's position as well as the features' position. The random motion slowly started catching up after the 8th time step and by the 50th time step, it has achieved a cost of about 15% higher than that of the adaptive strategy. When moving along the negative x-axis, the cost actually starts increasing after about the 15th time step, as the robot was so far away from the features that it lost more information by moving than it gained by sensing at each time step. This is due to the poor angular resolution of the sonar.

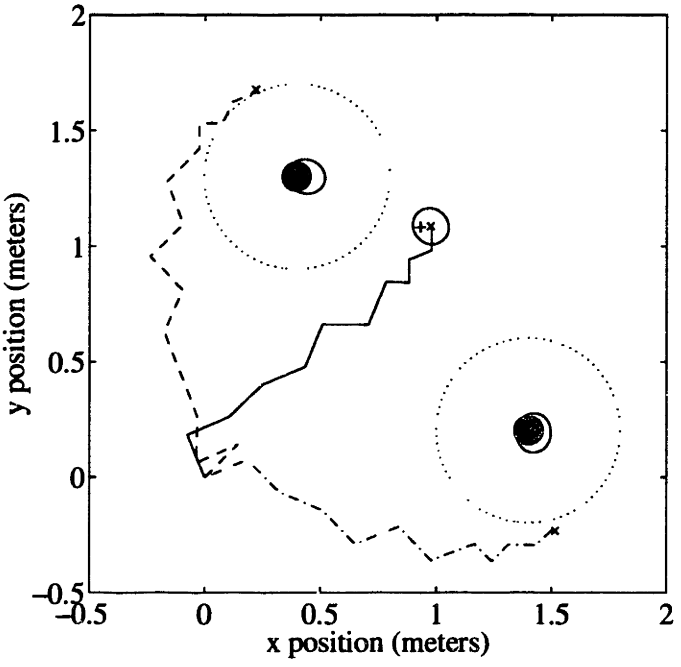

Figure 4-3: Three typical paths taken by the robot in simulations of adaptive ASM in the presence of two PVC tubes (filled circles) of radius 8.4 cm. Out of 1000 simulated runs, the robot chose to go to the left, to the right and through the middle with about equal frequency. The 95% confidence level of the map for the middle path is shown by the ellipses. (See text for symbols.)

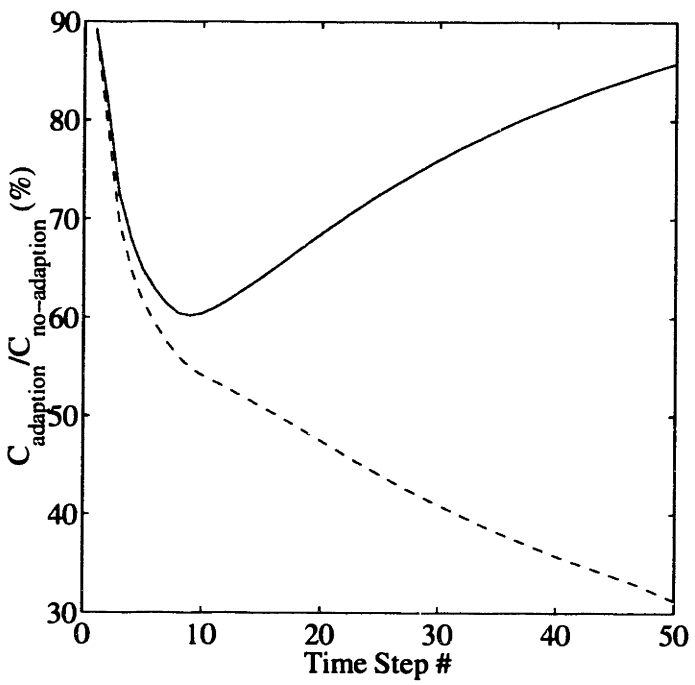

Figure 4-4: Simulation result showing the cost function $C(\mathbf{P})$ when performing adaptation divided by the cost function when *not* performing adaptation. Two cases in which no adaptation was performed are shown: the solid line denotes the case when the robot moved randomly, while the dashed line denotes the case when the robot moved along the negative x-axis.

4.4.2 Adaptive scanning and motion

In using sonar for mapping an environment, one is limited by the relatively slow speed with which measurements can be acquired. Thus, we imposed the additional constraint that the sonar could only scan an angle of 15° (that is, 50% more than the measurement bearing standard deviation) at each time step. The algorithm was required to decide where to direct the attention of the robot. The algorithm thereby adaptively decided where to look as well as where to move. This was implemented in the framework outlined above by adding an additional action u_θ to be controlled and solving Equation (4.13) given the constraints of the scan angle.

To compare the simulation results to the experiments and the previous simulations, the simulations were, as before, conducted over 50 time steps. However, under the adaptive strategy, only a 15° scanning angle was chosen at each time step. This is equivalent to obtaining 17 sonar returns. Without adaptation, a complete scan was taken at every time step, generating 400 sonar returns. Figure 4-5 shows the relative cost of adaptive sensing and motion with and without adaptation. The solid line denotes the case when the robot moved randomly without adaptation, while the dashed line denotes the case when the robot moved along the negative x-axis without adaptation as a function of sonar returns. The dotted vertical line indicates the point where the adaptive case was terminated as 50 time steps was reached. As can be seen from Figure 4-5, the adaptive method obtained a map with high confidence after relatively few sonar returns. After obtaining a total of 20000 returns, moving randomly and moving along the negative x-axis only achieved 93% and 33% of the confidence level that the adaptive method obtained with 850 sonar returns. Thus, the adaptive strategy required fewer measurements, and achieved a higher confidence level than any of the strategies without adaptation. The actual vehicle's motion was very similar to that shown in Figure 4-3.

Table 4.1 compares all the strategies on the basis of how many sonar returns and how many time steps are required before the map reached a specified confidence level.

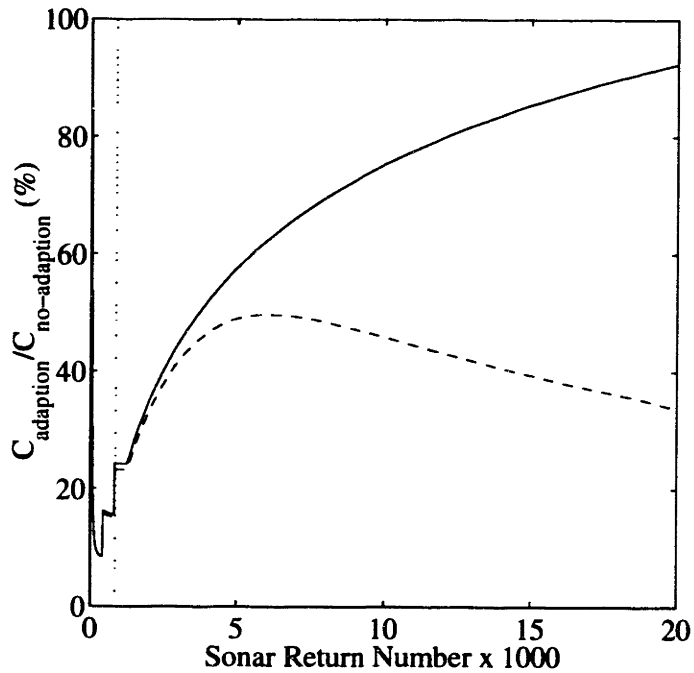

Figure 4-5: Advantage of adaptive sensing and motion control. The cost function when performing adaptation divided by the cost function when not performing adaptation is plotted. Two cases in which no adaptation was performed is shown: the solid line denotes the case when the robot moved randomly, while the dashed line denotes the case when the robot moved along the negative x-axis. The vertical dotted line indicates the point at which the adaptive method had completed its 50 time steps and terminated.

	$C = C_e$		$C = C_r$	
Strategy	Returns	Steps	Returns	Steps
Adaptive sensing and motion:	100	6	300	18
Adaptive motion:	1600	4	5200	13
No adaptation, random motion:	3200	8	20000	50
No adaptation, line motion:	3600	9	∞	∞

Table 4.1: Resources needed to achieve a given cost C in simulation. $C_e = 0.038\text{m}^2$ is the minimum cost achieved when moving in one direction during the experiment. $C_r = 0.0019\text{m}^2$ is the minimum cost achieved in the simulations when moving randomly.

As expected, the adaptive sensing and motion strategy required the fewest number of sonar returns to reach a given confidence level. However, it used more time steps

than the adaptive motion strategy alone, as under the adaptive motion and sensing strategy only one feature was measured at each time step, while under the adaptive motion strategy, both features were measured at each time step.

Figure 4-6 shows the sensor trajectory for a simulation involving eight objects. Figure 4-7 shows the east and north vehicle location errors and associated error bounds as a function of time during the simulation. After an initial loop around four of the objects is executed, the error bounds converge and the sensor wanders back-and-forth over a small area.

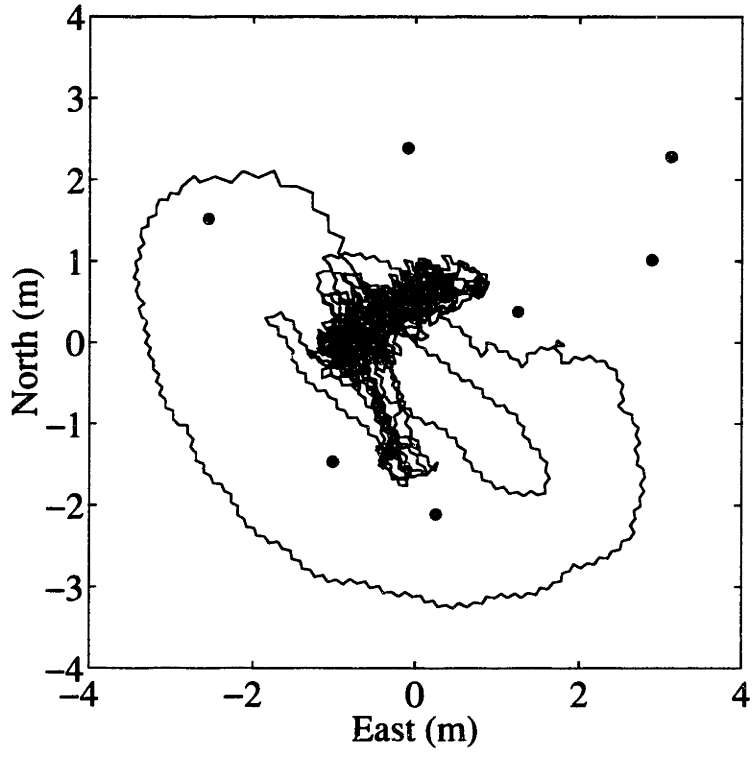

Figure 4-6: Path of vehicle performing adaptive motion among multiple objects.

4.5 Experimental results

Simulations provides an excellent tool for developing and testing of new algorithms as well as a good indication for the performance of a method in the real world. How-

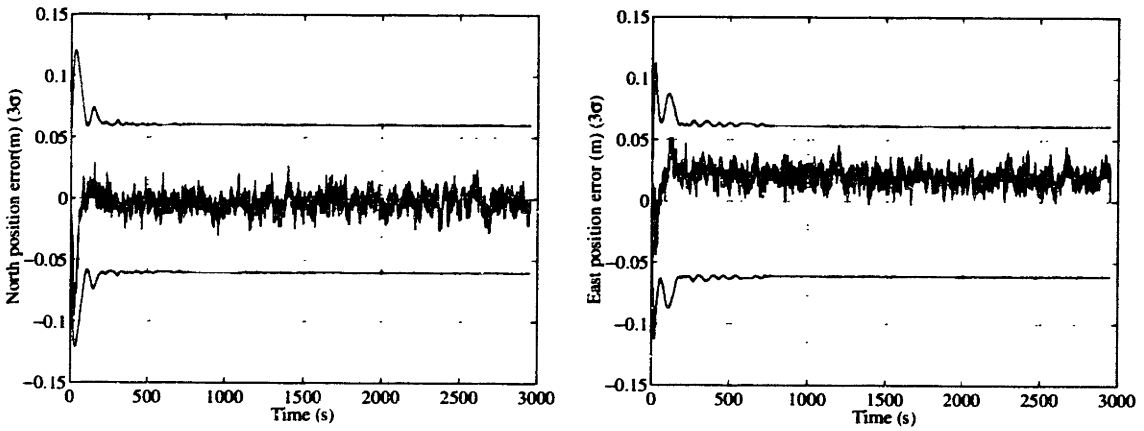

Figure 4-7: Position estimate and 3σ bounds for adaptive motion among multiple features. As can be seen, a steady state is reached after about 500 seconds.

ever, when operating in the real world there many events are difficult or impossible to incorporate in simulations. Thus, performing laboratory experiments gives valuable insight to a method and its validity. In the next two subsections, the adaptive augmented stochastic mapping developed in Section 4.3 is implemented on two experimental platforms: (1) a Nomad land robot, and (2) in a underwater testing tank. The experiments use the vehicle dynamic model of Equation (4.5) through Equation (4.7) and the adaption step is performed according to Equation (4.13).

4.5.1 Air sonar experimental results

The algorithm was implemented on a Nomad Scout robot [117] equipped with a 50 kHz Polaroid 6500 series ultrasonic sensor mounted on a stepping motor that rotated the sensor in 0.9 degree increments, as pictured in Figure 4-8. The error in the sensor and the vehicle odometry was assumed to be the same as that used in the simulations. The same constraints were employed as well. The sonar returns were assumed to be mainly specular, therefore regions of constant depth were extracted from the scans [69]. In these experiments, tracks were initiated from the first scan only, rather than using the DNN track initiator. Figure 4-9 shows a rough model

of the room and the configuration of the robot and features (PVC tubes of known radius) in the experimental setup. The dots indicate individual sonar returns of the Polaroid sensor from a complete scan of the environment. Each complete scan of the environment consisted of 400 sonar returns. In these experiments 15 motion steps were performed in each run.

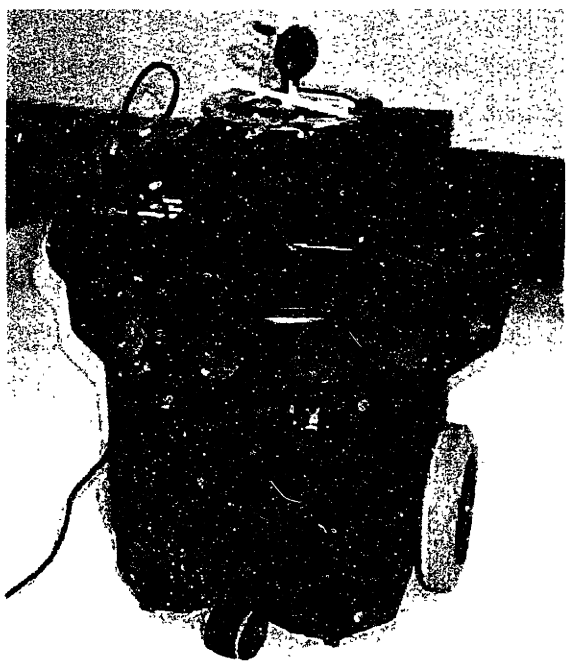

Figure 4-8: The Nomad Scout robot with the Polaroid ultrasonic sensor mounted on top.

Figure 4-10 shows the advantage of adaptation for a representative Nomad run, similar to the simulation of Figure 4-4. As can be seen from this figure, the advantage of performing adaptation is clear. Further, the experimental result was well within the one standard deviation bound of the simulated predicted result over 2000 runs.

