51

Integrating Mapping and Navigation

by

Christopher Michael Smith

S.B., Ocean Engineering (1991)
Massachusetts Institute of Technology

S.M., Ocean Engineering (1994)
Massachusetts Institute of Technology

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

Doctor of Philosophy in Ocean Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1998

Author .....
Department of Ocean Engineering
May 27, 1998

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

Accepted by .....
J. Kim Vandiver
Chairman, Committee on Graduate Students

Integrating Mapping and Navigation

by

Christopher Michael Smith

Submitted to the Department of Ocean Engineering
on May 27, 1998, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Ocean Engineering

Abstract

The primary obstacle for AUV navigation and mapping in unknown environments is the difficulty of effectively incorporating measurement data to aid in navigation. State of the art approaches to this concurrent mapping and localization (CML) problem fail because they do not fully account for data association, navigation, and prior model uncertainties. This thesis develops an integrated mapping and navigation (IMAN) algorithm that accounts for these uncertainties within a general, unified framework. A theory for non-separable hybrid estimation in the presence of navigation errors is formalized to address the novel structural aspects of this problem. Two primary focus points are maintained: enhancing the estimation and decision-making capabilities of a CML algorithm through appropriate representation of parametric uncertainty and event ambiguity and management of algorithmic complexity to maintain operability over short (tens of seconds) data sets. A Bayesian analysis is used to develop an appropriate prior knowledge representation strategy and to examine the fundamental constraints of CML. A functional implementation of IMAN is evaluated using Monte-Carlo simulation, and its performance is contrasted with alternative algorithms. This initial implementation of IMAN provides navigation and mapping performance comparable to the state of the art when complexity does not overwhelm the decision process. Although further research is required to provide robust recovery from highly ambiguous situations, IMAN is shown to be a valid generalized approach to CML. The extensions to estimation theory developed in this thesis provide a powerful new basis for reasoning about events within the filtering process that has a broad applicability to navigation, mapping, and control problems in robotics.

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

Acknowledgements

There have been so many people that have shaped my life and my thinking these past years. I would like to thank everyone, but will restrain myself to those special individuals whose impact has been immeasurable. You have my sincerest gratitude for all of your help and encouragement. I would like to begin by thanking my parents, my grandparents, and the rest of my family. They have supported me throughout the strange trip that my education has been, even when they didn't understand what I was doing. Their caring and belief in me have made it all possible. John Leonard has been so much more than the typical advisor. I know of no other professor who would put so much time and passion into true education. The long discussions and easy give-and-take he fostered have often been the springboard of inspiration. Intellectual growth is highly dependent on a good environment, and I was particularly lucky to be surrounded by a research group in which I could thrive. Drew Bennett and Jacob Feder provided a sense of cooperative advancement and skeptical advocacy that honed my ideas and gave me the intellectual confidence to tackle difficult problems head-on. Finally, I never could have made it to this point without Stoy Ray. Stoy has celebrated with me through the good times and kept me sane through the hard times. No matter how hectic everything else was, I always had a comfortable place at home where I could block it all out. Thank you all again for making all this work worthwhile.

Contents

List of Figures

List of Tables

List of Definitions

List of Notation

In the development of the integrated mapping and navigation algorithm and the description of its implementation, considerable use is made of notation. The variety of fields from which this research derives, moreover, make use of overlapping and conflicting standard notations. We have attempted to use a concise and consistent notation throughout this thesis. 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 used in general for sets, matrices, functions, and Jacobian. Finally, caligraphic letters are used as type identifiers. Because several indices are often needed to specify a variable, the following conventions have been adopted. Post-subscripts are reserved for a time index and also any conditioning factors. A common simplification for such conditioning is borrowed from the Kalman filtering literature. The subscript $k|k-1$ indicates an estimate at time k conditioned on all information through time $k-1$. Similarly, the subscript $k|k$ indicates an estimate at time k given all information through time k. Pre-subscripts are reserved for the primary index of a variable. If an additional index is required, it is indicated by a post-superscript. Pre-superscripts are used for type identifiers or to indicate a specific element of a vector variable. Two examples are provided to help clarify these conventions.

Example 1: ${}^{\mathcal{R}^+}_{t}x^{q}_{k|k-1,r\omega_k}$

This indicates a vehicle state estimate x, at time k, conditioned on all information through time $k-1$ and the r^{th} assignment, ω, from time k. The t^{th} vehicle state estimate from the q^{th} sub-tree level is specified. Also, the variable represents a partially updated state estimate, $\mathcal{R}^+$.

Example 2: ${}^{n}\xi$

This indicates the north, n, element of a feature state estimate ξ.

We now present the list of notation used in this thesis. Symbols are defined first, followed by Roman then Greek letters.

$\mapsto$: resolves to; the decision at the tail of the arrow is required to resolve to the hypothesis at the head of the arrow

$\twoheadrightarrow$: is representable as; the object at the tail of the arrow can be represented as (but is not equivalent to) the object at the head of the arrow

$\vdash$: has a; the object (or class) at the bar has or owns the object (or an instance of the class) at the dash

$\rightarrowtail$: derives; the class at head of the arrow derives from the class at the tail of the arrow

$\in$: is in; the object to the left is an element of the set to the right

a: index over a previous set of updated feature states

$\mathcal{A}$: type identifier for association hypotheses

b: index over a previous set of updated vehicle states

c: the ontological distance function

C_d: dependency compatibility function

C_D: dependency set compatibility function

$\mathcal{C}$: type identifier for the completion rate metric

d: a dependency

D: a dependency set

$\mathcal{D}$: type identifier for dispositional decisions regarding feature tracks

e: index over the dependencies in a dependency set

e: an error vector

E: a set of error vectors

f: a model dynamics function

F: the Jacobian of a model dynamics function

$\mathcal{F}$: type identifier for a feature class

g: an inverse measurement function

g^*: an innovation function

G: the Jacobian of an inverse measurement function

G^*: the Jacobian of an innovation function

$\mathcal{G}$: type identifier for the global error metric

h: a measurement function

H: the Jacobian of a measurement function

i: index over a set of features

I: an information matrix

j: index over a set of projected feature states

J: the Jacobian of a function

k: index over the set of time cycles

l: index over a set of projected vehicle states

L: a likelihood function

$\mathcal{L}$: type identifier for the relative error metric

m: index over a set of measurements

$\mathcal{M}$: type identifier for miss hypotheses

n: index over a set of updated feature states

N: the ordinality function, the number of elements in a set

$\mathcal{N}$: type identifier for new feature hypotheses

o: index over a set of decisions

$\mathcal{O}$: type identifier for ontological decisions

p: index over a set of hypotheses

P: an estimated state error covariance matrix

$\mathcal{P}$: type identifier for projected states

q: index over a set of sub-tree levels

Q: a process noise covariance matrix

r: index over a set of assignments

R: a measurement noise covariance matrix

$\mathcal{R}$: type identifier for assignment root states

$\mathcal{R}^+$: type identifier for partially updated states

s: index over a set of assignment root states

S: an innovations covariance matrix

$\mathcal{S}$: type identifier for spurious measurement hypotheses

t: index over the set of states in a sub-tree level

t: a continuous time variable

T: the duration of a time cycle

u: index over the set of hypotheses regarding a decision

U: a decision hypothesis index function, indexes hypotheses from a set that pertain to a given decision

$\mathcal{U}$: type identifier for updated states

v: index over the set of hypotheses in an assignment

V: an assignment hypothesis index function, indexes hypotheses from a set that pertain to a given assignment

$\mathcal{V}$: type identifier for a vehicle class

w: index over a set of updated vehicle states

$\mathcal{W}$: type identifier for measurement model classes

x: an estimated vehicle state

X: a set of estimated vehicle states

y: a vehicle track

Y: a set of vehicle tracks

z: a measurement

Z: a set of measurements

$\mathcal{Z}$: type identifier for measurement origin decisions

α: index over the set of leaves compatible with a hypothesis

β: index over the set of leaves compatible with an assignment

γ: a gating threshold

δ: a decision

δ^*: the Kronecker delta

Δ: a set of decisions

ε: an error

ζ: index over the set of error vectors

η: a feature track

H: a set of feature tracks

θ: a hypothesis, a pitch angle

ϑ: a class of hypotheses

Θ: a set of hypotheses

ι: index over the set of feature types

κ: index over the set of measurement types

λ: a clutter density parameter

μ: a measurement model

ν: an innovation

ξ: a vehicle estimated state

Ξ: a set of vehicle estimated states

π: ratio of area to diameter for a circle

ρ: a feature separation parameter

σ: the standard deviation of a distribution, particularly when identifying the probability of highest density regions

ϕ: a dynamic model

φ: a bearing angle

ψ: an angle, particularly a yaw angle

ω: an assignment

Ω: a set of assignments

Chapter 1

Introduction

Navigation is one of the fundamental capabilities required for the successful operation of autonomous underwater vehicles (AUVs), and in fact, for any mobile robot. Knowledge of the robot's position is essential for oceanographic sampling, mapping, and even simple transit. Traditional navigational techniques function reliably only in restricted mission domains. They require operation near a known acoustic array or allow missions of limited duration in unknown areas. Recent non-traditional navigation techniques have extended AUV operability to venues for which highly accurate bathymetric or gravimetric maps are available. There remains a need for navigation solutions which are valid in *a priori* unknown regions. In this thesis, we begin to explore what promises to be a rich source of such navigation solutions: the incorporation of environmental data to aid in navigation through concurrent mapping and localization (CML). Incorporating environmental data aids navigation by providing an otherwise unavailable source of absolute position information. While related techniques, such as multiple target tracking and stochastic mapping, can help in understanding the issues involved, their failure to address all of the sources of uncertainty intrinsic to concurrent mapping and localization limits their applicability. The current integrated mapping and navigation (IMAN) algorithm explicitly addresses the issues of data association uncertainty, vehicle navigational uncertainty, and prior

model uncertainty. The focus of this thesis is to enhance the navigational estimation process and to manage the computational complexity any such algorithm entails. By reducing, or even bounding, the navigational uncertainty of AUV operation in unknown regions, we extend the mission capabilities for marine robots.

1.1 The mine-countermeasures mission

Consider the task of mine-countermeasures (MCM) in shallow and very shallow water. Mine-countermeasures are the detection and neutralization of mine-like objects, usually with the goal of providing a safe avenue for landing on a beach [82, 14, 34]. While there are a variety of devices which may need to be cleared, there are certain characteristics which can be used to correctly identify mine-like objects with a high degree of certainty [92]. This mission is difficult for a number of reasons. Forgetting for a moment any difficulties in detecting mine-like objects, complete coverage of a specified region is necessary. The region to be de-mined is most likely unknown *a priori*. Also, operation in shallow and very shallow water (200 to 10 feet deep) entails possibly significant and hard-to-model environmental forces.

Traditional efforts in MCM have employed military personnel or trained marine mammals to identify mine-like objects [68, 72]. Novel approaches are desired to reduce the danger associated with such direct involvement while maintaining a high level of performance in mine clearing. Recent efforts have explored the use of crawling robots, particularly for very shallow water and surf zone MCM [43]. These approaches rely on randomized motion and large numbers of expendable robots to provide coverage for a specified region. While such crawling robots may be effective in the surf zone and for many mines in the very shallow water region, moored mines and larger coverage areas limit their applicability as a general solution to shallow water and very shallow water MCM. In these regions, autonomous underwater vehicles (AUVs) provide a more appropriate platform for mine-countermeasures.

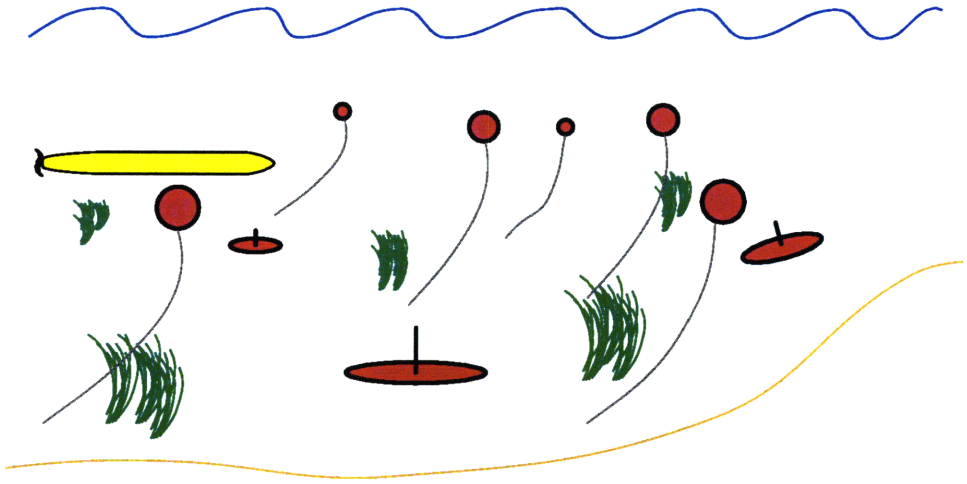


Figure 1-1: The mine-countermeasures mission. A vehicle must identify and map all mine-like objects in a specified region. In the shallow and very shallow water zones, mines may be on the bottom, moored, or even buried. Accurate mapping, and thus navigation, is necessary to enable neutralizing or avoiding the identified mines in the future.

One or more AUVs equipped with commercially-available sonar transducers can provide a solution to shallow water and very shallow water MCM if the problems of navigation are overcome. Figure 1-1 shows a simplified scenario for an AUV solution to MCM. The goal for the AUV is to detect all mine-like objects in the specified region and to map these objects accurately so that they may subsequently be neutralized. However, AUV navigation in such an *a priori* unknown region presents a significant problem. The pre-deployment of an acoustic array is often infeasible. Shallow water operations entail exposure to more volatile environmental forces, degrading the performance of dead-reckoned navigation. And yet accurate navigation is essential over the entire mission. Fewer vehicles (in comparison with the crawlers used in surf-zone MCM) call for a less random approach, necessitating navigational accuracy to ensure coverage of the entire specified area. Additionally, detected mine-like objects must be accurately mapped. Thus navigation is perhaps the key technology in realizing an AUV mine-countermeasures solution.

In a traditional conception of the mapping problem, that is, viewing mapping

and navigation separately, the map being produced is not involved in estimating the locations of the vehicle and detected objects. Dead reckoning is used for navigation, and this navigation information is used to estimate object locations. Detected objects are added to a map. Figure 1-2 illustrates this process and the extensions used to integrate mapping and navigation more fully. Existing concurrent mapping and localization techniques use the map to aid in estimating vehicle and object locations. This assumes that navigational events, such as measuring a particular object, can be resolved with certainty. Integrated mapping and navigation attempts to (1) detect navigational events correctly and (2) estimate vehicle and object locations. The increasing interdependence of the mapping and navigation tasks is designed to exploit what knowledge can be extracted from *a priori* models and new measurements without making unwarranted assumptions about the problem domain.

The evolution from traditional mapping to IMAN is a change from making assumptions about the exact nature of the environmental context of the vehicle to a more thorough reasoning using *a priori* models of what might form this environmental context. This change is designed to incorporate the uncertainties of navigation more realistically. However, this places a greater emphasis on prior knowledge representation and reasoning about navigational events. Considering navigational events in addition to vehicle and object states transforms the problem to a hybrid estimation space; we are estimating both continuous parameters (the vehicle and object states) and discrete hypotheses (the navigational events). While hybrid estimation is not new, the many types of uncertainty involved in navigation make this extension non-obvious. The challenge is to enhance estimation by more realistically representing the navigation problem while maintaining the computational complexity incurred by hybrid estimation at manageable levels.

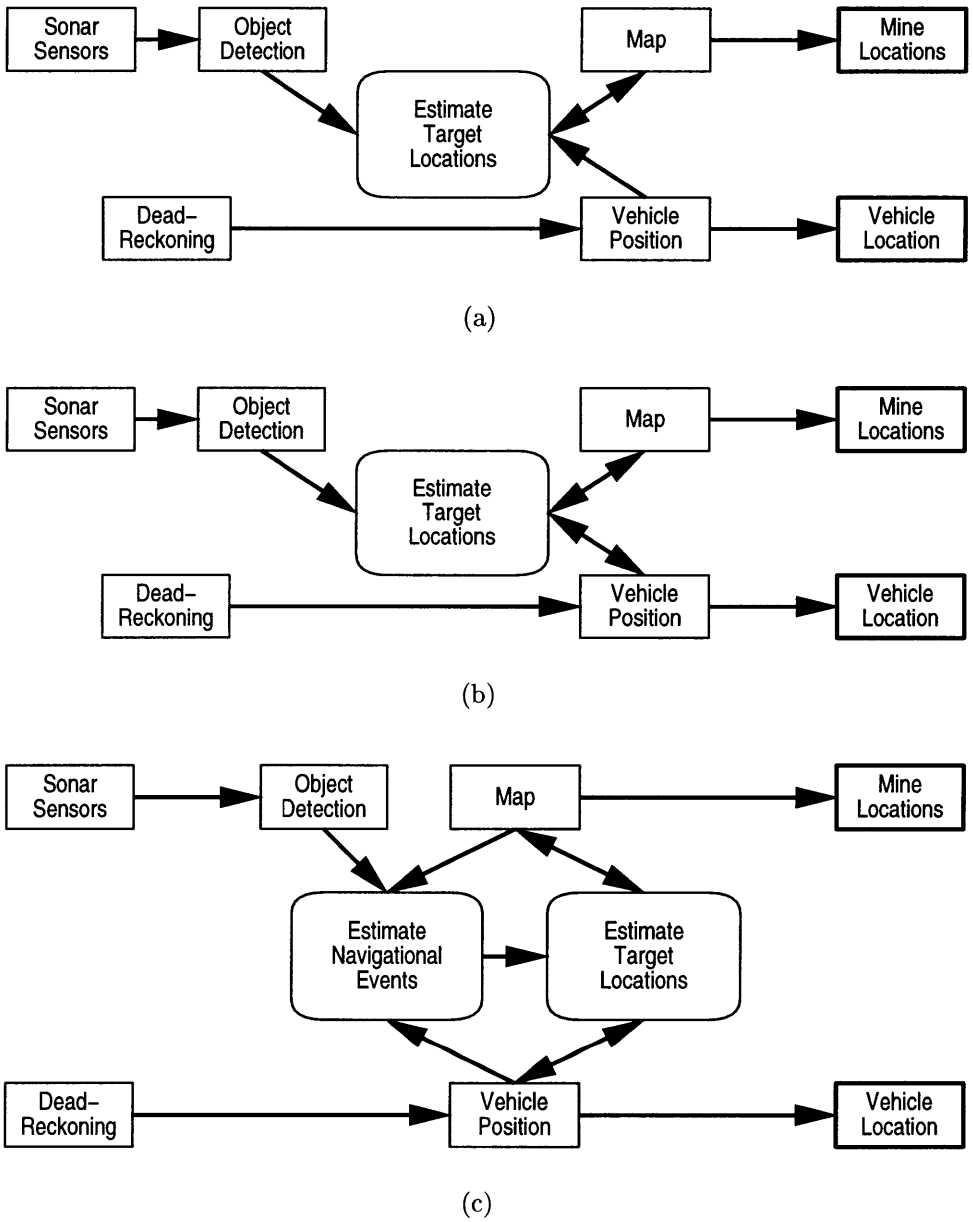


Figure 1-2: The process of mapping. (a) Traditional mapping techniques view mapping as separate from navigation. (b) Concurrent mapping and localization uses the evolving map to aid in estimating vehicle and object states. (c) Integrated mapping and navigation further integrates mapping and navigation. The evolving map is used along with estimated vehicle state to estimate both navigational events and vehicle and object state.

1.2 Integrating mapping and navigation

Incorporating environmental data in navigation and realistically representing the concurrent mapping and localization problem require a fundamental rethinking of current navigation techniques. Integrating mapping and navigation means revisiting sources of uncertainty in navigation and their impact on both state estimation and navigational event detection. There are three major sources of uncertainty which affect this hybrid estimation problem and which are not fully represented in existing navigation technologies: data association uncertainty, vehicle navigation uncertainty, and prior model uncertainty. The goal is to integrate mapping and navigation through the incorporation of environmental measurements into the navigation process and to explicitly represent these forms of uncertainty so that decisions are made in light not only of the estimates provided, but the presence of these uncertainties as well. IMAN can be thought of as a generalization of existing technologies which partially achieve this goal.

Stochastic mapping (SM) is a continuous estimation formulation which estimates (1) vehicle state and (2) feature states. However, associations between features, feature estimates, and measurements are assumed to be provided.

Multiple-hypothesis tracking (MHT) is a hybrid estimation technique which estimates (1) the set of features which are present, (2) data associations between measurements and proposed features, and (3) feature states. However, accurate vehicle location information is assumed to be available.

Concurrent mapping and localization (CML) is a general navigation problem which estimates (1) vehicle state, (2) the set of features which are present, (3) data associations between measurements and proposed features, and (4) feature states. Data association uncertainty, vehicle navigation uncertainty, and prior model uncertainty are explicitly represented.

Figure 1-3 shows the relationship between these three approaches. Below we consider the impact that these three major sources of uncertainty have on concurrent mapping and localization.

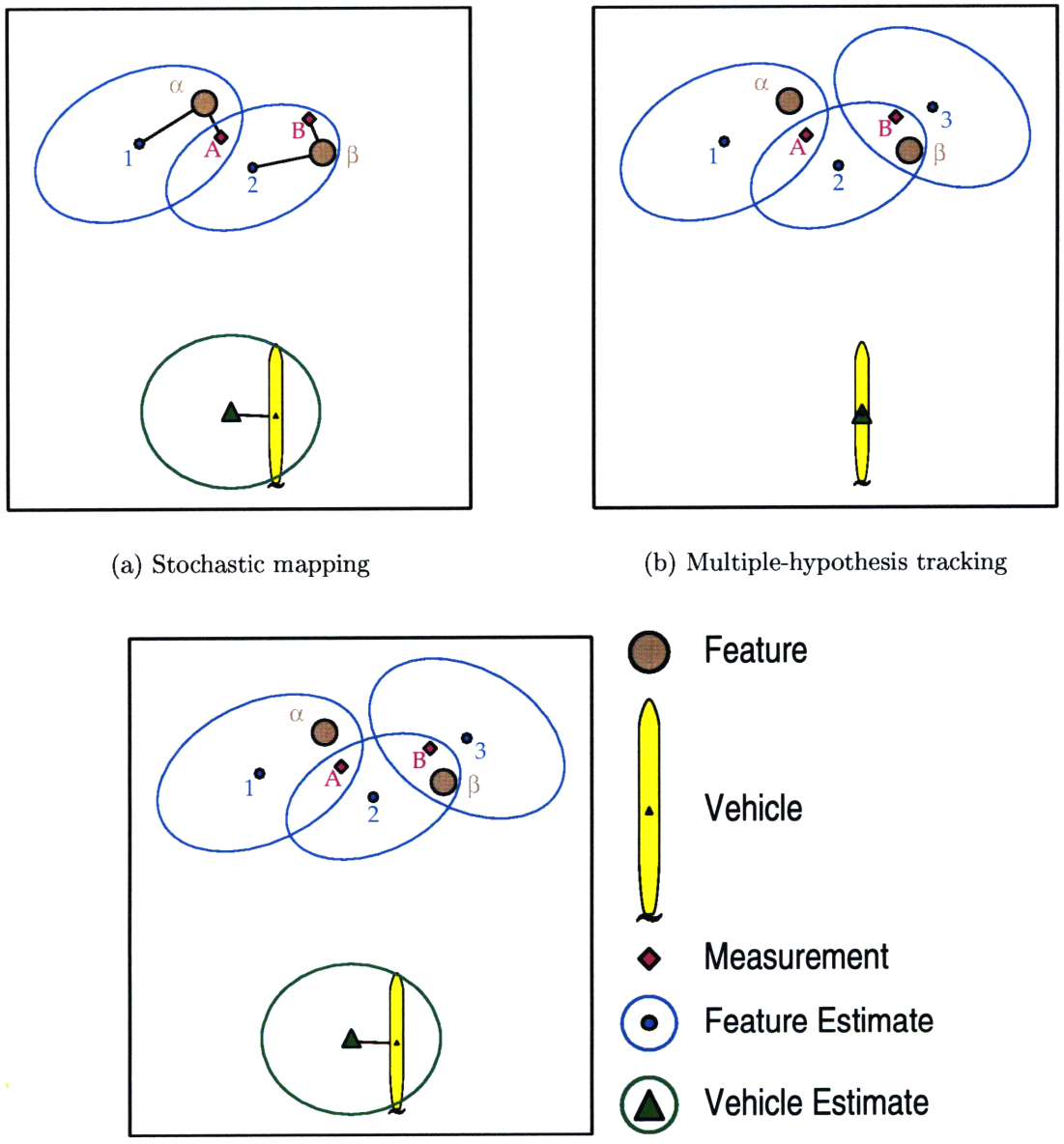
1.2.1 Data association uncertainty

Data association uncertainty is uncertainty about the source of a measurement or the status of a feature being tracked.

Data association is resolved in traditional navigation techniques by using beacons with distinct carrier frequencies as position references. In this way, measurement source identity is conveyed as part of the measurement itself. Objects in the environment, however, cannot in general be uniquely identified. Figure 1-4 illustrates this difficulty. Uncertainties and noise in both measurements and existing object state estimates lead to ambiguity in associating measurements with object models. The data association problem becomes less important as the density of returned measurements decreases. In such cases, the vehicle may be able to navigate with enough accuracy that simple gating techniques successfully disambiguate measurement source identity. However, real-world data is often degraded to the point where methods to cope with data association uncertainty explicitly are required. While data association uncertainty has been addressed in the multiple target tracking community [5], the presence of vehicle navigation uncertainty prevents the direct applicability of such techniques. IMAN requires a more general approach to modeling navigational events so that data associations may be correctly identified during concurrent mapping and localization.

1.2.2 Vehicle navigation uncertainty

Vehicle navigation uncertainty is a problem, particularly in extended missions, when there are no absolute position references available [41]. Inertial navigation systems



(a) Stochastic mapping

(b) Multiple-hypothesis tracking

(c) Concurrent mapping and localization

Figure 1-3: Approaches to navigation and mapping. Stochastic mapping (a) estimates both vehicle and feature states, but assumes that data association information is provided. Multiple-hypothesis track (b) estimates the data association possibilities, but assumes that vehicle location is known. Concurrent mapping and localization (c) is a generalization of these approaches which explicitly represents data association uncertainty, vehicle navigation uncertainty, and prior model uncertainty.

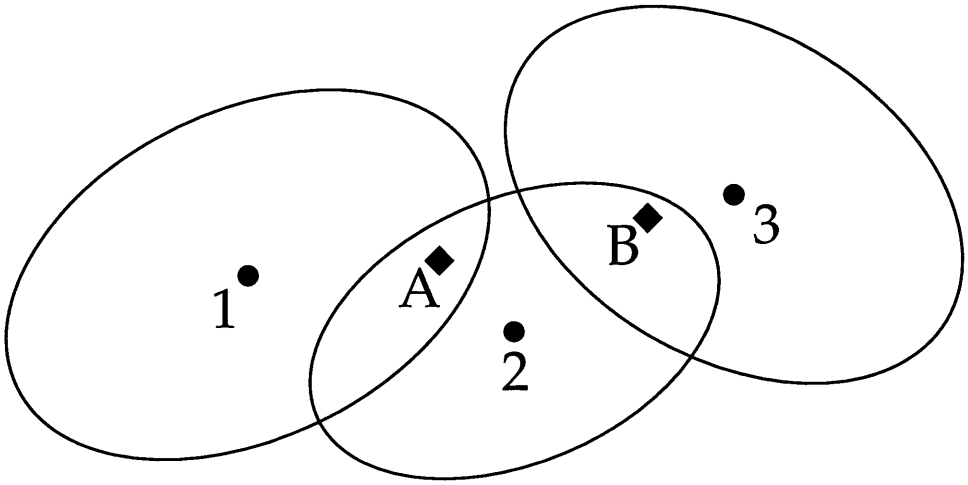


Figure 1-4: The data association problem. Because of navigational and measurement uncertainty, matching measurements to their origins may not be a simple task. The vehicle is modeling objects 1, 2, and 3. State estimates are shown with two-sigma error ellipses. Measurements A and B are detected. There is uncertainty about the origins of the measurements and the disposition of the modeled objects that is only resolved once data association decisions are made.

and Doppler velocity sensors may slow the growth of positional uncertainty, but inevitable measurement noise means that any system relying solely on dead reckoning for navigation will suffer unbounded error growth.

Vehicle navigation uncertainty is uncertainty regarding the vehicle velocity or orientation due to the finite accuracy of the navigational system, whether it is based on dead-reckoning or an inertial system. **Vehicle navigation uncertainty** leads to error growth in the estimated vehicle position.

While accuracy in navigational systems varies considerably, all AUVs are limited by navigation uncertainty in long missions unless external position references are utilized. Environmental objects offer a source of such position references. By observing environmental objects assumed to be stationary or moving with predictable trajectories, position uncertainty can be reduced and even bounded. It should be noted that such measurements offer only relative position information rather than an absolute, or global, position reference. This affects map representation and reasoning about

navigational events, as will be seen.

1.2.3 Prior model uncertainty

Finally, there are uncertainties in the prior models which are used.

> **Prior model uncertainty** refers to the mismatch between the model used to describe an environmental object or process and the actual phenomenology of vehicle measurements.

This may be because of model simplification, because instances of a class of environmental objects vary stochastically, or because a supposed object is being modeled to capture its phenomenology when the environmental context of the vehicle is not reducible to such a model. Existing navigation techniques tend to take advantage of simplistic representations of prior knowledge because of the assumptions they make about navigational events. The increased importance IMAN places on reasoning about navigational events requires a more realistic treatment of prior knowledge representation. Prior models of environmental objects and the measurement processes through which they are perceived are necessary to form reasonable hypotheses about navigational events.

1.3 Improving navigational estimation

Our investigation of integrated mapping and navigation will focus on two aspects of the problem which we feel are crucial to the success of initial implementations:

- enhancing the estimation and decision-making capabilities relative to state-of-the-art navigational algorithms and
- managing the algorithmic complexity of IMAN.

1.3.1 Enhanced estimation and decision-making

Current efforts at concurrent mapping and localization expand the Kalman filter to accommodate models of environmental features in addition to the vehicle state [94, 10, 44]. Decisions about data association, the detection of new tracks, etc. are resolved as they arise based on assumptions about the prior models involved. In the more traditional approach of the Kalman filter, any decision-making is hidden in either vehicle control inputs or the definitions of the dynamic and measurement models. As navigation algorithms are expanded to encompass a more flexible model of the environment and the vehicle, the role of these data association decisions becomes increasingly important. Such decisions are discrete not only in time (as the estimation algorithm may be), but also in possible outcomes. Data association decisions, for example, have a set of discrete navigational events as their possible outcomes. It is a hypothesis-testing or detection problem. Such detections are made by default in the traditional Kalman filter (by definition of the *a priori* model); they are made with little fanfare in stochastic mapping, which, while it must explicitly model the choices, does so in a simplistic and limiting way.