Figure 4-11 shows the advantage of performing adaptive motion and sensing over moving in a straight line along the negative x-axis for the Nomad Scout robot (solid line). The simulated result is shown by a dashed line along with the one standard deviation bounds for 2000 simulated runs. As can be seen, the experimental cost ratio was within the one standard deviation of the simulated value. The adaptive strategy

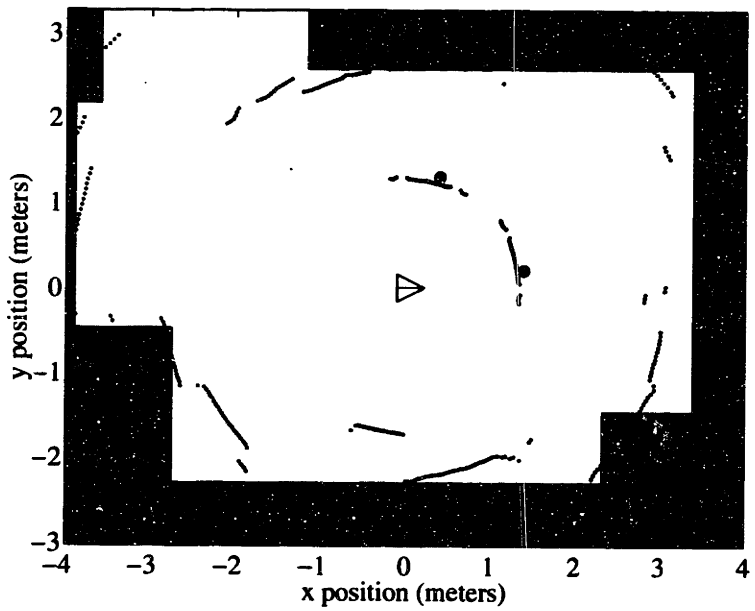

Figure 4-9: The returns from the Polaroid sensor (dots) with a rough room model superimposed. The robot is drawn as a triangle, the PVC tubes are indicated by small filled circles.

Strategy, $C = C_e$	Returns	Steps
Adaptive sensing and motion:	100	6
Adaptive motion:	2000	5
No adaptation, line motion:	6000	15

Table 4.2: Number of scans and time steps required to achieve a cost of $C_e = 0.038\text{m}^2$ or less for a representative experiment. The cost C_e was chosen to be the minimum cost achieved during the no adaptation strategy.

produced a high confidence map after relatively few measurements, consistent with the simulations in Figure 4-5. The non-adaptive method only reached a confidence level of 50% of the adaptive level, even after more than 20 times as many measurements were taken.

Table 4.2 shows the number of time steps and number of sonar returns required under each strategy to reach a map with some maximum specified cost C. As ex-

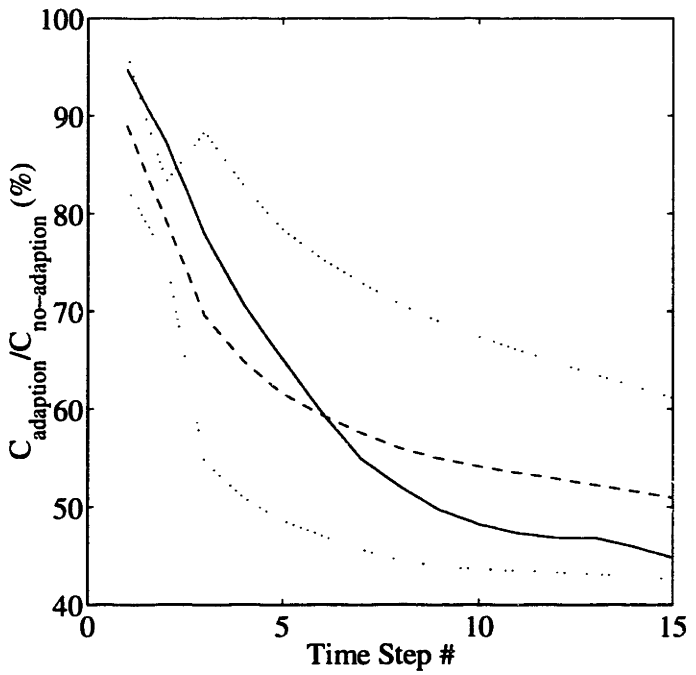

Figure 4-10: Comparison of cost function $C(\mathbf{P})$ with and without adaptation for a representative Nomad experiment. The solid line is the ratio of the cost when performing adaptation divided by the cost of moving in a straight line along the negative x-axis. The dashed line is the average simulation result over 2000 runs, with the dotted line denoting the one standard deviation bounds.

pected, the adaptive sensing and motion strategy required orders of magnitude less sonar returns than any of the other strategies. However, the adaptive motion strategy also used fewer time steps to reach a specified confidence level. Comparing this table for the experimental results to that of the simulation results of Table 4.1, we see that they are consistent.

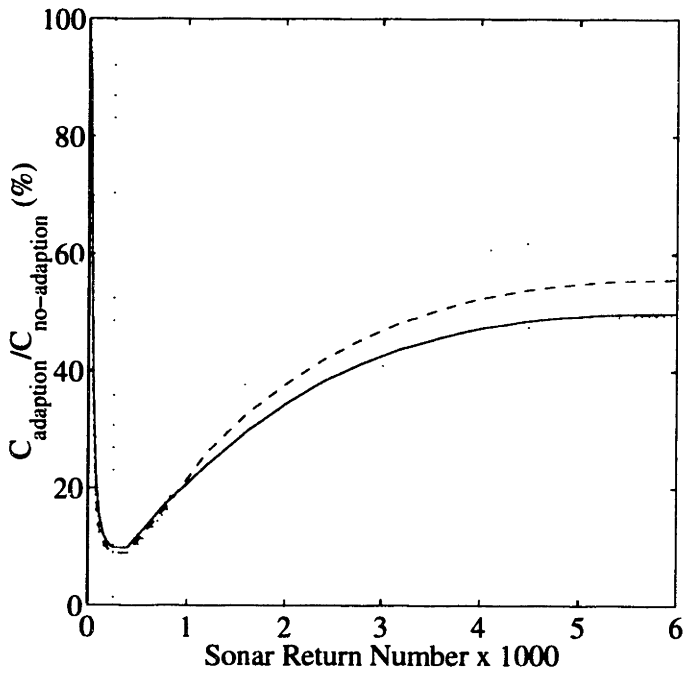

Figure 4-11: Advantage of adaptive sensing and motion in a representative Nomad experiment. The solid line is the cost of adaptive sensing and motion divided by the cost of moving in a straight line along the negative x-axis. The dashed line is the average of 2000 simulated runs and the dotted lines show the one standard deviation bound. The dotted vertical line indicates the time where the adaptive case terminated as 15 time steps were completed.

4.5.2 Underwater sonar experimental results

The second type of experiments conducted to test the adaptive stochastic mapping algorithm used a narrow-beam 675 kHz sector scan sonar mounted on a planar robotic positioning system, as shown in Figure 4-12. The positioning system was controlled by a Compumotor AT6450 controller card. The system was mounted on a 3.5 by 9.4 by 1.5 meter testing tank. The system executed on a PC running Matlab and C++ under WindowsNT. The C++ routines performed the interface to the sonar and the AT6450. The stochastic mapping algorithm was implemented in Matlab, which was also used for the the graphical user interface. The Matlab and C++ programs were integrated, resulting in a closed loop system for performing CML. At each time step, Equation (4.13) was minimized over the action space of the robot to choose the motion and scanning angles of the sensor.

The experiments were designed to simulate an underwater vehicle equipped with a sonar that can scan at any direction relative to the vehicle at each time step. Conducting complete 360° scans of the environment at every time step is slow with a mechanically scanned sonar and computationally expensive with an electronically scanned sonar. For these experiments, we envisioned a vehicle mounted with two sonars, one forward looking for obstacle avoidance, and one that can scan at any angle for localization purposes. The forward looking sonar was assumed to scan an angle of ±30°. The scanning sonar was limited to scan over $U_\theta = [-15°, 15°]$, at each time step. The scanning was performed in intervals of 0.15°. The sonars were modeled to have a standard deviation in bearing of 10.0° and 2.0 centimeters in range. Between each scan by the sonar, the vehicle could move between 15 cm and 30 cm. The lower limit signifies a minimum speed that the vehicle could move at before loosing controllability, while the upper limit signifies the maximum speed of the vehicle. The vehicle was constrained to only turn in increments of 22.5°. Further, we assumed that the vehicle was equipped with a dead-reckoning system with an accuracy of 10% of distance traveled and an accuracy of 1.0° in heading.

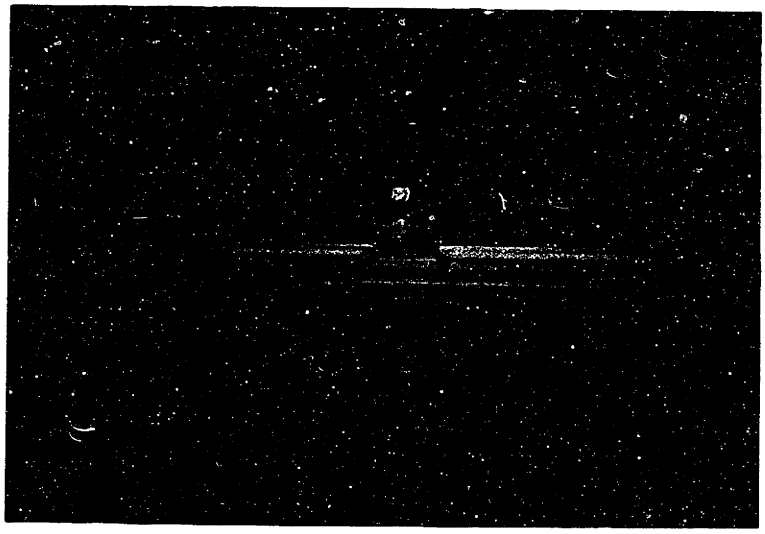

Figure 4-12: The planar robotic positioning system and sector-scan sonar used in the underwater sonar experiments. The water in the tank is approximately 1 meter deep. The transducer was translated and rotated in a horizontal plane mid-way through the water column.

Figure 4-13 shows a typical scan taken by the sonar from the origin. The crosses show individual sonar returns. The circles show the features (PVC tubes). The dotted circles around the features signify the minimum allowable distance between the vehicle and the features. The triangle shows the position of the sensor. Circular arc features were extracted from the sonar scans using the technique described in [83].

Figure 4-14 shows the sensor trajectory for a representative underwater sonar experiment with the complete algorithm. The sensor started at the origin and moved around the tank using the adaptive stochastic mapping algorithm to decide where to move and where to scan. Based on minimization of the cost function in Equation (4.13), the vehicle selected one target to scan to provide localization information. In addition, at each time step, the sonar was also scanned in front of the vehicle for obstacle avoidance. Figure 4-15 shows the x and y errors for the experiment and associated error bounds. No sonar measurements were obtained from approximately time step 75 to time step 85 due to a communication error between the sonar head and the host PC. Figure 4-16 shows the cost as a function of time. Solid vertical lines in

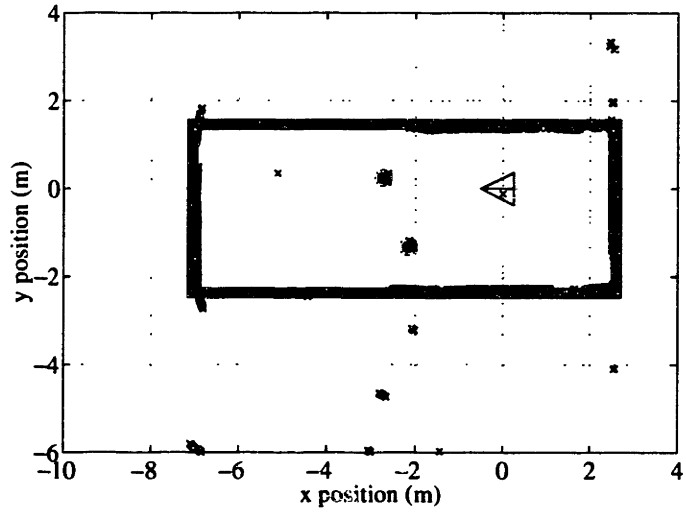

Figure 4-13: The returns from the underwater sonar for a 360° scan of the tank from the origin. The crosses shows individual returns. The small circles identify the position of the features (PVC tubes), with a dotted 5 cm outside circle drawn around them to signify the minimum allowable distance between the sonar and the features. The sonar was mounted on the carriage of the positioning system, which served as a "simulated AUV". The location of the sonar is shown by a triangle. The outline of the tank is shown in gray.

Figures 4-15 and 4-16 indicate the time steps when features of the environment were removed. Figure 4-17 plots the vehicle position error versus time for the stochastic mapping algorithm in comparison to dead-reckoning.

Figures 4-18 through 4-20 show the results of a non-adaptive experiment in which the vehicle moved in a straight line in the negative x direction with the two objects present throughout the experiment. Without adaptive motion, the observability of the features was degraded and the y estimate seems to diverge. Comparing Figure 4-19 with Figure 4-15 we observe that the non-adaptive strategy had a maximum 3σ confidence level of about 0.8 m and 0.2 m in the x and y positions respectively, while the maximum 3σ confidence level under adaption was about 0.17 m and 0.15 m in the y and x position respectively. A further indication of the advantage of the adaptive approach can be seen by comparing Figure 4-20 with Figure 4-17.

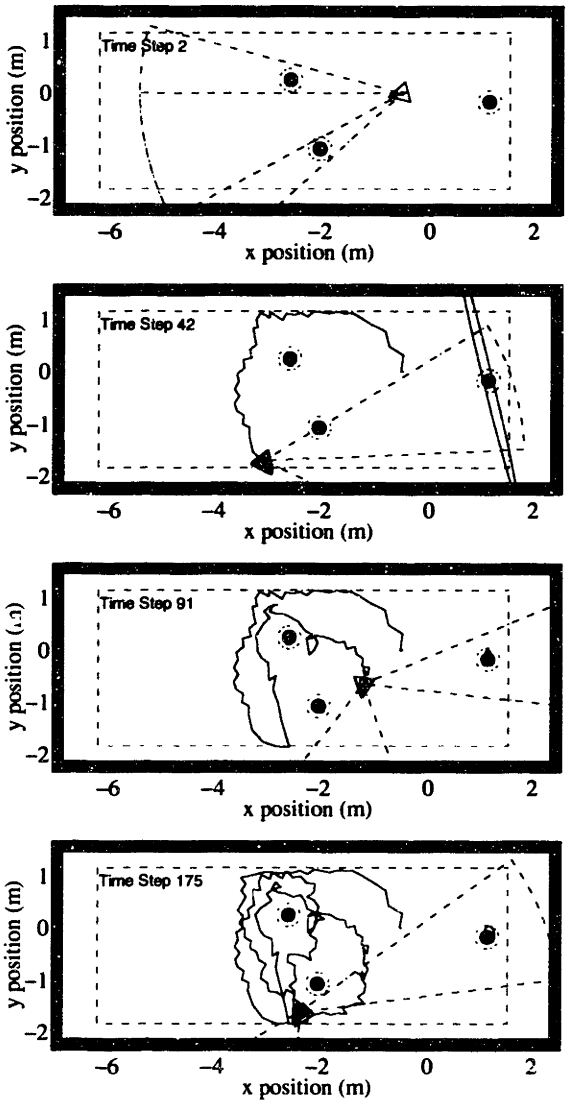

Figure 4-14: Time evolution of the sensor trajectory for an adaptive stochastic mapping experiment with the underwater sonar. The feature on the right was added at the 40th time step and the two left features were removed towards the end of the experiment. The vehicle started at the origin and moved adaptively through the environment to investigate different features in turn, maximizing Equation (4.13) at each time step. The filled circles designate the feature locations, and are surrounded by dotted circles which designate the "stand off" distance used by an obstacle avoidance routine. Similarly, the dashed-dotted rectangle designate the "stand off" distance to the tank walls. The dashed-dotted line represents the scanning region selected by the vehicle at the last time step. The large triangle designates the vehicle's position. The vehicle was constrained from moving outside the dashed-dotted lines to avoid collisions with the walls of the tank. Sonar returns originating from outside the dashed-dotted lines were rejected.

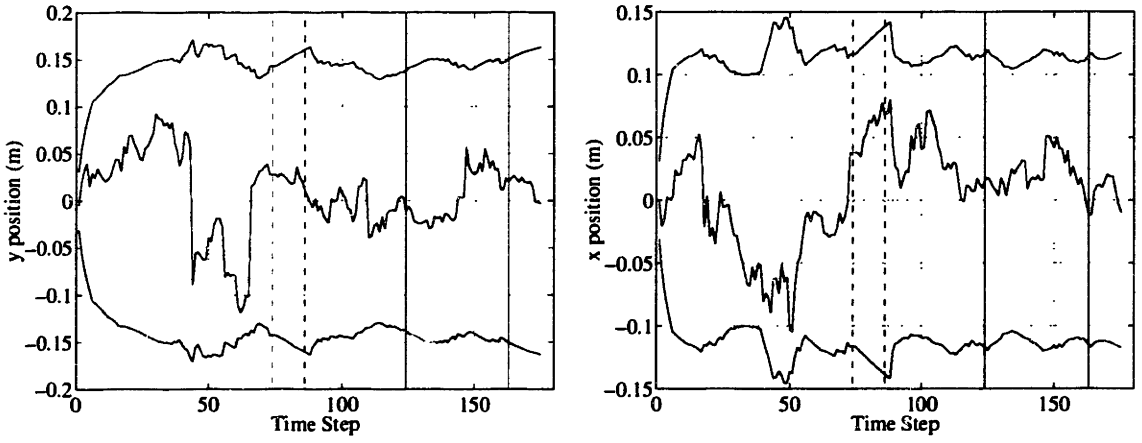

Figure 4-15: Position errors in the x and y directions and 3-σ confidence bounds for adaptive underwater experiment.

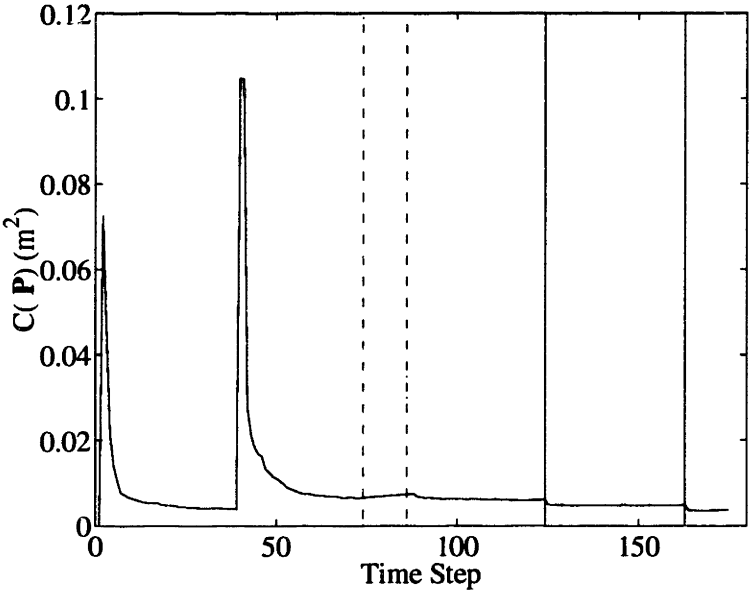

Figure 4-16: Cost as a function of time for adaptive underwater experiment. The cost increased at approximately the 40th time step when the third object was inserted into the tank and was observed for the first time. During the time interval between the two dashed vertical lines, no sonar data was obtained due to a serial communications problem between the PC and the sonar head. The two solid vertical lines designate the time steps during which features were removed from the tank, to simulate a dynamic environment.

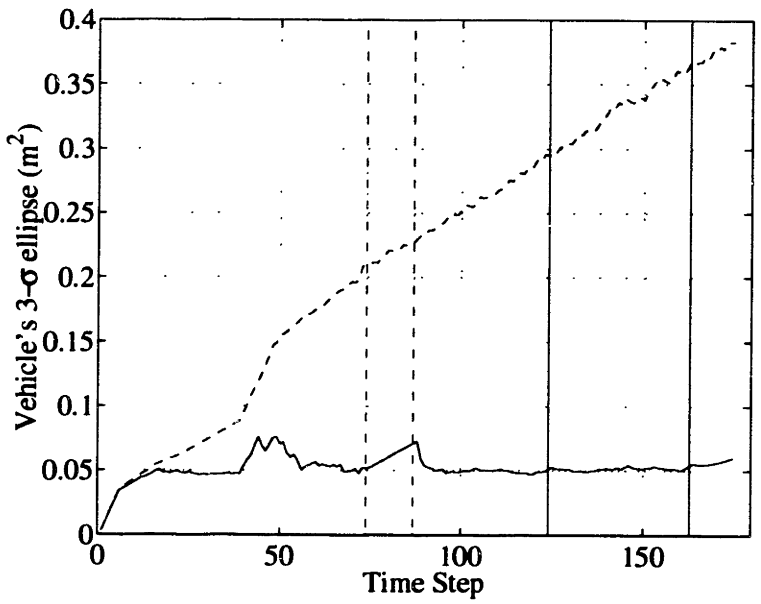

Figure 4-17: Vehicle position error versus time for dead-reckoning (dashed line) and the adaptive ASM algorithm (solid line). During the time interval between the two dashed vertical lines, no sonar data was obtained due to a serial communications problem between the PC and the sonar head. The two solid vertical lines designate the time steps when features were removed from the tank, to simulate a dynamic environment.

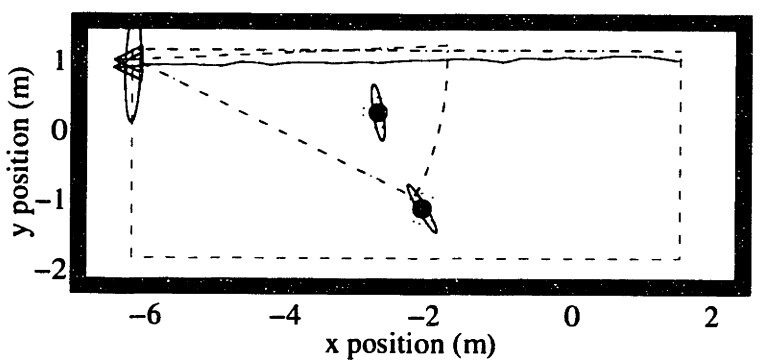

Figure 4-18: Sensor trajectory for a non-adaptive experiment in which the vehicle moved in a straight line. While accurate location information was obtained in the x direction, the CML process diverged in the y direction. The dashed-dotted rectangle indicates the "stand off" distance from the tank walls used by an obstacle avoidance routine.

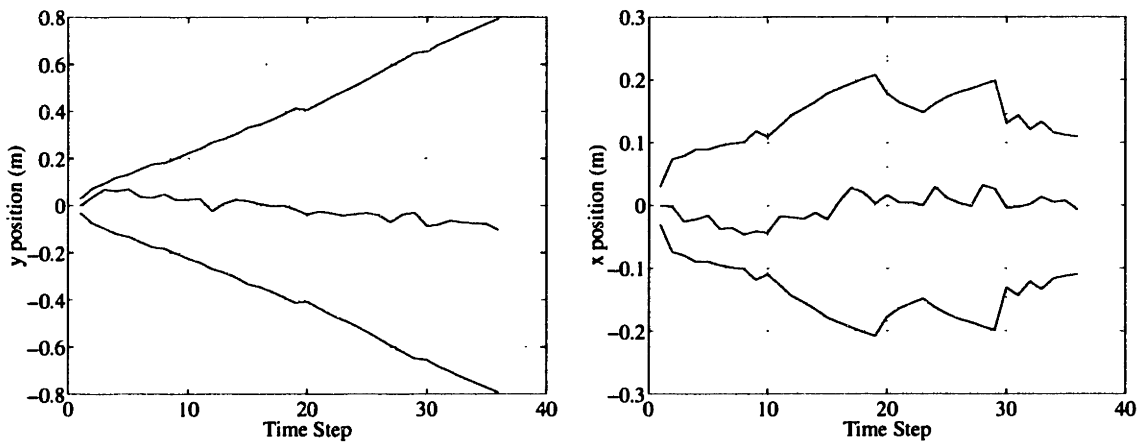

Figure 4-19: Position errors in the x and y directions and 3-σ confidence bounds for non-adaptive underwater experiment.

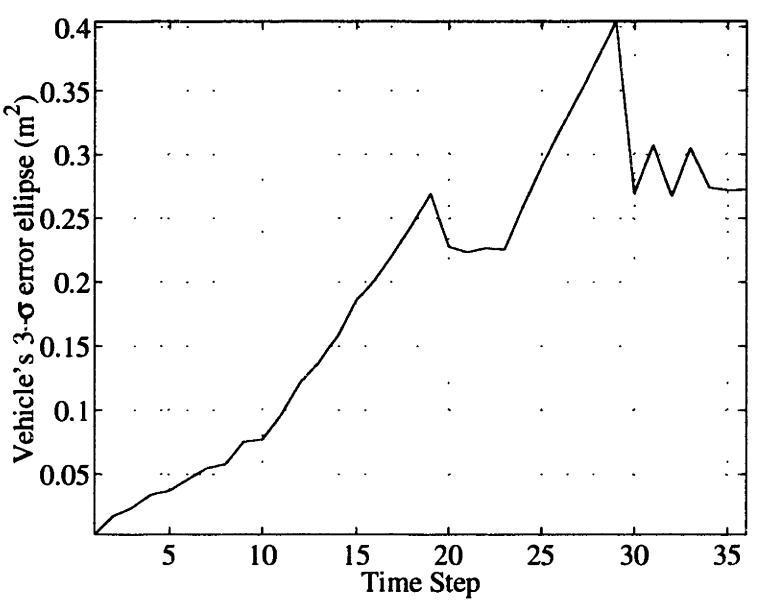

Figure 4-20: Vehicle error as a function of time for non-adaptive underwater experiment.

4.6 Conclusions

This chapter has considered the problem of adaptive action selection for feature-based concurrent mapping and localization for mobile robots. An adaptive action metric was incorporated within a stochastic mapping framework and used for adaptive action selection. The validity of the approach was tested via air and underwater experiments.

The proposed method enables adaptive concurrent mapping and localization (CML) in *a priori* unknown environments for any number of features. The adaptive method was based on choosing actions that, given the current knowledge, would maximize the information gained in the next measurement. This approach can easily be implemented as an extra step in a stochastic mapping algorithm for CML. The validity and usefulness of the approach were verified both in simulation and in experiments with air and underwater sonar data. Over a range of different operating conditions, the system exhibits a behavior in which it selectively explores different objects in the environment. This behavior is exhibited in Figure 4-21, which shows the sensor path for two different underwater sonar experiments under similar conditions.

Based on the air sonar experiments, we feel confident in the accuracy of the simulation in predicting experimental outcomes. For example, the three typical paths for the robot shown in Figure 4-3 are representative of both adaptive simulations and real data experiments, and the plots comparing the performance with and without adaptation are similar. The advantage of performing adaptive CML is notable when only adapting the motion of the vehicle (Figures 4-4 and 4-10). However, more substantial gains are obtained when performing adaptive motion *and* sensing. This is apparent from Figures 4-5 and 4-11, where the number of sonar returns required to obtain a given confidence level is an order of magnitude fewer than when not performing adaptation.

The adaptive sensing technique employed here is a *local* method. At each cycle, only the next action of the robot is considered. By predicting over an expanded time horizon, one can formulate global adaptive mapping and navigation. For example,

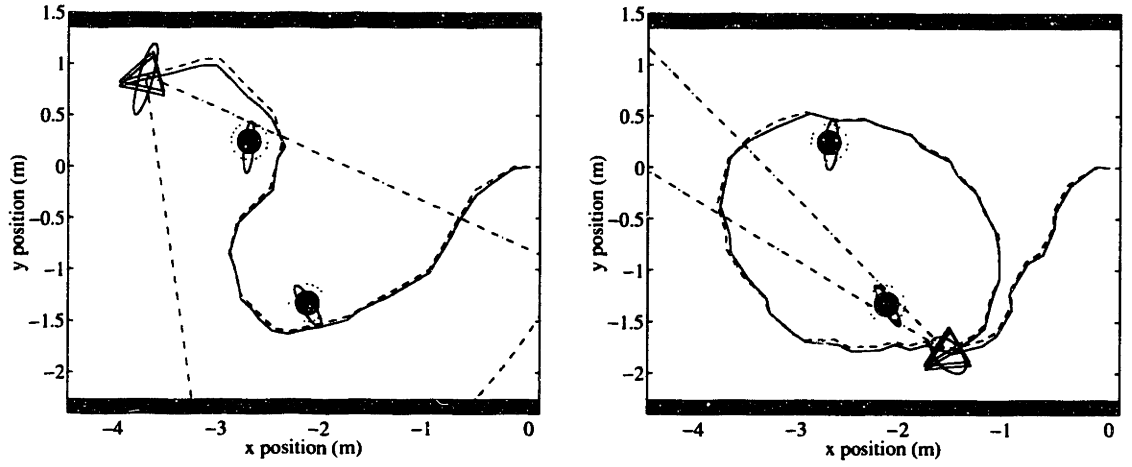

Figure 4-21: Two representative stochastic mapping experiments that exhibited adaptive behavior. In each figure, the solid line shows the estimated path of the sensor and the dashed line shows the actual path. The triangle indicates the final position of the sensor. The filled disks indicate the locations of the features (PVC tubes). The ellipses around the features and the sensor are the 3 σ contours, that is, the 99% highest confidence regions. The sonar view is indicated by the dashed-dotted line. In the left figure, the sensor had a scanning angle of $[-30°, 30°]$. The sensor started at (0,0) and then adaptively determined the path to take as well as the direction to scan, resulting in "exploratory" behavior. The sensor first moves over to one of the objects, turns, and moves around the second. The experiment illustrated in the right figure was similar, with the exception that the sonar was only able to scan n area of $[-7.5°, 7.5°]$ each time step. Again, we can see that exploratory behavior emerged as the sensor attempted to maximize the information it obtained about the environment. In these experiments, tracks were initiated from the first scan only, rather than using the DNN track initiator. In addition, the robot was constrained to turn a maximum of 30° at each time step in 15° increments.

one can consider how to determine the best path between the robot's current position and a desired goal position. Such methods, however, tends to cause the space of possible actions to become extremely large, and a computationally efficient method for searching the action space will be essential. Using the method proposed here will clearly rapidly become intractable. For example, if 20 possible actions are considered at each time step, 20^k actions must be considered at the k-th time step. Thus, a combination of the current method with a form of cost function related to the

distance to the goal might well be appropriate. For instance, one could perform adaptation only if the adaptation step causes at least a p-percentage improvement over moving towards the goal. One could also imagine a strategy for identifying the steady state of the system (which occurred after about 500 seconds in Figure 4-7), in which adaptation halted and the robot concentrated on reaching the goal position.