In this thesis, we enhance the decision-making ability for concurrent mapping and localization. This project is informed by similar multiple-hypothesis testing algorithms in the field of multiple-target tracking. The primary steps are to make these decisions more explicit and to move to a hybrid estimation space. We seek now not only to estimate the states of the vehicle and environmental objects, but also to test hypotheses about navigational events. Simply making these decisions more explicit opens considerable potential for making more adaptive and deliberative decisions about navigational events. We develop the explicit modeling of decisions regarding navigational events and examine how such a hybrid estimation scheme can be implemented while realistically incorporating navigational uncertainties. We evaluate this hybrid estimation scheme in absolute terms and in comparison with the more automatic decision-making structure of existing techniques.

1.3.2 Complexity management

The attraction of Kalman-based approaches to estimation is their computational economy. Stochastic mapping implementations, for example, use a Kalman filter whose order increases as additional environmental objects are incorporated. The number of the states in the filter grows linearly with the number of objects, but computation grows with the cube of the number of objects [69]. There has not been, to date, much consideration of map management or other methods of reducing computational complexity for stochastic mapping, possibly because implementations have thus far concentrated on applications with a limited number of targets and a sensible, if *ad hoc*, strategy for identifying new objects and deleting unnecessary models. The move to a hybrid estimation space entails a large increase in algorithmic complexity, due to the explicit modeling of alternative possibilities for environmental events and the additional complexities created by delaying the resolution of decisions about navigational events. While some work has been done in the multiple target tracking literature on managing complexity for multiple-hypothesis algorithms, these strategies are not directly applicable to the concurrent mapping and localization problem. Complexity management is therefore a crucial component in the implementation of integrated mapping and navigation.

In this thesis, a basic level of complexity management is achieved. The goal is to allow the practical consideration of portions of realistic missions. This is achieved, essentially allowing mission segments to be simulated. The computational complexity manifests itself not so much in slowing the algorithm as in requiring unsupportable amounts of storage in those cases where complexity management fails. The complexity management strategies presented in this thesis are an outgrowth of a system-level consideration of the problem and are posed within the theoretical framework of the project of enhanced estimation and decision-making. Because of this, complexity management is less a separate focus than an addition to the priorities considered in the hybrid estimation process. This, we hope, aligns IMAN, as a navigation scheme,

more closely with decision theory, something that should benefit subsequent implementations which rely more heavily on the adaptive decision-making potential of integrated mapping and navigation.

1.4 Expanded problem statement

Integrated mapping and navigation is truly a much larger goal than could be obtained in a single dissertation. The project of the present document is to formulate this larger problem of IMAN and make initial strides in implementing an algorithm for incorporating environmental data in navigation. As a result of this, we concentrate on a particular subset of problems facing the development of IMAN. We feel that the issues of prior knowledge representation, explicit representation of the various forms of uncertainty, and choice of environmental model type must be addressed at this early stage in the development of IMAN and so concentrate on them in both theoretical developments and implementation considerations. Within this refined field of inquiry, we also identify as research focus points the goals of enhancing the estimation and decision-making capabilities, or the navigational event discrimination, of the algorithm and of managing the algorithmic complexity. As a result, we identify the problem statement for the thesis:

> Formulate the incorporation of environmental data as an aid to AUV navigation through the concurrent mapping of environmental objects and localization of the vehicle and demonstrate the performance of an implementation of this algorithm, focusing in particular on enhanced estimation and decision-making and complexity management.

1.5 The potential of IMAN

We have considered how the mine-countermeasures problem benefits from the improved navigation offered by IMAN. There are a wealth of additional applications which also stand to gain from this navigation technology. Any AUV mission where absolute position references, such as a pre-deployed acoustic array, are unavailable can improve navigation performance by incorporating environmental information. Even when a prior map of a region is available, IMAN offers a way to combine measurements with the *a priori* map more optimally, improving performance over the use of a static map that is assumed to be correct. The incorporation of environmental information is of use in other robotics scenarios as well. Land robots often have access to GPS (the global positioning system), but its use may be unreliable for accurate navigation, particularly in cluttered environments [101, 42]. Robot navigation in underground environments, as an aid to mining operations [31] or for geophysical research [6], has no access to GPS and can reap substantial benefits from integrated mapping and navigation. Throughout the field of robotics, regardless of the alternative navigation aids which are currently available, IMAN can provide additional information to improve navigation and reasoning about navigational events.

With its greater ability to reason about navigational events, the IMAN framework has the potential to expand beyond the traditional boundaries of mapping and navigation. By improving the ability of the vehicle to understand and reason about its environmental context, IMAN can provide a framework for situated navigation. We traditionally think of mapping and navigation solely in terms of fixed landmarks and static environmental objects. By exploiting a more thorough representation of prior knowledge, IMAN could consider dynamic objects and more ephemeral aspects of the environmental context. Moving targets can not only be tracked, but prior models of possible behavior could lead to advances in the action recognition problem, in which the vehicle detects specific behaviors of another vehicle or agent through observation alone [12, 83, 56]. Models of physical oceanography may be verified or contrasted by

reasoning about model predictions and vehicle observations [26]. These are difficult problems, and IMAN alone does not solve them, but integrating mapping and navigation and providing a basis for reasoning about navigational (and, more generally, phenomenological) events establishes a framework for addressing hybrid estimation problems in the presence of navigation uncertainty.

1.6 Thesis scope

There are two ways to discuss the scope of what we have attempted and accomplished in this thesis. First, there is a problem-level description of the issues which must be faced in the thesis. We maintain that the essential issue for concurrent mapping and localization is the realistic modeling of the forms of uncertainty that obscure the detection and identification of navigational events, including data association uncertainty, vehicle navigation uncertainty, and prior model uncertainty. Second, the thesis scope can be described in terms of the primary focus points which form its goals. Our focus points are

- enhancing estimation and decision-making regarding navigational events and
- managing algorithmic complexity.

These inform the structure of the thesis and shape the results which are to be considered. To reiterate, within the general problem of navigation, we consider the explicit inclusion of data association, navigation, and model uncertainties for concurrent mapping and localization with the aim of enhancing estimation and decision-making and managing complexity.

There are a number of additional issues that we are not facing, are simplifying, or are simply ignoring. Among those factors not considered in this thesis are extended questions of and alternatives to feature modeling, comparison with intrinsically one-step data association methods such as probabilistic data association [5],

and alternative software implementation structures. Additionally, we ignore cross-model estimate correlations. A discussion of the implications of this, and possible alternatives which either re-incorporate or do not at any point ignore these cross-correlations, is included in Chapter 3. However, in the interest of focusing on an initial implementation and reducing computational complexity as much as possible, we ignore these correlations for the present. The developments in this thesis are, in spite of the issues which have been ignored or simplified, quite substantial. A generalized theoretical framework for integrated mapping and navigation is developed. Additionally, significant strides are made in implementing this theory. The product is a navigational algorithm competitive with the current state-of-the-art and offering greater potential for extension to more capable navigation technologies.

1.6.1 Preview of the results and conclusions

At this point, it may be of some use to the reader to provide a preview of our results and conclusions. This will, we hope, provide a useful mental reference of the problem scope and our goals as the thesis is presented. There are a number of difficulties in analyzing the performance of an algorithm such as IMAN that extends current capabilities in a rather fundamental way (in this case, through the explicit incorporation of a number of types of uncertainty which current algorithms are incapable of directly addressing). Because of this, the performance of IMAN is examined from a number of viewpoints. First, there is the question of absolute performance. How good is IMAN at preventing the vehicle from getting lost? What are the limitations on its operation? How robust is it in realistic situations? These questions are considered by examining IMAN performance in simulation. Additionally, there is the question of contrasting IMAN with the current state-of-the-art. In response, a limited comparison of IMAN with dead-reckoned mapping and an enhanced version of stochastic mapping is considered. Figure 1-5 presents a representative result comparing the performance of IMAN and a dead-reckoned navigation algorithm. We demonstrate

that IMAN is competitive with state-of-the-art technologies. Part of the attraction of IMAN, however, is its potential for dealing with more complicated missions in a unified way, a distinct advantage over the somewhat *ad hoc* extensions that current algorithms need to function at even modest levels of environmental complexity. Finally, the results presented provide a compelling case that significant progress has been made in enhancing estimation and decision-making and in managing algorithmic complexity while more realistically addressing the uncertainties endemic to AUV navigation in *a priori* unknown regions.

The major conclusions to be drawn from the results of this thesis are that IMAN is a valid approach to concurrent mapping and localization and that its enormous potential is realistically attainable. At the close of this project, we are still in the initial steps of developing truly integrated mapping and navigation. This unified framework for examining navigational issues through hybrid estimation of vehicle and feature models and navigational events provides a powerful paradigm for future research in feature ontology, behavior recognition, and navigation in highly unstructured and dynamic environments. The current implementation succeeds in enhancing the estimation and decision-making capabilities of state-of-the-art navigation technologies and manages complexity well enough to process realistic data sets of limited duration. The extensibility of the design of the IMAN framework ensures that future research can build on this powerful theoretical and implementational base.

1.7 Thesis overview

In this chapter, we have introduced the problem of concurrent mapping and localization and described the potential that integrated mapping and navigation can provide. We have enumerated the scope of the thesis and briefly considered the results and conclusions of this research.

In Chapter 2, we explore existing research in more detail as a way of establishing

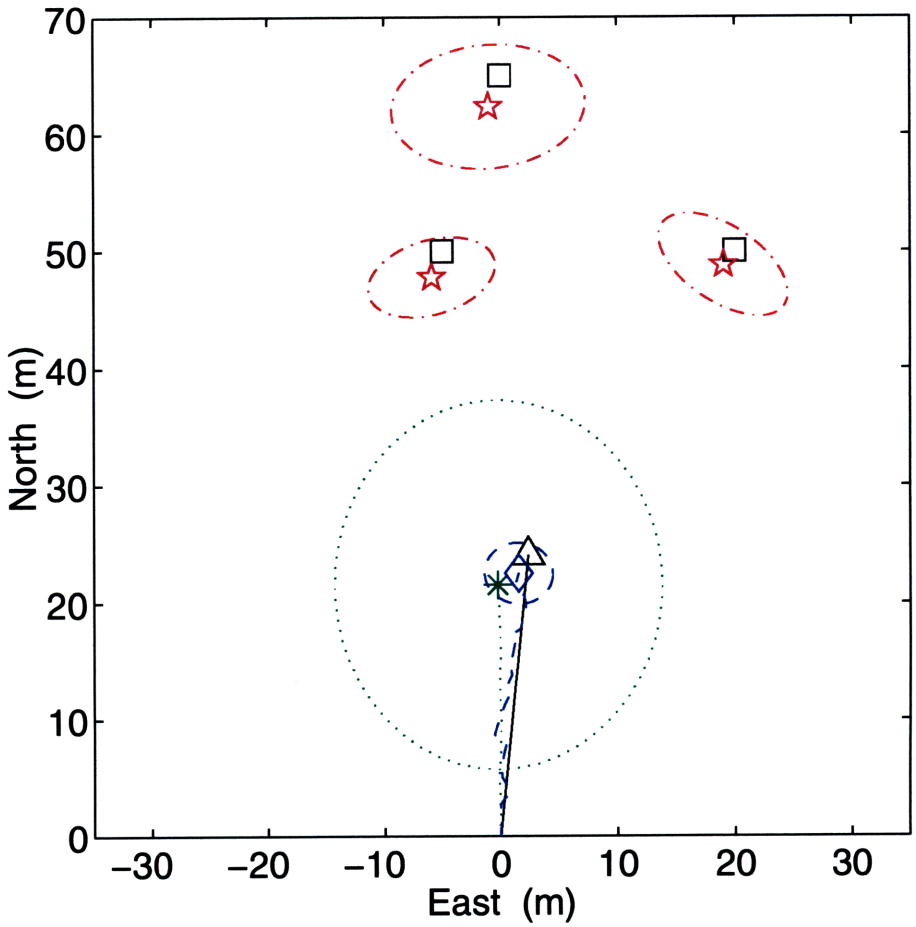


Figure 1-5: Comparison of IMAN and dead-reckoned navigation. Three point-like targets are used to improve the navigation estimate. Actual target locations are displayed as squares. The vehicle is attempting to swim directly north in the presence of an unknown cross-current. The actual vehicle path is shown as a solid line, with its final position indicated by a triangle. Uncertain relative measurements of the features are received each second. The dead-reckoned vehicle path is shown as a dotted line, as is its three-sigma error ellipse. The final dead-reckoned estimate of vehicle position is indicated by an asterisk. Using IMAN, vehicle navigation performance is improved and the features are mapped. The IMAN estimated vehicle path is shown by a dashed line, the final position estimate by a diamond, and the final three-sigma error ellipse by a dashed line. The final feature estimates are displayed as stars with their associated three-sigma error ellipses indicated by dashed-dotted lines. Feature measurements are degraded by non-unity probability of detection and the presence of clutter. IMAN successfully maps and models the features while rejecting spurious measurements and recovers vehicle motion in the presence of an unknown cross-current. Position uncertainty is greatly reduced by the incorporation of environmental data, using the three targets as positional references.

an appropriate approach to IMAN. We consider the current state-of-the-art in CML and the best results which have been produced. We examine related fields for the insights that similar problems can bring to the current project. Finally, integrated mapping and navigation is considered developmentally from a number of standpoints to render it within the existing research framework and to clarify its position relative to prior and future research.

Chapter 3 initiates the discussion of enhancing estimation and decision-making for concurrent mapping and localization. The theoretical foundations of hybrid estimation and the integrated mapping and navigation problem are developed within a Bayesian framework. Navigational event modeling for both ontological consideration of environmental objects and for dispositional questions relating to data association is presented. A probabilistic interpretation of the hybrid estimation problem is described and reduced to a recursive estimation strategy for projecting current possibilities and updating estimates as additional measurements become available.

Chapter 4 continues the discussion of enhancing estimation and decision-making, focusing on implementation issues. A C++ implementation is examined, illustrating a compelling object-oriented analysis of the problem which generalizes the current implementation to a framework for IMAN. Practical matters concerning model representation and decision tracking are also examined.

In Chapter 5, the impact of complexity management is considered. The developments of the previous two chapters are reconsidered with the goal of reducing algorithmic complexity and integrating complexity management and decision management at the theoretical level. The role of prior knowledge representation in generating algorithmic complexity is examined and methods for controlling this process adaptively are discussed.

In Chapter 6, the limitations of concurrent mapping and localization are explored. The growth of vehicle position error and its dependence on model uncertainty and navigational system uncertainty are examined. The information that can be extracted

from relative measurements is quantified and the overall utility of features as sources of information is considered for a common operational paradigm. The effects of data association uncertainty and the errors which this uncertainty causes are discussed. Each of these inquiries provides a look at how the kinds of uncertainty endemic to concurrent mapping and localization structure the navigation problem.

Chapter 7 analyzes the performance of the IMAN algorithm developed in this thesis. Appropriate performance metrics are derived. A number of contrasting algorithms are explored. The performance of IMAN and these contrasting algorithms in response to environmental clutter, track interaction, and number of features is presented. Some key topics are discussed to illustrate the directions in which the development of IMAN needs to proceed.

Finally, in Chapter 8, conclusions about the validity of IMAN are drawn. The capability of the current implementation and the enormous potential provided by the unified framework for hybrid estimation of vehicle state and navigational events are cited as compelling arguments for this approach to navigation in *a priori* unknown environments. A number of immediate research directions to refine the current implementation are considered. The long-term viability of this algorithm is also discussed, along with a variety of future research projects which will become increasingly realizable as the theoretical and implementational basis for IMAN is developed.

Chapter 2

Improving Concurrent Mapping and Localization

In this chapter, we consider the state of the art in concurrent mapping and localization research, examine related fields for ways to improve this level of performance, and justify integrated mapping and navigation, a novel approach to concurrent mapping and localization. We begin with a survey of existing CML implementations. While these implementations do not address the full problem as we have stated it, they offer insight into the research issues and implementation problems to be faced, as well as providing a performance benchmark. Several related research areas are discussed next. These illustrate ways in which current CML algorithms may be extended and provide a basis for choosing among alternative approaches to the problem. We then develop a particular approach to concurrent mapping and localization which we feel is well-suited to this problem domain. The basis for integrated mapping and navigation is described. IMAN is developed in several ways to illustrate its relation to the research discussed. Finally, we summarize our approach to solving the concurrent mapping and localization problem and the significant research issues involved and reiterate our focus points for this thesis.

2.1 Current implementations of CML

Concurrent mapping and localization is affected by uncertainties pertaining to both the continuous states to be estimated and discrete hypotheses regarding, for example, data association and the number of features present. Prior to this thesis, there has been no implementation which addresses data association uncertainty, navigation uncertainty, and model uncertainty in a unified theoretical framework. In Chapter 1, we motivated CML as a generalization of stochastic mapping and multiple-target tracking. The current research in CML is, however, united by a focus on navigation and the practical aspects of incorporating environmental information. Additionally, these implementations are all in the land robotics domain. Data association uncertainty, navigation uncertainty, and prior model uncertainty have all been identified as important factors in finding a solution to CML. The lack of a theoretical basis for dealing with these uncertainties together has, however, lead to a number of *ad hoc* extensions to traditional navigation and mapping technologies. Despite the limited success of these approaches, the need for an approach which can deal with all of these forms of uncertainty in a realistic and unified way remains apparent. These approaches do, however, underscore the ways in which uncertainty must inform a CML implementation and provide a clear grounding in the benefits of a unified approach to dealing with uncertainty.

Perhaps the most advanced implementation of a concurrent mapping and localization algorithm has been devised by Chatila *et al.* [53, 16, 46, 10, 15]. They have considered combining navigation and mapping of unknown regions in an office-like environment. This has led to simple feature-based environment modeling based on points (corners) and planes (walls). Their primary contribution [46, 76, 77] has been relocation fusion, a sub-optimal filter update strategy. Land robot navigation is often degraded by poor dead reckoning, a result of inadequate prior models of the nonlinearities involved in wheel slip, stiction, and related difficulties. Because of this, the projection model for the vehicle is much less accurate than the sonar measurement

and feature models. The traditional method for robustifying the extended Kalman filter to model errors is to artificially increase process noise to account for the additional uncertainty. However, the relatively poor information present in the vehicle model is then used to update the relatively more accurate models of the features, causing degraded performance. Mourtalier and Chatila address this problem in model uncertainty by sub-optimally updating the vehicle projected state before using this state to update the feature models, as illustrated in Figure 2-1. While this strategy in effect uses the measurement information twice when updating the features, performance gains are seen. Relocation fusion is thus a restriction of information flow between models based on prior decisions about their relative accuracy. Data association uncertainty has received somewhat less emphasis in the implementation. Initial work [76] relied on manual data matching, a feat only possible because of the limited data sets used. Subsequent work has automated this matching using simple nearest-neighbor matching [46].

Rencken [88] has also produced a concurrent mapping and localization algorithm. Simple results with point and plane models were produced. A focus was maintained on realistic and accurate modeling of sensor physics and vehicle dynamics. These results demonstrated that a CML algorithm using fixed sensors can bound position error growth, at least for limited missions.

Chong and Kleeman [20] have examined the use of local maps in concurrent mapping and localization. The problem they have addressed is reduced representation of inter-feature correlation by representing local groups of features together. A single correlation model is used to relate the uncertain models of points from two separate local maps. While the map representation strategies developed [20] remain simplistic, the assessment of map representation and improved understanding of the relationship between global and local mapping is key in addressing a number of issues.

Uhlmann [104] has examined some fundamental issues in concurrent mapping and localization. First, he (and later Julier [47, 86]) have explored the role of nonlinearities

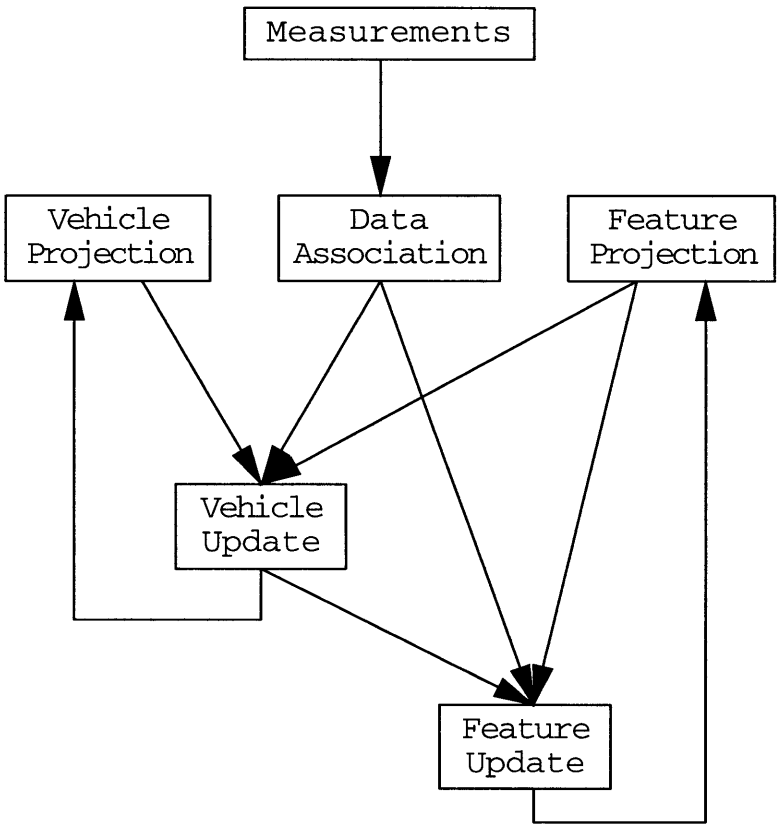


Figure 2-1: Process flow in relocation fusion. The optimal form of the Kalman filter requires that updates be carried out using projected states. Relocation fusion is a sub-optimal update technique designed to account for model uncertainty. The vehicle state is updated first using projected vehicle and feature states. The feature states are then updated using the projected feature states and the *updated* vehicle state. Although this method overestimates the amount of information received (effectively using information more than once), the restriction of information flow from the uncertain vehicle model more than compensates, resulting in a better overall performance.

in degrading filter performance. They have developed a κ-filter, which is essentially a simple bootstrap filter for providing a more complete description of probability distributions as they undergo nonlinear transformations. The focus was not to extend the probability models beyond the usual assumptions of Gaussianity, but to improve the fit of the Gaussian approximations used. Uhlmann has also developed a technique called covariance intersection [104] which attempts to provide a conservative update estimate when cross-correlations are ignored. Cross-correlation terms add significantly to the complexity of CML algorithms, and this technique has the potential to prevent filter divergence when correlations are ignored. However, the use of covariance intersection in some ways masks the map representation problem. It is possible that additional work on map representation can be used to improve system performance and to deepen our understanding of the relationship between relative and global maps.

Summary of the state of the art

None of these representations has addressed the combined problems of data association, navigation, and prior model uncertainties. Techniques such as relocation fusion, covariance intersection, and κ-filtering are all essentially *ad hoc* extensions which appeal for a general framework within which to pose the concurrent mapping and localization problem. The primary shortcoming of state of the art techniques is the representation of data association uncertainty. In the next section, we consider stochastic mapping, a forerunner of these CML approaches, and multiple-hypothesis tracking to see how different fields have handled these types of uncertainty in isolation.

2.2 Extending the state of the art

The current state of the art in concurrent mapping and localization clearly points out the need for a more general approach to the problem. Our project of improving the

theoretical basis for CML begins with a consideration of the theory underlying the mapping and tracking problems. The field of stochastic mapping provides insight into the continuous estimation problem in the presence of interacting uncertain models. Stochastic mapping, in addition, is the immediate precursor (at least in spirit) of many of the state of the art CML implementations considered above. The added consideration of data association uncertainty expands the problem to include discrete, as well as continuous, estimation. Multiple-target tracking has developed a number of techniques to implement hybrid estimation and provides a deeper understanding of some of its key aspects. In this section, we consider stochastic mapping and multiple-target tracking in the context of CML.

2.2.1 The basis of continuous estimation

Stochastic mapping [95, 93] is the mapping and localization technique underlying the state of the art implementations described above. It is an extension of Kalman filtering and models the entire map with a single state vector and associated error covariance, as shown in Figure 2-2. Although vehicle navigation error and (to an extent) prior model error are accounted for, data association information is assumed. When new features are encountered, the estimate and covariance vectors are extended to include the new estimate. A principle focus in the development of stochastic mapping has been the transformation of models through numerous points of view to allow correct identification when features are revisited. Despite this, all decision-making is assumed to happen independently, at a higher level of abstraction.

Implementations of stochastic mapping [88] often use simple nearest-neighbor processing to match measurements to feature models. This allows consideration of features for which absolute identification information is not available. This approach is prone to errors in data association, which can lead to estimation errors and filter divergence.

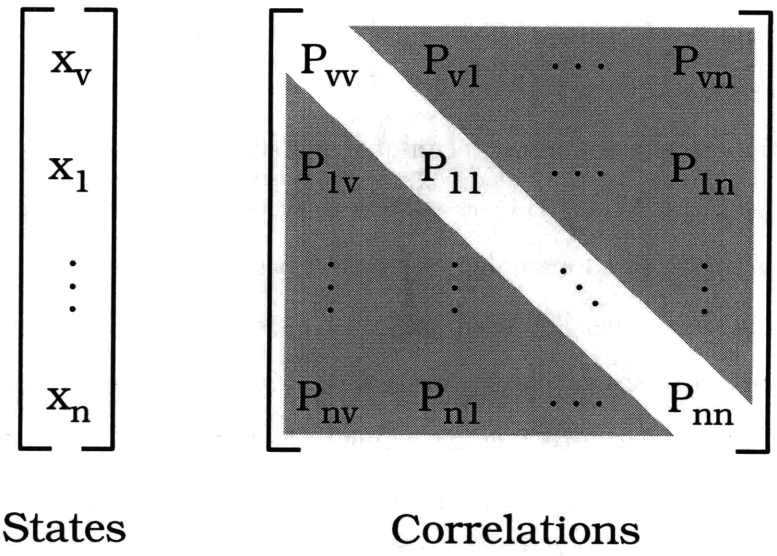


Figure 2-2: Structure of the state estimates in stochastic mapping. The entire map is represented using a single state vector formed of the states of the vehicle and the individual features. The associated estimated error covariance matrix contains both model covariances (the diagonal blocks) and cross-model correlations (the off-diagonal blocks, shaded in the diagram).

2.2.2 The basis of hybrid estimation

While stochastic mapping concerns itself only with the continuous estimation (that is, parametric estimation) problem, data association uncertainties require fundamentally discrete (or event-based) estimation. Data association has been extensively considered in the field of multiple-target tracking, where higher target densities, clutter, and maneuvering targets provide considerable data association uncertainty. However, multiple-target tracking techniques assume that sensor location is known at all times, an assumption that does not obtain during uncertain navigation.

One approach to data association in multiple-target tracking is to separate the discrete problems of estimating the number of targets and matching measurements to target tracks from the continuous filtering problem. This approach is typified by the probabilistic data association filter [5, 18]. While this approach is often implementationally compact, it suffers from an inherent smearing of the data. Also, the effects of errors are often difficult to identify. The limitations of the probabilistic

data association filter are most evident in cluttered environments and when moving targets cross paths.

An alternative data association technique, multiple-hypothesis tracking (MHT), was pioneered by Reid [87]. MHT postulates multiple competing hypotheses to explain the measurement phenomenology based on prior target models. As additional information is obtained, the likelihood ratios of these competing hypotheses shift to favor the correct data association hypothesis. By incorporating information from a more extended time period, MHT is more able to recover from ambiguous situations, such as path crossing and dense clutter. The theoretical basis for MHT was further developed by Chong, Chang, and Mori [17, 75]. Leonard [62] addressed model-based localization and sonar map-building using a Kalman filtering approach. Subsequently, Cox and Leonard [24] implemented a solution to land robotic map-building using MHT. Kurien [58] considered the theoretical extension of MHT to address both tracking and identification. Moran [74] extended MHT to consider curved two-dimensional surfaces in addition to corners and planes and applied this algorithm to underwater sonar measurements. It was demonstrated [73] that navigational uncertainty led to filter divergence. Barker *et al.* [7] consider the MHT problem using a probabilistic representation of prior terrain knowledge and discrete constraints, but maintain a mixed model; targets are feature-like, but other environmental objects are field-like.

Similar work has been done on a more field-based level for image processing and the use of image data to extract information [60].

2.3 The integrated mapping and navigation approach

In this section, we summarize previous research in a number of fields to justify a novel approach to CML, integrated mapping and navigation. IMAN is a unified theoretical framework for explicitly addressing data association uncertainty, navigation uncer-

tainty, and prior model uncertainty in concurrent mapping and localization. Key features of the IMAN approach include:

- extended Kalman filtering,
- feature-based environment modeling,
- non-separable hybrid estimation, and
- delayed decision-making.

The extended Kalman filter is an extension of filtering theory for nonlinear dynamic and measurement models. The relative feature measurements typically available in concurrent mapping and localization are quite nonlinear. Additionally, the vehicle dynamic model is nonlinear. Because of the nonlinearities, the assumption of Gaussianity does not obtain either.[1] The EKF is a point estimate of the actual process. The limitations of such an approach are well known [51, 99, 4]. The EKF is particularly prone to errors when the covariance estimate is large, which is of course when such errors are most harmful. The popularity of the extended Kalman filter in spite of these limitations is due to its compact form, its similarity to the linear Kalman filter, and its ease of implementation. A number of alternative are available [99, 27, 84]. A Bayesian interpretation of the filtering problem leads to a natural way to address nonlinearities [90]. The problem is that such approaches can seldom be specified in closed form. The field of nonlinear estimation has also produced a number of relevant results, including some analytic techniques for assessing the goodness-of-fit of estimations for nonlinear and non-Gaussian probability distributions [55, 54].

Feature-based modeling is common in the current CML implementations, but this is in part due to the development of these techniques within the land robotics community and for the purpose of mapping office-like environments, which are easily represented in terms of simple features. Existing work in the modeling of the

[1] For example, a distribution which is Gaussian in measurement space will have curves of constant probability in Cartesian space which are shaped like bananas or pears.

underwater environment provides a less clear manifest for a feature-based approach. Stewart [100] has developed a stochastic framework for mapping using a cell-based map. Such approaches are also referred to as field-based approaches. Even navigational approaches claiming to use environmental 'features' and maps [9, 103, 49] often use extended bathymetric structures, such as contour lines. Such an approach is significantly different in spirit than a true feature-based modeling approach. The fundamental aspect of features is that they are distinct (although sometimes not clearly) and that they are compact enough to be identified (*in toto*) using measurement sets on the scale of the sensor footprint. Larger structures can serve as landmarks, but fail to provide a model scale that is useful in terms of formulating hypotheses about feature dispositions, a case of not being able to see the forest for the kelp.