While this chapter has focused on an adaptive sensing metric for improved state estimation performance in CML, we believe that other types of adaptive strategies can help answer many of the important problems in mobile robot navigation. Some open issues are the development of adaptive strategies to identify environmental features useful as navigational landmarks, operation in dynamic environments and the integration of CML with a path-planner.

In conclusion, adaptive sensing can save time and energy, reduce the amount of data that needs to be acquired, improve state estimation robustness, ease data association ambiguity, prevent divergence, and facilitate recovery from errors.

4.7 Summary

This chapter has addressed the problem of how to perform concurrent mapping and localization (CML) adaptively using sonar. A metric for adaptive sensing which is defined in terms of Fisher information and represents the sum of the areas of the error ellipses of the vehicle and feature estimates in the map was introduced. Predicted sensor readings and expected dead-reckoning errors are used to estimate the metric for each potential action of the robot, and the action which yields the lowest cost (i.e., the maximum information) is selected. This technique has been demonstrated via simulations, in-air sonar experiments, and underwater sonar experiments. The vehicle tends to explore selectively different objects in the environment.

The technique of adaptive CML is a promising venue for increased CML performance. Through adaptive CML, the vehicle autonomously decides where to move.

However, moving in a straight line to the goal position might cause collisions with other objects in the environment. The question of path-planning is thus addressed in the next chapter.

Chapter 5

Path planning in dynamic environments

Path planning is a key technology for the safe autonomous operation of mobile robots. Motivated by fluid analogies, artificial harmonic potentials can eliminate local minima problems in planar path planning. In this chapter, simple analytical solutions to harmonic potentials are derived using tools from fluid mechanics, and are applied to planning among multiple objects in dynamic environments. These closed-form solutions enable real-time computation of paths from local information.

5.1 Introduction

The previous chapters have addressed navigation techniques for mobile robots. In particular, the focus has been on feature based approaches to mapping an *a priori* unknown environment while concurrently using that map to localize the robot. The success of these approaches enables the robot to answer the questions "what is my map?" and "where am I?". However, once answers to these questions are found, the next challenging problem for the robot is to find a collision free path to its goal. The field of mobile robot path planning addresses this problem, thereby providing

the robot with answers to the question "how do I get there?". The path planning problem becomes particularly challenging in dynamic environments. Mobile robot path planning in dynamic environments is the focus of this chapter.

The problem of path planning for a robot that avoids collision with objects in a known environment has been studied extensively. Such a map can be obtained by the CML techniques described in earlier chapters. In path planning, the task is to find a trajectory which will bring a mobile robot from an initial position $\mathbf{x}_i$, to the final position $\mathbf{x}_f$, while avoiding obstacles. In this chapter we will limit ourself to path planning in two-dimensions.

Approaches to the path planning problem can be divided into global and local algorithms. Several methods using a combination of both have also been suggested. In such methods, either a local method is assisted by a global planner when needed (that is, when the robot gets stuck) as in [94], or paths from all points in space to the goal are defined by some potential field method [59]. In the potential field methods, the position and shape of all obstacles in a given region are assumed to be known, for example through CML, and the potential function is constructed using this information. Thus, only local calculations at each point in space are required for the robot to find the direction it should move. Initially, potential field methods had the drawback that a robot could get stuck in local minima [59, 91, 28]. This problem has in recent years been solved by using potentials that have their local extrema on the boundaries of obstacles by the use of harmonic potentials, that is, functions satisfying Laplace's equation, $\nabla^2\phi = 0$, often motivated by a fluid or electrostatic analog [57, 116, 33, 32]. The disadvantage of these methods is that the Laplacian has to be solved over the whole state space, and the convergence when solved numerically is very slow. On an $L \times L$ grid the computation time scales as L^4, thus making it difficult to find solutions in real-time for dynamic environments.

Two notable exceptions to the use of numerical solutions of harmonic potentials are described in references [49, 60]. Guldner and Utkin [49] use analytical solutions

for objects having a simple shape, making it computationally inexpensive. However, they only consider static environments and only the object closest to the robot. This has the drawback that the robot is not influenced by any other objects in space. Kim and Khosla [60] use the panel method from computational hydrodynamics. This gives an approximate closed-form solution over the entire space given an arbitrarily shaped polygon, however, they restrict themselves to static environments.

In this chapter, we extend the potential field method to *dynamic* environments for real-time path planning in two dimensions. We also introduce analytical solutions for multiple objects in dynamic environments.

In the Section 5.2 we draw the analog between fluid flow and path planning in two dimensions. We also introduce several methods for defining objects in an analytical form, and a new method for defining the goal position of the robot. Section 5.3 extends the method to dynamic environments, while Section 5.4 presents concluding remarks.

5.2 The fluid analogy

Harmonic potentials have the great advantage that they achieve their extremum only on the boundary of objects. Thus no local minima will occur in the admissible configuration space, and a path generated by following a steepest gradient descent is guaranteed to reach the goal without hitting any objects in the domain, if the goal is reachable. This property of harmonic potentials is the extremum property [82]. Such a path is in fluid mechanics termed a streamline, and defines the path of a fluid particle. (There is one streamline for every object in a flow where the velocity vanishes, termed the stagnation line and the velocity vanishes at the stagnation point on the boundary of the object. However, this line has zero measure, and the stagnation point is a saddle point, so these points, in the absence of large friction, do not cause a problem when using streamlines as robot paths.) In using harmonic potentials in two-dimensional

path planning it is useful to draw the analog to invicid incompressible irrotational fluid flow, termed *potential flow*, and thereby use different powerful tools and properties from hydrodynamics [57, 60, 82, 58] and intuition. Let the fluid velocity at $\mathbf{x}$ be $\mathbf{U}(\mathbf{x})$. The assumption that the flow is irrotational states that the vorticity vanishes, that is:

$$\nabla \times \mathbf{U} = 0 \quad \Leftrightarrow \quad \mathbf{U} = \nabla\phi(\mathbf{x}) \,, \tag{5.1}$$

where $\phi(\mathbf{x})$ is a velocity potential. Since the fluid is assumed to be incompressible, the continuity equation can be written:

$$\nabla \cdot \mathbf{U} = 0 \,. \tag{5.2}$$

Combining Equation (5.1) and (5.2) we arrive at the Laplace equation for the velocity potential:

$$\nabla^2\phi = 0 \,. \tag{5.3}$$

On obstacle boundaries, the boundary condition of Laplace's equation is given by the impenetrability of obstacle boundaries, called von Neumann boundary conditions, which can be expressed as

$$\mathbf{n} \cdot \mathbf{U} = \mathbf{n} \cdot \nabla\phi = 0 \,, \tag{5.4}$$

where $\mathbf{n}$ is the unit normal vector on the boundary. In addition, there are different flows describing potential flows, such as, sources, sinks, and uniform flows. Potential flow represents an *ideal* fluid flow, where viscosity is ignored.

In Section 5.2.1 we describe several basic methods for defining objects in an analytical form. In Section 5.2.2 we describe our methods and approximations made for using closed form solutions of Laplace's equation to handle multiple objects. Sec-

tion 5.2.3 introduces a new method for defining the initial position and the goal position of the robot.

5.2.1 Modeling of objects

In this section we outline the different methods used for modeling of objects in potential flows.

From standard fluid texts on potential flows, such as the book by Milne-Thomson [82], solutions for various shaped objects can be found in closed form. In fluid dynamics, it is convenient to work with *complex* potentials, rather than just the velocity potential in order to utilize the properties of conformal mapping of complex variables for two-dimensional problems. The complex potential consists of the velocity potential $\phi(\mathbf{x})$, as defined above, and a function called the stream function $\psi(\mathbf{x})$, which is constant on a path of a fluid particle, that is, on a streamline. The complex potential is defined as

$$w = \phi + i\psi = f(z) \,, \quad z = x + iy \,; \tag{5.5}$$

where $i^2 = -1$.

The velocity field, $\mathbf{U} = [u \;\; v]$ can be found from either ϕ, ψ or w by

$$\begin{aligned} u &= \frac{\partial \phi}{\partial x} = \frac{\partial \psi}{\partial y} = -\Re\left(\frac{dw}{dz}\right) , \\ v &= \frac{\partial \phi}{\partial y} = -\frac{\partial \psi}{\partial x} = \Im\left(\frac{dw}{dz}\right) . \end{aligned} \tag{5.6}$$

We here outline the four major methods to define various potential flows: simple flows, use of specific theorems, conformal mapping and a panel method. We here use U to represent a velocity and r to represent a length.

1. SIMPLE FLOWS: The potential for some simple flows can be found by trial and

error, such as that of a uniform flow

$$w = -Uze^{-i\alpha} . \tag{5.7}$$

Using Equation (5.6) we find $u = U\cos\alpha$, $v = U\sin\alpha$, which we see is the uniform flow of magnitude U making an angle α with the positive x-axis.

2. SPECIFIC THEOREMS: Laplace's equation has been studied extensively, as it appears in many physical problems. Thus, many special theorems for modeling of objects apply. For example, the complex potential

$$w = -U\left(ze^{-i\alpha} + \frac{r^2e^{i\alpha}}{z - z_0}\right) . \tag{5.8}$$

represent the flow past a circular cylinder centered at $z_0 = x_0 + iy_0$, with radius r in a uniform stream with velocity U inclined at an angle α with the positive x-direction. This result follows directly from the circle theorem described in [82].

3. CONFORMAL MAPPING: A powerful technique for obtaining flows around objects of more complex shape in two dimension, is by the use of conformal mappings [82, 87, 48]. One of the most useful mappings is the Joukowski transformation:

$$z = \xi + \frac{r^2}{4\xi} , \tag{5.9}$$

by which we can map the ξ-plane to the z-plane and vice versa. If we take a circle of radius $r = \frac{1}{2}(a + b)$ in the ξ-plane, this transformation will map the circle into an ellipse with major axis a and minor axis b in the z-plane. That

is, the circle in the ξ-plane is given by Equation (5.8) as:

$$w = -U\left(\xi e^{-i\alpha} + \frac{(a+b)^2 e^{i\alpha}}{4\xi}\right) , \tag{5.10}$$

and solving Equation (5.9) for ξ we get

$$\xi = \frac{1}{2}(z \pm \sqrt{z^2 - r^2}) , \quad r^2 = a^2 - b^2 . \tag{5.11}$$

From these two equations we can find dw/dz and thus the velocity field around an ellipse.

4. PANEL METHOD: In panel methods [60, 31, 84] a body of unspecified shape may be generated by adding to a uniform flow a linear combination of singularities including sources, sinks, doublets, and vortices. This is done by approximating the shape of the object by a finite number of line segments in two-dimensions (three dimensional panel methods also exist), called *panels*, each of which consists of a uniform distribution of singularities of a certain kind. We here only outline the panel method based on a uniform distribution of sources, first introduced by Hess and Smith [53], and refer the reader to the cited books and articles for derivations of other panel methods.

   In this method, the potential ϕ_j of panel j is given by its midpoint (a_j, b_j), the angle θ_j it makes with the oncoming uniform stream, the panel length $2L_j$ and the panel strength λ_j by:

$$\phi_j(x,y) = \frac{\lambda_j}{4\pi}\int_{-L}^{L} \ln[(x-x_j)^2 + (y-y_j)^2]^2 dl_j , \tag{5.12}$$

   where $x_j = a_j + l_j \cos\theta_j$ and $y_j = b_j + l_j \sin\theta_j$.

   The N panels constituting a polygon are numbered in a *clockwise* direction. Each of the sides of the polygon is made up of one or more panels. In practice, the largest panel has the length of the minimum thickness of the polygon. Each

panel has a common end point with each of its two neighboring panels. Thus, the potential for the polygon in the uniform flow, ϕ_u, at point (x, y) is:

$$\phi(x,y) = \phi_u + \sum_{j=1}^{N} \phi_j \, . \tag{5.13}$$

The task is to determine $\lambda_1, ..., \lambda_N$. This is done by solving N linearly independent equations given by the requirement that the outward normal velocity is zero (i.e. Equation (5.4)) on the midpoint, (a_j, b_j), of each of the N panels. Hence, the N equations are given by

$$\frac{\partial}{\partial \mathbf{n}_j} \phi(x,y)|_{(a_j,b_j)} = 0 \, , \quad \text{for } j = 1, ..., N \, , \tag{5.14}$$

We also find that the method introduced by [60] of having $\mathbf{n} \cdot \mathbf{U} \geq 0$ on the boundaries instead of Equation (5.4) to be appropriate. Notice that the von Neumann boundary condition is only satisfied at the midpoint of each panel, thus the method is only approximate. When all λ's have been determined, the velocity field (without the uniform flow) is given by:

$$\begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \begin{bmatrix} \partial/\partial x \\ \partial/\partial y \end{bmatrix} \sum_{j=1}^{N} \phi_j = \sum_{j=1}^{N} \frac{\lambda_j}{4\pi} R_{\theta_j}^{-1} \begin{bmatrix} 2\tan^{-1}\left(\frac{y_r+L_j}{x_r}\right) - 2\tan^{-1}\left(\frac{y_r-L_j}{x_r}\right) \\ \ln\left(\frac{x_r^2+(y_r+L_j)^2}{x_r^2+(y_r-L_j)^2}\right) \end{bmatrix} ,$$

where

$$\begin{bmatrix} x_r \\ y_r \end{bmatrix} = R_{\theta_j} \begin{bmatrix} x - a_j \\ y - b_j \end{bmatrix} , \; R_{\theta_j} = \begin{bmatrix} \sin\phi_j & -\cos\phi_j \\ \cos\phi_j & \sin\phi_j \end{bmatrix} .$$

For not very complex shapes, less than $N = 20$ panels gives a very good ap-

proximation and keeps the computational cost low.

5.2.2 Multiple objects

All the methods for modeling of objects outlined in section 5.2.1, with some exceptions, only give the analytical solution for *one* object in a uniform flow. Due to the linearity of Laplace's equation, the superposition of several different solutions will also be a solution. However, the superposition of solutions to Laplace's equation for objects in a flow will deform the contour where Equation (5.4) is satisfied from the actual boundaries of the objects, which again may cause intersection of robot path (that is, streamlines) and the objects. Thus, modifications have to be performed in order to use the methods of section 5.2.1 for robot path planning. We here present a new method for path planning in an environment with several objects where the path is influenced by all known objects in the environment, and allows for moving obstacles and obstacles that change size.

This problem can be solved as follows: If the robot is very close to an object the robot must first of all avoid that object. This can be achieved by considering the flow field near an object as the resulting flow around the object in a fictitious uniform flow, for which the solution is known exactly. When the robot is far away from any objects, the superposition of solutions is close to the exact solution, while superposition and the extremum property ensures that Laplace's equation is satisfied and that the robot will not get stuck in a local minimum. In between these two regions, a transition from the field close to the object and far from the object is used. Thus, by this method, a harmonic potential is obtained that satisfies the boundary conditions. However, this potential is an approximate solution to the potential flow. This approximation does not cause a problem as the analog to potential flow is merely a tool for intuition, and gives us tools from fluid mechanics for solving Laplace's equation.

More explicitly, let $\phi_{i-1} = \phi_e + \sum_k^{i-1} \phi_{o_k}$ define the velocity potential before the introduction of object i. This potential is composed of ϕ_e which is the "external"

field which will define the goal position, and $\sum_k^{i-1} \phi_{o_k}$ which defines the potential for all the $i-1$ objects that already exist. Let ϕ_{u_i} be the potential of a uniform flow, and let ϕ_{ou_i} be the potential which defines the flow exactly around the object in the uniform flow ϕ_{u_i}. Then $\phi_{o_i} \equiv \phi_{ou_i} - \phi_{u_i}$ is the potential associated with the object in the absence of the uniform flow. We first place a region around the object where we impose that the potential is ϕ_{ou_i} (region A). In implementation, this region is typically chosen to be of width equal to the distance a robot can move in a little more than one time step around the objects' boundary. Thus, once inside this region, the solution is exact for the given object geometry, that is, Equation (5.4) is satisfied on the object boundary, and there are no local extrema by the extremum property. Outside this region, we design an outer region where we make a continuous linear transition, with continuous first order derivatives, from the exact potential of the object in the uniform flow, ϕ_{ou_i}, to the sum of the potential prior to the introduction of object i and the potential of the object without the uniform flow, $\phi_{i-1}+\phi_{o_i}$ (region B). In implementation, region B is typically chosen to be of width of the length a robot can move in a couple of time steps. Outside this transition region, the potential is then $\phi_i = \phi_{i-1} + \phi_{o_i}$ (region C). By superposition, since ϕ_{i-1}, ϕ_{o_i} and ϕ_{u_i} are all solutions to Laplace's equation, any linear combination of them is also a solution. Thus, the potential is a harmonic potential in all three regions, and by the extremum property there are no local extrema other than on the boundary of the object.

The potential in region C is an approximation of the potential of a fluid. Since, by definition, a robot in region C is at a safe distance from any object, the approximated solution for the fluid flow is appropriate for a robot's path. The sectioning of the regions around an object is depicted in Figure 5-1.

If two objects are so close that their B-regions overlap, one can let the path be determined by the mean of the field from both objects. This will work since the objects are far enough away so that the robot can navigate robustly in this region, which by definition is a B-region. If two objects are so close together that their A-

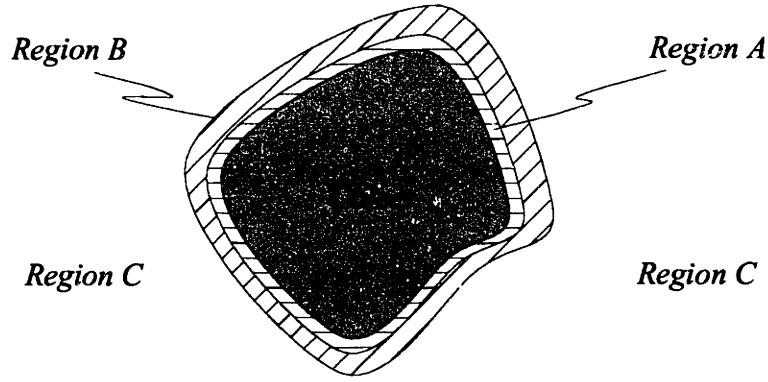

Figure 5-1: An object modeled with a harmonic potential and the different zones defined around it in order to enable path-planning among multiple objects.

regions overlap, it is not safe for a robot to operate in this area, since it is less than the distance the robot can move in one time step from the objects, thus making it liable to collide with one of the objects. In such cases, to make the planner more robust, we view all objects that are so close that their A regions overlap as a single object.

In Figure 5-2 we show several paths for a robot whose goal is just to move from the left to the right given different initial vertical position. These paths are analogous to the stream lines for an ideal fluid. The dotted lines shows the outer border of region B, while the dashed-dotted region shows the outer border of region A. Figure 5-2(a) shows the paths when the two objects are so far away from each other that they do not have overlapping B regions. (b) shows the robot paths if the B regions (but not A regions) overlap. (c) shows the paths when the objects are so close that they can be considered as a single object (that is, their A regions overlap or the objects touch).

5.2.3 Defining the external field

Traditionally, a point source has been placed at the robot's initial position while a point sink has been place at the robot's goal position in order to make the goal the global minimum and the initial position the global maximum of the operating space [57, 116, 33]. However, this method has the disadvantage that the field can become

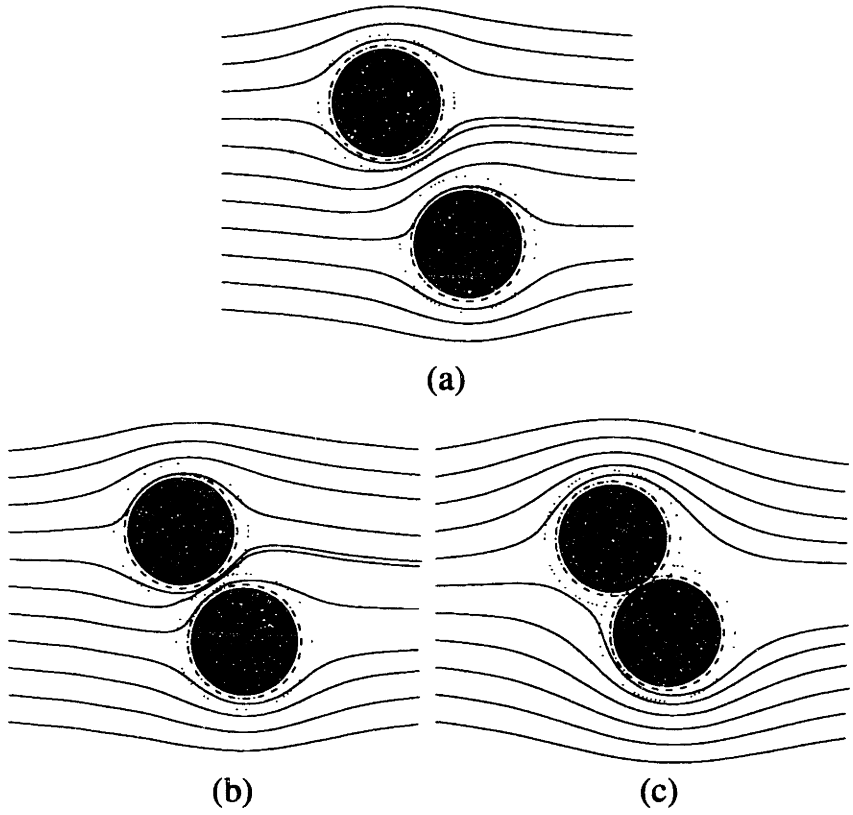

Figure 5-2: Robot paths, from left to right, past objects for different initial positions.

arbitrary small when the robot is far away from its goal and initial position, while the field tends to infinity at the goal and initial position, thus, making the method vulnerable to numerical noise. Kim and Khosla [60] used a uniform field directed from the initial point to the goal sink, avoiding a point source at the initial position. This eliminates the near zero field in between goal and robot, but not the singularity of the goal position.

We approach this problem by the use of a uniform field always directed from the robot's current position to the goal position with magnitude U. More precisely, we use the complex potential

$$w = -Uze^{-i\alpha} \ , \ \alpha = \tan^{-1}\left(\frac{y_f - y}{x_f - x}\right) \ , \tag{5.15}$$

as the underlying external potential (that is, $\phi_e = \Re(w)$). Here (x, y) is the robot's current position and (x_f, y_f) its goal position. Thus $\phi_{u_i} = \phi_e$ for all i by definition. Notice that w has no extrema. The advantage of this method is that the external field will always have magnitude U and that $\phi_{u_i} = \phi_e$ for all times. The uniform field used, is rotated with time, however, it is always directed parallel to a straight line from the robot to its goal. This type of motion is analog to the Dog curve problem [73], thus the time variation of the field and robot paths will be continuous, and allows for implementation of moving goals in a natural manner.

A disadvantage of this method is that the solution for a uniform flow from any direction around any object in the environment must be known. Thus, objects for which the panel method is used should have the panel strengths precomputed for *any* direction uniform flow.

5.3 Dynamic systems

In this section, path planning in static environments is generalized to path planning in dynamic environments. The same general methods for defining objects in space as outlined previously will be used, but with some simple changes in order to incorporate translation (section 5.3.1), expansion and contraction (section 5.3.2), and rotation of objects (section 5.3.3). Section 5.3.4 describes a method for dealing with objects that come in contact and thereby closes existing paths.

5.3.1 Translation of objects

In this section we show how to find the velocity field of an object moving with a velocity $\mathbf{V}$ in a uniform flow $\mathbf{U}$. We first define an inertial reference frame X-Y. Second, we define a coordinate system x-y which is fixed on the moving object, and moves with a velocity $\mathbf{V} = [V_X \;\; V_Y]$ with respect to the reference frame X-Y. Since the object is moving we need to ensure that the von Neumann boundary condition is

satisfied on the surface of the object in the *x-y coordinate system* in order to ensure that a path does not pass through the object. When this is achieved, we can transform the field back to the X-Y system. The uniform flow past the object, as seen in the x-y coordinate system is:

$$\mathbf{U_o} = \mathbf{U} - \mathbf{V} \,. \tag{5.16}$$

Notice that $\mathbf{U_o}$ is also a uniform field, but has different magnitude and direction than $\mathbf{U}$. Thus we can find the velocity field around this object using any of the methods outlined in section 5.2.1. In particular, assume that the velocity at time t at a point $(x(t), y(t))$ is $\boldsymbol{v} = [v_x \;\; v_y]$ in the x-y coordinate system. In the X-Y coordinate system, this point's coordinates are given by:

$$\begin{aligned} X(t) &= X(0) + x(t) + V_X t \,, \\ Y(t) &= Y(0) + y(t) + V_Y t \,, \end{aligned} \tag{5.17}$$

where $X(0), Y(0)$ is the coordinates in the X-Y coordinate system of the origin of the x-y coordinate system. The velocity $\mathbf{u}$ at this point in the X-Y system is then simply

$$\mathbf{u} = \boldsymbol{v} + \mathbf{V} \,. \tag{5.18}$$

As a simple example, we can find the velocity field when a circular cylinder with velocity $\mathbf{V}$ moves in a fluid with velocity $\mathbf{U}$. Using Equation (5.8) we find

$$v_x - iv_y = ||\mathbf{U} - \mathbf{V}|| \left(e^{-i\alpha} - \frac{r^2 e^{i\alpha}}{z - z_0} \right) \,, \tag{5.19}$$

where r is the radius of the cylinder, α is the angle that $\mathbf{U} - \mathbf{V}$ makes with respect to the X-axis, and $z_0 = X_0 + iY_0$ is the center of the cylinder. The velocity field is then simply found by combining equations (5.18) and (5.19).

Figure 5-3 shows a circular robot starting at the left whose goal is marked by

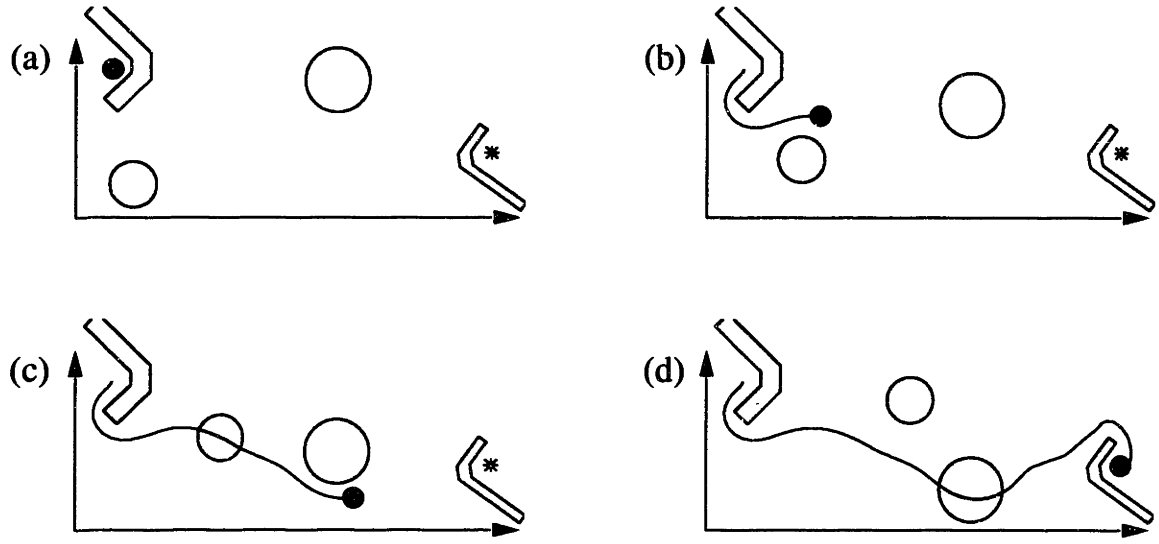

Figure 5-3: Robot avoiding moving circular cylinders and static walls.

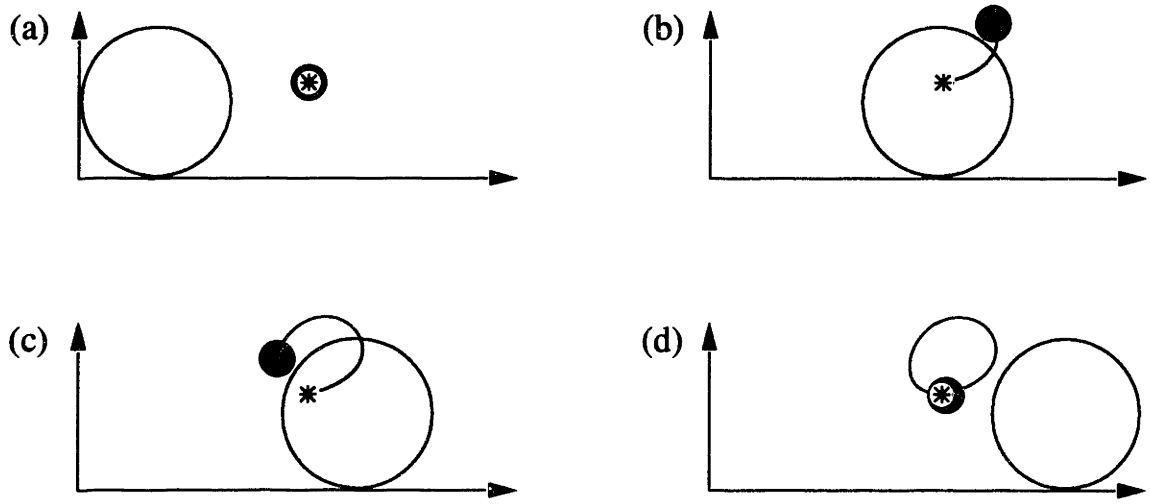

Figure 5-4: Robot avoiding circular cylinder crossing the robot's goal '*'.

a '*'. A moving circular cylinder starts at the bottom left, and moves to the left and up, and another moving cylinder starts at the upper middle and moves straight down. The two L-shaped objects are static and were defined using the panel method with 18 panels each. The line in the drawing represents the robot's path. As can be seen from these figures, the robot manages to maneuver in the dynamic environment without hitting the moving objects or the static walls, while still reaching its goal. For

this particular plot, each computation took about 0.05 seconds using non-optimized matlab code on a Sparc 5. It is expected that the algorithm can be speeded up considerably, as only about 10^2 operations are required for each of the circles, and only about 10^4 operation for each of the L-shaped objects, due to the simple analytical forms, to obtain the velocity field at the robot's current position.

Figure 5-4 shows how a robot that initially is at rest at its goal, marked by a '*' (Figure 5-4(a)), while a larger circular moving object moves from left to right. As the circular object is approaching, the robot starts to move away from its goal in order to avoid the moving object, part (b) and (c). When the object has moved away from the robot's goal, the robot can again reach its goal position, as shown in part (d).