Hybrid estimation is encountered in a variety of contexts [96, 33]. It is most common in manufacturing and intelligent control systems, where the continuous plant models are driven using discrete control inputs [66]. Under such conditions, the separability of the discrete and continuous parts may be exploited to aid in analyzing the problem [5]. In filtering problems that must address event-based estimation, discrete and continuous portions of the problem are also often separated. This is accomplished in one of two ways. First, discrete decisions may be forced or probabilistically smeared instantaneously relative to the continuous estimation process. This is found in probabilistic data association filtering [5] and interacting multiple model filtering [29]. This simplification often leads to discrete estimation errors due either to the data smearing or the forced resolution of discrete estimation before a clear winner is identifiable. Second, multiple instances of the continuous filter may be instantiated for each discrete possibility, as in multiple model filtering [3]. This requires considerable storage and computation requirements, making multiple model filtering ineffective as the discrete portion of the estimation problem becomes complex.

The ability to delay decision-making is closely related to the hybrid estimation problem and its resolution. Comparisons of MHT and probabilistic data associa-

tion have demonstrated [29] that delaying decision-making reduces the occurrence and severity of data association errors. This allows MHT algorithms to perform in multiple-target tracking problems which lead to filter divergence or track loss when probabilistic data association filtering is used.

2.4 Summary

In this chapter, we have considered existing research to situate the current project of improving concurrent mapping and navigation. State of the art implementations were discussed, and the limitations they experience due to simplifying assumptions about the data association problem were highlighted. Extensions to the state of the art were considered by examining the fields of stochastic mapping and multiple-hypothesis tracking, which form useful bases for continuous estimation in mapping and navigation and hybrid estimation, respectively. Finally, a novel approach to concurrent mapping and navigation was developed. Integrated mapping and navigation utilizes feature-based environmental modeling and non-separable hybrid estimation to capture more fully the uncertainties that come to bear in concurrent mapping and localization. The key aspects of the IMAN approach are justified in the existing research of a number of fields, including filtering, estimation, tracking, and control. In the next chapter, we develop the theory for an integrated mapping and localization algorithm.

Chapter 3

Integrated Mapping and Navigation

In this chapter, we develop the theoretical basis for integrated mapping and navigation (IMAN). We begin by discussing hybrid estimation as it applies to concurrent mapping and localization (CML). The role of navigational events in estimating vehicle and feature position is considered, and the basic structural elements of hybrid estimation in the presence of vehicle position uncertainty are described. We then turn to the problem of prior knowledge representation. The role of vehicle, feature, and measurement models is examined. In this thesis, we make use of two dynamic models: (1) a simple model for a survey-class AUV and (2) a general model for two-dimensional point-like features. These models are presented along with models for dead reckoning and relative sonar measurements. Next, we examine some theoretical issues raised by discrete estimation in the presence of sensor location uncertainty. The dependency of continuous estimates on discrete decisions and the probabilistic basis of discrete estimation are detailed. Issues relating to the continuous estimation problem are then considered. A Bayesian interpretation of the estimation problem is discussed, and the role of cross-model correlation and map representation is addressed. Finally, the process of integrated mapping and navigation is presented. The

theoretical basis for each step of this process is detailed and an illustrative example is developed throughout.

3.1 Hybrid estimation and concurrent mapping and localization

Hybrid estimation is estimation over both discrete and continuous degrees of freedom [75, 81]. Discrete degrees of freedom in estimation are decisions among a discrete number of hypotheses. Continuous degrees of freedom estimate the value of continuously-varying parameters.

> **Hybrid estimation** is a process in which both discrete and continuous quantities are estimated.

Hybrid estimation is used in a wide variety of applications, but often some degree of separability between the discrete and continuous degrees of freedom can be exploited. Much of the work on hybrid systems maintains continuous plant models but simplifies the controller through a discrete specification [96, 81]. In multiple-model estimation schemes, continuous models are run in parallel to represent the outcomes of various discrete events or decisions [3]. Many algorithms which must model discrete events, such as probabilistic data association filtering [5] and interacting multiple model estimation [29], force decision to be made within a single time cycle, effectively decoupling the discrete degrees of freedom. Such separation of the discrete and continuous parts of the estimation problem is undesirable for IMAN. The discrete navigational events considered are reflections of discrete events in the world (or our model of the world). Forcing one-step decisions would prevent delayed decision-making, removing any advantage from the additional accumulation of evidence. Running parallel continuous models would simply be too computationally intensive. Concurrent mapping and

localization calls for a hybrid estimation scheme in which the discrete and continuous degrees of freedom interact. The challenge is to understand how this interaction should inform the structure and process of integrated mapping and navigation.

3.1.1 Navigational events

Traditional Kalman filtering is designed to find the (in some sense) optimal way to incorporate measurements with dynamic projection, assuming that we have an exact model of what is happening. However, there are often cases when "what is happening" needs to be estimated as well. The process of navigation, and in particular, concurrent mapping and localization, is characterized by discrete events. These events can include: (1) changes in the behavior of the vehicle or the features being modeled and (2) measurements.

> A **navigational event** is an event that can be modeled in more than one way. **Navigational events** are the primary source of discrete uncertainty in concurrent mapping and navigation.

In each of these cases, the navigator must decide, either explicitly or implicitly, what has occurred and how to modify the estimates of vehicle and feature state in response. Traditionally, a single model is used for the system dynamics [48, 4]. This dynamic model is assumed to hold at all times, describing the evolution of the system, and therefore, the estimates, through time. Likewise, a single measurement model is used. Each measurement (or set of measurements) is assumed to have a given explanation, including a specification of the states upon which the measurement depends. The move to a more general model, as required in concurrent mapping and localization, involves considering different possible outcomes for these cases, i.e. projection and updating. In this thesis, we for the most part retain the assumption of a given system dynamics model. This results in a one-to-one mapping when projecting vehicle and feature state estimates. We do, however, regard measurements as navigational

events. Data association uncertainty means that the states upon which each measurement depends are not given. The result is that multiple models regarding the data association of measurements must be considered.

3.1.2 General aspects of hybrid estimation

By considering discrete events in addition to parametric states in the estimation problem, we move from continuous estimation to hybrid estimation. As discussed in Section 2.3, previous research indicates that non-separable hybrid estimation is the appropriate approach for the problem of concurrent mapping and localization. This has the effect of changing the structure of the problem from that used for stochastic mapping, separable hybrid estimation, or multiple hypothesis tracking. In stochastic mapping, a single state vector is used to describe the state of the vehicle and features at any given time. This state vector has a single associated covariance matrix to estimate the error in the state estimate. The consideration of measurements as navigational events and the discrete estimation of these events does not necessarily contradict this structure. In multiple-model estimation, multiple versions of this global state vector and associated error covariance matrix are maintained to account for each possible resolution of all of the uncertain events. Such an approach, however, becomes rapidly uncomputable as the number of events to be estimated increases. To account for this, and to leave the structure open to extension regarding the inclusion of vehicle and feature dynamics as navigational events, we separate the vehicle state estimate and each of the feature estimates, as shown in Figure 3-1. Instead of a single global state vector, the vehicle and each proposed feature have a corresponding state vector and associated error covariance matrix. We consider the evolution of each feature and the vehicle independently. This change in structure complicates the specification of cross-model correlations, the off-diagonal blocks of the stochastic mapping error covariance matrix. Such matrices can of course be included at the cost of considerable added complexity. The role of these cross-model correlations will be

$$\begin{bmatrix} x_v \\ x_1 \\ \vdots \\ x_n \end{bmatrix} \qquad \begin{bmatrix} P_v & P_{v1} & \cdots & P_{vn} \\ P_{1v}^T & P_1 & \cdots & P_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ P_{vn}^T & P_{1n}^T & \cdots & P_n \end{bmatrix}$$

(a) SM state (b) SM covariance

$$x_v \quad P_v \qquad\qquad \begin{matrix} x_1 & P_1 \\ & \vdots \\ x_n & P_n \end{matrix}$$

(c) IMAN vehicle (d) IMAN features

Figure 3-1: The structure of state estimates. In stochastic mapping (SM), a single state vector (a) is used to represent the vehicle and all features. An associated covariance matrix (b) provides the model covariances (the diagonal blocks) as well as cross-model correlations (the off-diagonal blocks). In integrated mapping and navigation (IMAN), vehicle and individual feature models are separated, as in (c) and (d). State estimates and covariances pertain to a particular model; cross-model correlations are not represented.

discussed in detail in Section 3.4.2.

3.1.3 Features

Integrated mapping and navigation incorporates feature-based environment modeling. This is in line with the modeling strategies of existing concurrent mapping and localization implementations and multiple-target tracking.

A **feature** is a distinctive object in the environment that can be modeled to aid in navigation. A **feature** model captures the phenomenology of measurements that arise from the object being modeled as a **feature**.

However, it could be argued that AUV navigation occurs in very different environments than those usually encountered in multiple-target tracking applications. Bathymetric features seem the most likely candidates (at least initially) for AUV navigation in *a priori* unknown environments. Previous research however, has typically used a field-based, or cell-based, approach for modeling underwater environments [100, 30]. The representation of such environments in terms of discrete features of limited dimension has some potential advantages. First, process modeling and reasoning about the behavior and interaction of these features is, from a high-level standpoint, best accomplished with a feature-based approach. If the features being modeled are too large or too small, effective reasoning about navigational events and the features would be difficult, if not impossible, to provide.[1] Second, the consideration of features, rather than fields, allows a greater flexibility in the resolution at which discrete events are considered. Finally, using features as models improves the information available by estimating both *from where* and *from what* measurements are received.

There is an additional ontological concern in prescribing a feature-based environment modeling paradigm. There may not, in fact, be any such thing as features in the environment. To answer this, we appeal to a phenomenological viewpoint. The feature models that are used to characterize the environment are noumenal observers, in other words, they describe the experienced phenomenology without resorting to an epistemological imperative.

A **noumenal observer** is a model that captures the measurement phenomenology associated with a particular environmental occurrence. By noumenal, the real nature of the occurrence is not implied, only a prior model of the measurements that result from the occurrence.

[1] A balance needs to be maintained between sensor range and footprint; measurement and navigational event frequency; and feature dynamics, extent, and density. The focus of this thesis remains on the development of hybrid estimation for concurrent mapping and localization rather than these issues of feature modeling. As more extensive and varied feature models are developed, these tuning issues will become more immediate.

This means that, from a theoretical standpoint, feature ontology, that is, the specification of feature models, is best determined empirically. In this thesis, we concentrate on the use of point-like features. Such features can be found in realistic data sets and correspond to environmental occurrences such as lobster pots, small man-made devices such mines and buoys, and even small discrete rock outcroppings [67]. Because feature models can be regarded as noumenal observers, there may be no such thing as the *actual* state of a feature, only estimates of varying quality. The success of a feature model is determined by its predictive ability in accounting for the phenomenology presented by the measurements.

3.1.4 States

As discussed above, the integrated mapping and navigation approach concentrates on features and the vehicle individually rather than specifying the global state. Additionally, there may in fact be no actual feature states. This change in the structure of the estimation problem results in some altered nomenclature for discussing the vehicle and feature estimates. We consider an estimate to be the estimate of a single vehicle or feature. For example, the estimate of a point-like feature might consist of its north and east coordinates. This estimate will have an associated error covariance matrix, which estimates the error of the estimate and the correlations between each estimate variable within the model, that is, the particular vehicle or feature model. Cross-model correlations, if included, are considered separately. Because of the discrete degrees of freedom in the hybrid estimation problem, there may be multiple estimates for a feature or the vehicle at any given time. Thus each estimate will also have an associated likelihood representing an estimate of the absolute probability of that feature or vehicle estimate being the correct estimate of the model, i.e. representing the absolute probability that the discrete decision resolutions used to generate that estimate are correct. This collection of a model state estimate, its associated error covariance, and its likelihood, form an estimate.

An **estimate** is composed of the state estimate for a particular vehicle or feature model, its associated covariance matrix, and the likelihood that the decision resolutions used in generating the state estimate are correct.

Each estimate also has a context within the discrete estimation problem. This is specified by a dependency on some subset of the decisions that are as yet unresolved. The specification of the required resolution of these decisions for an estimate to hold is a dependency set.

A **dependency set** is the set of decision resolutions that are required for an estimate to be valid.

An estimate in context, that is, an estimate and its dependency set, is a state.

A **state** is an estimate in the context of the discrete estimation problem. A **state** consists of an estimate and its dependency set.

Note that there will be one or more states for each vehicle and proposed feature at any given time. Also, given a specific, consistent, resolution of all decisions, there will be a single state for each model (the vehicle and each proposed feature) given that set of decision resolutions. The state is thus the basic unit within the hybrid estimation problem.

3.1.5 Tracks

At any given time, the vehicle or a proposed feature may have multiple possible states. However, this set of possible states evolves in a logical way through time.

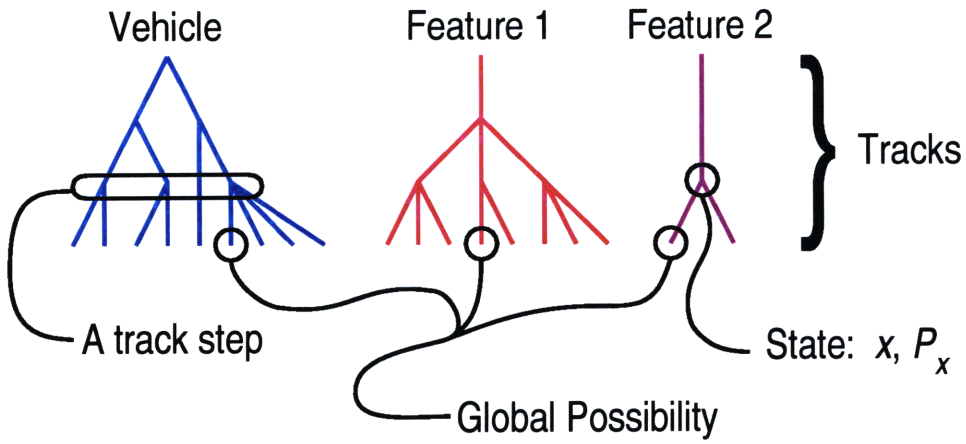


Figure 3-2: The structure of the IMAN estimation problem. Each model, that is, the vehicle and each proposed feature, is modeled by a track. Tracks provide a dual representation as trees and track steps. Track trees capture the causal relationships among states. Track steps capture the functional relationships of the estimation process. Each node of a track tree is a state. Global possibilities contain a single compatible state from each track.

A **track** is the structured set of possible states for the vehicle or a proposed feature through time.

The concept of tracks was introduced in multiple hypothesis tracking [75]. Under integrated mapping and navigation, each track is representable as a tree of states. As the feature estimate evolves, states are modified. If multiple hypotheses about state updating (or projection) are considered, multiple child states are included for that node state. The tree structure of the track captures the causality of both the continuous and discrete degrees of freedom of the estimation problem. In addition to this structure, a track can also be represented as a set of possible states at any particular point in the estimation process. As in traditional Kalman filtering, the estimation process is divided into projection and update steps for each time cycle. A track will contain a set of possible states for each of these steps. This set is referred to as a track step. A given track step might thus contain the set of possible updated vehicle states for a specific time cycle. Track steps, which do not contain any sense of

causality for the track, can be compared using the dependency sets of their individual states. Thus each vehicle and proposed feature is estimated as a track with this dual representation of track tree and track steps. Figure 3-2 illustrates the tree structure of tracks and the relationships among the concepts of track, state, and feature.

3.2 Prior knowledge representation

With the integrated mapping and navigation approach, vehicle and feature models are separated, and greater emphasis is placed on reasoning about how to appropriately model navigational events regarding these separate models. This places greater currency in the representation of these models and their manipulation, so that extended reasoning may be accomplished efficiently and appropriately given the circumstances. Thus, prior knowledge representation takes a more central role in IMAN than it has in existing concurrent mapping and localization implementations. Prior knowledge representation is also important for the case of AUV navigation because submarine bathymetry is an inherently more complex environment than the simple office environments, consisting of walls and corners, often considered [53, 20, 16, 15, 11]. In specifying models, there are three areas which must be considered. The first is system models for individual features. By interpreting features as noumenal observers of the environment, we assert that there are classes of features that may be estimated using one or more parametric prior models. The specification of these feature models is referred to as feature ontology.

> **Ontology** is the specification of the different models that may be used to represent the vehicle or an environmental occurrence.

The second consideration for prior knowledge representation is specification of one or more dynamic models for the vehicle. Vehicle modeling has received considerable attention in the context of control and planning [36, 65, 37, 89, 32]. The third

and final consideration for prior knowledge representation is measurement modeling. The vehicle is in general involved in multiple types of measurement. Some measurements give rise to no discrete uncertainty. These are typically proprioceptive measurements (measurements regarding only the dynamic state of the vehicle itself), local measurements (such as temperature, conductivity, pressure, or optical backscatter), or field-based exteroceptive measurements (such as Doppler velocity profiles). Such measurements can be appropriately handled within the traditional Kalman filtering paradigm. For integrated mapping and navigation, such models are separately specified, but are in essence identical to the usual Kalman update techniques. Other measurements are relative measurements involving the state of multiple models within the IMAN estimation scheme. These are usually subject to uncertainty about what proposed feature, if any, has participated in the measurement, and as such are the primary focus of the hybrid estimation process.

In this thesis, we develop two models. The first is used to represent point-like features; the second models the dynamics of a survey-class AUV. Two measurement models are also developed. Dead reckoning measurements are measurements of the vehicle speed, depth, and orientation. There is no discrete uncertainty associated with these dead reckoning measurements. Also, a relative sonar measurement model is specified for the detection of point-like features using a forward-looking sonar array. These sonar measurements are subject to discrete uncertainty regarding what feature, if any, is involved in the measurement.

3.2.1 Feature ontology

Feature ontology is the specification of parametric models which correspond to classes of supposed features in the environment. The purpose of such a model is to accurately describe the behavior (or one type of behavior) for the feature in question. This corresponds to the specification of the state projection step in Kalman filtering.

A **dynamic model** is a model of a possible dynamic behavior for a vehicle or feature. The specification of a **dynamic model** enables estimates to be projected through time.

A general implementation of IMAN could consider multiple competing hypotheses about how to model feature behavior. This might be necessary to account for changes in behavior (such as the sudden maneuvers of a target [107, 21]) or changes in feature type due to a refined understanding of its associated phenomenology (such as revising a planar model to a curved surface representation upon detecting a region of non-zero curvature [74]). In this thesis, such ontological events are not entertained. Projection of a given state is a one-to-one mapping based on a specified feature model. Each track that is proposed is associated with a specific feature model to be used for projection.

Point-like features

In this thesis, we consider a simplified environment model containing a single feature class: points. Point-like features can be extracted from actual sonar data and correspond to a variety of causative agents, ranging from lobster pots to rock outcroppings [67]. Such points are considered to be nearly stationary over the time during which the vehicle observes them. Each point is entirely specified by its north and east coordinates within the global reference frame, that is,

$$^{\mathcal{F}_{pt}}\xi = \begin{bmatrix} ^{n}\xi \\ ^{e}\xi \end{bmatrix}, \tag{3.1}$$

where $\mathcal{F}_{pt}$ denotes the point feature model. The point feature model is thus:

$$^{\mathcal{F}_{pt}}\phi_k : \quad \xi_{k+1} = {}^{\mathcal{F}_{pt}}f\left(\xi_k\right) + w_k, \text{where} \tag{3.2}$$

$$^{\mathcal{F}_{pt}}f\left(\xi_k\right) = \xi_k. \tag{3.3}$$

The feature model is denoted $^{\mathcal{F}_{pt}}\phi$; the time index k denotes the projection implied in Equation 3.2. Since point-like features are assumed to be nearly stationary, however, the point model is independent of time,

$$^{\mathcal{F}_{pt}}\phi_k = {}^{\mathcal{F}_{pt}}\phi \quad \forall \; k. \tag{3.4}$$

In addition to the assumed dynamics, $^{\mathcal{F}_{pt}}\phi$ includes an additive noise term, w_k. This is assumed to be a sampled Gaussian white noise process and is present for two reasons. First, the environmental object being modeled may have an associated stochastic component when modeled as a point. This is due to finite dimension, varying view angle, and other properties that may not be resolved parametrically by the feature model. Second, this process noise in included to account for model uncertainty and has the effect of robustifying the estimation of the process [1]. Without the additional noise, the feature estimate would become overconfident in its knowledge of the feature, leading to divergence [1]. The additional noise provides a method of discounting current knowledge of the feature to allow the estimate to adapt to unmodeled systematic variation in the feature.

The usual assumptions about the process noise are made [39]. The noise is assumed to be a white sequence and to be uncorrelated with the initial feature estimate, prior estimates of the vehicle and any other features, and any other noise processes. Additionally, the noise is assumed to be zero-mean, stationary, and Gaussian, with a covariance given by

$$\mathrm{E}\left[w_k \, w_k^T\right] = {}^{\mathcal{F}_{pt}}Q = \begin{bmatrix} {}^{n}w & 0 \\ 0 & {}^{e}w \end{bmatrix}. \tag{3.5}$$

Table 3.1: Point dynamic model ${}^{\mathcal{F}_{pt}}\phi$ parameters.

Parameter	Symbol	Value
north process noise covariance	${}^{n}w$	2 m^2
east process noise covariance	${}^{e}w$	2 m^2

For all of the simulated runs in this thesis, a point process noise covariance of

$$ {}^{\mathcal{F}_{pt}}Q = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \text{ m}^2 \tag{3.6} $$

is assumed, as shown in Table 3.1. This value is somewhat arbitrary, but was chosen based on simulated runs to achieve desired convergence characteristics for the feature estimate. Note that increasing the feature coordinate process noise would result in attaching more weight to new measurements, while decreasing this value would attach greater worth to the existing estimate. Setting the process noise too low can result in filter divergence due to overconfidence in the current estimate.

3.2.2 Vehicle ontology

The specification of a vehicle model is essentially identical to the specification of a feature model. The additional considerations are that a more complex model of the vehicle may be possible because of the availability of additional information (e.g. control inputs) and the fact that the vehicle may participate in additional measurement modalities.

A simple vehicle model

AUV control, particularly for survey-class vehicles, is typified by motion along line-segments. The vehicle moves in a straight line, occasionally performing maneuvers. Because of this operational paradigm, simple dynamic models can usually capture

vehicle behavior quite well. Our assumed model of vehicle motion is essentially a steady-state kinematic model. The vehicle is assumed to travel at a constant speed in the direction in which it is heading. Model uncertainty is compensated by added process noise and proprioceptive measurements (heading, pitch, depth, and speed). The vehicle state variables are the north and east coordinates in the global reference frame and the depth, speed, pitch, and yaw, that is,

$$x = \begin{bmatrix} {}^{n}x & {}^{e}x & {}^{z}x & {}^{u}x & {}^{\theta}x & {}^{\psi}x \end{bmatrix}^{T}, \tag{3.7}$$

where $\mathcal{V}$ denotes the vehicle dynamic model. The vehicle dynamic model is

$${}^{\mathcal{V}}\phi_k : \quad x_{k+1} = {}^{\mathcal{V}}f_k\left(x_k\right) + w_k, \tag{3.8}$$

where ${}^{\mathcal{V}}f_k$ is the vehicle dynamic function and w_k is additive process noise. The vehicle dynamic function is:

$${}^{\mathcal{V}}f_k\left(x_k\right) = \begin{bmatrix} {}^{n}x_k + {}^{u}x_k \cos\left({}^{\psi}x_k\right) \cos\left({}^{\theta}x_k\right) \\ {}^{e}x_k + {}^{u}x_k \sin\left({}^{\psi}x_k\right) \cos\left({}^{\theta}x_k\right) \\ {}^{z}x_k - {}^{u}x_k \sin\left({}^{\theta}x_k\right) \\ {}^{u}x_k \\ {}^{\theta}x_k \\ {}^{\psi}x_k \end{bmatrix}. \tag{3.9}$$

Although the dynamic model is nonlinear, we can linearize the vehicle model in the neighborhood of a given operating point using its Jacobian. The Jacobian of this

vehicle dynamic function with respect to the vehicle state is given by

$$
{}^{\nu}F = \frac{\partial}{\partial x}\,{}^{\nu}f\left(x\right) = \begin{bmatrix}
1 & 0 & 0 & \cos\left({}^{\psi}x\right)\cos\left({}^{\theta}x\right) & \\
0 & 1 & 0 & \sin\left({}^{\psi}x\right)\cos\left({}^{\theta}x\right) & \\
0 & 0 & 1 & -\sin\left({}^{\theta}x\right) & \\
0 & 0 & 0 & 1 & \\
0 & 0 & 0 & 0 & \\
0 & 0 & 0 & 0 &
\end{bmatrix}
\begin{bmatrix}
-{}^{u}x\cos\left({}^{\psi}x\right)\sin\left({}^{\theta}x\right) & -{}^{u}x\sin\left({}^{\psi}x\right)\cos\left({}^{\theta}x\right) \\
-{}^{u}x\sin\left({}^{\psi}x\right)\sin\left({}^{\theta}x\right) & {}^{u}x\cos\left({}^{\psi}x\right)\cos\left({}^{\theta}x\right) \\
0 & -{}^{u}x\left(\cos{}^{\theta}x\right) \\
0 & 0 \\
1 & 0 \\
0 & 1
\end{bmatrix}, \tag{3.10}
$$

where the Jacobian is understood to be evaluated at a specific operating point x. As with the point model, the vehicle model is assumed to be independent of time:

$$
{}^{\nu}\phi_k = {}^{\nu}\phi \quad \forall \; k. \tag{3.11}
$$

The vehicle model process noise is assumed to be a sampled stationary zero-mean Gaussian white noise process with covariance

$$
\mathrm{E}\left[w_k\, w_k^T\right] = {}^{\nu}Q = \begin{bmatrix}
{}^{n}w & 0 & 0 & 0 & 0 & 0 \\
0 & {}^{e}w & 0 & 0 & 0 & 0 \\
0 & 0 & {}^{z}w & 0 & 0 & 0 \\
0 & 0 & 0 & {}^{u}w & 0 & 0 \\
0 & 0 & 0 & 0 & {}^{\theta}w & 0 \\
0 & 0 & 0 & 0 & 0 & {}^{\psi}w
\end{bmatrix}. \tag{3.12}
$$

Table 3.2: Vehicle dynamic model ${}^{v}\phi$ parameters.

Parameter	Symbol	Value
north process noise variance	${}^{n}w$	0.5 m^2
east process noise variance	${}^{e}w$	0.5 m^2
depth process noise variance	${}^{z}w$	0.0025 m^2
speed process noise variance	${}^{u}w$	0.01 $\frac{m^2}{s^2}$
pitch process noise variance	${}^{\theta}w$	0.0008 rad^2
yaw process noise variance	${}^{\psi}w$	0.0008 rad^2

The process noise is assumed independent of the initial vehicle state and other noise processes. Table 3.2 shows the values used for the process noise in the vehicle model for all of the runs in this thesis.

3.2.3 Measurement modeling

In addition to the dynamic models used to reason about vehicle and feature projection and behavior, measurement models must also be specified. Measurement models capture the physics and the logical possibilities of sensing processes.

> A **measurement model** captures the prior knowledge about the physics and logical possibilities for a specific sensing process. **Measurement models** for relative measurements must also consider the relationships among the dynamic models of the objects involved in the measurement.

Measurements are, of course, subject to uncertainty. In the usual continuous sense, measurements are parametrically uncertain. When measurements involve multiple models, however, there may additionally be discrete uncertainty about measurement origin. The specification of a measurement model thus involves not only the model of the underlying sensing physics, but a broader understanding of the measurement channel so that deliberation about the possible navigational events associated with a measurement can be enumerated without conditioning on the specific features or

models involved.

A measurement **channel** is the environmental substrate in which measurements are taken. In the case of acoustic sensors, for example, the **channel** is represented by the acoustical properties of the water column which serves as a medium of propagation.

In this thesis, we consider two measurement modalities. The first involves only the vehicle model, and consists of dead reckoning measurements. These dead reckoning measurements do not contain discrete uncertainty and can be processed as in the usual Kalman framework. Additionally, we consider sonar measurements using a high resolution forward-looking sonar [67]. These measurements may involve the vehicle and the feature models and are subject to data association uncertainty.

Dead reckoning measurements

The vehicle measures its depth, speed, pitch, and heading during each time cycle. The dead reckoning measurement model is denoted ${}^{\mathcal{W}_{dr}}\mu$. The measurements are based on the vehicle state:

$$ {}^{\mathcal{W}_{dr}}\mu_k : \quad z_k = {}^{\mathcal{W}_{dr}}h_k\left(x_k\right) + v_k, \tag{3.13} $$

where ${}^{\mathcal{W}_{dr}}\mu_k$ is the measurement model at time k, z_k is the dead reckoning measurement vector, x_k is the vehicle state, and v_k is additive measurement noise. The dead reckoning measurements involve direct measurements of the vehicle's depth, speed, pitch, and yaw:

$$ z_k = \left[{}^{z}z_k \quad {}^{u}z_k \quad {}^{\theta}z_k \quad {}^{\psi}z_k \right]^T. \tag{3.14} $$

Table 3.3: Dead reckoning measurement model ${}^{\mathcal{W}_{dr}}\mu$ parameters.

Parameter	Symbol	Value
depth measurement noise variance	${}^{z}v$	25 cm^2
speed measurement noise variance	${}^{u}v$	0.25 $\frac{\text{m}^2}{\text{s}^2}$
pitch measurement noise variance	${}^{\theta}v$	0.5 deg^2
yaw measurement noise variance	${}^{\psi}v$	0.5 deg^2

The measurement function is given by

$$ {}^{\mathcal{W}_{dr}}h_k\left(x_k\right) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{n}x_k \\ {}^{e}x_k \\ {}^{z}x_k \\ {}^{u}x_k \\ {}^{\theta}x_k \\ {}^{\psi}x_k \end{bmatrix}. \tag{3.15} $$

The measurement noise v_k is assumed to be a sampled zero-mean Gaussian white-noise process with covariance

$$ \mathrm{E}\left[v_k\, v_k^T\right] = {}^{\mathcal{W}_{dr}}R_k = \begin{bmatrix} {}^{z}v & 0 & 0 & 0 \\ 0 & {}^{u}v & 0 & 0 \\ 0 & 0 & {}^{\theta}v & 0 \\ 0 & 0 & 0 & {}^{\psi}v \end{bmatrix}. \tag{3.16} $$

Also, the dead reckoning measurement model is assumed to be independent of time, allowing the dependence of ${}^{\mathcal{W}_{dr}}\mu$, ${}^{\mathcal{W}_{dr}}h$, and ${}^{\mathcal{W}_{dr}}R$ on k to be dropped. Throughout this thesis, we will assume a measurement noise covariance with parameters as listed in Table 3.3.