5.3.2 Expanding objects

Objects that expand or contract are simple to model. The only requirement is that the von Neumann boundary condition is satisfied on the dynamically changing boundary. Thus, if a boundary expands/contracts with a velocity V the normal velocity of the boundary must be V. The magnitude of the normal vector is then V on the boundary and is made to drop off with the inverse of the distance to the boundary, as potentials obeying Laplace's equation drops off with the inverse of the distance. Such objects might be useful in situations where, for example, one needs to avoid liquids that are dripping on a surface, or to incorporate time-varying safety zones.

Figure 5-5 shows how the robot navigates around an expanding object using this method.

5.3.3 Rotating objects

Determining the velocity potential for a rotating object is more complicated. It can, however, be shown that the stream function on the boundary of an object rotating with an angular velocity ω around the origin, and a translational velocity

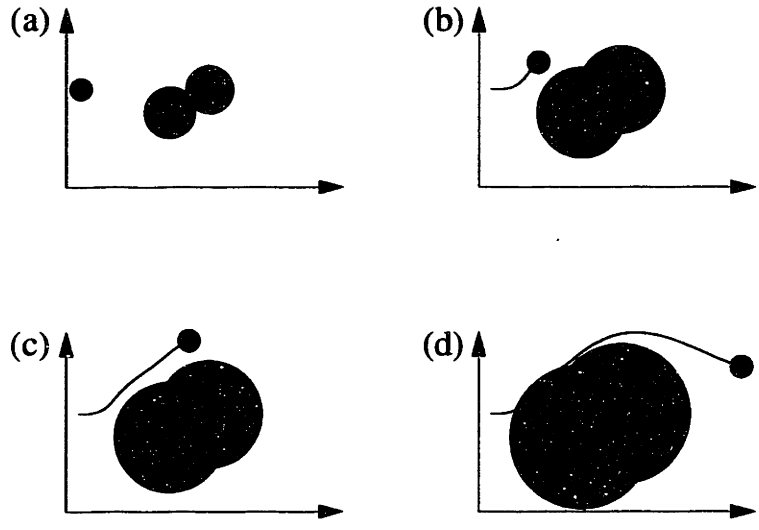

Figure 5-5: Robot moving over an expanding object.

$\mathbf{U} = [U_x \;\; iU_y] = Ue^{-i\alpha}$ in a *static* fluid, is given by (see [82] for a proof):

$$2i\psi = -Uze^{i\alpha} + U\bar{z}e^{i\alpha} + i\omega z\bar{z}\,, \tag{5.20}$$

where $\bar{z}$ is the complex conjugate of z.

Now, if we let the domain outside a contour C in the z-plane be mapped conformally on to the outside of the unit circle $|\xi| = 1$ in a complex ξ-plane, by the relation

$$z = f(\xi)\,, \tag{5.21}$$

the points at infinity in the z-plane and ξ-plane correspond. Therefore, for the liquid to be at rest at infinity, the complex potential w cannot contain positive powers of z (or ξ) when expanded in a power series in z (or ξ). Now, we define a general point on the unit circle boundary by $\sigma = e^{i\theta}$, where the stream function in Equation (5.20) must apply. Thus, by equations (5.20) and (5.21), the stream function ψ on the

boundary C becomes

$$\begin{aligned} 2i\psi &= B(\sigma) \\ &= -Uf(\sigma)e^{-i\alpha} + U\bar{f}(1/\sigma)e^{i\alpha} + \\ &\quad i\omega f(\sigma)\bar{f}(1/\sigma) \, . \end{aligned} \tag{5.22}$$

The function $B(\sigma)$ is called the *boundary function* and can be expanded in negative powers of σ, grouped in $B_1(\sigma)$, and in positive powers of σ, grouped in $B_2(\sigma)$. That is, $B(\sigma) = B_1(\sigma) + B_2(\sigma)$. It can further be shown that, if the *fluid* is moving with velocity $-\mathbf{U}$ and the *object* is rotating with angular velocity ω about the point z_0, the complex potential that satisfies the boundary condition (Equation (5.20)) is then (see [82] for proof):

$$w = B_1(\xi) + Uf(\xi)e^{-i\alpha} - i\omega\bar{z}_0 z \, . \tag{5.23}$$

To get the velocity field for the contour C rotating about the point z_0 in a fluid flowing uniformly with velocity $Ue^{-i\alpha}$ we use Equation (5.6).

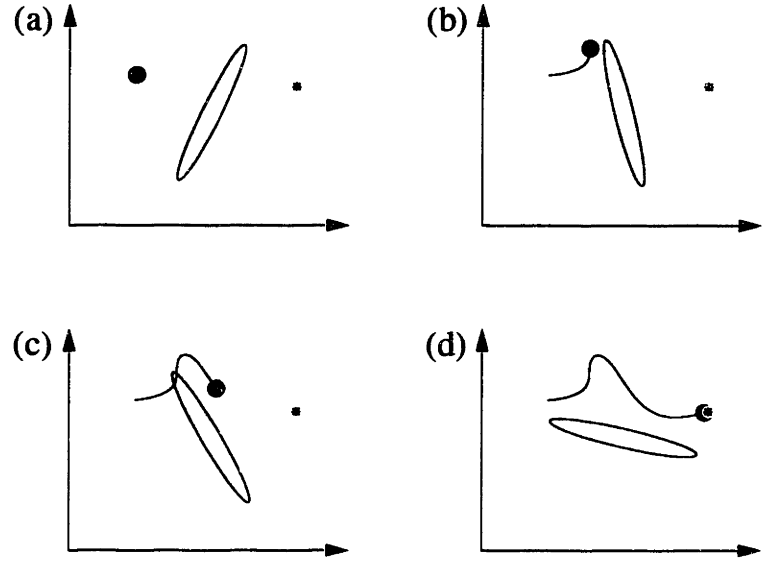

Figure 5-6: The robot navigates to its goal at '*', avoiding the rotating ellipse. The robot's goal is so high that the robot is forced to go above the rotating ellipse.

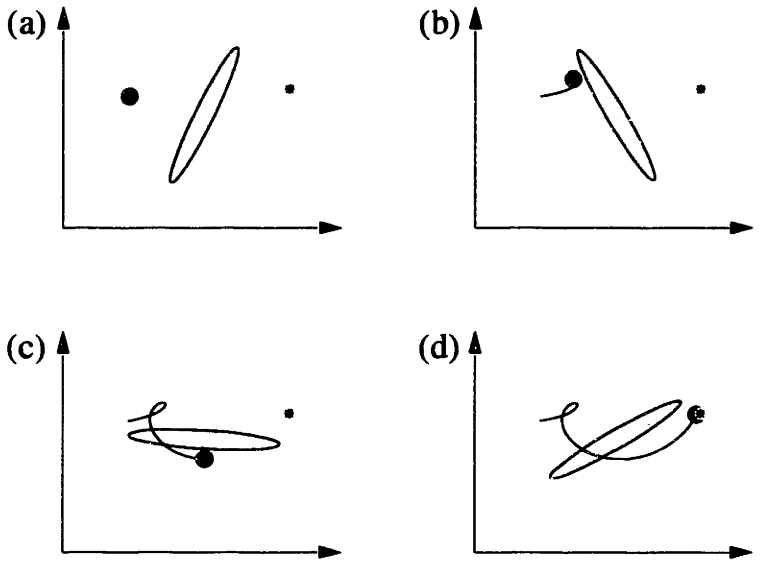

Figure 5-7: The robot navigates to its goal at '*', avoiding the rotating ellipse. The robot at first tries to go above the ellipse, but is forced to go below the ellipse.

The effect of a rotating object can be illustrated by two simple examples where we let a robot navigate to its goal in the presence of a rotating ellipse. Figures 5-6 and 5-7 illustrate how a circular robot navigates around a rotating ellipse, depending on its goal position. The initial position of the robot is to the left, and is the same for both cases. The initial position and the angular velocity of the ellipse is also the same in both figures. In Figure 5-6 the robot moves in the correct general direction initially and passes above the rotating ellipse. In Figure 5-7, however, the robot is initially on a path to pass above the ellipse (part (b)), but it is unable to, and is forced to move below the ellipse (part (c)) and to its goal in part (d).

5.3.4 Interaction among obstacles

The main problem left to solve is how to deal with situations where objects move in such a manner as to close an existing path. A solution to this problem is to add circulation around the objects. By changing the tangential velocity on the boundary, we are able to move the streamlines around the object. In particular we are able to

move the stagnation lines. As mentioned, all objects have one and only one stagnation line. The problem is, therefore, to make a smooth transition from two stagnation lines to one when two objects come in contact and cut a path. This is achieved by making the surface of the objects "rotate" (that is, add circulation) in opposite directions as the two objects approach each other. More quantitatively, one rotates the surface of the object by a speed V_s such that the tangential velocity at the point of contact vanishes. That is, if the tangential velocity at the point of contact *prior* to the rotation of the surface is V_t, the condition of vanishing tangential velocity is achieved by letting $V_s = -V_t$. Thus, when two objects, A and B, come in contact one lets $V_{s_A} = -V_{t_A}$ and $V_{s_B} = -V_{t_B}$. This, in conjunction with the method of defining multiple objects, as defined in section 5.2.2, will conserve the validity of Equation (5.4) on the objects boundaries. When two objects are separated by a distance δ we want a smaller circulation on the objects to enable a point robot to move between the two objects. Thus, the speed of the object's surface is given by:

$$V_s = -(1 - \frac{\delta}{\Delta})V_t \,, \tag{5.24}$$

where Δ is a free parameter that determines how far away two objects can be before rotations of the surfaces are introduced. In general the relative distance between every object in the environment is needed, which is computationally expensive. However, by introducing another region D outside region B, (see Figure 5-1) to each object, the computational cost becomes small. If a robot is outside all D regions or inside the D region of just one object, it is at a safe distance from any interacting objects, thus no rotation of object surfaces is needed. If, however, the robot is inside region D of *two* objects, Equation (5.24) must be employed. Here, Δ should be chosen large enough so that no discontinuities arises. If the robot is in the D region of more than two objects, where Equation (5.24) might lead to a contradiction, Equation (5.24) should be applied to the two objects closest to the robot.

An example of using this method is given in Figure 5-8, where a point robot starts

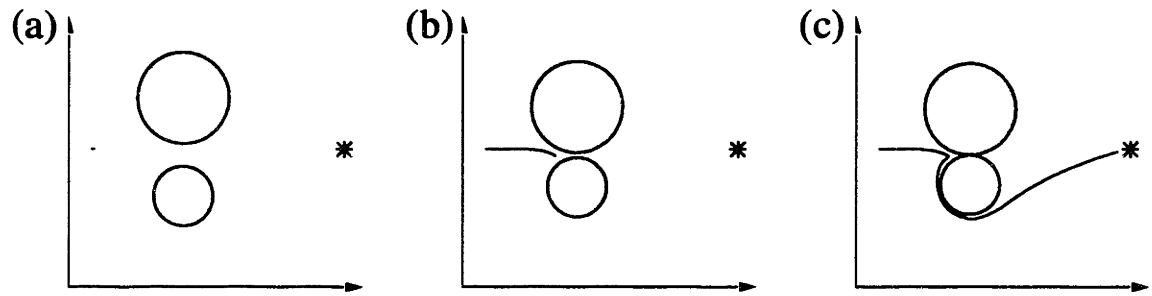

Figure 5-8: A point robot moves from left to its goal at '*'.

at the left, and the goal position is at the right marked by a '*'. The two circles move against each other. Initially the robot believes it can take the shortest path between the two objects (Figure 5-8 (a)–(b)). However, as the robot moves closer to the two objects and the two objects are approaching each other, the circulation around each object is increased and the stagnation lines of both are moved to the center line between the two objects. At the instant when the two objects touch, the two stagnation lines merge into one. At (b) the robot feels this effect and realizes it can not pass between the two objects, and takes the shorter path around the smaller circle to its goal in part (c).

5.4 Concluding remarks

In this chapter we described a new method based on the use of harmonic functions for real-time path planning in dynamic environments. This is possible through the use of *analytical* solutions to Laplace's equation in dynamic environments, making real-time path planning possible, and through the use of special modifications that allows for several objects to be modeled analytically in a dynamic environment.

In static cases, a harmonic potential will have no local minima, a path will not intersect with any object and a convergence to the goal position is guaranteed if the goal position is reachable. The same features of harmonic potentials carry over to the dynamic case at every instant. That is, if there exists a path at the current time

leading to the goal that does not intersect with any obstacle, the robot will move along this path. In dynamic environments it is not possible to guarantee convergence to the goal position or collision avoidance for all dynamic environments over some time t. This path planner only considers the possible paths at each instant, thus making it possible to trap it by alternately opening and closing existing paths, making it confused as to which path to take. Further, if the goal is within a semi-closed region, such as inside a U-shape, the goal cannot be reached. This is a property of an ideal fluid. Since fluid does not flow through the U-shape, the flow will go around it, thus separating the flows inside and outside the U.

Several extensions to this work are possible, and experimental evaluation should also be performed. The utilization of harmonic potentials for control of robots with multiple degrees of freedom should be developed. Further, taking the curvature of the streamlines into account, and moving away from streamlines with high curvature is an attractive approach. This can be seen as locally applying elastic bands on the path, similarly to [94]. Extensions to n-dimensions should also be studied.

5.5 Summary

This chapter has presented a potential field based approach to path planning in *dynamic* environments based on fluid analogies. Harmonic potentials have the properties that they have no local minima and that a path will be guaranteed to converge to the goal position in the static case. The same property of harmonic potentials hold in dynamic environments, however obstacle avoidance and convergence to the goal can not be guaranteed for all possible situations. For instance, the robot could be trapped in a closed room.

The technique developed here presents an attractive solution the problem faced by a robot performing path planning in a dynamic environment.

Chapter 6

Conclusions and future research

This chapter gives a summary of the thesis, reviews the contributions and presents suggestions for future research.

6.1 Thesis summary

This thesis has investigated new methods for mobile robot navigation. The navigation problem is for a mobile robot to determine the structure of an *a priori* unknown environment and use this knowledge to localize globally. This problem is that of concurrent mapping and localization (CML), which *simultaneously* poses the questions "what is my map?" and "where am I?" to the robot. These questions are particularly crucial for autonomous underwater vehicles (AUVs), as *a priori* maps or external localization systems are rarely available to aid in the navigation process.

Through the extension of state-of-the-art techniques of feature-based CML, the theory and method of augmented stochastic mapping was developed in Chapter 2, and its long-term performance was assessed in simulations. The ability of ASM to perform CML was shown in simulations, experiments and post-processing of oceanic data.

A solution to the long standing issue of computational complexity of feature-

based CML was developed in Chapter 3, and verified by simulations and experiments. The method reduces the $\mathcal{O}(n^3)$ computational complexity of conventional (optimal) methods to one that is constant in computational requirements, while achieving a CML accuracy close to the optimal solution.

Increasing the performance of feature-based CML through adaptive strategies was investigated in Chapter 4, and resulted in the development of adaptive augmented stochastic mapping. In this technique, the vehicle performs CML *and* decides where to move and where to direct its attention in order to increase the robustness, speed and energy efficiency of CML. The validity of the approach was shown through theory, simulations and experiments.