Sonar measurements

A second measurement model is used to describe relative measurements taken with a forward-looking sonar. The sonar data is processed to extract measurements which might correspond to features. The origin of these measurements is uncertain, however. A measurement might have come from any of the currently modeled features or from a new (i.e. previously unknown) feature, or it may be the result of an unmodeled phenomenon. Measurements arising from unmodeled phenomena are termed clutter and may arise from unuseable (or unmodelable) features, sonar channel reverberation or multipath, or noise in the sensor itself. The sonar measurement model is denoted ${}^{w_s}\mu$. Given that a measurement arises at time k from a relative measurement of the vehicle with state x_k and a feature i with state ξ_k^i, the sonar measurement provides a noisy measurement of the range ${}^{r}z$ and bearing ${}^{\varphi}z$:

$$ {}^{w_s}\mu_k : \quad z_k = {}^{w_s}h_k\left(x_k, \xi_k^i\right) + v_k = \begin{bmatrix} {}^{r}z_k \\ {}^{\varphi}z_k \end{bmatrix}. \tag{3.17} $$

If we regard the vehicle state as consisting of the north and east coordinates and heading,

$$ x_k = \begin{bmatrix} {}^{n}x_k & {}^{e}x_k & {}^{\psi}x_k \end{bmatrix}^T, \tag{3.18} $$

and the feature state as consisting of the north and east coordinates of the feature,

$$ \xi_k^i = \begin{bmatrix} {}^{n}\xi_k^i & {}^{e}\xi_k^i \end{bmatrix}^T, \tag{3.19} $$

then the measurement function can be written as

$$ {}^{w_s}h_k\left(x_k, \xi_k^i\right) = \begin{bmatrix} \sqrt{\left({}^{n}\xi_k^i - {}^{n}x_k\right)^2 - \left({}^{e}\xi_k^i - {}^{e}x_k\right)^2} \\ \arctan\left(\frac{{}^{e}\xi_k^i - {}^{e}x_k}{{}^{n}\xi_k^i - {}^{n}x_k}\right) - {}^{\psi}x_k \end{bmatrix}. \tag{3.20} $$

Table 3.4: Sonar measurement model ${}^{\mathcal{W}_s}\mu$ parameters.

Parameter	Symbol	Value
range measurement noise covariance	${}^{r}v$	0.5 m^2
bearing measurement noise covariance	${}^{\varphi}v$	5 deg^2

We adopt the convention that if ${}^{n}\xi_k^i = {}^{n}x_k$, the expected (that is, disregarding measurement error) bearing is zero. Since we are using separate vehicle and feature models, the measurement function can be linearized based (separately) on vehicle and feature state. The Jacobian of this measurement function with respect to the vehicle state is

$$ {}^{\mathcal{W}_s}_{x}H_k = \begin{bmatrix} -\frac{{}^{n}\xi_k^i - {}^{n}x_k}{\sqrt{\left({}^{n}\xi_k^i - {}^{n}x_k\right)^2 - \left({}^{e}\xi_k^i - {}^{e}x_k\right)^2}} & -\frac{{}^{e}\xi_k^i - {}^{e}x_k}{\sqrt{\left({}^{n}\xi_k^i - {}^{n}x_k\right)^2 - \left({}^{e}\xi_k^i - {}^{e}x_k\right)^2}} & 0 \\ \frac{{}^{e}\xi_k^i - {}^{e}x_k}{\left({}^{n}\xi_k^i - {}^{n}x_k\right)^2 - \left({}^{e}\xi_k^i - {}^{e}x_k\right)^2} & -\frac{{}^{n}\xi_k^i - {}^{n}x_k}{\left({}^{n}\xi_k^i - {}^{n}x_k\right)^2 - \left({}^{e}\xi_k^i - {}^{e}x_k\right)^2} & -1 \end{bmatrix} \tag{3.21} $$

and the Jacobian with respect to the feature state is

$$ {}^{\mathcal{W}_s}_{\xi^i}H_k = \begin{bmatrix} \frac{{}^{n}\xi_k^i - {}^{n}x_k}{\sqrt{\left({}^{n}\xi_k^i - {}^{n}x_k\right)^2 - \left({}^{e}\xi_k^i - {}^{e}x_k\right)^2}} & \frac{{}^{e}\xi_k^i - {}^{e}x_k}{\sqrt{\left({}^{n}\xi_k^i - {}^{n}x_k\right)^2 - \left({}^{e}\xi_k^i - {}^{e}x_k\right)^2}} \\ -\frac{{}^{e}\xi_k^i - {}^{e}x_k}{\left({}^{n}\xi_k^i - {}^{n}x_k\right)^2 - \left({}^{e}\xi_k^i - {}^{e}x_k\right)^2} & \frac{{}^{n}\xi_k^i - {}^{n}x_k}{\left({}^{n}\xi_k^i - {}^{n}x_k\right)^2 - \left({}^{e}\xi_k^i - {}^{e}x_k\right)^2} \end{bmatrix}. \tag{3.22} $$

The measurement noise is assumed to be a sampled zero-mean Gaussian white noise process with covariance

$$ \mathrm{E}\left[v_k\, v_k^T\right] = {}^{\mathcal{W}_s}R_k = \begin{bmatrix} {}^{r}v & 0 \\ 0 & {}^{\varphi}v \end{bmatrix}. \tag{3.23} $$

This noise process is assumed to be independent of the initial vehicle and feature states and other noise processes. Table 3.4 shows the components of the sonar measurement model measurement noise covariance matrix.

3.3 Issues in discrete estimation

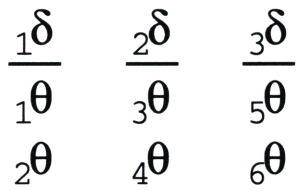
3.3.1 Decisions

In the continuous estimation problem, there are no compatibility problems introduced by specifying the PDF of an estimate. Consequent probabilistic manipulations can be chained together with little problem, e.g. in the recursive formulation of the Kalman filter. In the case of discrete estimation, however, one is dealing with exclusive events rather than parametric uncertainty. Because of this, estimates may depend on the outcome of previous events in incompatible ways. For example, assume that at the previous time cycle the vehicle either received a spurious measurement or measured a new feature. The possible estimates for the current time cycle are conditioned on the outcome of this decision. Estimates which assume that the measurement was spurious should not consider updates from the proposed new feature; they depend on the previous decision in an incompatible way. Because of this, each state must keep track of its decision dependencies. Only compatible states may be considered together. Incompatible states are part of different "what if..." scenarios and cannot occur together. This concept is illustrated in Figure 3-3.

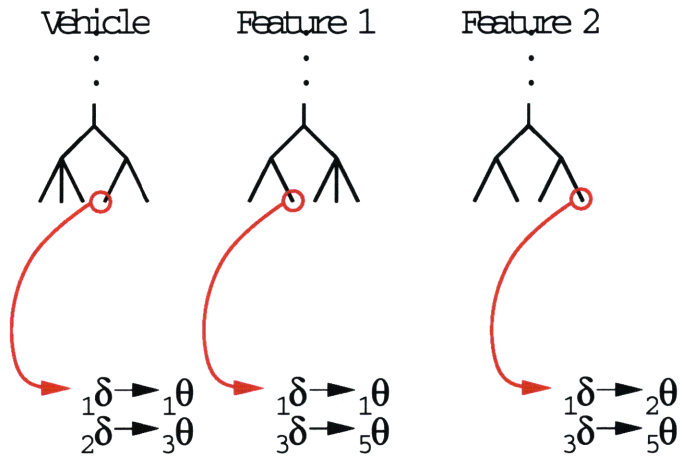
Recall that a navigational event is an event that contains some discrete uncertainty. The possibilities regarding each navigational event are represented as a decision.

A **decision** is an enumeration of the modeling alternatives being considered in the representation of a navigational event.

At each time step k, there will, in general, be a number of decisions ${}^{o}\delta_k$, where the index o ranges over this set of decisions. We use a set notation to indicate the set of



(a) Consider three decisions, $_1\delta$, $_2\delta$, and $_3\delta$. Let each have two possible hypotheses as shown above.



(b) The compatibility of states can be determined by considering their dependency sets. Here we consider a state from each tree (the vehicle, feature one, and feature two). The selected states from the vehicle and feature one are compatible because all decisions that they mutually depend on ($_1\delta$) are required to resolve to the same hypothesis ($_1\theta$). The selected states from the vehicle and feature two, however, are incompatible because the require different resolutions for decision $_1\delta$.

Figure 3-3: Decision dependencies and compatibility. Two states are compatible if and only if they resolve all decisions upon which they mutually depend with the same hypothesis.

decisions Δ_k that contains these decisions, indexed by o:

$$\Delta_k = \{{}_o\delta_k\}_o \,. \tag{3.24}$$

There are three types of decisions which we consider: feature track disposition decisions ${}^{\mathcal{D}}\delta$, measurement origin decisions ${}^{\mathcal{Z}}\delta$, and feature or vehicle ontological decisions ${}^{\mathcal{O}}\delta$.

Dispositional decisions are decisions about what has happened to a particular track.

Origin decisions are decisions about the origin of a particular measurement.

Ontological decisions are decisions about the dynamic behavior or model representation of a particular track.

As noted above, we focus on dispositional and origin decisions in this thesis.

Each decision has a set of hypotheses representing the possible models for that navigational event,

$$\Theta_{{}_o\delta_k} = \{{}_u\theta^o_k\}_u \,, \tag{3.25}$$

where u indexes over the hypotheses for decision o.

A **hypothesis** is a possible resolution for a decision. A **hypothesis** is a proposed model for the navigational event represented by the decision.

A hypothesis may resolve more than one decision. We denote the set of *all* hypotheses, that is, all hypotheses from all decision, at time k as

$$\Theta_k = \{{}_p\theta_k\}_p, \tag{3.26}$$

where p indexes over all of the hypotheses for time k. We also define a decision indexing function U which recovers the set of hypotheses for a particular decision from the set of all hypotheses for a given time:

$$\Theta_{{}_o\delta_k} = \{{}_u\theta_k^o\} = U\left({}_o\delta_k, \Theta_k\right) = U\left({}_o\delta_k, \{{}_p\theta_k\}_p\right). \tag{3.27}$$

3.3.2 Decision dependencies

Estimates that are enumerated based on possible hypotheses for a given decision depend on those hypotheses. When a choice among competing hypotheses is made, we refer to the decision being resolved to a particular hypothesis.

A **dependency** is the assertion that a decision must resolve to a particular hypothesis.

Thus, estimates have dependencies for the resolution of decisions. We denote a dependency by

$$d = {}_o\delta_k \mapsto {}_u\theta_k^o, \tag{3.28}$$

meaning that the dependency d requires that decision ${}_o\delta_k$ resolve to hypothesis ${}_u\theta_k^o$. We additionally define δ_d as the decision referred to by dependency d and θ_d as the hypothesis required by d.

A given state x has a set of dependencies

$$D_x = \{_e d\}_e \,, \tag{3.29}$$

where e indexes over all of the dependencies in the set, i.e. all of the dependencies for the state x. Note that, since hypotheses for a given decision are mutually exclusive, a state can have at most one dependency for a given decision. Two dependencies are incompatible if they require different resolutions for the same decision. We define a dependency compatibility function C_d, which is zero if the two dependencies are compatible and one if they are not:

$$C_d\left({}_{e_1}d, {}_{e_2}d\right) = \begin{cases} 1 & \text{if } \delta_{{}_{e_1}d} = \delta_{{}_{e_2}d} \text{ and } \theta_{{}_{e_1}d} \neq \theta_{{}_{e_2}d}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.30}$$

We can quantify an ontological distance c, that is, a distance in discrete estimation space, for a pair of dependency sets as the number of incompatible dependencies that they contain:

$$c\left(D_1, D_2\right) = \sum_{e_1} \sum_{e_2} C_d\left({}_{e_1}d, {}_{e_2}d\right), \tag{3.31}$$

where e_1 indexes over all of the dependencies in dependency set D_1 and e_2 indexes over all of the dependecies in dependency set D_2. The dependency set compatibility function C_D is true for a pair of dependency sets if and only if the ontological distance between them is zero:

$$C_D\left(D_1, D_2\right) = \begin{cases} \text{true} & c\left(D_1, D_2\right) = 0, \\ \text{false} & \text{otherwise.} \end{cases} \tag{3.32}$$

Note that compatible dependency sets require the same resolution for all decisions that they mutually depend on.

$$C_D(D_1, D_2) = \text{true} \quad \text{iff} \quad \theta_{e_1 d} = \theta_{e_2 d} \quad \forall \quad e_1, e_2 \mid \delta_{e_1 d} = \delta_{e_2 d}\,, \tag{3.33}$$

where e_1 indexes the dependency set D_1 and e_2 indexes the dependency set D_2.

3.4 Issues in continuous estimation

3.4.1 A Bayesian interpretation of the Kalman filter

In this section, we consider the general problem of projecting and updating when the vehicle and feature models are separated. This is done within a Bayesian framework; however, the results are related to the extended Kalman filter equations, which are typically used to implement CML. The purpose of this rederivation is three-fold: to examine the effects of considering the models separately, to examine the role of cross-model correlations, and to provide deeper understanding of the Bayesian approach to the estimation procedure, which will be of use as we consider the implementation of the hybrid estimation problem with IMAN. There are typically two steps in recursive estimation problems. First, the existing estimate is projected to the current time. Second, this projected estimate is updated with any measurements that have been taken.

Bayesian state projection

First we consider the projection of an estimate to predict the state of a model at some future time. We assume that we have an existing estimate $x_{k-1|k-1}$, where the estimate is of the state at time $k-1$ and incorporates all measurements through time

$k-1$.[2] This estimate is a description of the probability density function (PDF) of the actual state, if one exists, $p\left(x_{k-1} \middle| Z^{k-1}\right)$, where we adopt the convention that Z^{k-1} refers to the set of all measurements through time $k-1$.[3] The estimate provides both a state estimate

$$\hat{x}_{k-1|k-1} = \mathrm{E}\left[x_{k-1} \middle| Z^{k-1}\right] \tag{3.34}$$

and a covariance matrix

$$P_{k-1|k-1} = \mathrm{E}\left[\left(x_{k-1} - \mathrm{E}\left[x_{k-1}\right]\right)\left(x_{k-1} - \mathrm{E}\left[x_{k-1}\right]\right)^T \middle| Z^{k-1}\right]. \tag{3.35}$$

The dynamic model ϕ is assumed to contain deterministic dynamics f corrupted by white noise w,

$$\phi = \phi\left(f, w\right). \tag{3.36}$$

This model provides a stochastic description of the evolution of the PDF of the state. We consider the noise to be additive, providing the following discrete (in time) dynamic model:

$$x_k = f\left(x_{k-1}\right) + w, \tag{3.37}$$

where k is a time index. We also make the usual assumptions about the process noise [39]; that is, it is a sampled zero-mean white Gaussian process, and it is independent of the initial state estimate and other noise processes. The noise process is

[2]Throughout this section, we refer to a general state as x. Note that the projection of feature states ξ is identical.

[3]The set of all measurements taken *at* time $k-1$ would be indicated by Z_{k-1}.

assumed to have a covariance

$$\mathrm{E}\left[ww^T\right] = Q. \tag{3.38}$$

The goal of projection is to characterize the projected PDF $p\left(x_k|\, Z^{k-1}, \phi\right)$. This is not straightforward if the dynamic function f is nonlinear. The extended Kalman filter [39, 4, 99] addresses this problem by linearizing the state dynamics around the estimate

$$f\left(x_{k-1}\right) \approx f\left(\hat{x}_{k-1|k-1}\right) + F_{\hat{x}_{k-1|k-1}}\left(x_{k-1} - \hat{x}_{k-1}\right), \quad \text{where} \tag{3.39}$$

$$F_{\hat{x}_{k-1|k-1}} = \left.\frac{\partial f}{\partial x}\right|_{x=\hat{x}_{k-1|k-1}} \tag{3.40}$$

is the Jacobian of the dynamic function. This simplification has two effects. First, nonlinearities in the dynamic function are ignored. Second, because of the assumed linearity, Gaussianity of the estimate is assumed to be preserved, even if this is not the case. A number of methods exist to improve this estimation process for nonlinear systems [99, 84, 27, 39, 28, 91]; such procedures may improve the robustness and quality of the estimate, but do not fundamentally alter the development of the hybrid estimation process found here.

Using these assumptions, the projected density $p\left(x_k|\, Z^{k-1}, \phi\right)$ is Gaussian and can be characterized by its mean and covariance. The mean can be determined without linearization to provide a projected estimate

$$\hat{x}_{k|k-1} = \mathrm{E}\left[x_k|\, Z^{k-1}, \phi\right] = f\left(\hat{x}_{k-1|k-1}\right). \tag{3.41}$$

The covariance of the error of this estimate, using the linearized dynamic function, is

then

$$P_{k|k-1} = \mathrm{E}\left[\left(x_k - \hat{x}_{k|k-1}\right)\left(x_k - \hat{x}_{k|k-1}\right)^T\right] = F_{\hat{x}_{k-1|k-1}} P_{k-1|k-1} F^T_{\hat{x}_{k-1|k-1}} + Q. \quad (3.42)$$

This is also the covariance of the projected density;

$$P_{k|k-1} = \mathrm{E}\left[\left(x_k - \mathrm{E}\left[x_k\right]\right)\left(x_k - \mathrm{E}\left[x_k\right]\right)^T \middle| Z^{k-1}\right]. \quad (3.43)$$

Bayesian measurement updating

Now we consider the update step, the combination of the projected estimate with a measurement. The projected state $x_{k|k-1}$ contains the mean

$$\hat{x}_{k|k-1} = \mathrm{E}\left[x_k \middle| Z^{k-1}\right] \quad (3.44)$$

and covariance

$$P_{k|k-1} = \mathrm{E}\left[\left(x_k - \mathrm{E}\left[x_k\right]\right)\left(x_k - \mathrm{E}\left[x_k\right]\right)^T \middle| Z^{k-1}\right] \quad (3.45)$$

of the state vector PDF $p\left(x_k \middle| Z^{k-1}\right)$ at time k based on all information through time $k-1$. We also have a set of measurements Z_k from the current time k. We can consider these measurements sequentially or grouped by measurement type with no change to the final updated state estimate [98, 50]. We will refere to updating with a particular measurement (or block of measurements) ${}_{m_1}z_k$ out of the set of all measurements for time k,

$$Z_k = \left\{{}_m z_k\right\}_m. \quad (3.46)$$

We denote the set of measurements already used for updating

$${}_{m_1-1}Z_k = \left\{{}_m z_k\right\}_{m<m_1}. \quad (3.47)$$

First, we consider the single model case, in which only the vehicle participates in the measurement. This occurs, for example, when incorporating dead reckoning measurements. A measurement model μ provides an expected measurement function h and a description of the measurement noise v:

$$\mu = \mu(h, v). \tag{3.48}$$

We assume that the measurement noise is additive, providing the following measurement equation:

$$_{m_1}z_k = h(x_k) + v_k. \tag{3.49}$$

The measurement noise is assumed to be a sampled zero-mean white Gaussian process that is independent of the state variables and other noise processes and has a covariance

$$\mathrm{E}\left[v_k\, v_k^T\right] = R. \tag{3.50}$$

Our goal is to characterize the *a posteriori* density

$$p\left(x_k|_{m_1}Z_k, Z^{k-1}\right) = p\left(x_k|_{m_1}z_k, {}_{m_1-1}Z_k, Z^{k-1}\right), \tag{3.51}$$

that is, to estimate the state x_k based on all of the measurements taken at earlier times Z^{k-1}, all of the measurements taken at this time which have already been processed $_{m_1-1}Z_k$, and the current measurement being processed $_{m_1}z_k$. The current estimate $\hat{x}_{k|k-1,m_1-1}$ is thus a partially updated PDF $p\left(x_k|\,_{m_1-1}Z_k, Z^{k-1}\right)$. Using Bayes rule, we can write the posterior density as

$$p\left(x_k|\,_{m_1}z_k, {}_{m_1-1}Z_k, Z^{k-1}\right) = \frac{p\left(_{m_1}z_k|\,x_k, {}_{m_1-1}Z_k, Z^{k-1}\right) p\left(x_k|\,_{m_1-1}Z_k, Z^{k-1}\right)}{p\left(_{m_1}z_k|\,_{m_1-1}Z_k, Z^{k-1}\right)}. \tag{3.52}$$

Our assumptions about the measurement noise allow this to be reduced to

$$p\left(x_k|\,{}_{m_1}Z_k, Z^{k-1}\right) = \frac{p\left({}_{m_1}z_k|\,x_k, \mu\right) p\left(x_k|\,{}_{m_1-1}Z_k, Z^{k-1}\right)}{p\left({}_{m_1}z_k|\,\mu\right)}. \tag{3.53}$$

The conditional measurement density $p\left({}_{m_1}z_k|\,x_k, \mu\right)$ is given by the measurement model; it is Gaussian with mean $h\left(x_k\right)$ and covariance R.[4] The total measurement density $p\left({}_{m_1}z_k|\,\mu\right)$ is evaluated by linearizing the measurement function h around the estimated state;

$$h\left(x_k\right) \approx h\left(\hat{x}_{k|k-1,m_1-1}\right) + H_{\hat{x}_{k|k-1,m_1-1}}\left(x_k - \hat{x}_{k|k-1,m_1-1}\right), \tag{3.54}$$

where

$$H_{\hat{x}_{k|k-1,m_1-1}} = \left.\frac{\partial h}{\partial x}\right|_{x=\hat{x}_{k|k-1,m_1-1}} \tag{3.55}$$

is the Jacobian of the measurement function with respect to the state vector. The measurement PDF is thus approximated by a Gaussian with a mean $h\left(\hat{x}_{k|k-1,m_1-1}\right)$ and a covariance $H_{\hat{x}_{k|k-1,m_1-1}} P_{k|k-1,m_1-1} H^T_{\hat{x}_{k|k-1,m_1-1}}$. Substituting these values into Equation 3.53 provides

$$\begin{aligned} p\left(x_k|\,{}_{m_1}Z_k, Z^{k-1}\right) = {} & \frac{\left|H_{\hat{x}} P_{k|k-1} H_{\hat{x}} + R\right|^{1/2}}{(2\pi)^{n/2}\left|P_{k|k-1}\right|^{1/2}|R|^{1/2}} \\ & \cdot \exp\left\{-\frac{1}{2}\left[\left(x_k - \hat{x}\right)^T P^{-1}_{k|k-1}\left(x_k - \hat{x}\right) + \left({}_{m_1}z_k - h\left(x_k\right)\right)^T R^{-1}\left({}_{m_1}z_k - h\left(x_k\right)\right)\right.\right. \\ & \left.\left. - \left({}_{m_1}z_k - h\left(\hat{x}\right)\right)^T \left(H_{\hat{x}} P_{k|k-1} H^T_{\hat{x}} + R\right)^{-1}\left({}_{m_1}z_k - h\left(\hat{x}\right)\right)\right]\right\} \end{aligned} \tag{3.56}$$

where n is the order of the state vector and we have used $\hat{x}$ to mean $\hat{x}_{k|k-1,m_1-1}$ and $P_{k|k-1}$ to mean $P_{k|k-1,m_1-1}$ for clarity. Completing the squares, Equation 3.56 is seen

[4] This is often written as $p_v\left({}_{m_1}z_k - h\left(x_k\right)\right)$, meaning that the quantity ${}_{m_1}z_k - h\left(x_k\right)$ is distributed with the PDF of the measurement noise v.

to be a Gaussian with mean

$$\hat{x}_{k|k-1,m_1} = \mathrm{E}\left[x_k \middle|\, {}_{m_1}Z_k, Z^{k-1}\right] = \hat{x}_{k|k-1,m_1-1} + P_{k|k-1,m_1-1}H^T_{\hat{x}_{k|k-1,m_1-1}} \left(H_{\hat{x}_{k|k-1,m_1-1}} P_{k|k-1,m_1-1} H^T_{\hat{x}_{k|k-1,m_1-1}} + R\right)^{-1} \cdot \left({}_{m_1}z_k - h\left(\hat{x}_{k|k-1,m_1-1}\right)\right) \quad (3.57)$$

and covariance

$$P_{k|k-1,m_1} = P_{k|k-1,m_1-1} - P_{k|k-1,m_1-1}H^T_{\hat{x}_{k|k-1,m_1-1}} \cdot \left(H_{\hat{x}_{k|k-1,m_1-1}} P_{k|k-1,m_1-1} H^T_{\hat{x}_{k|k-1,m_1-1}} + R\right)^{-1} H_{\hat{x}_{k|k-1,m_1-1}} P_{k|k-1,m_1-1}, \quad (3.58)$$

which is of course the Kalman update equation.

For relative measurements, both vehicle state x_k and feature state ξ_k are involved in the measurement ${}_{m_1}z_k$ according to a measurement equation

$${}_{m_1}z_k = h\left(x_k, \xi_k\right) + v_k. \quad (3.59)$$

Such relative updates proceed in a similar manner, with the exception that, due to the separation of the vehicle and feature models, the measurement function is now linearized as

$$h\left(x_k, \xi_k\right) \approx h\left(\hat{x}_{k|k-1,m_1-1}, \hat{\xi}_{k|k-1,m_1-1}\right) + H_{\hat{x}_{k|k-1,m_1-1}}\left(x_k - \hat{x}_{k|k-1,m_1-1}\right) + H_{\hat{\xi}_{k|k-1,m_1-1}}\left(\xi_k - \hat{\xi}_{k|k-1,m_1-1}\right), \quad (3.60)$$

where

$$H_{\hat{x}_{k|k-1,m_1-1}} = \left.\frac{\partial h}{\partial x}\right|_{x=\hat{x}_{k|k-1,m_1-1}} \quad (3.61)$$

as before and

$$H_{\hat{\xi}_{k|k-1,m_1-1}} = \left.\frac{\partial h}{\partial \xi}\right|_{\xi=\hat{\xi}_{k|k-1,m_1-1}} \tag{3.62}$$

is the Jacobian of h with respect to the feature state estimate $\hat{\xi}_{k|k-1,m_1-1}$. The innovations,

$$\nu = {}_{m_1}z_k - h\left(\hat{x}_{k|k-1,m_1-1}, \hat{\xi}_{k|k-1,m_1-1}\right), \tag{3.63}$$

are distributed identically to $p\left({}_{m_1}z_k \middle| x_k, \xi_k, \mu\right)$, but with zero mean. The covariance of the innovations for the case of a relative measurement is

$$\begin{aligned} S\left(\hat{x}_{k|k-1,m_1-1},\ \hat{\xi}_{k|k-1,m_1-1},\ {}_{m_1}z_k\right) = \mathrm{E}\left[\nu\nu^T\right] = \\ H_{\hat{x}_{k|k-1,m_1-1}}\ {}_xP_{k|k-1,m_1-1}\ H^T_{\hat{x}_{k|k-1,m_1-1}} \\ + H_{\hat{\xi}_{k|k-1,m_1-1}}\ {}_\xi P_{k|k-1,m_1-1}\ H_{\hat{\xi}_{k|k-1,m_1-1}} + R, \end{aligned} \tag{3.64}$$

where ${}_xP_{k|k-1,m_1-1}$ is the vehicle state covariance and ${}_\xi P_{k|k-1,m_1-1}$ is the feature state covariance.

The updated vehicle state is then approximated by a Gaussian with mean

$$\begin{aligned} \hat{x}_{k|k-1,m_1} = \hat{x}_{k|k-1,m_1-1} + {}_xP_{k|k-1,m_1-1}\ H_{\hat{x}_{k|k-1,m_1-1}} \\ \cdot S\left(\hat{x}_{k|k-1,m_1-1},\ \hat{\xi}_{k|k-1,m_1-1},\ {}_{m_1}z_k\right)^{-1} \\ \cdot \left({}_{m_1}z_k - h\left(\hat{x}_{k|k-1,m_1-1}, \hat{\xi}_{k|k-1,m_1-1}\right)\right) \end{aligned} \tag{3.65}$$

and covariance

$$
\begin{aligned}
{}_{x}P_{k|k-1,m_1} = {}_{x}P_{k|k-1,m_1-1} - {}_{x}P_{k|k-1,m_1-1}\, H_{\hat{x}_{k|k-1,m_1-1}} \\
\cdot S\left(\hat{x}_{k|k-1,m_1-1},\ \hat{\xi}_{k|k-1,m_1-1},\ {}_{m_1}z_k\right)^{-1} \\
\cdot H_{\hat{x}_{k|k-1,m_1-1}}\ {}_{x}P_{k|k-1,m_1-1}. \quad (3.66)
\end{aligned}
$$

Similarly, the updated feature state is approximated by a Gaussian with mean

$$
\begin{aligned}
\hat{\xi}_{k|k-1,m_1} = \hat{\xi}_{k|k-1,m_1-1} + {}_{\xi}P_{k|k-1,m_1-1}\, H_{\hat{\xi}_{k|k-1,m_1-1}} \\
\cdot S\left(\hat{x}_{k|k-1,m_1-1},\ \hat{\xi}_{k|k-1,m_1-1},\ {}_{m_1}z_k\right)^{-1} \\
\cdot \left({}_{m_1}z_k - h\left(\hat{x}_{k|k-1,m_1-1}, \hat{\xi}_{k|k-1,m_1-1}\right)\right) \quad (3.67)
\end{aligned}
$$

and covariance

$$
\begin{aligned}
{}_{\xi}P_{k|k-1,m_1} = {}_{\xi}P_{k|k-1,m_1-1} - {}_{\xi}P_{k|k-1,m_1-1}\, H_{\hat{\xi}_{k|k-1,m_1-1}} \\
\cdot S\left(\hat{x}_{k|k-1,m_1-1},\ \hat{\xi}_{k|k-1,m_1-1},\ {}_{m_1}z_k\right)^{-1} \\
\cdot H_{\hat{\xi}_{k|k-1,m_1-1}}\ {}_{\xi}P_{k|k-1,m_1-1}. \quad (3.68)
\end{aligned}
$$

3.4.2 The role of cross-model correlations

Cross-model correlations have been shown to be quite important in preventing divergence in CML when global position information is needed [16]. However, the entire story is not told by the global position information. By ignoring cross-model correlations, the global map more easily slips relative to the vehicle and feature estimates.

Map slip is the slip of relative position information, that is, the geometric relationships among the vehicle and features being modeled, with respect to the global frame. This **map slip** results in increased global positioning error with no corresponding increase in the estimate of the relative positions of the vehicle and features.

However, the relative position information, that is, the geometric relationships among features and the vehicle, may be accurately modeled without the inclusion of the cross-model correlations.[5] This is because the over-estimation of the information about global position which the cross-model correlations discount is in fact valid information about the relative position of the objects in question. The primary reason for ignoring cross-model correlations for the present is complexity, both representational and computational. This has been an issue in multiple hypothesis tracking, and the separation of models in integrated mapping and navigation only exacerbates the problem of representing correlations. There is ongoing research [104] on methods for conservatively estimating global estimates without maintaining cross-model covariance information. Techniques such as covariance intersection have the potential to correct for ignored cross-model correlations with minimal computational overhead.

In spite of this, we feel that there are significant lessons to be learned from a more in depth examination of the map representation problem and the relationship between global and relative position estimation. This is particularly the case when the vehicle position is very uncertain. In such situations, a good case can be made for relative mapping as the appropriate representation strategy to improve navigation. A full consideration of the map representation problem is beyond the scope of this thesis, but will likely provide an interesting research subject for some time. In this thesis, we are more directly concerned with the representation of multiple hypotheses within a framework of uncertain navigation and mapping. Cross-model correlations

[5]Note that the inclusion of cross-model correlations does not, by itself, prevent map slip. Ignoring these correlations, however, does allow map slip to occur more easily and can lead to increased global estimation errors.

are ignored in order to augment the computability of the integrated mapping and navigation algorithm. We note, however, that a number of alternatives exist for recovering this information either in post-processing, as a post-decision-resolution step, or integrally within the algorithm. The post-processing recovery of global position estimates including cross-model correlation effects can be accomplished by using a stochastic mapping algorithm, which ignores data association uncertainty, based on the data association decisions made by the IMAN algorithm. This technique remains sub-optimal because it neglects the possibility that the incorporation of cross-model correlation effects could affect the decision-making process. A similar, but more timely, implementation would recalculate estiamtes to include cross-model correlation information after decisions have been resolved. Although corrected estimates would not be available for the current time, cross-model correlations would be included until the most recent unresolved decision. The impact of cross-model correlations on decisions could be regained to some extent by resetting the unresolved portion of the track trees when some error threshold is exceeded in comparing the most recent estimates with and without the inclusion of cross-model correlations. Finally, covariance intersection [104] may provide the capability of recovering conservative, if sub-optimal, estimates of the vehicle and feature positions without retaining the cross-model correlation information or re-processing the estimates.

3.5 The process of IMAN

Traditional recursive estimation techniques are split into two processes: projection and updating. In integrated mapping and navigation, the updating step is expanded to incorporate the discrete part of the hybrid estimation process, as shown in Figure 3-4. This discrete estimation has two distinct phases. The first, which occurs before the continuous updating step, is the enumeration of the discrete possibilities of how estimated states may be updated based on the set of measurements taken. After con-

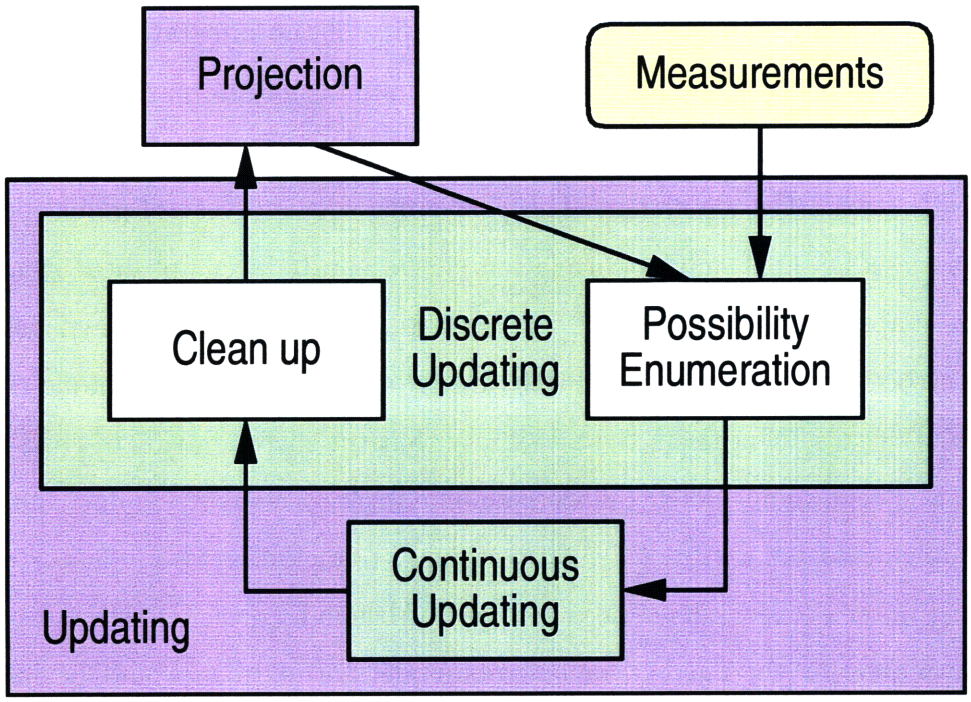


Figure 3-4: Overview of the IMAN process. Recursive estimation is divided into two steps: projection and updating. In integrated mapping and navigation, the update step contains discrete, as well as continuous, components. There are two discrete update activities: possibility enumeration and clean-up.

tinuous updating, an additional discrete step cleans up the resulting estimated states, removing those discrete possibilities that are to be rejected. The current implementation of IMAN only considers discrete possibilities regarding updating; however, a more general scheme would enable discrete estimation regarding projection, as well.

The process of IMAN is shown in greater detail in Figure 3-5. Discrete possibility enumeration for updating begins by determining what measurement-to-track associations, or match hypotheses, to consider.

A **match hypothesis** is an hypothesis associating a measurement with a particular feature track.

Based on this information, the possibilities for feature tracks are fully enumerated. The vehicle is involved with all of the measurements together; the vehicle track must

therefore be enumerated based of global assignments rather than individual hypotheses.

> An **assignment** is a globally consistent set of hypotheses for a given update cycle. Each **assignment** represents a plausible resolution for *all* update decisions for that cycle.

The vehicle track is enumerated based on the set of possible assignments. After the continuous updating of the feature and vehicle tracks, a discrete clean-up step, pruning, is performed. Pruning removes discrete possibilities that are to be rejected. Chapter 5 discusses pruning and other complexity management issues.

As noted in Section 3.1, the vehicle and each proposed feature are described by tracks, which have a dual representation as trees and track steps. We denote the vehicle track y, and sets of vehicle states X. The set of feature tracks is denoted

$$H = \{{}_i\eta\}_i \,, \tag{3.69}$$

where i indexes over each feature track ${}_i\eta$. Sets of feature states for feature ${}_i\eta$ are denoted ${}_i\Xi$. Indices to indicate time or conditioning factors are included as necessary. For a cycle of the IMAN process at time k, we begin with the updated results from the previous cycle and a set of new measurements to be incorporated:

- feature tracks $H_{k-1|k-1} = \left\{{}_i\eta_{k-1|k-1}\right\}_i$, with
  - updated feature states for each track ${}_i^{u}\Xi_{k-1|k-1} = \left\{{}_a\xi^i_{k-1|k-1}\right\}_a$,
- vehicle track y, with
  - updated vehicle states ${}^{u}X_{k-1|k-1} = \left\{{}_bx_{k-1|k-1}\right\}_b$, and
- measurements taken at time k, $Z_k = \{{}_mz_k\}_m$.

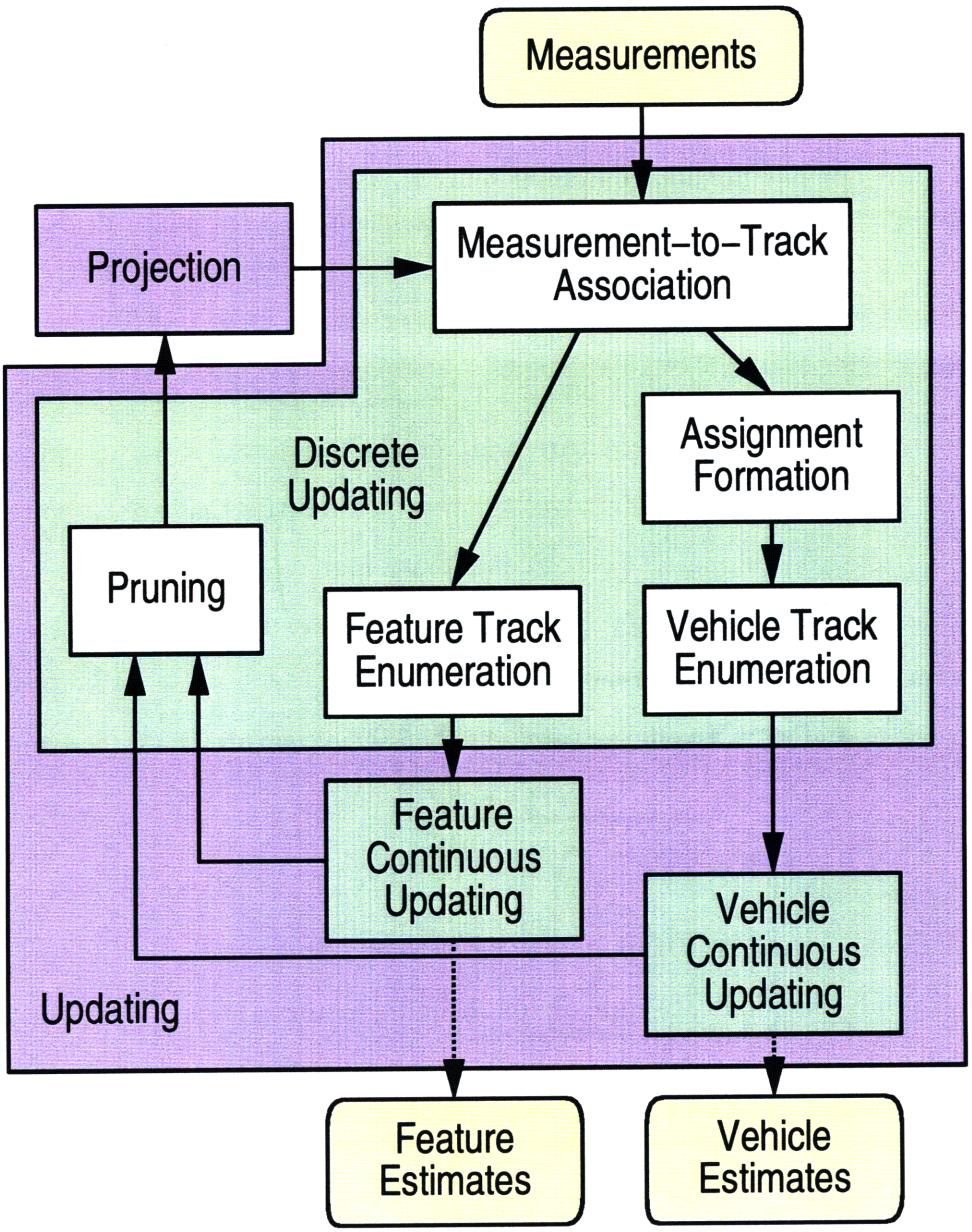


Figure 3-5: The process of integrated mapping and navigation. The discrete updating step is shown in detail. Projected state estimates and measurements are used to determine measurement-to-track associations, or match hypotheses. Feature tracks are fully enumerated, and then updated, based on these possible matches. The vehicle must be enumerated based on global assignments, rather than individual hypotheses. Pruning is used to remove rejected discrete possibilities, for example, possibilities with very low likelihoods.

Below, we consider the steps in the IMAN process cycle. For each step, we examine what existing elements are used and what the product of the step is. The theoretical basis for the process is described and novel aspects are highlighted. Furthermore, we develop an example throughout this discussion to illustrate each step. In this example, we initially assume a vehicle and a single feature track, each with two existing possible states, as shown in Figure 3-6.

3.5.1 Projection

Track projection is the process of predicting a model state based on a previous estimate and some behavioral model. In this thesis, we assume that projection proceeds without any discrete uncertainty. Each proposed track, including the vehicle track, has a single dynamic model which describes the behavior of its state estimates through time. The set of possible updated states from the previous cycle are projected, using this dynamic model, to a set of possible predicted states.

For the vehicle track y, the existing estimates $X_{k-1|k-1}$ and the vehicle model

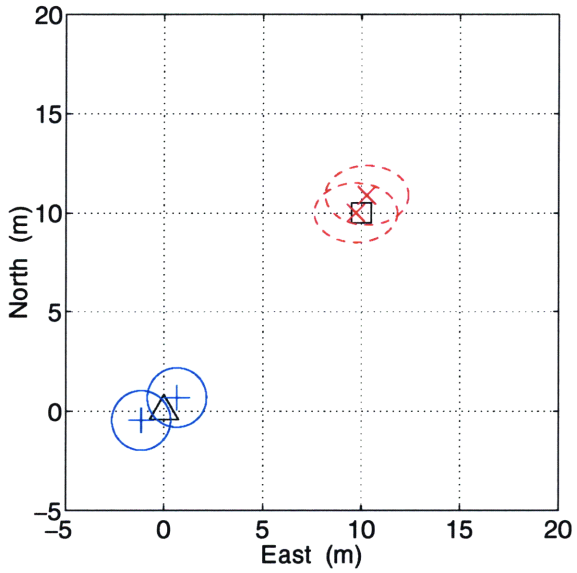
$$ {}_y\phi = {}^{\nu}\phi \tag{3.70} $$

are used to provide the set of projected estimates $X_{k|k-1}$. The specifics of the vehicle model used in this thesis are presented above in Section 3.2.2. Using an extended Kalman filter implementation, the dynamic function

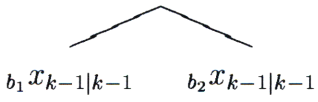
$$ {}_yf = {}^{\nu}f, \tag{3.71} $$

its Jacobian with respect to a specific vehicle state estimate

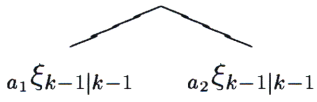
$$ {}_xF^y_{k-1|k-1} = \left.\frac{\partial\, {}_yf}{\partial x}\right|_{x=\hat{x}_{k-1|k-1}}, \tag{3.72} $$



(a) Initial states. The vehicle, indicated by a triangle, is located at (0,0); there is a single feature, indicated by a square, at (10,10). Vehicle track estimates are indicated by a cross; their three-sigma highest density regions (HDRs) by a solid line. Each feature track estimate is indicated by an x; its three-sigma HDR by a dashed line.



(b) Vehicle track tree



(c) Feature track tree

Figure 3-6: IMAN example initial states. A vehicle and a single feature are considered. At the start of the cycle, each track contains two updated estimates from a previous time cycle.

and the noise process ${}_{y}w$ with covariance

$$ {}_{y}Q = {}^{\nu}Q \tag{3.73} $$

are used to provide a Gaussian approximation for the projected states. The projected densities are characterized by means

$$ {}_{l}\hat{x}_{k|k-1} = {}_{y}f\left({}_{b}\hat{x}_{k-1|k-1}\right), \quad \forall\, b, \tag{3.74} $$

where b indexes over the set of existing (updated) states:

$$ X_{k-1|k-1} = \left\{ {}_{b}x_{k-1|k-1} \right\}_{b}. \tag{3.75} $$

The covariances of the estimates are

$$ {}_{lx}P^{y}_{k|k-1} = {}_{bx}F^{y}_{k-1|k-1}\ {}_{bx}P^{y}_{k-1|k-1}\ {}_{bx}F^{y}_{k-1|k-1}{}^{T} + {}_{y}Q, \quad \forall\, b. \tag{3.76} $$

Since there is no discrete uncertainty, there is a one-to-one mapping between the set of previous states $\left\{ {}_{b}x_{k-1|k-1} \right\}_{b}$ and the set of projected states $\left\{ {}_{l}x_{k|k-1} \right\}_{l}$.

A similar process is used to project each feature track. Each track ${}_{i}\eta$ makes use of the point-like feature model

$$ {}_{i\eta}\phi = {}^{\mathcal{F}_{pt}}\phi, \tag{3.77} $$

which, as developed in Section 3.2.1, has a dynamic function

$$ {}_{i\eta}f = {}^{\mathcal{F}_{pt}}f \tag{3.78} $$

with a Jacobian

$$ {}_{\xi}F^{{}_{i}\eta}_{k-1|k-1} = \left. \frac{\partial\, {}_{i}\eta f}{\partial \xi} \right|_{\xi=\hat{\xi}_{k-1|k-1}} \tag{3.79} $$

and a noise process ${}_{i}\eta w$ with covariance

$$ {}_{i}\eta Q = {}^{\mathcal{F}_{pt}}Q. \tag{3.80} $$

The projected densities are assumed to be Gaussian with means

$$ {}_{j}\hat{\xi}^{{}_{i}\eta}_{k|k-1} = {}_{i}\eta f\left({}_{a}\hat{\xi}^{{}_{i}\eta}_{k-1|k-1}\right), \quad \forall\, a, i, \tag{3.81} $$

and covariances

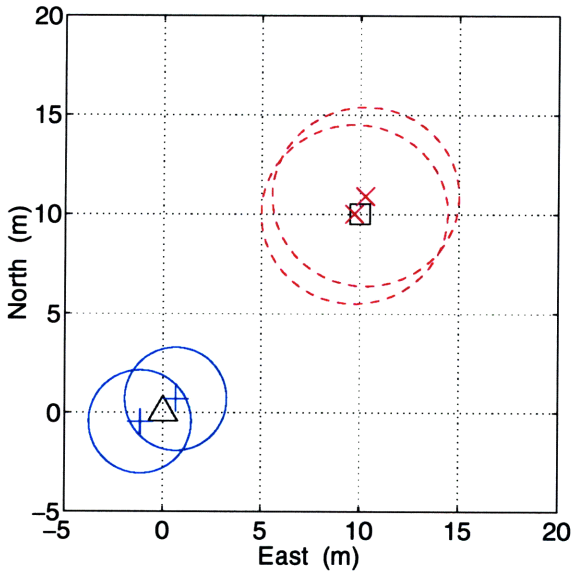
$$ {}_{j\xi}P^{{}_{i}\eta}_{k|k-1} = {}_{a\xi}F^{{}_{i}\eta}_{k-1|k-1}\ {}_{a\xi}P^{{}_{i}\eta}_{k-1|k-1}\ {}_{a\xi}F^{{}_{i}\eta}_{k-1|k-1}{}^{T} + {}_{i}\eta Q, \quad \forall a, i. \tag{3.82} $$

Again, projection is a one-to-one mapping between the previous updates $\left\{ {}_{a}\xi^{{}_{i}\eta}_{k-1|k-1} \right\}_{a,i}$ and the projected states $\left\{ {}_{j}\xi^{{}_{i}\eta}_{k|k-1} \right\}_{j,i}$.

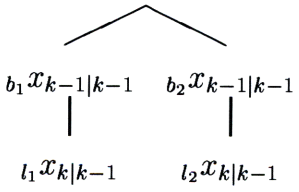
In our example, each existing state yields one projected state. In this case, the vehicle is assumed to be (nominally) stationary, so the only effect of projection is to increase the uncertainty for each of the states (due to process noise). The projected state estimates and the vehicle and feature track trees after the projection step are shown in Figure 3-7.

3.5.2 Measurement-to-track association

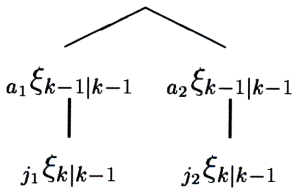
Measurement-to-track association is the process of determining possible match hypotheses, or data associations possibly linking a measurement to a particular feature track. This is the first step in the discrete estimation process determining the navigational events surrounding the incorporation of measurements. There are two stages



(a) Projected states. The vehicle, indicated by a triangle, is located at (0,0); there is a single feature, indicated by a square, at (10,10). Vehicle track estimates are indicated by a cross; their three-sigma HDRs by a solid line. Each feature track estimate is indicated by an ex; its three-sigma HDR by a dashed line.



(b) Vehicle track tree



(c) Feature track tree

Figure 3-7: IMAN example projected states. Model uncertainty, represented as process noise, results in increased uncertainty, that is, larger HDRs, for each estimate after projection. The projection process is a one-to-one mapping, since no discrete uncertainty is considered regarding vehicle or feature dynamics.

to this step. First, measurements are gated using vehicle and feature estimates so that unlikely matches are not considered (essentially, the prior probability of such unlikely matches is set to zero). Second, a probability needs to be assigned to any matches which gate.

We must first consider what match hypotheses to entertain. Each possible combination of vehicle track y, feature track ${}_i\eta$, and measurement ${}_m z_k$ is examined. The vehicle track y has a set of possible projected states

$$X_{k|k-1} = \left\{ {}_l x_{k|k-1} \right\}_l . \tag{3.83}$$

Each feature ${}_i\eta$ has a set of possible projected states

$${}_i\Xi_{k|k-1} = \left\{ {}_j\xi^i_{k|k-1} \right\}_j . \tag{3.84}$$

The goal is to produce the set of match, or association, hypotheses ${}^{\mathcal{A}}\Theta$ to be considered.

Each state has a dependency set associated with it. Only compatible states can be compared. Recall that the dependency set compatibility function C_D can be used to test for compatibility.[6] A vehicle state ${}_l x_{k|k-1}$ and a feature state ${}_j\xi^i_{k|k-1}$ are compatible if

$$C_D \left(D_{{}_l x_{k|k-1}}, D_{{}_j \xi^i_{k|k-1}} \right) = \text{true}. \tag{3.85}$$

The measurement model

$${}_m\mu_k = {}^{\mathcal{W}_s}\mu \tag{3.86}$$

is used to characterize the expected measurement $\hat{z}_{k|j,l}$ for each compatible pair of

[6] See Section 3.3.2.

vehicle and feature states, $\left({}_{l}x_{k|k-1}, {}_{j}\xi^{i}_{k|k-1}\right)$. The sonar measurement model ${}^{\mathcal{W}_s}\mu$ used in this thesis is described in Section 3.2.3, and provides an expected measurement function ${}_{m}h_k$, its Jacobians with respect to given vehicle and feature state estimates, ${}_{l}{}^{x}H^{m}_{k}$ and ${}_{j}{}^{\xi^i}H^{m}_{k}$, and the measurement noise process ${}_{m}v_k$ with covariance

$$ {}_{m}R_k = {}^{\mathcal{W}_s}R. \tag{3.87} $$

This measurement model provides a Gaussian expected measurement estimate with mean

$$ \hat{z}_{k|j,l} = {}_{m}h_k \left({}_{l}x_{k|k-1}, {}_{j}\xi^{i}_{k|k-1}\right), \tag{3.88} $$

and covariance

$$ {}_{m}S_{k|j,l} = {}_{l}{}^{x}H^{m}_{k}\ {}_{l}{}^{x}P_{k|k-1}\ {}_{l}{}^{x}H^{mT}_{k} + {}_{j}{}^{\xi^i}H^{m}_{k}\ {}_{j}{}^{\xi}P^{i}_{k}\ {}_{j}{}^{\xi^i}H^{mT}_{k} + {}_{m}R_k. \tag{3.89} $$

The measurement innovation is defined as

$$ {}_{m}\nu_{k|j,l} = {}_{m}z_k - \hat{z}_{k|j,l}, \tag{3.90} $$

and also has covariance ${}_{m}S_{k|j,l}$. The Mahalanobis distance between the actual and estimated measurements is the square of the innovation normalized by its covariance, $\nu^T S^{-1}\nu$. This is used, as is common [5, 17][7], to define a measurement gate,

$$ {}_{m}\nu^{T}_{k|j,l}\ {}_{m}S^{-1}_{k|j,l}\ {}_{m}\nu_{k|j,l} \leq {}_{\mu}\gamma. \tag{3.91} $$

[7] A number of alternative strategies have been developed which attempt to more effectively reduce unlikely matches [35, 22, 64] or adapt gating the threshold based on additional information [40].

Throughout this thesis, a measurement gate constant of

$$w_{s\mu}\gamma = 4 \tag{3.92}$$

is used. This means that a measurement that actually came from the specified vehicle and feature states according to the sonar measurement model would have an 87% probability of falling within the gate.

If a measurement gates with any compatible combination of vehicle and feature states, a match hypothesis is formed for the feature track and the measurement. That is, we entertain the possibility that the measurement comes from that feature. Note that the Mahalanobis distance of the innovation is a chi-square distributed random variable with degrees of freedom equal to the order of the innovation vector. The chi-square cumulative density function P_{χ^2}, which can be computed using standard algorithms [85], can be used to evaluate the probability that the innovation is no larger than the realized value ν. We define the probability of a match $P_{\mathcal{A}}$ as the *maximum* probability over all pairs of compatible vehicle and feature states that gate with the measurement,

$$\begin{aligned} P_{\mathcal{A}}\left({}_i\eta, {}_m z_k\right) &= \max_{j,l} P_{\chi^2}\left({}_m\nu_{k|j,l}^T \; {}_m S_{k|j,l}^{-1} \; {}_m\nu_{k|j,l}\right) \\ \forall\, j,l \;\Big|\; C_D\left(D_{l x_{k|k-1}}, D_{j \xi^i_{k|k-1}}\right) &= \text{true} \quad \text{and} \quad {}_m\nu_{k|j,l}^T \; {}_m S_{k|j,l}^{-1} \; {}_m\nu_{k|j,l} \leq w_{s\mu}\gamma \,. \end{aligned} \tag{3.93}$$

In our example, we assume that all pairs of vehicle and feature states are compatible. Since there are two state estimates for each track, that is, the vehicle and the feature, there will be four pairs of states to consider when checking for a match hypothesis between a given measurement and the feature track. Two measurements are received, as shown in Figure 3-8. These measurements are compared with the expected measurement densities for the four pairs of vehicle and feature states. Note that this comparison takes place in measurement space, but is equivalent to innova-

tion space except for a translation. Measurements must fall within at least one of these expected measurement gates for a match hypothesis to be considered. Both measurements gate with at least one pair of states, so two match hypotheses are formed.

3.5.3 Hypothesis formation

Feature track enumeration is the enumeration of decidedly possible hypotheses regarding the disposition of the proposed feature tracks being estimated. At each time cycle, hypotheses are formed about the disposition of each feature track. Each proposed track ${}_{i}\eta$ forms a dispositional decision ${}^{\mathcal{D}}_{i\eta}\delta_k$ to consider possibilities about what happened to that feature during the time cycle. The goal is to determine all of the hypotheses that should be considered for this dispositional decision

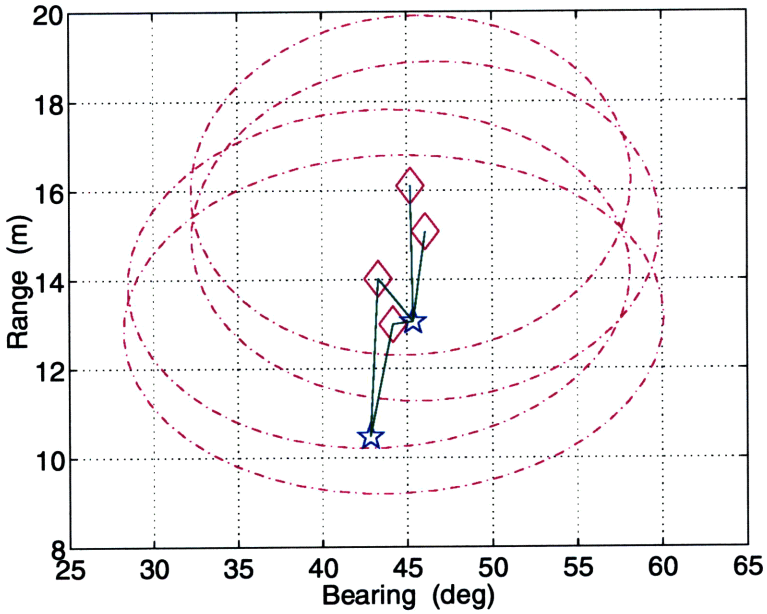
$$\Theta_{i\eta\delta_k} = \{{}_{u}\theta^{i\eta}_{k}\}_{u}\,, \tag{3.94}$$

where u indexes over all of the hypotheses to be considered in decision ${}_{i\eta}\delta_k$. The first step in enumerating the feature tracks is measurement-to-track association, described above. Once the set of match hypotheses has been found, additional hypotheses about the disposition of features and the origin of measurements can be postulated. The following types of hypotheses can be made regarding feature disposition:

- **match** ${}^{\mathcal{A}}\vartheta$–the feature has given rise to a (single) specific measurement and
- **miss** ${}^{\mathcal{M}}\vartheta$–the feature has not given rise to any measurement.

A feature track may have any number of match hypotheses, but must have zero or one miss hypotheses.

Decisions are also formed about the origin of measurements. Each measurement ${}_{m}z_k$ forms an origin decision ${}^{\mathcal{O}}_{mz}\delta_k$ to consider hypotheses regarding its origin. The



(a) Measurements and expected measurement gates. Each of the four compatible pairs of vehicle and feature states generates an expected measurement, indicated by a diamond, with a gating region shown by a dashed-dotted line. Two measurements, indicated by stars, are compared with these gating regions. Positive gates, or associations, are indicated by solid lines.

	$j_1\xi_{k\|k-1}$	$j_2\xi_{k\|k-1}$
$l_1x_{k\|k-1}$	m_1z_k, m_2z_k	m_1z_k, m_2z_k
$l_2x_{k\|k-1}$	m_1z_k	m_1z_k

(b) A table of associations, indicating which measurements gate with each pair of vehicle and feature states.

Figure 3-8: IMAN example gating. Two measurements are received and compared with state estimates in measurement space. The first measurement gates with all four pairs of vehicle and feature states, while the second only gates with two of the pairs. Match hypotheses are formed associating both measurements with the feature.

goal is to enumerate the full set of hypotheses to be considered for each origin decision

$$\Theta_{{}_{m}z\delta_k} = \{{}_{u}\theta_k^{m}{}^{z}\}_{u}, \tag{3.95}$$

where u indexes over all of the hypotheses to be considered for decision ${}_{m}z\delta_k$. The following types of hypotheses can be made regarding measurement origin:

- **match** ${}^{\mathcal{A}}\vartheta$–the measurement has arisen from a (single) specific feature that is currently being tracked,
- **new** ${}^{\mathcal{N}_\iota}\vartheta$–the measurement has arisen from a (single) specific feature of type ι that is not currently known, and
- **spurious** ${}^{\mathcal{S}}\vartheta$–the measurement does not reflect any useful relative measurement of a feature (known or unknown).

A measurement may have any number of match or new hypotheses, but may only have zero or one spurious hypotheses.

The specification of what hypotheses to form in a given situation is part of the prior model definition for the tracks and measurements. Match hypotheses are formed based on a gating procedure, using continuous models to eliminate unlikely discrete possibilities. In the most thorough enumeration, each feature track would also form a miss hypothesis and each measurement would also form a spurious hypothesis and new hypotheses for each possible feature class. Often, this extensive formation of hypotheses is unnecessary. In Section 5.3, a number of hypothesis formation strategies are considered. Reducing the hypotheses formed based on prior models is a fundamental technique for complexity management.