The techniques of Chapter 2, 3 and Chapter 4 provides solutions to the questions "what is my map?" and "where am I?". The question of "where am I going?", which is part of the broader problem of autonomous mobile robot operations, was implicitly addressed in Chapter 4. In addition to these three questions, there is one important remaining question to be addressed for the mobile robot — the question of "how do I get there?". This is the topic of Chapter 5, where path-planning in dynamic environments was addressed through theoretical developments and simulations.

The truly autonomous operation of mobile robots is crucial for the utility that society can achieve from robotic technologies and new technology in general. As we see it, the current bottleneck in achieving this goal is the limit of navigational techniques. We believe that the contributions of this thesis have advanced this technology and brought us closer to that goal.

Having given a brief summary of the thesis, we will now turn to the contributions.

6.2 Contributions

The main contributions of this thesis are

- An analysis of the long-term performance of a stochastic mapping based algorithm for performing concurrent mapping and localization.
- A solution to the problem of map scaling encountered by stochastic mapping based on the use of a decoupled map representation, consisting of multiple globally-referenced submaps, and the use of consistent methods for transitioning vehicle position information between submaps.
- A method for performing adaptive concurrent mapping and localization based on maximizing the Fisher information gained from the next action of the robot.
- A technique for performing path planning in dynamic environments based on fluid analogies enabling real-time computation of collision free paths to the goal from local information.

The following sections gives a summary of each and its importance in advancing mobile robot navigation.

6.2.1 Long-term performance analysis of CML

Before the presentation of the work in Chapter 2, two important questions had not been addressed in the existing literature, namely

1. What is the long-term behavior of feature-based concurrent mapping and localization?
2. Is stochastic mapping a viable technique for AUV navigation?

Stochastic mapping does not present the means of performing track initiation, data association and track deletion, which are all paramount for the successful operation

of CML in experiments and in the field. Through the development of the delayed nearest neighbor track initiator, a gated nearest neighbor data association filter and a logic based track deleter, these issues were addressed and integrated with stochastic mapping, culminating in the technique of augmented stochastic mapping (ASM). Tank experiments and post-processing of oceanic data verified the validity of ASM in providing a CML algorithm powerful enough to conduct CML in experiments and in the field.

Non-linear estimation algorithms might well provide seemingly consistent, bounded error in the short run. Thus, previous published results in the literature did not provide a convincing result of the long-term behavior of CML. We presented mission runs that are much longer in duration than any previously published. The ASM algorithm showed consistent, bounded estimates, thereby establishing its performance for long term missions.

6.2.2 Map scaling

One of the major obstacles to CML is the rapidly increasing computational complexity required to map an increasing number of features. Current techniques can only track about one hundred features in real time, using any commercial personal computer. Chapter 3 provides a solution to this long-standing issue in mobile robot navigation by the introduction of decoupled stochastic mapping (DSM). Consistent, globally bounded errors are achieved by DSM for any number of features in constant computation time and memory requirements. DSM was shown to provide faster than real-time CML in environments that were orders of magnitude larger than any previously reported in the literature.

The key ideas behind DSM are the use of a decoupled map representation, consisting of multiple globally-referenced submaps, and to employ consistent methods for transitioning vehicle position information between submaps.

Two DSM methods were introduced: multi-pass DSM and single-pass DSM.

Multi-pass DSM enables the robust and consistent reduction of global errors between submaps to achieve a performance bound that is comparable to the result obtained using full covariance stochastic mapping. The method of single-pass DSM is suitable for missions in which only a single pass through the survey area is anticipated. Single-pass DSM results in slow, linearly growing errors with distance from the starting point. For a single pass through the survey region, single-pass DSM achieves smaller error bounds than multi-pass DSM. However, continued surveying of the region does not reduce the global error of the submaps as in multi-pass DSM.

6.2.3 Adaptive concurrent mapping and localization

A information based framework for performing adaptive concurrent mapping and localization was developed, resulting in the technique of adaptive augmented stochastic mapping (adaptive ASM). The increased performance, speed and energy efficiency of adaptive ASM was confirmed by simulations and experiments. Further, an interesting exploratory behavior of the robot emerged during adaptive ASM.

6.2.4 Path planning in dynamic environments

Dynamic environments pose a particularly difficult problem for the autonomous mobile robot. Conventional approaches may cause the robot to get stuck, infinitely computing new collision free paths from the current position to the goal as the environment changes continuously. We introduced a novel harmonic potential based path planner for operating among multiple objects in dynamic environments.

6.3 Future research

During this investigation of mobile robot navigation, we have gained significant insight into the challenges as well as the opportunities available through these new techniques. Below we provide suggestions for future research that we believe are of

great importance for mobile robot navigation and indicate, where appropriate, possible solutions.

6.3.1 Data association

The problem of associating measurements with features in cluttered environments remains a challenging and important problem for any estimation problem that faces data association ambiguities. The hybrid estimation approach to CML of Integrated mapping and navigation (IMAN) [105], provides one venue for addressing this problem. In IMAN, multiple hypotheses are generated for each measurement, and a stochastic mapping algorithm is employed for each consistent set of hypotheses causing an enormous computational burden. However, there might well be simpler and more robust techniques available. On interesting technique arises from closer examination of Equation (3.14). This quantity is the uncertainty of the map (all the features) if a perfect vehicle estimate was provided. Thus, data association can be achieved with the *removal* of the uncertainty of the vehicle that existed in the *initiation* of the feature estimate. The uncertainty accumulated during the initiation phase can be eliminated by re-passing the returns used for the initiation process through the stochastic mapping algorithm. This approach, we believe, provides a theoretically simpler, and computationally less complex approach to data association than IMAN. One of the central causes of the extreme complexity of IMAN was the complexity arising from vehicle uncertainty. Thus, the method outline here might provide a more viable approach.

6.3.2 Feature extraction

Any feature-based approach to navigation is crucially dependent on being able to identify features in the environment. This remains an extremely important area of research. Adaptive strategies for improving feature identification can be advantageous. For instance, if trying to identify whether an object is a wall or a corner,

moving parallel to the whole rather than perpendicular to the wall would make the identification easier. The development of improved sonar technologies and the study of bats and dolphins may greatly increase the ability to extract environmental features from the seabed.

6.3.3 Cooperative mapping and navigation

Cooperative mapping has the potential to greatly increase and simplify error identification, error recovery, and state estimation and opens up a whole new and exciting field of AUV research. Using a team of AUVs to to conduct surveys of an area can provide more robustness than using a single vehicle as the success of the mission is not critically dependent on a single vehicle.

The decoupled stochastic mapping method developed in Chapter 3, provides a natural way of achieving this. Each time a vehicle is about to create a new submap, it queries all the other vehicles in the area to determine if any of them have previously mapped the area. If none have, it creates a new submap, otherwise it retrieves the appropriate submap for another vehicle and performs a cross-map vehicle update into this map if the two vehicles have not exchanged submaps earlier, otherwise it performs a cross-map vehicle relocation into this map. This is possible as DSM provides *globally* referenced maps, thus enabling submap usage between multiple vehicles. This is *not* possible through other techniques presented in the literature.

In addition to traditional means of detecting errors, cooperative mapping and navigation provides another potent modality. Vehicles can periodically communicate their position and error bounds to each other, then estimate their distances with respect to each other through the time of flight of the communication signals between them. If an inconsistency arises, one or more vehicles has an erroneous estimate. The vehicle(s) with an erroneous estimate can be identified through consistency checks.

A vehicle that knows it has a faulty estimate can relocate itself globally using the other vehicles and their error bounds to triangulate its own position and error

bound. If available, a submap for the area that the faulted vehicle is operating in can be provided by one of the other vehicles, thereby providing the faulted vehicle with a consistent map, while only locally losing information by the removal of the old submap.

State estimation can be improved by letting each vehicle communicate their current estimate and covariance to all other vehicles. Once this information is obtained, each vehicle can estimate its relative distance to the other vehicles and combine this estimate with the estimate provided by the other vehicles, thereby improving the estimate of its own position. DSM provides a framework of reasoning about how to achieve this, as care must be taken not to reuse information.

6.3.4 Field experiments

The end goal of mobile robot navigation research, and AUV navigation research in particular, is to achieve accurate navigation in the real world with actual AUVs. Thus experimental verifications and testing using AUVs are of great importance. In addition to dealing with all the technical issues involved in operating an AUV, extension of the models presented here to three dimensions, improved feature extraction and data association might be necessary.

With the introduction of DSM, autonomous large scale indoor navigation by a mobile robot, such as the Nomad mobile robot used in Chapter 4, can be achieved with the techniques presented here and the use of standard techniques for identifying indoor features, such as walls, corners and edges [69].

6.4 Summary

This chapter has provided a brief summary of the thesis and restated its main contributions of providing: (1) a long-term performance analysis of the stochastic mapping approach to concurrent mapping and localization, (2) a solution to the map scaling

problem, (3) a method for performing adaptive concurrent mapping and localization, and (4) a technique for performing path planning in dynamic environments. Issues for future research have been suggested.

Through the contributions of this thesis, the mobile robot, and autonomous underwater vehicles in particular, are not left at the mercy of the navigational techniques of the 18th century — integration of velocity and heading measurements. Although there are still important issues to be resolved, the navigational techniques presented here provide a basis for performing accurate, consistent and safe navigation for the mobile robot operating in large-scale, *a priori* unknown environments.

Bibliography

[1] W. Au. *The Sonar of Dolphins.* New York: Springer-Verlag, 1993.

[2] R. K. Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):996–1005, August 1988.

[3] Y. Bar-Shalom, editor. *Multitarget Multisensor Tracking: Advanced Applications, Volume II.* Boston: Artech House, 1992.

[4] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association.* Academic Press, 1988.

[5] Y. Bar-Shalom and X. R. Li. *Estimation and Tracking: Principles, Techniques, and Software.* Artech House, 1993.

[6] B. Barshan and R. Kuc. ROBAT: A sonar-based mobile robot for bat-like prey capture. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 274–279, Nice,France, May 1992.

[7] J. G. Bellingham, editor. *Scientific and Environmental Data Collection with Autonomous Underwater Vehicles*, Cambridge, MA, March 1992. MIT Sea Grant Marine Industry Collegium.

[8] J. G. Bellingham, C. Chryssostomidis, M. Deffenbaugh, J. J. Leonard, and H. Schmidt. Arctic under-ice survey operations. In *Proc. Int. Symp. on Unmanned Untethered Submersible Technology*, pages 50–59, 1993.

[9] J. G. Bellingham and T. R. Consi. Robots underwater — ongoing research at MIT. *Sea Technology*, May 1990.

[10] J. G. Bellingham, T. R. Consi, C. Goudey, and J. Bales. Monitoring of Boston Harbor and Massachusetts Bay with autonomous vehicles. Technical report, Sea Grant College Program, Massachusetts Institute of Technology, August 1991.

[11] J. G. Bellingham, T. R. Consi, C. Goudey, J. W. Bales, and C. Chryssostomidis. Performance characteristics of the Odyssey AUV. In *Proc. Int. Symp. on Unmanned Untethered Submersible Technology*, pages 37–49, 1993.

[12] J. G. Bellingham, C. A. Goudey, T. R. Consi, J. W. Bales, D. K. Atwood, J. J. Leonard, and C. Chryssostomidis. A second generation survey AUV. In *IEEE Conference on Autonomous Underwater Vehicles*, Cambridge, MA, 1994.

[13] J. G. Bellingham, J. J. Leonard, J. Vaganay, C. Goudey, D. Atwood, T. Consi, J. Bales, H. Schmidt, and C. Chryssostomidis. AUV operations in the arctic. In *Proceedings of the Sea Ice Mechanics and Arctic Modeling Workshop*, Anchorage, Alaska, USA, April 1995.

[14] J. G. Bellingham and J. S. Willcox. Optimizing AUV oceanographic surveys. In *IEEE Conference on Autonomous Underwater Vehicles*, Monterey, CA, 1996.

[15] O. Bergem. A multibeam sonar based positioning system for an AUV. In *Eighth International Symposium on Unmanned Untethered Submersible Technology (AUSI)*, September 1993.

[16] J. O. Berger. *Statistical Decision Theory and Bayesian Analysis.* Springer-Verlag, second edition, 1985.

[17] S. Betgé-Brezetz, P. Hébert, R. Chatila, and M. Devy. Uncertain map making in natural environments. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1048–1053, April 1996.

[18] A. Blake and A. Yuille, editors. *Active Vision.* MIT Press, 1992.

[19] O. Bozma and R. Kuc. Characterizing pulses reflected from rough surfaces using ultrasound. *J. Acoustical Society of America*, 89(6):2519–2531, June 1991.

[20] O. Bozma and R. Kuc. Single sensor sonar map building based on physical principles of reflection. In *Proc. IEEE Int. Workshop on Intelligent Robots and Systems*, 1991.

[21] R. A. Brooks. The whole iguana. In *Robotics Science.* MIT Press, 1989.

[22] D. K. Brown. Damn the mines! *Proceedings of the U. S. Naval Institute*, 118(3):45–50, March 1992.

[23] W. Burgard, A.B. Cremers, D. Fox, D. Haehnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. Experiences with an interactive museum tour-guide robot. *Artificial Intelligence*, 1999.

[24] R. N. Carpenter. Concurrent mapping and localization using forward look sonar. In *IEEE AUV*, Cambridge, MA, August 1998.

[25] R. N. Carpenter, J. J. Leonard, and H. J. S. Feder. Concurrent mapping and localization using forward look sonar. Technical Report Marine Robotics Laboratory Technical Memorandum 99-2, Massachusetts Institute of Technology, 1999.

[26] J. A. Castellanos, J. D. Tards, and G. Schmidt. Building a global map of the environment of a mobile robot: The importance of correlations. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1053–1059, 1997.

[27] Naval Underwater Warfare Center. NUWC DIVNPT UUV homepage, March 1999.

[28] H. Chang. A new technique to handle local minimum for imperfect potential field based motion planning. *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 108–112, April 1996.

[29] K. S. Chong and L. Kleeman. Sonar based map building in large indoor environments. Technical Report MECSE-1997-1, Department of Electrical and Computer Systems Engineering, Monash University, 1997.

[30] K. S. Chong and L. Kleeman. Sonar feature map building for a mobile robot. In *Proc. IEEE Int. Conf. Robotics and Automation*, 1997.

[31] C-Y. Chow. *An Introduction to Computational Fluid Mechanics.* John Wiley & Sons, 1979.

[32] C. I. Connolly. Harmonic functions and collision probabilities. *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 3015–3019, May 1994.

[33] C. I. Connolly, J. B. Burns, and R. Weiss. Path planning using laplace's equation. *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 2101–2106, May 1990.

[34] I. J. Cox and J. J. Leonard. Modeling a dynamic environment using a Bayesian multiple hypothesis approach. *Artificial Intelligence*, 66(2):311–344, April 1994.

[35] I. J. Cox and G. T. Wilfong. *Autonomous Robot Vehicles.* Springer-Verlag, 1990.

[36] M. W. M. G. Dissanayake, P. Newman, H. F. Durrant-Whyte, S. Clark, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. In *Sixth International Symposium on Experimental Robotics*, March 1999.

[37] A. Elfes. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, RA-3(3):249–265, June 1987.

[38] H. J. S. Feder and J. J. Leonard. Decoupled stochastic mapping part II: performance analysis, 1999. Submitted for consideration for publication to the IEEE Transactions on Robotics and Automation.

[39] H. J. S. Feder, J. J. Leonard, and R. N. Carpenter. Concurrent mapping and localization using forward look sonar. Technical Report Marine Robotics Laboratory Technical Memorandum 99-3, Massachusetts Institute of Technology, 1999.