In our example, we have a single feature η and two measurements, ${}_{m_1}z_k$ and ${}_{m_2}z_k$. The feature track has a dispositional decision ${}_{\eta}\delta_k$. The measurements have origin decisions ${}_{m_1}z\delta_k$ and ${}_{m_2}z\delta_k$, respectively. During gating, we determined that a match hypotheses should be formed associating both of these measurements with the feature.

$$ {}_{\eta}\delta_k = \left\{ \begin{array}{l} {}_{p_1}\theta_k = {}^{\mathcal{A}}\vartheta\left(\eta, {}_{m_1}z_k\right) \\ {}_{p_2}\theta_k = {}^{\mathcal{A}}\vartheta\left(\eta, {}_{m_2}z_k\right) \\ {}_{p_3}\theta_k = {}^{\mathcal{M}}\vartheta\left(\eta, k\right) \end{array} \right\} $$

(a) Feature η disposition

$$ {}_{m_1 z}\delta_k = \left\{ \begin{array}{l} {}_{p_1}\theta_k = {}^{\mathcal{A}}\vartheta\left(\eta, {}_{m_1}z_k\right) \\ {}_{p_4}\theta_k = {}^{\mathcal{S}}\vartheta\left({}_{m_1}z_k\right) \end{array} \right\} \qquad {}_{m_2 z}\delta_k = \left\{ \begin{array}{l} {}_{p_2}\theta_k = {}^{\mathcal{A}}\vartheta\left(\eta, {}_{m_2}z_k\right) \\ {}_{p_5}\theta_k = {}^{\mathcal{S}}\vartheta\left({}_{m_3}z_k\right) \end{array} \right\} $$

(b) Measurement ${}_{m_1}z_k$ origin

(c) Measurement ${}_{m_2}z_k$ origin

Figure 3-9: IMAN example hypothesis formation. Match hypotheses are considered associating both measurements with the feature track. The feature track also entertains a miss hypothesis. Each measurement also forms a spurious hypothesis. The fully-enumerated decisions for the feature disposition and the measurement origins are shown.

We also consider a miss hypothesis for the feature track and spurious hypotheses for each feature. Figure 3-9 shows the enumerated decisions after hypothesis generation.

3.5.4 Feature track updating

Once the discrete possibilities have been enumerated, updated feature track estimates based on these possibilities must be produced. During the measurement-to-track association step, we compare each possible projected vehicle state ${}_{\iota}x_{k|k-1}$ and each possible projected feature state ${}_{\jmath}\xi^i_{k|k-1}$, attempting to gate each measurement ${}_{m}z_k$ with these pairs of states. If the measurement gates with this pair of vehicle and feature states, a match hypothesis is formed. An updated state estimate is also produced based on these projected vehicle and feature states and the measurement. Although there will only be a single match hypothesis for a given feature track and measurement, there may be multiple updated states based on that hypothesis, depending on the sets of predicted vehicle and feature states. The differences between these updated feature states is in dependency on previous decisions. As noted above, once all decisions are

resolved, a single track estimate will remain.

Each match hypothesis formed associates the vehicle, a feature ${}_i\eta$, and a measurement ${}_m z_k$. Recall that pairs of vehicle and feature states $\left({}_l x_{k|k-1}, {}_j\xi^i_{k|k-1}\right)$ gate with the measurement ${}_m z_k$ if and only if (1) the states are compatible,

$$C_D\left(D_{{}_l x_{k|k-1}}, D_{{}_j\xi^i_{k|k-1}}\right) = \text{true}, \tag{3.96}$$

and (2) the Mahalanobis distance between the actual and expected measurements falls below a gating threshold specified by the measurement model ${}_{m z_k}\mu$,

$${}_m\nu^T_{k|j,l}\ {}_m S^{-1}_{k|j,l}\ {}_m\nu_{k|j,l} \leq {}_{m z_k}\mu\gamma. \tag{3.97}$$

For each pair of state where this is the case, we form an updated feature state ${}_n\xi^i_{k|k}$, using an extended Kalman filter and the measurement model ${}_{m z_k}\mu$.[8] The Kalman gain is given by

$${}_m W_{k|j,l} = {}_j\xi P^{i\eta}_{k|k-1}\ {}_j\xi H^{i\eta}_{k|k-1}{}^T\ {}_m S^{-1}_{k|j,l}, \tag{3.98}$$

leading to an updated state that is assumed to be Gaussian and has mean

$${}_n\hat{\xi}^i_{k|k} = {}_j\hat{\xi}^i_{k|k-1} + {}_m W_{k|j,l}\ {}_m\nu_{k|j,l} \quad \text{and covariance} \tag{3.99}$$

$${}_{n\xi}P^{i\eta}_{k|k} = {}_j\xi P^{i\eta}_{k|k-1} - {}_m W_{k|j,l}\ {}_m S^{-1}_{k|j,l}\ {}_m W^T_{k|j,l}. \tag{3.100}$$

Note that for track updating there is not, in general, a one-to-one correspondence between possible projected state and possible updated state.

The dependency set of the updated feature state is the union of the dependency sets of the vehicle and feature states used to produce it, plus dependencies on the

[8] See Section 3.4.1.

match hypothesis associating the feature track and the measurement. We denote the match hypothesis associating feature ${}_i\eta$ and measurement ${}_mz_k$ as

$$ {}_{p_1}\theta_k = {}^{\mathcal{A}}\vartheta\left({}_i\eta, {}_mz_k\right). \tag{3.101} $$

The dependency ${}_{e_1}d$ requires the resolution of the feature track dispositional decision ${}_i\eta\delta_k$ to this match hypothesis:

$$ {}_{e_1}d = {}_i\eta\delta_k \mapsto {}_{p_1}\theta_k. \tag{3.102} $$

Dependency ${}_{e_2}d$ requires the resolution of the measurement origin decision ${}_mz\delta_k$ with this match hypothesis:

$$ {}_{e_2}d = {}_mz\delta_k \mapsto {}_{p_1}\theta_k. \tag{3.103} $$

Then the dependency sets for the updated feature states are

$$ D_{{}_n\xi^i_{k|k}} = D_{{}_j\xi^i_{k|k-1}} + D_{{}_lx_{k|k-1}} + {}_{e_1}d + {}_{e_2}d, \quad \forall\, n, i. \tag{3.104} $$

If a miss hypothesis is generated, a single updated feature state is produced for each projected state in the track, ${}_j\xi^i_{k|k-1}$, where j indexes over all of the projected track states. In this case the feature state estimate and covariance are unchanged:

$$ {}_n\hat{\xi}^i_{k|k} = {}_j\hat{\xi}^i_{k|k-1}, \tag{3.105} $$

and

$$ {}_{n\xi}P^{i\eta}_{k|k} = {}_{j\xi}P^{i\eta}_{k|k-1}. \tag{3.106} $$

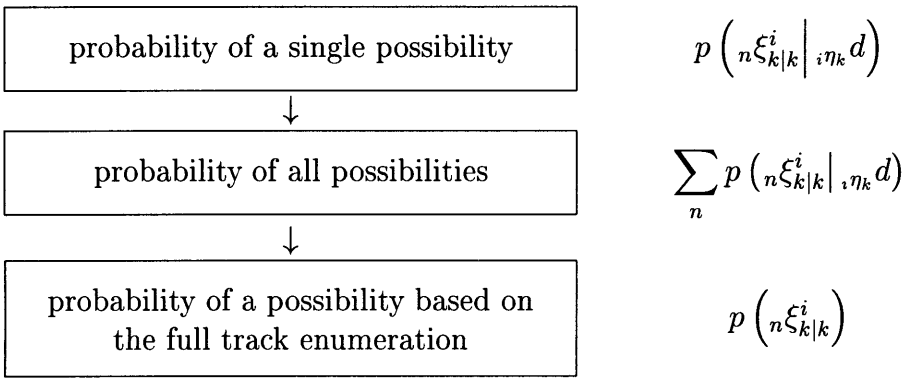


Figure 3-10: Likelihood calculation for updated feature states. First, the probability of individual states is estimated based on the events they depend on and the likelihoods of any states used to produce them. The sum of these probabilities is calculated, determining the total probability for the possibilities being considered. This total probability is used to normalize the updated state likelihoods, producing the probabilities of the updated states conditioned on the outcome of the track enumeration process.

We denote the miss hypothesis for track ${}_i\eta$ at time k as

$$ {}_{p_2}\theta_k = {}^{\mathcal{M}}\vartheta\left({}_i\eta, k\right). \tag{3.107} $$

The dependency ${}_{e_3}d$ requires the resolution of the feature track dispositional decision ${}_i\eta\delta_k$ to this miss hypothesis:

$$ {}_{e_3}d = {}_i\eta\delta_k \mapsto {}_{p_2}\theta_k. \tag{3.108} $$

The dependency set for an updated state based on a miss hypothesis is the union of the dependency set of the projected state and the dependency of the feature dispositional decision on the miss hypothesis,

$$ D_{{}_n\xi^i_{k|k}} = D_{{}_j\xi^i_{k|k-1}} + {}_{e_3}d. \tag{3.109} $$

The likelihoods of each of the possible updated states must now be found. This

occurs in two steps, as shown in Figure 3-10. First, the probability of the state based only on the navigational events that give rise to it is estimated. Second, as a result of the enumeration of events, we assert that these states form an exhaustive, mutually exclusive set of feature states. That is, possibilities that are not enumerated are considered impossible. This allows the normalization of the updated states based on the total probability of the enumerated states. The probability of each updated feature state that is based on a match hypothesis conditioned on the outcome of the feature dispositional decision ${}_{i\eta}\delta_k$ is the product of the probabilities of the projected vehicle and feature states used to form the updated state and the probability of the match hypothesis. We denote the match hypothesis

$$ {}_{p_1}\theta_k = {}^{\mathcal{A}}\vartheta\left({}_{i}\eta, {}_{m}z_k\right). \tag{3.110} $$

The likelihood of the updated state based only on itself can be written

$$ p\left({}_{n}\xi^i_{k|k}\middle|\, {}_{i\eta_k}d = {}_{i\eta}\delta_k \mapsto {}_{p_1}\theta_k\right) = p\left({}_{j}\xi^i_{k|k-1}\right) p\left({}_{l}x_{k|k-1}\right) p\left({}_{p_1}\theta_k\right), \tag{3.111} $$

where $p\left({}_{j}\xi^i_{k|k-1}\right)$ and $p\left({}_{l}x_{k|k-1}\right)$ are the likelihoods of the projected feature and vehicle states, respectively, used to produce the updated state and $p\left({}_{p_1}\theta_k\right)$ is the probability of match for the gating process,

$$ p\left({}_{p_1}\theta_k\right) = \frac{1}{(2\pi)^{N({}_{m}z_k)/2}\left|{}_{m}S_{k|j,l}\right|^{1/2}} \exp\left\{-\frac{1}{2}\, {}_{m}\nu^T_{k|j,l}\, {}_{m}S^{-1}_{k|j,l}\, {}_{m}\nu_{k|j,l}\right\}, \tag{3.112} $$

where $N\left({}_{m}z_k\right)$ is the order of the measurement ${}_{m}z_k$.

The probability of each updated feature state based on a miss hypothesis is given by the product of the projected feature state and the probability of miss for the measurement model,

$$ p\left({}_{n}\xi^i_{k|k}\middle|\, {}_{i\eta_k}d = {}_{i\eta}\delta_k \mapsto {}_{p_2}\theta_k\right) = p\left({}_{j}\xi^i_{k|k-1}\right)\left(1 - {}_{m z_k\mu}P^{\iota}_{D}\right), \tag{3.113} $$

where ${}_{p_2}\theta_k$ is the miss hypothesis,

$$ {}_{p_2}\theta_k = {}^{\mathcal{M}}\vartheta\left({}_i\eta, k\right), \tag{3.114} $$

and ${}_{m z_k}\mu P_D^{\iota}$ is the probability of detection for feature type ι specified by the measurement model ${}_{m z_k}\mu$.

The set of possible updated states for the track,

$$ {}_i\Xi_{k|k} = \left\{ {}_n\xi^i_{k|k} \right\}_n, \tag{3.115} $$

where n indexes over all of the updates that have been formed, is considered a full enumeration of the possible events in updating the track. That is, they form an exhaustive, mutually-exclusive set of events. The likelihood of each updated state, conditioned on the total enumeration of the feature track, is the normalized probability,

$$ p\left({}_n\xi^i_{k|k}\right) = \frac{p\left({}_n\xi^i_{k|k} \middle|\, {}_i\eta_k d \right)}{\sum_n p\left({}_n\xi^i_{k|k} \middle|\, {}_i\eta_k d \right)}. \tag{3.116} $$

In our example, we begin feature updating with two possible projected feature states, ${}_{j_1}\xi_{k|k-1}$ and ${}_{j_2}\xi_{k|k-1}$. There are two measurements, ${}_{m_1}z_k$ and ${}_{m_2}z_k$, and two possible projected vehicle states, ${}_{l_1}x_{k|k-1}$ and ${}_{l_2}x_{k|k-1}$. Three hypotheses are formed to enumerate the feature dispositional decision ${}_{\eta}\delta_k$. Both measurements gate with at least one pair of projected vehicle and feature states, leading to match hypotheses

$$ {}_{p_1}\theta_k = {}^{\mathcal{A}}\vartheta\left(\eta, {}_{m_1}z_k\right) \quad \text{and} \tag{3.117} $$

$$ {}_{p_2}\theta_k = {}^{\mathcal{A}}\vartheta\left(\eta, {}_{m_2}z_k\right). \tag{3.118} $$

A miss hypothesis for the feature track is also formed:

$$_{p_3}\theta_k = {}^{\mathcal{M}}\vartheta\,(\eta, k)\,. \tag{3.119}$$

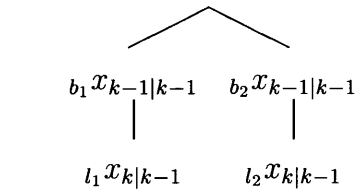
Based on these hypotheses and the results of gating[9], eight updated feature states are produced, as shown in Figure 3-11. These are based on the following combinations of projected feature state, hypothesis, and projected vehicle state and measurement, if any:

1. $_{n_1}\xi_{k|k}$: $_{j_1}\xi_{k|k}$, $_{p_1}\theta_k$, $_{l_1}x_{k|k}$, and $_{m_1}z_k$;
2. $_{n_2}\xi_{k|k}$: $_{j_1}\xi_{k|k}$, $_{p_1}\theta_k$, $_{l_2}x_{k|k}$, and $_{m_1}z_k$;
3. $_{n_3}\xi_{k|k}$: $_{j_1}\xi_{k|k}$, $_{p_2}\theta_k$, $_{l_1}x_{k|k}$, and $_{m_2}z_k$;
4. $_{n_4}\xi_{k|k}$: $_{j_1}\xi_{k|k}$ and $_{p_3}\theta_k$;
5. $_{n_5}\xi_{k|k}$: $_{j_2}\xi_{k|k}$, $_{p_1}\theta_k$, $_{l_1}x_{k|k}$, and $_{m_1}z_k$;
6. $_{n_6}\xi_{k|k}$: $_{j_2}\xi_{k|k}$, $_{p_1}\theta_k$, $_{l_2}x_{k|k}$, and $_{m_1}z_k$;
7. $_{n_7}\xi_{k|k}$: $_{j_2}\xi_{k|k}$, $_{p_2}\theta_k$, $_{l_1}x_{k|k}$, and $_{m_2}z_k$; and
8. $_{n_8}\xi_{k|k}$: $_{j_2}\xi_{k|k}$ and $_{p_3}\theta_k$.

3.5.5 Assignment formation

Feature track updating is relatively simple because of the assumption that each track is involved in a single dispositional decision, that is, each feature gives rise to at most one measurement. Feature track enumeration is thus independent for each track. This is not the case when enumerating the vehicle track. The vehicle is invested in all of the dispositional decisions as it takes part in every measurement event. Because

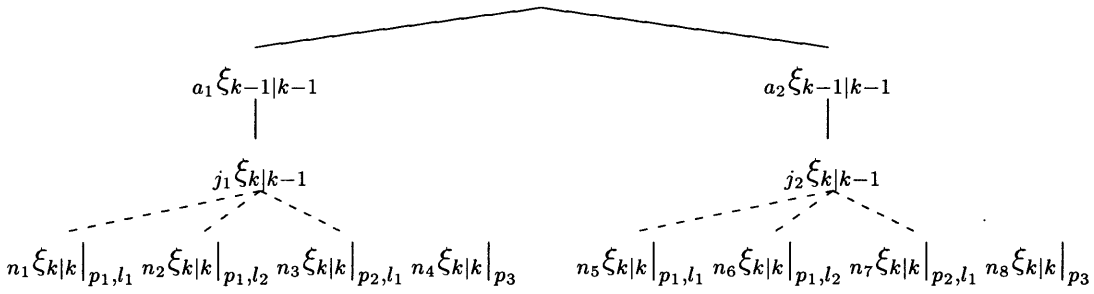
[9]See Section 3.5.2.



(a) Projected vehicle track tree

$$ {}_{\eta}\Theta_k = \left\{ \begin{array}{c} {}_{p_1}\theta_k = {}^{\mathcal{A}}\vartheta\left(\eta, {}_{m_1}z_k\right) \\ {}_{p_2}\theta_k = {}^{\mathcal{A}}\vartheta\left(\eta, {}_{m_2}z_k\right) \\ {}_{p_1}\theta_k = {}^{\mathcal{M}}\vartheta\left(\eta\right) \end{array} \right\} $$

(b) Feature dispositional hypotheses



(c) Updated feature track tree

Figure 3-11: IMAN example feature updating. The projected vehicle track (a) and possible dispositional hypotheses for the feature track (b) are used to form updated feature states (c). The indices of the hypothesis and projected feature state, if any, that are used in forming each updated feature state are indicated after each updated state. Updates based on match hypotheses are shown with dashed lines. Updates based on miss hypotheses are shown with dotted lines.

of this, hypotheses regarding individual decisions cannot be treated independently. Instead, global assignments which resolve the set of all dispositional decisions for the current time cycle are needed. An assignment is a set of hypotheses which consistently resolves all decisions in a given set.

For time k, the set of all dispositional and origin decisions is denoted

$$\Delta_k = \{{}_o\delta_k\}_o\,, \tag{3.120}$$

where o indexes over all of the decisions. Each decision ${}_o\delta_k$ has a set of hypotheses that possibly resolve it,

$${}_o\Theta_k = \{{}_u\theta_k^o\}_u\,, \tag{3.121}$$

where u indexes over all of the hypotheses that potentially resolve decision ${}_o\delta_k$. Individual hypotheses can resolve more than one decision. For example, a match hypothesis resolves both a feature dispositional decision and a measurement origin decision. The set of all hypotheses at time k is the union of these decision hypothesis sets for all decisions at time k:

$$\Theta_k = \{{}_p\theta_k\}_p = \sum_o {}_o\Theta_k. \tag{3.122}$$

The decision hypothesis indexing function U can be used to recover and index the hypotheses regarding a particular decision,

$${}_o\Theta_k = U\left({}_o\delta_k, \Theta_k\right). \tag{3.123}$$

The result of assignment formation is a set of assignments that represent all of

the global possibilities being considered for a given time cycle,

$$\Omega_k = \{{}_r\omega_k\}_r \,. \tag{3.124}$$

Each assignment consistently resolves all of the decisions in Δ_k. An assignment ${}_r\omega_k$ can be represented as a set of hypotheses:

$${}_r\omega_k \twoheadrightarrow \Theta_k^r = \{{}_v\theta_k^r\}_v \,, \tag{3.125}$$

where v indexes over all of the hypotheses used by assignment ${}_r\omega_k$. The assignment hypothesis indexing function V can recover and index the hypotheses used by an assignment from the set of all hypotheses,

$$\Theta_k^r = V\left({}_r\omega_k, \Theta_k\right). \tag{3.126}$$

Assignments can also be represented as dependency sets. An assignment ${}_r\omega_k$ specifies the resolution of every decision for time k. These dependencies are denoted as

$${}_r\omega_k \twoheadrightarrow D_{{}_r\omega_k} = \{{}_o\delta_k \mapsto {}_v\theta_k^r\}_o \,, \tag{3.127}$$

where every decision o resolves to some hypothesis in Θ_k^r. Every hypothesis in Θ_k^r resolves some decision in Δ_k. There must be only one hypothesis for each decision in the set of hypotheses used by an assignment, that is, the intersection of a decision hypothesis set ${}_o\Theta_k$ and an assignment hypothesis set Θ_k^r must be a single hypothesis for all decisions in Δ_k and all assignments in Ω_k. The Kronecker delta $\delta_{u,v}^*$ is defined

as

$$\delta^*_{u,v} = \begin{cases} 1 & u = v, \\ 0 & \text{otherwise.} \end{cases} \tag{3.128}$$

Using this definition, we can specify necessary and sufficient conditions for assignment validity. If an assignment ${}_r\omega_k$ uses hypotheses $\{{}_v\theta^r_k\}_v$, and each decision ${}_o\delta_k$ can be resolved by the hypotheses $\{{}_u\delta^o_k\}_u$, then we must have

$$\sum_u \sum_v \delta^*_{u,v} = 1 \quad \forall\, u, v, o. \tag{3.129}$$

IMAN uses a simple brute force method to enumerate all possible assignments given a set of decisions and their possible resolving hypotheses. Other methods have been developed for the efficient generation of assignments [2, 70, 106, 105, 52, 13, 78]. In particular, Murty's algorithm [78] has been implemented to aid in assignment formation in multiple hypothesis tracking [70, 25, 63]. This algorithm is capable of efficiently generating a partial, ranked enumeration of assignments based on some likelihood estimation strategy. However, care should be taken, since any likelihood used for such a method is necessarily based on the probabilities of individual assignments, rather than the probabilities of assignments conditioned on the results of vehicle track enumeration.

In our example, there are three decisions for the current time cycle:

- ${}_\eta\delta_k$: the feature dispositional decision,
- ${}_{m_1 z}\delta_k$: the origin decision for measurement ${}_{m_1}z_k$, and
- ${}_{m_2 z}\delta_k$: the origin decision for measurement ${}_{m_2}z_k$.

Figure 3-12 shows the enumeration of these decisions and the three assignments that are formed based on these decisions and the hypotheses that possibly resolve them.

$$\Theta_k = \begin{Bmatrix} {}_{p_1}\theta_k & = & {}^{\mathcal{A}}\vartheta\,(\eta, {}_{m_1}z_k) \\ {}_{p_2}\theta_k & = & {}^{\mathcal{A}}\vartheta\,(\eta, {}_{m_2}z_k) \\ {}_{p_3}\theta_k & = & {}^{\mathcal{M}}\vartheta\,(\eta, k) \\ {}_{p_4}\theta_k & = & {}^{\mathcal{S}}\vartheta\,({}_{m_1}z_k) \\ {}_{p_5}\theta_k & = & {}^{\mathcal{S}}\vartheta\,({}_{m_2}z_k) \end{Bmatrix}$$

(a) Hypotheses

$${}_{\eta\delta}\Theta_k = \begin{Bmatrix} {}_{p_1}\theta_k \\ {}_{p_2}\theta_k \\ {}_{p_3}\theta_k \end{Bmatrix}$$

(b) Feature decision

$${}_{m_1 z}\Theta_k = \begin{Bmatrix} {}_{p_1}\theta_k \\ {}_{p_4}\theta_k \end{Bmatrix}$$

(c) Measurement m_1 decision

$${}_{m_2 z}\Theta_k = \begin{Bmatrix} {}_{p_2}\theta_k \\ {}_{p_5}\theta_k \end{Bmatrix}$$

(d) Measurement m_2 decision

$$\Omega_k = \begin{Bmatrix} {}_{r_1}\omega_k & \twoheadrightarrow & \{{}_{p_1}\theta_k, {}_{p_5}\theta_k\} \\ {}_{r_2}\omega_k & \twoheadrightarrow & \{{}_{p_2}\theta_k, {}_{p_4}\theta_k\} \\ {}_{r_3}\omega_k & \twoheadrightarrow & \{{}_{p_3}\theta_k, {}_{p_4}\theta_k, {}_{p_5}\theta_k\} \end{Bmatrix}$$

(e) Assignments

Figure 3-12: IMAN example assignment formation. Five hypotheses (a) are used in enumerating the feature dispositional decision (b) and the two measurement origin decisions, (c) and (d). Three assignments (e) are formed based on these hypotheses and decisions.

3.5.6 Vehicle track enumeration

Vehicle track enumeration is the process of extending the vehicle track tree from the set of projected vehicle states ${}^{\mathcal{P}}X_{k|k-1}$ to a set of updated vehicle states ${}^{\mathcal{U}}X_{k|k}$ based on the set of assignments Ω_k, the measurements taken Z_k, and the projected feature states Ξ_k. Specifying the updated states involves both forming the particular estimates and evaluating the likelihood of each updated state. The enumeration of the vehicle track depends on the global assignments rather than individual hypotheses. Some hypotheses, such as a miss hypothesis, do not alter the value of a vehicle estimate, but may affect its likelihood. The only type of hypothesis that also affects the value of the vehicle estimate is a match hypothesis. Because of this, vehicle track enumeration and updating are carried out in three parts, as shown in Figure 3-13. First, assignment root states ${}^{\mathcal{R}}X_{k|k-1,\Omega_k}$ are formed. These root states incorporate information about the assignments that have been formed Ω_k. Second, the match hypotheses in each assignment are used to enumerate possibilities for updating. Each assignment root state functions as the root of a sub-tree. The goal is to enumerate all possible updates of a vehicle estimate, a feature estimate, and a measurement. There is a sub-tree level for each match hypothesis in the assignment during this process. Third, once the specific updating possibilities have been enumerated, the measurements are incorporated through continuous updating. Finally, the likelihoods of the updated vehicle states are calculated.

Assignment root state formation

To aid in enumerating the vehicle track tree, assignment root states are used as an intermediary step.

> An **assignment root state** incorporates a specific assignment into the vehicle track enumeration process. Vehicle track trees are updated on the basis of **assignment root states**.

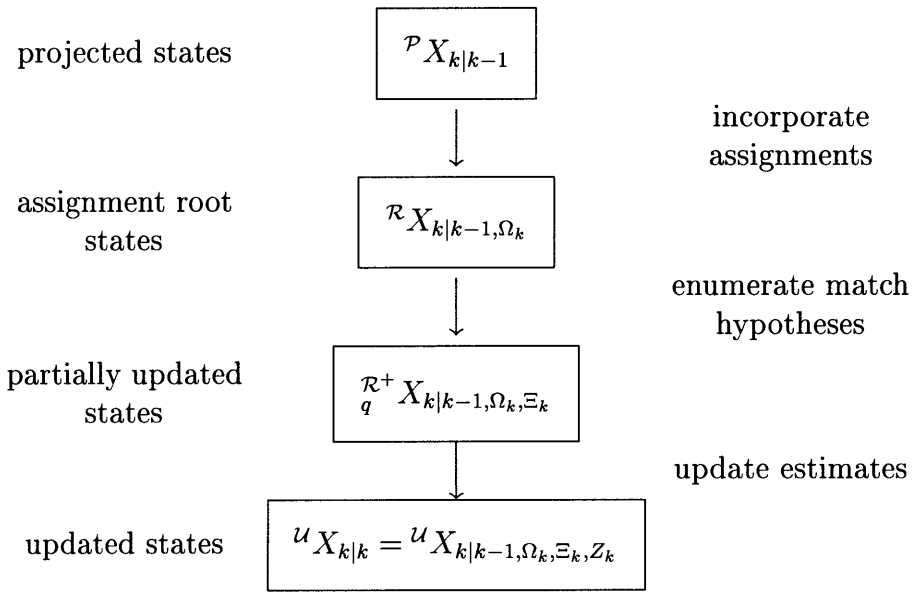


Figure 3-13: Vehicle track enumeration and updating. There are three steps in enumerating and updating the vehicle track. First, assignment information is incorporated to produce assignment root states. Second, match hypothesis update possibilities are enumerated, producing partially updated states. Third, continuous updating finishes the incorporation of measurement information to form the updated vehicle states.

We start with the the set of possible projected vehicle states

$$^{\mathcal{P}}X_{k|k-1} = \left\{ {}_{l}x_{k|k-1} \right\}_{l}, \tag{3.130}$$

where l indexes over all of the projected states. There is a set of possible assignments for the current time cycle,

$$\Omega_k = \left\{ {}_{r}\omega_k \right\}_{r}. \tag{3.131}$$

For each combination of projected vehicle state ${}_{l}x_{k|k-1}$ and assignment ${}_{r}\omega_k$, an assignment root state ${}_{s}x_{k|k-1,l,r}$ is formed. The value of the estimate is unchanged:

$${}_{s}\hat{x}_{k|k-1,l,r} = {}_{l}\hat{x}_{k|k-1}, \tag{3.132}$$

and

$${}_{s}{}^{x}P^{y}_{k|k-1,l,r} = {}_{l}{}^{x}P^{y}_{k|k-1}. \tag{3.133}$$

The dependency set of the assignment root state ${}_{s}x_{k|k-1,l,r}$ is the union of the dependency set of the projected vehicle state ${}_{l}x_{k|k-1}$ and the dependency set representation of the assignment ${}_{r}\omega_k$:

$$D_{{}_{s}x_{k|k-1,l,r}} = D_{{}_{l}x_{k|k-1}} + D_{{}_{r}\omega_l}. \tag{3.134}$$

The likelihoods of the assignment vehicle states are initially set to the prior likelihoods of the assignments that are used to form them, that is,

$$p\left({}_{s}x_{k|k-1,l,r}\right) = p\left({}_{r}\omega_k\right). \tag{3.135}$$

The *a priori* likelihood of an assignment is based on the likelihoods of the hypothesis types that it contains. An assignment ${}_r\omega_k$ is representable as a set of hypotheses

$$ {}_r\Theta_k = \{{}_v\theta_k^r\}_v \,. \tag{3.136} $$

The *a priori* probability for an hypothesis ${}_v\theta_k^r$ depends on what type of hypothesis it is. There are four possible types of hypothesis:

- ${}^{\mathcal{A}}\vartheta$: a match hypothesis,
- ${}^{\mathcal{M}}\vartheta$: a miss hypothesis,
- ${}^{\mathcal{S}}\vartheta$: a spurious measurement hypotheses, and
- ${}^{\mathcal{N}_\iota}\vartheta$: a new feature hypothesis for feature type ι.

The specifics of hypothesis formation are part of the prior model definitions for the feature dynamic models and the measurement models. IMAN typically uses a delayed track initiation strategy which precludes the use of new feature hypotheses during track enumeration. A feature of type ι has a dynamic model ${}_\iota\phi$. A measurement of type κ has a measurement model ${}_\kappa\mu$. Each dynamic model ${}_\iota\phi$ specifies a probability of detection P_D^ι, which is the probability that a known feature within the sensor footprint gives rise to a measurement during any given time cycle. The probability of observing a new feature P_N^ι is also specified when track initiation is considered during vehicle track enumeration. Each measurement model ${}_\kappa\mu$ specifies a probability of false alarm P_F^κ, which is the probability that a given return does not arise from a known feature. An alternative prior model representation uses the probability of false alarm to denote the probability that a given return does not correspond to any feature.

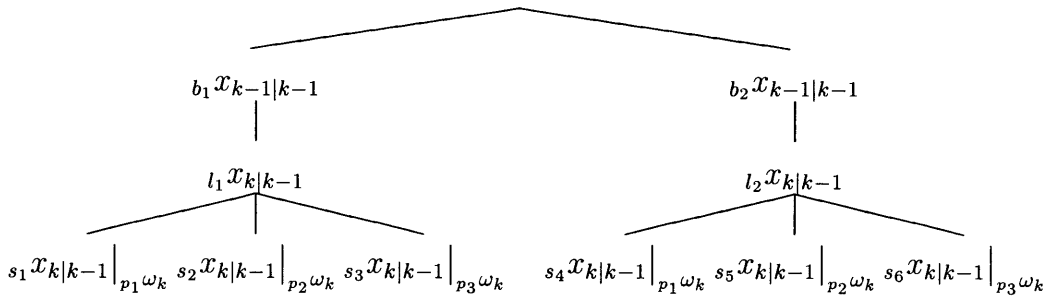


Figure 3-14: IMAN example assignment root state formation. The two projected vehicle states and three possible assignments lead to six possible assignment root states. The assignment upon which each assignment root state is conditioned is indicated.

Based on these specified quantities, the *a priori* hypothesis likelihood is given by

$$p\left({}_p\theta_k\right) = \begin{cases} P_D^\iota\left(1 - P_F^\kappa\right) & {}_p\theta_k = {}^{\mathcal{A}}\vartheta\left({}_i\eta^\iota, {}_m z_k^\kappa\right) \\ \left(1 - P_D^\iota\right) & {}_p\theta_k = {}^{\mathcal{M}}\vartheta\left({}_i\eta^\iota, k\right) \\ P_F^\kappa & {}_p\theta_k = {}^{\mathcal{S}}\vartheta\left({}_m z_k^\kappa\right) \\ P_N^\iota & {}_p\theta_k = {}^{\mathcal{N}_\iota}\vartheta. \end{cases} \tag{3.137}$$

The *a priori* probability of an assignment ${}_r\omega_k$ is the product of the probabilities of its hypotheses ${}_r\Theta_k$,

$$p\left({}_r\omega_k\right) = \prod_v p\left({}_v\theta_k^r\right), \tag{3.138}$$

where v indexes over all of the hypotheses used in assignment ${}_r\omega_k$.

In our example, there are two projected vehicle states and three possible assignments, leading to six assignment root states, as shown in Figure 3-14.

Sub-tree growth

Once assignment root states have been formed, updating possibilities due to any match hypotheses are enumerated. The assignment root state ${}_s x_{k|k-1}$ forms the root

of a sub-tree. The assignment root state is based on an assignment ${}_r\omega_k$. The enumeration for each match hypothesis is independent and propagates the sub-tree through one level. Match hypothesis enumeration starts with a partially updated state at the previous sub-tree level ${}_{t_1}x^{q-1}_{k|k-1}$, where q is the current sub-tree level, and a match hypothesis ${}_v\theta^r_k = {}^{\mathcal{A}}\vartheta$ to be enumerated on sub-tree level q. The goal is a set of partially updated states,

$$\underset{q}{\overset{\mathcal{R}^+}{}} X_{k|k-1,t_1} = \left\{ {}_{t_2}x^q_{k|k-1,t_1} \right\}_{t_2}. \tag{3.139}$$

The set of all possible state for this sub-tree level is denoted ${}_qX_{k|k-1}$. The match hypothesis ${}_v\theta^r_k$ associated feature track ${}_i\eta$ and measurement ${}_mz_k$,

$${}_v\theta^r_k = {}^{\mathcal{A}}\vartheta\left({}_i\eta, {}_mz_k\right). \tag{3.140}$$

The feature track ${}_i\eta$ has a set of possible projected states

$${}_i\Xi_{k|k-1} = \left\{ {}_j\xi^i_{k|k-1} \right\}_j. \tag{3.141}$$

A partially updated state ${}_{t_2}x^q_{k|k-1,t_1}$ is formed for each projected feature state ${}_j\xi^i_{k|k-1}$ if and only if (1) the previous partially updated vehicle state ${}_{t_1}x^{q-1}_{k|k-1}$ and the projected feature state are compatible,

$$C_D\left(D_{{}_{t_1}x^{q-1}_{k|k-1}}, D_{{}_j\xi^i_{k|k-1}}\right) = \text{true}, \tag{3.142}$$

and (2) the measurement ${}_mz_k$ successfully gates with this combination of states,

$${}_m\nu^T_{k|j,t_1}\ {}_mS^{-1}_{k|j,t_1}\ {}_m\nu_{k|j,t_1} \leq \mu\gamma, \tag{3.143}$$

where μ is the measurement model for measurement ${}_mz_k$. The partially updated state ${}_{t_2}x^q_{k|k-1,t_1}$ is formed using an extended Kalman filter. This continuous updating

step is discussed below in Section 3.5.7. The likelihood of the state ${}_{t_2}x^q_{k|k-1,t_1}$ is the product of the likelihood of the parent state ${}_{t_1}x^{q-1}_{k|k-1}$ and the match probability for the update, as defined in Section 3.5.4. If there are no possible projected feature states which meet these criteria, the entire sub-tree, including the assignment root state, is discarded.

In our example, there are six assignment root states. Four of these assignment root states depend on one match hypothesis; the other two do not depend on any match hypotheses. Therefore, there will be a single level of sub-tree growth, as shown in Figure 3-15. One of the assignment root states ${}_{s_5}x_{k|k-1}$ has no gating feature state to update with, so that sub-tree is removed.

Updated vehicle state likelihood

The final step in the discrete enumeration of vehicle updated states is the calculation of likelihoods for the updated states. Each assignment root state ${}_sx_{k|k-1}$ initially has a likelihood equal to the *a priori* probability of the assignment ${}_r\omega_k$ on which it is based,

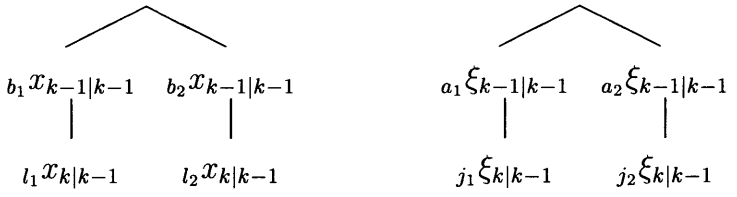
$$p\left({}_sx_{k|k-1}\right) = p\left({}_r\omega_k\right). \tag{3.144}$$

During sub-tree growth, partially updated state likelihoods are multiplied by the gating probabilities as matches are considered. The total set of vehicle updated states is denoted

$$X_{k|k} = \left\{{}_wx_{k|k}\right\}_w, \tag{3.145}$$

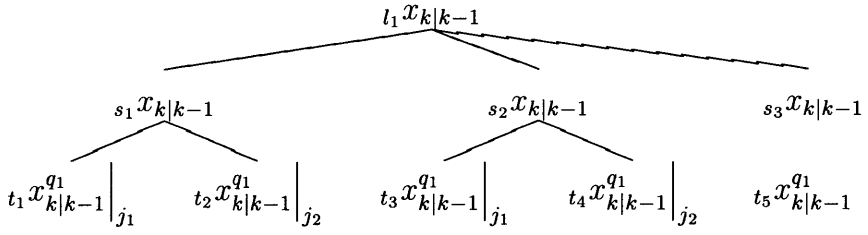
where w indexes over all of the vehicle track tree leaves. The set of leaves which are compatible with an assignment ${}_r\omega_k$ is denoted

$$X_{k|k,{}_r\omega_k} = \left\{{}_\beta x_{k|k}\right\}_\beta. \tag{3.146}$$

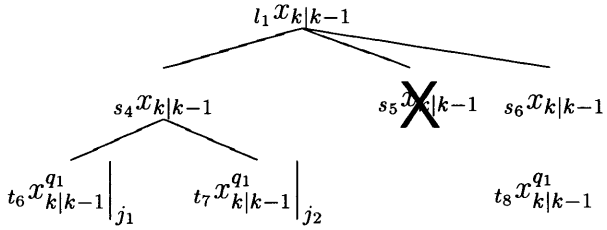


(a) Vehicle track tree

(b) Feature track tree



(c) Sub-tree growth from $_{l_1}x_{k|k-1}$



(d) Sub-tree growth from $_{l_2}x_{k|k-1}$

Figure 3-15: IMAN example sub-tree growth. The projected vehicle track tree (a) and the projected feature track tree (b) are shown for convenience. Sub-tree growth for the vehicle track tree is shown in (c) and (d). Feature states used in vehicle state updating are indicated. Vehicle states that do not depend on any match hypotheses are unchanged; their propagation is indicated by a dotted line. The assignment root state $_{s_5}x_{k|k-1}$ is not compatible with any of the projected feature states, and so its sub-tree is removed.

The total probability for this hypothesis is then the sum of the likelihoods of these leaf states,

$$p\left({}_{r}\omega_{k|k}\right) = \sum_{\beta} p\left({}_{\beta}x_{k|k}\right), \tag{3.147}$$

where the subscript $k|k$ on the assignment ${}_{r}\omega_{k|k}$ indicates that this probability is conditioned on the results of vehicle track enumeration. Each assignment root state likelihood is then set to the product of the likelihood of its parent state and the normalized likelihood of the assignment conditioned on track enumeration,

$$p\left({}_{s}x_{k|k-1,l,r}\right) = p\left({}_{l}x_{k|k-1}\right) \frac{p\left({}_{r}\omega_{k|k}\right)}{\sum_{r} p\left({}_{r}\omega_{k|k}\right)}. \tag{3.148}$$

The remainder of the sub-tree is renormalized to retain the likelihood ratios among child states while ensuring that the sum of the likelihoods of the child states are equal to the likelihood of the parent state.

3.5.7 Vehicle track updating

During sub-tree growth, continuous updating is needed to incorporate measurement updates for the vehicle states. In each case, a partial updated vehicle state ${}_{t}^{\mathcal{R}+}x_{k|k-1}^{q-1}$ is to be updated using a projected feature state ${}_{j}^{\mathcal{P}}\xi_{k|k-1}^{i}$ and a measurement ${}_{m}z_{k}$, based on the match hypothesis

$${}_{v}\theta_{k}^{r} = {}^{\mathcal{A}}\vartheta\left({}_{i}\eta, {}_{m}z_{k}\right). \tag{3.149}$$

Bayesian updating using a Kalman filter is used to produce a vehicle state ${}_{t}x_{k|k-1}^{q}$ as discussed in Section 3.4.1. The Kalman gain is

$${}_{m}W_{k|j,t,q-1} = {}_{tx^{q-1}}P_{k|k-1}^{y} \; {}_{tx^{q-1}}H_{k|k-1}^{y}{}^{T} \; {}_{m}S_{k|j,t,q-1}^{-1}. \tag{3.150}$$

This leads to a partially updated vehicle state ${}_t x^q_{k|k-1}$ which is assumed to be Gaussian with mean

$$ {}_t\hat{x}^q_{k|k-1} = {}_t\hat{x}^{q-1}_{k|k-1} + {}_m W_{k|k-1,j,t,q-1}\, {}_m\nu_{k|k-1,j,t,q-1}, \tag{3.151} $$

and covariance

$$ {}_{tx^q}P^y_{k|k-1} = {}_{tx^{q-1}}P^y_{k|k-1} - {}_m W_{k|k-1,j,t,q-1}\, {}_m S^{-1}_{k|j,t,q-1}\, {}_m W^T_{k|k-1,j,t,q-1}. \tag{3.152} $$

Figure 3-16 shows the possible updated states in our example. There are eight possible updated vehicle states and eight possible updated feature states.

3.6 Summary

In this chapter, we have developed the integrated mapping and navigation algorithm. Some general features of concurrent mapping and localization and hybrid estimation problems in general were identified and their impact on the structure of the integrated mapping and navigation solution was discussed. The importance of prior knowledge representation was highlighted and the models developed for use in this thesis were developed, including dynamic models for survey-class AUVs and point-like features and measurement models for dead reckoning measurements and sonar measurements of point-like features. Various issues pertaining to the continuous and discrete portions of the estimation problem were detailed. Finally, the integrated mapping and navigation algorithm was presented. A Bayesian interpretation of projection and updating was provided, anchoring the continuous portion of IMAN in the existing techniques of extended Kalman filtering and target tracking. The discrete estimation problem was highlighted, including measurement-to-track association, assignment formation, and the enumeration of feature and vehicle tracks. The evaluation of state likelihoods based on this discrete estimation process was also developed. In the next chapter,

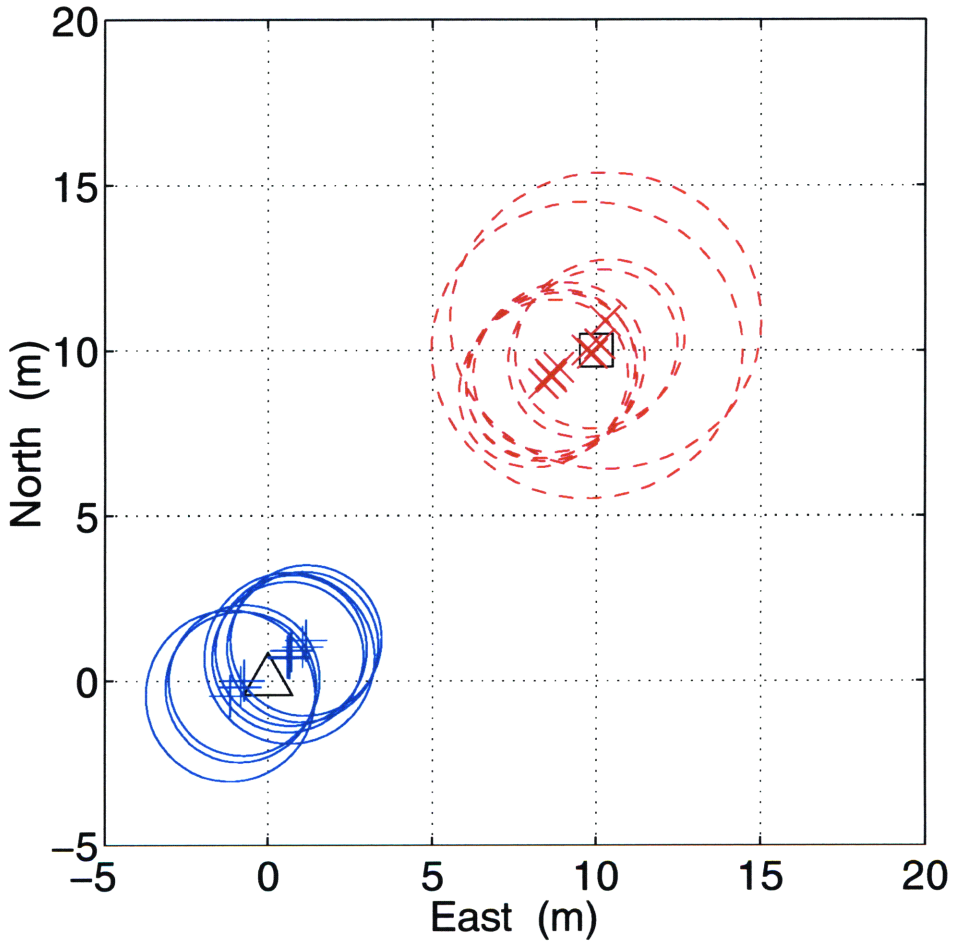


Figure 3-16: IMAN example updated states. Based on the vehicle and feature track enumerations, there are eight possible updated vehicle states and eight possible updated feature states.

we will examine some issues regarding the implementation of the integrated mapping and navigation algorithm.

Chapter 4

Implementing Integrated Mapping and Navigation

In this chapter, we consider the implementation of the integrated mapping and navigation algorithm. We begin by outlining a philosophy of implementation which stresses modularity based on an understanding of the problem structure and extensibility in order to enable the ready incorporation of algorithmic alternatives. An object-oriented analysis of the IMAN problem is described, providing a road map for a C++ implementation of the algorithm.[1] The implementation of prior models is considered, along with specific packages for modeling point-like features and survey-class AUVs. Implementation issues and structure for handling decision-based reasoning are detailed next. Finally state management requirements are outlined and the implementation of the IMAN interface is described.

The implementation of IMAN was accomplished under a particular programming philosophy. First and foremost, the structure of the implementation should mirror to the greatest possible extent the theoretical development. In this way, a calculus of the problem domain can be developed synergistically through theoretical develop-

[1]See Tuohy [102] for a consideration of the general requirements and implementation issues involved in AUV simulation.

Table 4.1: Responsibility of software packages developed for IMAN.

Package	Responsibility
`cmlhp`	Hypothesis formation and decision-based reasoning
`cmlpr`	Prior knowledge representation
`cmlpt`	Point-like feature modeling
`cmlst`	State and tree management and external interface
`cmlv`	Survey-class AUV modeling

ment and an analysis of implementation structure. Second, the overall complexity of the problem requires a modular approach to implementation. The interdependency of structural elements needs to be minimized to ensure a developmentally robust implementation. Finally, emphasis is placed on the extensibility of the implementation, and considerable thought concerning possible future development has informed the analysis of the problem. This philosophy of design is based, in part, on a pattern-based approach to large-scale software systems [59, 38, 79].

4.1 Analysis of IMAN structure

The main responsibilities of the implementation are divided among three software packages. Prior knowledge representation, including implementation of dynamic models and measurement models, is handled in the `cmlpr` package. Reasoning about hypotheses, decisions, and assignments is handled in the `cmlhp` package. The `cmlst` package provides management of states and tree structures as well as the external interface for the algorithm. Two additional packages provide specific model definitions: `cmlv` for survey-class AUVs and `cmlpt` for point-like features. The responsibilities of these packages is summarized in Table 4.1 and the hierarchical nature of their dependencies is illustrated in Figure 4-1.

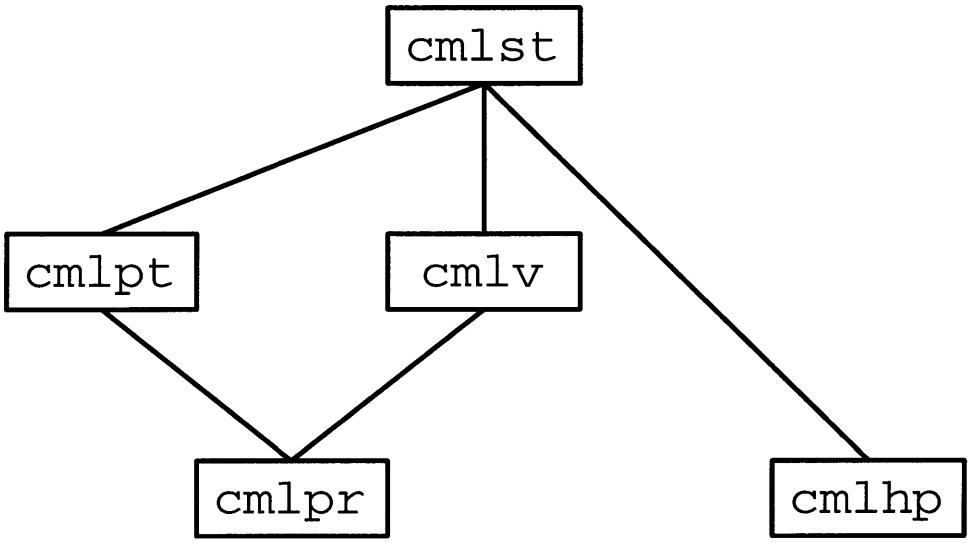


Figure 4-1: The hierarchy of software packages developed for IMAN. Prior knowledge representation is implemented in `cmlpr`. The `cmlhp` package, which contains classes used in hypothesis formation and decision-based reasoning, is also a fundamental package. Specific model types, such as the vehicle model package `cmlv` and the point model package `cmlpt`, make use of the classes in `cmlpr`. State and tree management and the interface to the algorithm are handled in the `cmlst` package, which depends on all of the other packages.

4.2 Prior knowledge representation: the `cmlpr` package

The `cmlpr` package is responsible for issues of prior knowledge representation. Figure 4-2 shows the hierarchical dependency of the software components in `cmlpr`. There are three main functions of a prior model. First, it must represent the projection of a state estimate through time using some dynamic model. Second, it must appropriately update the state estimate when new information becomes available. Third, it must be able to initialize an estimate for a new feature based on one or more measurements. For the case of CML, each of these steps may have a discrete as well as a continuous component, and thus the result of any of these functions may be a set of estimates rather than a single one.

The three main functional definitions (projection, updating, and initialization) are implemented by separate protocol[2] classes: `Projector`, `Updater`, and `Initializer`. The `Projector` and `Updater` components are fundamental, that is, they have no dependency on other components. To allow for the consideration of ontological decisions, `Projector` instances are organized in Markov networks. This structure is encapsulated in the `Network` component. An `Observer` is merely a `Projector` with the additional context provided by a `Network`. Thus a `Projector` provides a model for projection according to a given dynamic model, situated `Observers` allow for changing dynamic types as well. The `Initializer` component depends on the `Observer` component only to situate the resulting estimate within the dynamic model network. The extended Kalman filter algorithm is implemented in three abstract components, `EKFProjector`, `EKFUpdater`, and `EKFInitializer`, that inherit the three basic filtering protocols. These require further derivation to provide model specific information, such as noise parameters.

[2] A protocol class is merely an interface definition. In C++, this means that the class will contain only pure abstract functions (i.e. there are no concrete functions or data members) [38].

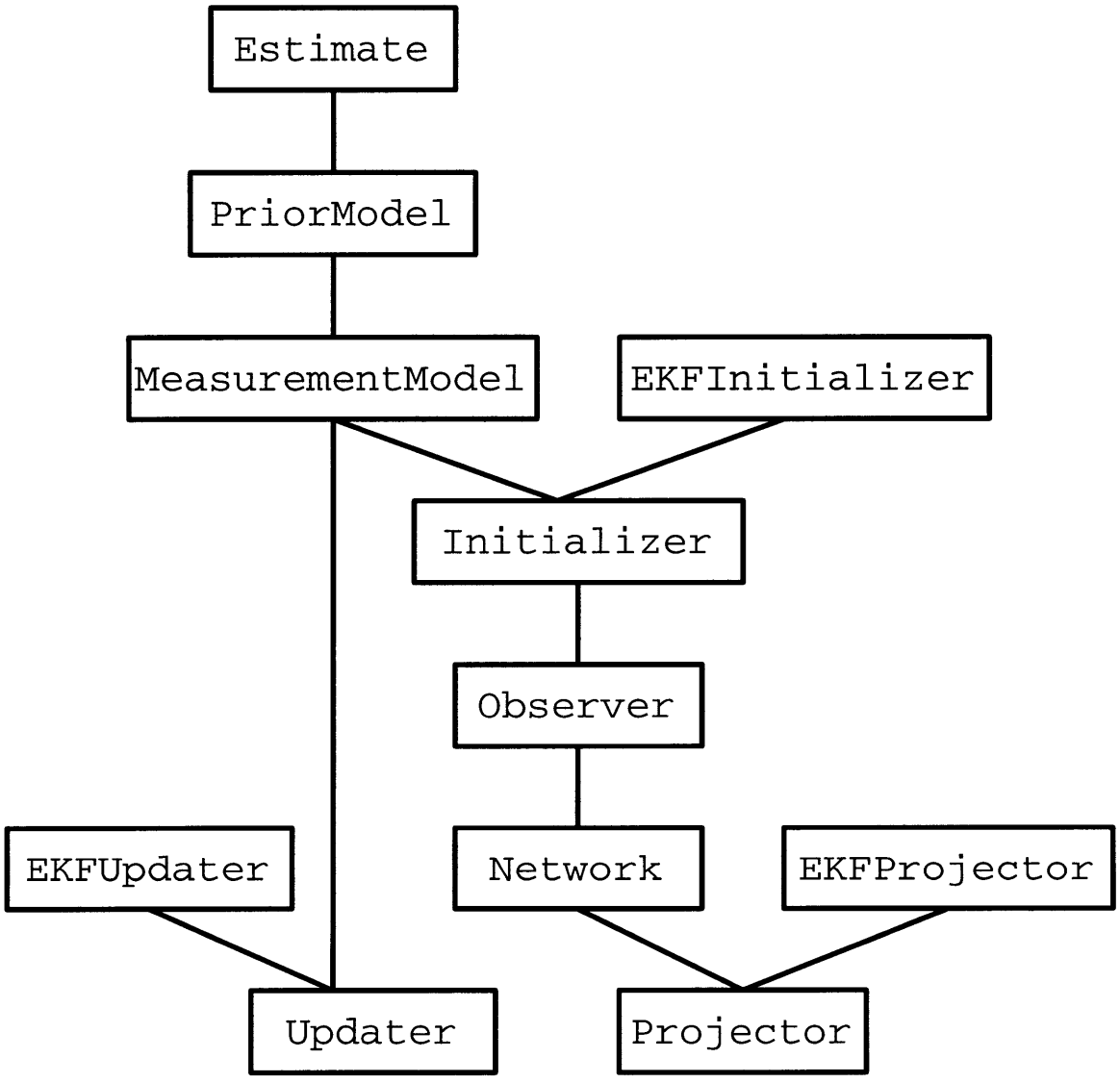


Figure 4-2: The hierarchy of classes in the cmlpr package.

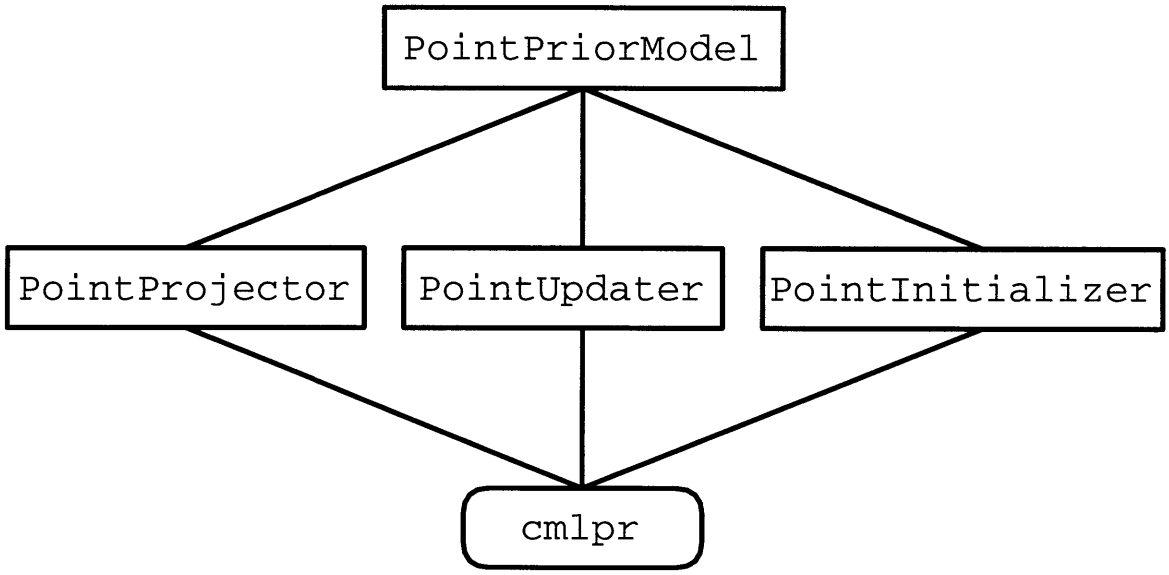


Figure 4-3: The hierarchy of classes in the `cmlpt` package.

The `MeasurementModel` component provides projection, updating, and initialization interfaces for a given measurement process. Channel-level prior information, that is, information about the sensory channel being used, may also be included in a `MeasurementModel`. The `PriorModel` component contains all of the prior information regarding a particular feature (or vehicle) type. It may consist of a number of `MeasurementModels` if more than one sensory modality is utilized. An `Estimate` is a state estimate for a feature or vehicle model. It contains an estimated state vector, an associated error covariance matrix, an estimated likelihood, and information regarding the prior model which produced it.

4.3 Modeling point-like features: the `cmlpt` package

The `cmlpt` package provides complete definitions for modeling point-like environmental features. The hierarchy of classes in the `cmlpt` package is shown in Figure 4-3. Concrete components are provided for each of the three filtering protocols:

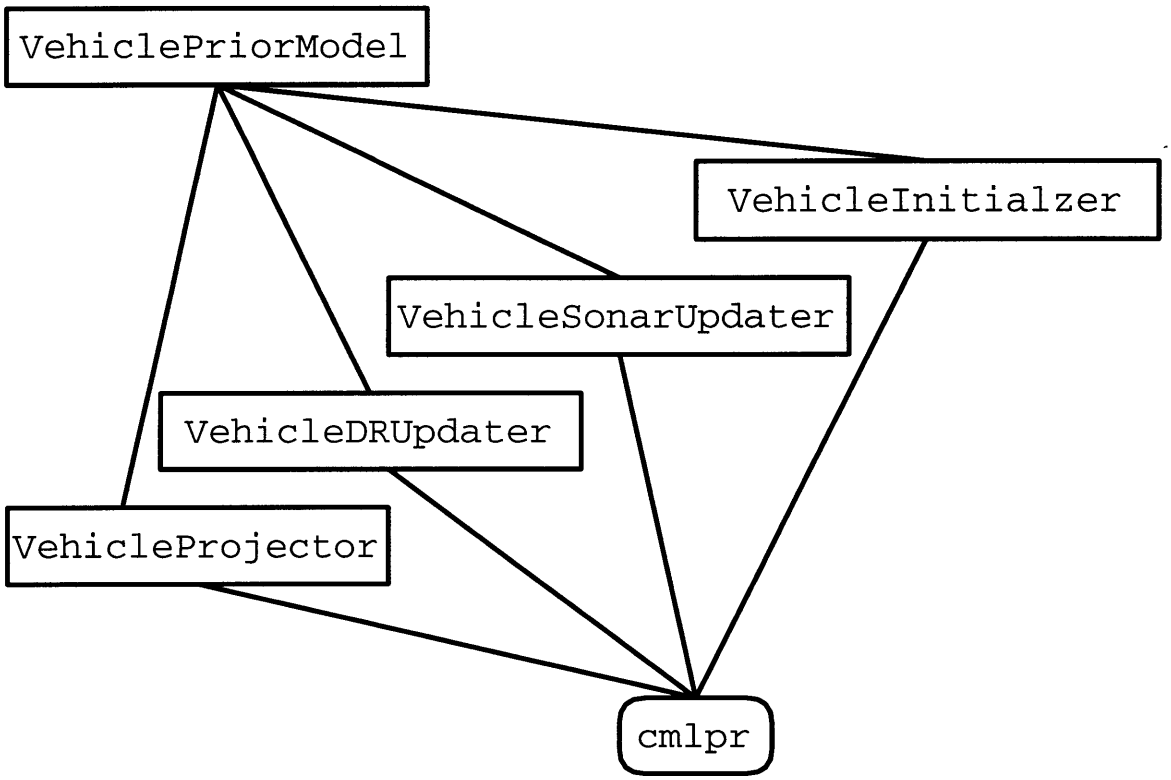


Figure 4-4: The hierarchy of components in the cmlv package.