[40] H. J. S. Feder, J. J. Leonard, and C. M. Smith. Adaptive concurrent mapping and localization using sonar. In *Proc. IEEE Int. Workshop on Intelligent Robots and Systems*, Victoria, B.C., Canada, October 1998.

[41] H. J. S. Feder, J. J. Leonard, and C. M. Smith. Adaptive mapping and navigation using sonar. Technical Report 98-3, Marine Robotics Laboratory, Massachusetts Institute of Technology, 1998. Available from `http://oe.mit.edu/~jleonard/pubs/tech_report983.pdf`.

[42] H. J. S. Feder, J. J. Leonard, and C. M. Smith. Adaptive sensing for terrain aided navigation. In *IEEE Oceans*, Nice, France, September 1998.

[43] H. J. S. Feder, J. J. Leonard, and C. M. Smith. Adaptive mobile robot navigation and mapping. *Int. J. Robotics Research*, July 1999. To Appear.

[44] H. J. S. Feder and J.-J. E. Slotine. Real-time path planning using harmonic potentials in dynamic environments. In *Proc. IEEE Int. Conf. Robotics and Automation*, April 1997.

[45] H. J. S. Feder, C. M. Smith, and J. J. Leonard. Incorporating environmental measurements in navigation. In *IEEE AUV*, Cambridge, MA, August 1998.

[46] A. C. Gelb. *Applied Optimal Estimation.* The MIT Press, 1973.

[47] E. Geyer, P. Creamer, J. D'Appolito, and R. Gains. Characteristics and capabilities of navigation systems for unmanned untethered submersibles. In *Proc. Int. Symp. on Unmanned Untethered Submersible Technology*, pages 320–347, 1987.

[48] M. O. González. *Complex Analysis, Selected Topics.* Marcel Dekker, Inc., 1991.

[49] J. Guldner and V. I. Utkin. Sliding mode control for an obstacle avoidance strategy based on an harmonic potential field. *Proc. of the 32d Conf. on Decision and Control*, pages 424–429, December 1993.

[50] G. Hager. *Task-directed Sensor Fusion and Planning: A Computational Approach.* Boston: Kluwer Academic Publishers, 1990.

[51] P. Hébert, S. Betgé-Brezetz, and R. Chatila. Decoupling odometry and exteroceptive perception in building a global world map of a mobile robot: The use of local maps. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 757–764, April 1996.

[52] D. B. Heckman and R. C. Abbott. An acoustic navigation technique. In *IEEE Oceans '73*, pages 591–595, 1973.

[53] J. L. Hess and A. M. O. Smith. Calculation of potential flow about arbitrary bodies. *Progress in Aeronatical Science*, 8:1–138, 1966.

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

[55] D. Johnson and D. Dudgeon. *Array Signal Processing.* Prentice Hall, 1993.

[56] L. P. Kaelbling, A. R. Cassandra, and J. A. Kurien. Acting under uncertainty: Discrete Bayesian models for mobile-robot navigation. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1996.

[57] D. Keymeulen and J. Decuyper. The fluid dynamics applied to mobile robot motion, the stream field method. *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 378–385, May 1994.

[58] D. Keymeulen and J. Decuyper. Self-organized system for the motion planning of mobile robots. *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 3369–3374, April 1996.

[59] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. of Robot. Res.*, 5:90–98, Spring 1986.

[60] J-O. Kim and P. Khosla. Real-time obstacle avoidance using harmonic potential functions. *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 790–796, April 1991.

[61] R. Kuc. Fusing binaural sonar information for object recognition. In *IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 727–735, 1996.

[62] B. J. Kuipers and Y. Byun. A robot exploration and mapping strategy based on a semantic heirarchy of spatial representations. *Robotics and Autonomous Systems*, pages 47–63, 1991.

[63] T. Kurien. Issues in the design of practical multitarget tracking algorithms. In Y. Bar-Shalom, editor, *Multitarget-Multisensor Tracking: Advanced Applications*, pages 43–83. Boston: Artech House, 1990.

[64] M. Kuristsky and M. Goldstein. Inertial navigation. In I. Cox and G. Wilfong, editors, *Autonomous Robot Vehicles*. Springer-Verlag, 1990.

[65] J. J. Leonard, A. A. Bennett, C. M. Smith, and H. J. S. Feder. Autonomous underwater vehicle navigation. In *IEEE ICRA Workshop on Navigation of Outdoor Autonomous Vehicles*, Leuven, Belguim, May 1998.

[66] J. J. Leonard, A. A. Bennett, C. M. Smith, and H. J. S. Feder. Autonomous underwater vehicle navigation. Technical Report Marine Robotics Laboratory Technical Memorandum 98-1, MIT, 1998.

[67] J. J. Leonard, I. J. Cox, and H. F. Durrant-Whyte. Dynamic map building for an autonomous mobile robot. *Int. J. Robotics Research*, 11(4):286-298, August 1992.

[68] J. J. Leonard and H. F. Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *Proc. IEEE Int. Workshop on Intelligent Robots and Systems*, pages 1442–1447, Osaka, Japan, 1991.

[69] J. J. Leonard and H. F. Durrant-Whyte. *Directed Sonar Sensing for Mobile Robot Navigation.* Boston: Kluwer Academic Publishers, 1992.

[70] J. J. Leonard and H. J. S. Feder. Decoupled stochastic mapping. Technical Report Marine Robotics Laboratory 99-1, Massachusetts Institute of Technology, 1999.

[71] J. J. Leonard and H. J. S. Feder. Decoupled stochastic mapping part I: theory, 1999. Submitted for consideration for publication to the IEEE Transactions on Robotics and Automation.

[72] T. Lozano-Pérez. Foreword. In I. J. Cox and G. T. Wilfong, editors, *Autonomous Robot Vehicles.* Springer-Verlag, 1989.

[73] D. G. Luenberger. *Introduction to Dynamic Systems, Theory, Models, & Applications.* John Wiley & Sons, 1979.

[74] E. S. Maloney, editor. *Dutton's Navigation and Piloting.* Annapolis, MD: Naval Institute Press, 1985.

[75] J. S. Manyika and H. F. Durrant-Whyte. *Data Fusion and Sensor Management: A decentralized information-theoretic approach.* New York: Ellis Horwoord, 1994.

[76] M. J. Mataric. Environment learning using a distributed representation. In *Proc. IEEE Int. Conf. Robotics and Automation*, page 402, 1990.

[77] M. J. Mataric. Integration of representation into goal-driven behavior-based robots. *IEEE Trans. Robotics and Automation*, 8(3):304–312, 1992.

[78] M. J. Mataric. Behavior-based control: Examples from navigation, learning, and group behavior. *Journal of Experimental and Theoretical Artificial Intelligence*, 9(2-3), 1997. special issue on Software Architectures for Physical Agents.

[79] M. B. May. Gravity navigation. In *Record of the 1978 Position Location and Navigation Symposium*, pages 212–218, San Diego, CA, USA, November 1978.

[80] M. Medeiros and R. Carpenter. High resolution array signal processing for AUVs. In *AUV 96*, pages 10–15, 1996.

[81] P. H. Milne. *Underwater Acoustic Positioning Systems.* London: E. F. N. Spon, 1983.

[82] L. M. Milne-Thomson. *Theoretical Hydrodynanamics.* The Macmillan Company, 4 edition, 1960.

[83] B. A. Moran, J. J. Leonard, and C. Chryssostomidis. Curved shape reconstruction using multiple hypothesis tracking. *IEEE J. Ocean Engineering*, 22(4):625–638, 1997.

[84] J. Moran. *An Introduction to Theoretical and Computational Aerodynamics.* John Wiley & Sons, 1984.

[85] H. Moravec. The stanford cart and the cmu rover. *IEEE Transactions on Industrial Electronics*, 1983.

[86] H. P. Moravec and A. Elfes. High resolution maps from wide angle sonar. In *Proc. IEEE Int. Conf. Robotics and Automation*, 1985.

[87] Morse and Feshbach. *Numerical Methods in Theoretical Physics II.* McGraw-Hill Book Company, Inc., 1953.

[88] P. Moutarlier and R. Chatila. An experimental system for incremental environment modeling by an autonomous mobile robot. In *1st International Symposium on Experimental Robotics*, Montreal, June 1989.

[89] P. Moutarlier and R. Chatila. Stochastic multi-sensory data fusion for mobile robot location and environment modeling. In *5th Int. Symposium on Robotics Research*, pages 207–216, 1989.

[90] P. Moutarlier and R. Chatila. Stochastic multisensory data fusion for mobile robot location and environment modeling. In *5th Int. Symposium on Robotics Research*, Tokyo, 1989.

[91] W. S. Newman and N. Hogan. High speed robot control and obstacle avoidance using dynamic potential functions. *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 14–24, November 1987.

[92] Office of Naval Research. Very shallow water and surf zone MCM. http://www.onr.navy.mil/sci_tech/ocean/info/VSWSZMCM/, March 1998.

[93] A. Papoulis. *Probability, Random Variables and Stochastic Processes.* New York: McGraw-Hill, third edition edition, 1991.

[94] S. Quinlan and O. Khatib. Elastic bands: Connecting path planning and control. *Proc. IEEE Int. Conf. on Robotics and Automation*, 2:802–807, 1993.

[95] D. B. Reid. An algorithm for tracking multiple targets. *IEEE Trans. on Automatic Control*, AC-24(6), Dec. 1979.

[96] W. D. Rencken. Concurrent localisation and map building for mobile robots using ultrasonic sensors. In *Proc. IEEE Int. Workshop on Intelligent Robots and Systems*, pages 2192–2197, Yokohama, Japan, 1993.

[97] Reson, Inc. Commercial offer: SEABAT 8001 forward-look sonar system, 1998. `http://www.reson.com`.

[98] H. Roitblat, W. Au, P. Nachtigall, R. Shizumura, and F. Moons. Sonar recognition of targets embedded in sediment. In *Proceedings of the Autonomous Vehicles in Mine Countermeasures Symposium*, pages 6–188 - 6–200. Naval Postgraduate School, 1995.

[99] S. Russell and P. Norvig. *Artificial Intelligence A Modern Approach.* Prentice Hall, 1995.

[100] S. Russell and E. Wefald. *Do the right thing : studies in limited rationality.* MIT Press, 1995.

[101] H. Shatkay and L. P. Kaelbling. Heading in the right direction. In *Proceedings of the Fifteenth International Conference on Machine Learning*, 1998.

[102] J. A. Simmons, P. A. Saillant, and S. P. Dear. Through a bat's ear. *IEEE Spectrum*, pages 46–48, March 1992.

[103] H. Singh. *An entropic framework for AUV sensor modelling.* PhD thesis, Massachusetts Institute of Technology, 1995.

[104] H. Singh, D. Yoerger, R. Bachmayer, A. Bradley, and W. Stewart. Sonar mapping with the autonomous benthic explorer (ABE). In *Proc. Int. Symp. on Unmanned Untethered Submersible Technology*, pages 367–375, September 1995.

[105] C. M. Smith. *Integrating Mapping and Navigation.* PhD thesis, Massachusetts Institute of Technology, 1998.

[106] C. M. Smith, H. J. S. Feder, and J. J. Leonard. Making difficult decisions autonomously: The impact of integrated mapping and navigation. In *IEEE AUV*, Cambridge, MA, August 1998.

[107] C. M. Smith, H. J. S. Feder, and J. J. Leonard. Multiple target tracking with navigation uncertainty. In *IEEE Int. Conference on Decision and Control (CDC)*, Tampa, FL, December 1998.

[108] C. M. Smith and J. J. Leonard. A multiple hypothesis approach to concurrent mapping and localization for autonomous underwater vehicles. In *International Conference on Field and Service Robotics*, Sydney, Australia, 1997.

[109] C. M. Smith, J. J. Leonard, A. A. Bennett, and C. Shaw. Feature-based concurrent mapping and localization for autonomous underwater vehicles. In *IEEE Oceans*, 1997.

[110] R. Smith, M. Self, and P. Cheeseman. A stochastic map for uncertain spatial relationships. In *4th International Symposium on Robotics Research.* MIT Press, 1987.

[111] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In I. Cox and G. Wilfong, editors, *Autonomous Robot Vehicles.* Springer-Verlag, 1990.

[112] D. Sobel. *Longitude: The True Story of a Lone Genius Who Solved the Greatest Scientific Problem of His Time.* Penguin Books, 1995.

[113] W. K. Stewart. *Multisensor Modeling Underwater with Uncertain Information.* PhD thesis, Massachusetts Institute of Technology, 1988.

[114] W. Kenneth Stewart. Three-dimensional stochastic modeling using sonar sensing for undersea robotics. *Autonomous Robots*, 3:121–143, 1996.

[115] N. Størkersen, J. Kristensen, A. Indreeide, J. Seim, and T. Glancy. Hugin — UUV for seabed surveying. *Sea Technology*, February 1998.

[116] L. Tarassenko and A. Blake. Analogue computation of collison-free paths. *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 540–545, April 1991.

[117] Nomadic Technologies. `http://www.robots.com`, 1998.

[118] R. S. Thau. *Reliably mapping a robot's environment using fast vision and local, but not global, metric data.* PhD thesis, Massachusetts Institute of Technology, 1997.

[119] S. Thrun, D. Fox, and W. Burgard. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31:29–53, 1998.

[120] S. Thrun, J.-S. Gutmann, D. Fox, W. Bugard, and B. J. Kuipers. Integrating topological and metric maps for mobile robot navigation: A statistical approach. In *AAAI-98*, 1998.

[121] S. T. Tuohy, J. J. Leonard, J. G. Bellingham, N. M. Patrikalakis, and C. Chryssostomidis. Map based navigation for autonomous underwater vehicles. *International Journal of Offshore and Polar Engineering*, 6(1):9–18, March 1996.

[122] C. Tyren. Magnetic anomalies as a reference for ground-speed and map-matching navigation. *The Journal of Navigation*, 35(2):242–254, May 1982.

[123] J. K. Uhlmann, S. J. Julier, M. Csorba, and H. F. Durrant-Whyte. A culminating advance in the theory and practice of data fusion, filtering, and decentralized estimation, 1996. Available from `http://ait.nrl.navy.mil/people/uhlmann/CovInt.html`.

[124] J. K. Uhlmann, S. J. Julier and M. Csorba. Nondivergent Simultaneous Map Building and Localisation using Covariance Intersection. Navigation and Control Technologies for Unmanned Systems II, 1997.

[125] R. Urick. *Principles of Underwater Sound.* New York: McGraw-Hill, 1983.

[126] J. S. Willcox, Y. Zhang, J. G. Bellingham, and J. Marshall. AUV survey design applied to oceanic deep convection. In *IEEE Oceans*, pages 949–954, 1996.

[127] B. Yamauchi and P. Langley. Place recognition in dynamic environments. *Journal of Robotic Systems*, 14(2):107–120, 1997.

THESIS PROCESSING SLIP

FIXED FIELD: ill. _____ name _____

index _____ biblio _____

▶ COPIES: (Archives) Aero Dewey (Eng) Hum

Lindgren Music Rotch Science

TITLE VARIES: ▶ ☐ _____

NAME VARIES: ▶ ☐ _____

IMPRINT: (COPYRIGHT) _____

▶ COLLATION: 242 P.

▶ ADD: DEGREE: _____ ▶ DEPT.: _____

SUPERVISORS: _____

NOTES:

cat'r: date:

page: 5178

▶ DEPT: M.E.

▶ YEAR: 1999 ▶ DEGREE: Ph.D.

▶ NAME: FEDER, Hans Jacob Sverdrup