PointProjector, PointUpdater, and PointInitializer. Full specification of the prior model is handled in PointPriorModel. This includes specification of the Markov network of Projectors, which is in this case a trivial network with a single node. Use of the point model is completely transparent except for the PointPriorModel constructor, which is used to initialize a prior model using particular noise and channel parameters.

4.4 Vehicle modeling: the cmlv package

The cmlv package provides a simple model for survey-class AUVs. The hierarchy of components in the cmlv package is shown in Figure 4-4. Projection and initialization are handled by the VehicleProjector and VehicleInitializer components, respectively. The vehicle model is involved in two separate measurement modali-

ties. The `VehicleDRUpdater` component provides proper updating for dead reckoning measurements (depth, speed, pitch, and yaw). The `VehicleSonarUpdater` component handles estimate updating with sonar returns hypothesized to have some from environmental features. The structure of the vehicle prior model is transparently contained in the `VehiclePriorModel` component. User interface occurs through the constructor of `VehiclePriorModel`.

4.5 Decision-based reasoning: the `cmlhp` package

The `cmlhp` package encapsulates the handling of hypotheses and decision-based reasoning. The `Hypothesis` component represents the trifold relationship of the measurement process: the sensing object, the sensed object, and the measurement itself. Actual objects are handled symbolically using identification flags, removing the dependency of hypothesis formation on prior knowledge representation. The `Decision` component captures the mechanics of hypothesis enumeration for particular navigational events. Asserted decision resolutions are encapsulated in the `Dependency` component, which is used primarily to track decision dependencies for states. Two auxiliary components, `Possibility` and `PossibilityStack`, encapsulate the logic of combining and comparing hypotheses. This allows consideration of consistent groups of hypotheses, but is entirely transparent outside the `cmlhp` package. The `Assignment` component handles assignment formation and processing. It also provides for the representation of possible `Assignments` as dependency sets.

4.6 State management: the `cmlst` package

The `cmlst` package handles management of the problem structure, including states, trees, and tracks. The basic structural unit is the `State` component which situates an estimate within the dependency structure of the estimation problem. The `StateTree` component encapsulates the representation of possible causal links between states

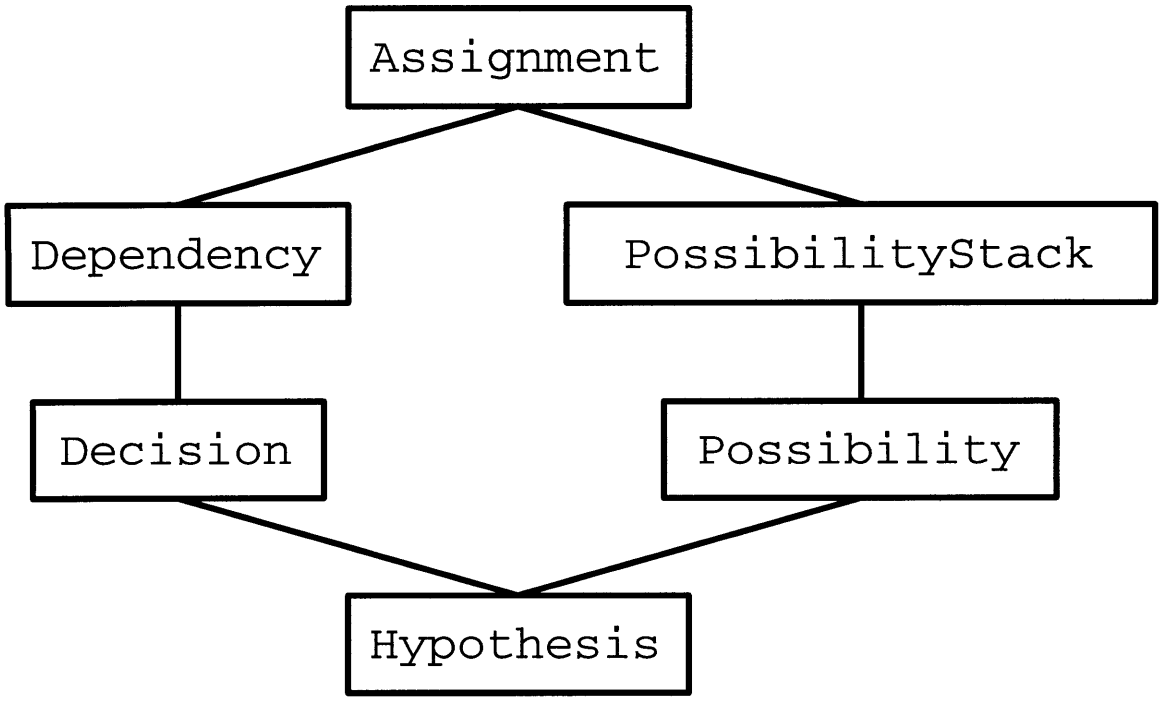


Figure 4-5: The hierarchy of components in the cmlhp package

using a tree structure. It also provides for reliable clean up of invalid states during the process of hypothesis or state rejection. The TrackStep component provides an alternative representation of states as sets of possible estimates for a particular track at given points in the process of IMAN. The TrackBase component provides a unified interface to this dual representation. The Measurement component encapsulates the measurement process and decisions about measurement origin. Handling of groups of measurements is provided by the MeasurementSet component. Feature track management is handled in the Track component, and vehicle track management is handled in the VehicleTrack component. Each of these two components has an associated component, TrackInitiator and VehicleInitiator respectively, to manage decisions and estimates regarding track instantiation. The set of proposed features is managed by the FeatureMap component. The external interface for the algorithm is handled in the Situation component.

A number of process-related aspects of the IMAN algorithm are not readily visible

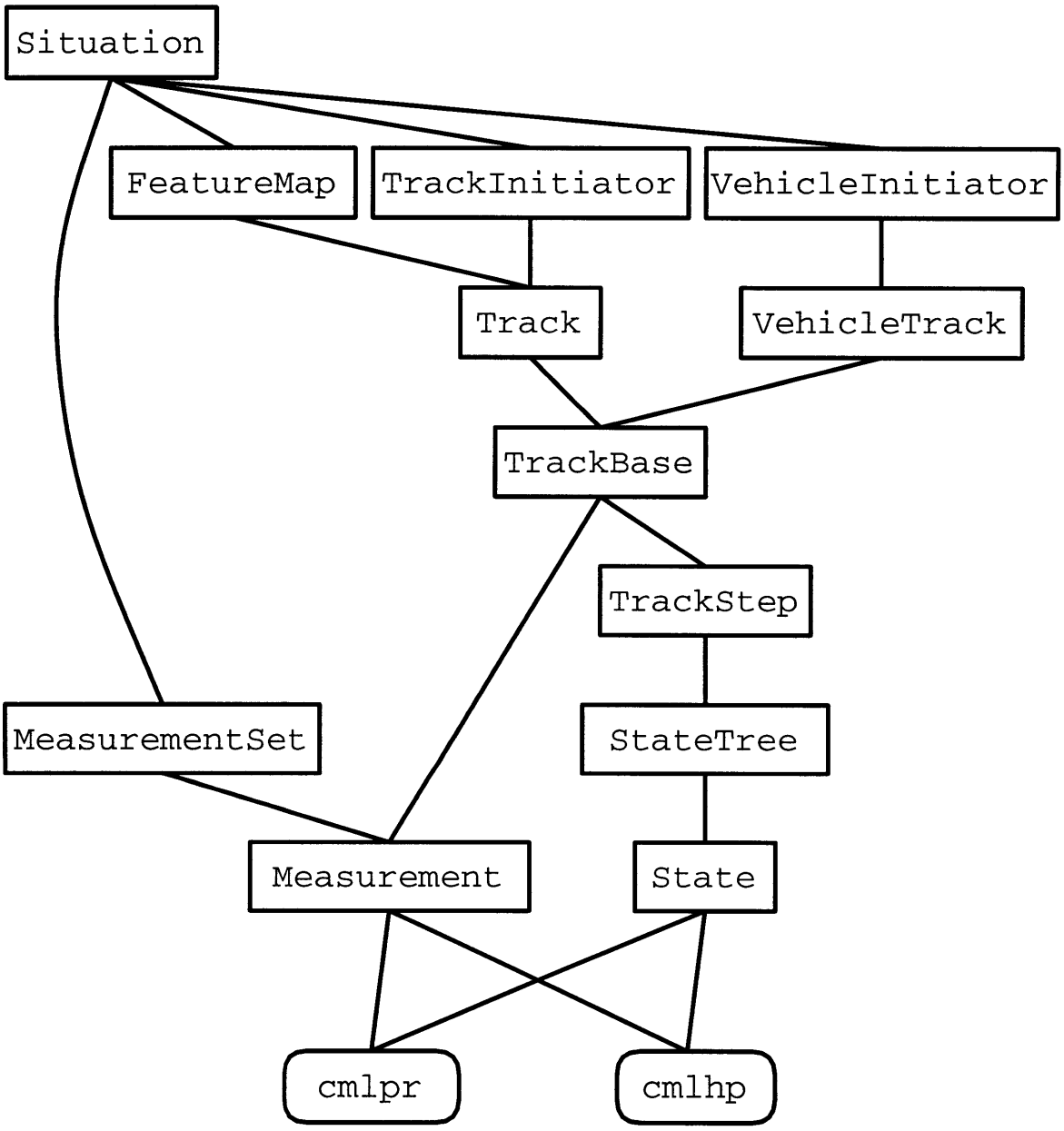


Figure 4-6: The hierarchy of components in the `cmlst` package

within the structurally-based decomposition provided above. Below we examine the location and impact of the possibility enumeration and pruning processes.

4.6.1 Possibility enumeration

Possibility enumeration is the formation of competing hypotheses to offer explanations regarding a navigational event. Data association uncertainties about measurement events are of a small number of types: match, miss, spurious measurement, and new feature hypotheses. However, gating and model-based possibility restrictions, forms of complexity management, will inform the particular choice of hypotheses formed. Currently, the enumeration logic for forming hypotheses is handled in the `Track` component. As more refined feature ontologies and measurement models are developed, this process will need to migrate to the `cmlpr` package so that it may be more closely aligned with the feature definitions. The `Track` component will then require the ability to extract and manipulate this information from the prior model specification.

4.6.2 Pruning

Pruning is the likelihood-based rejection of possibilities based on collected evidence. The primary (and usually only) pruning method used by IMAN is n-backscan assignment pruning, which identified rejected hypotheses based on the current support for competing assignments. Currently, this is handled in the `Situation` component, with some auxiliary functionality (e.g. for cleaning up rejected possibilities) in the `Track`, `VehicleTrack`, `TrackStep`, and `StateTree` components. A more general representation that provides increased encapsulation of the pruning process may be possible and should be considered during enhancements to the current implementation.

4.7 Summary

In this chapter, we have explored the implementation of the integrated mapping and navigation algorithm. Implementation and theoretical development were seen to provide a synergism for understanding the problem structure and for formalizing the concepts of non-separable hybrid estimation and concurrent mapping and localization. The current implementation has been developed with maintainability, modularity, and extensibility is mind. The IMAN algorithm implementation is contained in five software packages. The `cmlpr` package handles prior knowledge representation, implementation of the extended Kalman filter, and the interaction of dynamic and measurement models in reasoning about navigational events. The `cmlpt` package contains the prior model information for a range of point-like feature classes. The `cmlv` package provides simple modeling of survey-class AUVs. Hypotheses, decisions, dependencies, and assignments, that is, the structural elements used in reasoning about decisions regarding navigational events, are implemented in the `cmlhp` package. Finally, the `cmlst` package provides management of the structural elements used in estimation, including states, trees, and tracks, as well as the external interface for the algorithm. We have discussed the object-oriented decomposition of the algorithm and its implementation in these five packages. Comments were made about the ease of use of the current implementation (both in operation and development), and suggestions for further refinements were discussed. In the next chapter, we consider the task of complexity management and its theoretical and structural implications.

Chapter 5

Managing Complexity

In this chapter, issues in the management of complexity are examined. All multiple-hypothesis algorithms (i.e. non-separable hybrid estimation problems) must address the cost of considering alternative possibilities. The number of possible cases can (and does) grow extremely quickly. Additionally, the whole project of considering multiple possibilities is undertaken to allow an eventual choice from among those possibilities. Thus, complexity management is closely intertwined with the discrete estimation problem. We begin by considering the role of complexity management in the integrated mapping and navigation algorithm. The impact of complexity management choices is considered, and the basis for choosing among states, hypotheses, and assignments is detailed. We then turn to pruning, or likelihood-based possibility reduction. Pruning is the primary form of complexity management in many multiple-hypothesis tracking implementations. The variety of techniques used in these cases is considered. The application of pruning techniques to integrated mapping and navigation is discussed, valid methods are described, and implementation problems are noted. The increased complexity brought about by the explicit inclusion of vehicle position uncertainty demands more extensive complexity management than pruning can provide. We identify track number as a driving term in the proliferation of possibilities. A number of track initiation techniques are considered, including a delayed

track initiation strategy which significantly reduces complexity in a variety of circumstances. We also consider complexity that arises through track interaction and data association ambiguity. This subdominant effect limits IMAN performance once track initiation has been addressed. Potential solutions to the track interaction problem are outlined.

5.1 Managing algorithmic complexity

Uncertainty gives rise to complexity. In the case of continuous estimation, estimators are typically described in terms of a probability density function (PDF) for the state variables. The PDF varies over the entire state space. This is in general too complex for solution [97, 99]. In the case of linear Gaussian systems, the PDF is fully described by its mean and covariance.[1] For systems which are nonlinear and/or non-Gaussian, this approximation is not exact. In such cases, the first two moments of the posterior density do not provide a full characterization and the transformation of densities may be inaccurate. Despite this, linearization and the assumption of Gaussianity are common to render the problem computable.[2]

5.1.1 The need for complexity management

In a similar way, discrete estimation problems, in which multiple possible events are considered within a probabilistic framework, need simplifying assumptions to ensure computability. However, convenient descriptions of multiple possibilities are not compactly describable for typical cases. This has two primary results. First, individual possibilities must be represented separately, requiring extensive storage and computational resources as the number of possibilities increases. Second, some of these

[1] In the Kalman filter, this information is provided by the estimate and the estimated error covariance. It can be shown that these values are in fact the mean and covariance of the posterior density $p(x|z)$ [4].

[2] See Section 3.4 for a more extensive discussion of modeling errors in continuous estimation.

possibilities must be rejected for typical problems to remain computable. Complexity management is the coordinated project of possibility rejection. The reduction in proposed possibilities that complexity management engenders provides two benefits. The first is of course that the algorithm is more likely to remain computable. The second is that identifying a single estimate of vehicle or feature state becomes easier. Multiple possibilities can be thought of as modeling flexibility. There is a conflict between this modeling flexibility and computability. The goal of complexity management is to handle this trade-off in the most appropriate way, maintaining enough modeling flexibility to capture the phenomenology of the sensor measurements while rendering the algorithm computable.

5.1.2 The effect of choosing among possibilities

There are a number of ways to make choices about which discrete possibilities to retain and which to reject. Choices can be made concerning the basis for accepting or rejecting possibilities. There are two basic cases. First, possibilities may be accepted or rejected based on the estimated likelihood that they in fact represent the state of the world (or the best model of the world). In this case, modelable possibilities are evaluated as to the likelihood that the navigational events they depend on have occurred. Second, the types of possibilities considered may be limited based on some set of navigational events prescribed within prior models for the system. In this case, certain possibilities might or might not be considered based on the models used to describe the system. Choices can also be based on a number of structural constructs within the IMAN algorithm, such as states, hypotheses, and assignments.

5.1.3 Choosing and rejecting states

States may at first seem a likely candidate for complexity management. They form the basic unit of likelihood calculation. The likelihoods of hypotheses and assignments are computed (primarily) in terms of the states which support them. A problem

arises, however, in maintaining consistency throughout the estimation problem. The impact of individual states on the web of interdependencies is difficult to gauge without exhaustive consideration of other states. The dependencies of various estimates are described by the decision resolutions that they require. This makes hypotheses and decisions a more natural (and efficient) choice for likelihood-based possibility rejection. On the other hand, states can provide a great deal of control during possibility enumeration. While possibility enumeration regarding navigational events is also more naturally a function of decisions regard those events, state-based possibility restriction, or screening, can impact the effective complexity engendered by decision enumeration, particularly during enumeration of feature track trees.

Screening is the model-based restriction of possibilities.

For example, gating is essentially a state-based method for enumerating match hypotheses regarding measurements. In summary, states are not a good candidate for likelihood-based possibility rejection. They can, however, be instrumental in restricting possibilities during track enumeration, particularly when multiple interacting models are involved, due to their strong links to the prior dynamic and measurement models.

5.1.4 Hypotheses and decisions

Hypotheses are the basis for possibility enumeration. Decisions enumerate the set of possibilities for specific navigational events by forming hypotheses. Thus, hypotheses are in some sense the 'correct' unit for likelihood-based possibility rejection. However, care must be taken in specifying hypotheses for rejection, because the hypotheses are not characterized in terms of their role in the dependency web of the system. Thus, while hypotheses should be the subject of likelihood-based methods, choices among hypotheses must be made at a more global level. Hypotheses are also a primary avenue

for screening during the enumeration of decisions, particularly when measurement models are involved.

5.1.5 Choosing and rejecting assignments

Assignments are in essence descriptions of the global dependency web, capturing the possible global occurrences. Thus assignments are the appropriate unit by which to make choices among hypotheses, despite the fact that their likelihood is largely derived from the state enumeration process. Also, as they are an enumeration of possible global occurrences given the set of hypotheses and decisions, there is no way to restrict possible assignment formation without the potential for rejecting validly enumerated occurrences out of hand.

Pruning is the likelihood-based rejection of possibilities.

In some sense, states, hypotheses, and assignments form a continuum from local specification toward global specification. Pruning requires a global perspective to account for the dependency web of the system of estimates, and so should be based on assignment likelihoods. Likelihood-based possibility rejection choices made at the hypothesis and state levels ignore this dependency structure and so run the risk of removing necessary states or hypotheses, resulting in system-level inconsistencies (e.g. a rejection of the vehicle track). The process of possibility enumeration is not, in contrast, teleological. The basis for possibility enumeration lies in the prior model representation for the system. Thus states, as the most direct interface to the vehicle and feature prior models, are the proper substrate for screening. Additional restrictions may arise in a less local sense due to models of the channel or overall environment. Such restrictions, which are based on prior measurement models, are properly applied during hypothesis instantiation. Carrying out screening at an

inappropriate level can lead to inconsistent treatment of the phenomenology when model-specific possibilities are inappropriately rejected.

5.2 Pruning

Pruning is the likelihood-based rejection of discrete estimation elements. In this section, we examine a number of methods for pruning which have been developed for use in multiple-hypothesis tracking. The application of similar techniques to integrated mapping and navigation is then considered. Key lessons from pruning implementations are described, and some potential avenues for further development in likelihood-based possibility rejection for IMAN are discussed.

5.2.1 Pruning in multiple-hypothesis tracking

There are a number of structural difference between multiple-hypothesis tracking and IMAN. Since a single (exact) possible vehicle is considered, dependencies are not spread through the process of vehicle updating. This has two major effects. First, the data association problem can often be reduced by clustering interacting features together. This can reduce the global complexity. Second, tracks can be considered together. In fact, there is typically a single tree of possibilities for each cluster, rather than a tree of possibilities for each target track [74, 63, 23]. The terms hypothesis and assignment are given somewhat different meanings in MHT. Assignments are often referred to as cluster hypotheses. Individual decisions are not tracked other than during hypothesis enumeration. Because of clustering, trees can be pruned independently (i.e. there are no dependency relationships across trees). Additionally each state represents the state estimate for the entire cluster, rather than the individual feature. Thus each MHT state encompasses what would be a complete set of possible resolved tracks in IMAN. The arguments regarding the appropriateness of pruning states, hypotheses, and assignments do not hold for MHT states, which do

contain all dependency information by virtue of clustering. Despite this, much can be learned from MHT pruning techniques. All of these techniques reject some set of MHT states based on their likelihood. The differences are in the way the likelihood threshold for MHT state rejection is chosen.

The ultimate result of decision-making among competing hypotheses is the resolved track.

> A **resolved track** is a track of state estimates in which all decisions have been resolved and is thus a tree with no branchings.

In multiple-hypothesis tracking, the resolution of all tracks at a given time is provided by selecting a single state (per cluster) at that time. One of the primary reasons to delay decision-making is in hope that subsequent evidence will further distinguish possibilities. In general, this becomes less likely the longer decisions are delayed due to limitations in temporal correlation between events. Additionally, allowing decisions to remain unresolved for extended periods can lead to extremes in complexity growth.[3] A natural solution to both of these issues is to force a decision after a pre-determined number of time cycles. This is referred to as n-backscan pruning. At the level of the cluster tree corresponding to n time steps before the present cycle, the most likely state is selected. All alternative states at that tree level are rejected (i.e. removed from the tree and from consideration). Note that n-backstep pruning is not sensitive to the decision ambiguity present, that is, the likelihood ratios involved.

In MHT, the width of the cluster trees, that is, the number of states at a given level, is a good indicator of the algorithmic complexity involved (in terms of storage and computation resources required, for example) [24]. k-best pruning limits tree width by specifying the maximum number of states that can be considered. The k

[3]The extreme of this tactic is running multiple models to account for alternative hypotheses. This technique has been successfully used when the number of competing hypotheses is strictly limited [3], but requires too much storage and computation to be effective for either MHT or IMAN.

states with the greatest likelihoods are retained; all other states for the given tree level are rejected.

Two additional techniques reject states whose likelihood lie below a threshold. Threshold pruning is the rejection of all states whose likelihood is below a specified level. Ratio pruning specifies a likelihood ratio. Each state is compared to the state with the maximum likelihood. Any state whose likelihood ratio lies below the specified ratio threshold is rejected. Efficient computational routines exist for enumerating only the most likely assignments [78, 70].

5.2.2 Pruning in integrated mapping and navigation

As noted above, there are a number of structural differences between MHT and IMAN which complicate the application of pruning. IMAN states have no global sense of the dependencies of the system; they only retain information about their individual dependencies. Trees are used to structure estimates on a track level, rather than globally (or by cluster). Discrete vehicle uncertainty prevents clustering and engenders a complex web of dependencies across the system. Because of this, more than states need to be considered during pruning. We have identified assignments as the appropriate level of abstraction for reasoning about pruning. The results of assignment-based pruning are the retention or rejection of specific hypotheses. This global take on pruning is required to ensure that system consistency is maintained.

n-backscan assignment pruning

The primary technique used for likelihood-based possibility rejection in integrated mapping and navigation is n-backscan assignment pruning. The decisions about navigational events n time steps before the present time cycle k are compiled,

$$\Delta_{k-n} = \{{}_o\delta_{k-n}\}_o . \tag{5.1}$$

The set of assignments which provide consistent, exhaustive resolution of this set of decisions

$$\Omega_{k-n|k} = \left\{ {}_r\omega_{k-n|k} \right\}, \tag{5.2}$$

is calculated, where the time dependency $k-n|k$ indicates an assignment regarding the decisions formed at time $k-n$ based on the system structure at time k. Each of these assignments can be represented as a dependency set,

$${}_r\omega_{k-n|k} \twoheadrightarrow D_{{}_r\omega_{k-n|k}}. \tag{5.3}$$

The vehicle takes part in all decisions, so the vehicle track tree represents a set of states based on the global assignments. The set of leaf states in the vehicle tree is

$${}^u X_k = \left\{ {}^u_w x_k \right\}_w. \tag{5.4}$$

The support for each assignment can then be calculated as the sum of the likelihoods of the states whose dependency sets are compatible with the dependency set representation of the assignment,

$$p\left({}_r\omega_{k-n|k}\right) = \sum_w p\left({}_w x_k\right) \quad \forall\, w \left| C_D\left(D_{{}_w x_k}, D_{{}_r\omega_{k-n|k}}\right) = \text{true}\,. \right. \tag{5.5}$$

The assignments are then ranked according to their likelihood. The likelihood of the most likely assignment is taken as a cutoff value. Assignments whose likelihoods are less than the cutoff value are rejected. The selected assignment is denoted ${}^+\omega_{k-n|k}$. $\Theta_{k-1|k}$ is the set of all hypotheses for the unresolved decisions formed at time $k-n$. $\Theta^+_{k-n|k}$ is the set of all hypotheses used by assignment ${}^+\omega_{k-n|k}$. The set of rejected

hypotheses is then the difference between these sets

$$\Theta^{-}_{k-n|k} = \Theta_{k-n|k} - \Theta^{+}_{k-n|k}. \tag{5.6}$$

Once a set of rejected hypotheses has been obtained, the actual pruning can take place. Each state (on every track tree) that depends on a rejected hypothesis, that is, has the hypothesis as the resolution of any decision in its dependency set, is removed along with all causally related states (i.e. child states in its track tree).[4] Due to the removal of states, additional hypotheses may be rejected (or asserted) as the support for hypotheses across all the decisions is altered. For this reason, a recursive clean up of invalidated hypotheses is necessary. After each removal of a set of rejected hypotheses, all hypotheses and decisions are examined for support. Any hypothesis that has no remaining support is rejected. Also, if a decision is resolved (i.e. if it has a single remaining hypothesis) as a result of hypothesis removal, its remaining hypothesis is asserted. Any decision containing an asserted hypothesis must reject any competing hypotheses, and therefore resolve to the asserted hypothesis. This process in continued until there are no additional rejected hypotheses.

In MHT algorithms, n-backscan pruning is usually implemented for $n = 2$ or 3 [74, 57, 17], but useful delayed decision-making has been shown with $n = 1$ [29]. Due to the increased complexity of the IMAN algorithm, most of the experiments for this thesis use n-backscan pruning with $n = 1$.

State-based likelihood pruning

One of the limitations of assignment-based pruning is that the number of hypotheses and states is not directly controlled. Threshold and ratio pruning were implemented to prune states on a tree-by-tree (that is, a track-by-track) basis. In all cases, significant failure rates were encountered in which the system became inconsistent. System

[4]Note that these child states necessarily depend on the rejected hypothesis as well.

inconsistency occurs when no global possibilities for resolving all tracks exist. Improved, but still unacceptable, failure rates were encountered using state-based pruning of the feature track trees, but not the vehicle track tree. This emphasizes the role of assignments as the appropriate basis for likelihood-based possibility rejection. System consistency simply cannot be guaranteed without considering the global impact of dependencies.

Decision-making

One possible extension to the pruning techniques implemented in this thesis is to provide more reasoning about decisions themselves. It may be possible to implement threshold or ratio pruning to decisions while maintaining system consistency. This is because hypotheses can be rejected singly, and the dependency effects of the rejection can be propagated before continuing the pruning process. Such a method would provide greater control over the decision-making process, but would entail additional book-keeping to ensure system consistency.

5.3 Track initiation

While pruning can be effective in limiting complexity growth, it is insufficient to render IMAN computable. For this reason, model-based possibility restriction, or screening, is necessary. The dominant term in algorithmic complexity is the number of proposed tracks. In this section, we discuss the mechanisms by which feature track ordinality drives algorithmic complexity and consider a number of approaches to complexity management that focus on screening strategies.

5.3.1 Track number and complexity

In multiple-hypothesis tracking, feature tracks become interdependent only when they are part of the same cluster, that is, when they are mutually involved in ambiguous

data association decisions. Clusters can be handled independently, so complexity growth is dominated by cluster size rather than global feature ordinality. The enumeration of multiple possible vehicle states in integrated mapping and navigation destroys any chance of clustering tracks. Since vehicle states are enumerated on the basis of assignments (rather than individual hypotheses), vehicle states take part in all navigational event decisions (i.e. all feature disposition and measurement origin decisions). Because of this, vehicle state enumeration is very sensitive to the number of possible assignments. Independent decisions combine combinatorially during assignment formation, resulting in a growth of the number of vehicle states in proportion with the product of the assignment ordinality over all 'clusters', or independent decision groups. Thus, track formation itself can radically affect algorithmic complexity.

5.3.2 Basic track initiation techniques

Since track initiation is so vital in determining algorithmic complexity, it is the focus of our model-based possibility reduction efforts. The problem is as follows. As measurements are taken, match hypotheses are considered. Match hypotheses involve existing feature tracks and are closely tied to the prior models of features and measurements. Once this state-dependent process (match hypothesis generation) is complete, additional hypotheses are enumerated. The basis of these additional hypotheses is channel-level modeling in the measurement models. A channel is the environmental space within which a given sensing modality operates. Channel-level modeling includes prior models of feature density, probability of detection, probability of false alarm, and the logical process by which non-state-dependent possibility enumeration occurs. Track initiation is the instantiation of a new track based on the possibility that a given measurement (or set of measurements) has come from a previously unmodeled feature. Here we are concerned with the question of when it is appropriate to entertain a new feature hypothesis. Three basic track initiation

Table 5.1: Channel parameters used in the comparison of track initiation techniques.

Parameter		Value
Probability of detection	P_D	1
Probability of false alarm	P_F	0
Probability of correct match given detection	$P_{M\|D}$	1

techniques are detailed below: (1) an obvious approach, (2) a more clever brute force approach, and (3) an approach with restricted hypothesis formation. We compare their performance for a simple example. We assume that the environment consists of n distinct targets. By distinct, we mean that gating is effective in resolving all data association ambiguity.[5] Table 5.1 lists additional channel-level parameters assumed as part of the prior measurement model. No pruning is performed to illustrate the extreme complexity which track initiation can engender. Algorithmic complexity will be measured by the number of possible vehicle states w which are considered at each time step.

Obvious

Brute force methods entertain the possibility that each measurement might come from a new feature. An obvious implementation is to initiate a new track for each combination of possible vehicle state and measurement. Additionally, the possibility that each measurement is spurious is also entertained.

During the first time cycle, each measurement origin decision has two hypotheses: the measurement is spurious and the measurement comes from a new feature. The decisions are independent (from our assumption of distinct features), so the number

[5]For this case, where data association ambiguity is not a factor, a multiple hypothesis approach is unnecessary. The example is useful for comparing track initiation techniques, however, because it highlights complexity growth even in the absence of feature interaction, and thus independent of state-based hypothesis enumeration.

of possible assignments is

$$N\left(\Omega_1\right)_{\text{ob}} = 2^n. \tag{5.7}$$

There are no complications in matching, so the number of possible vehicle states after the first measurement is

$${}_{\text{ob}}w_1 = 2^n. \tag{5.8}$$

For each subsequent measurement, n measurements are received. Each measurements gates only with the 'correct' track(s). Each measurement has three possible explanations: it is spurious, it matches one of the existing proposed features, or it represents a new feature. The number of possible vehicle states grows as follows:

$${}_{\text{ob}}w_{k+1} = {}_{\text{ob}}w_k \left({}_{\text{ob}}w_k + 2\right)^n. \tag{5.9}$$

Clever brute force

A clever brute-force implementation reduces the number of tracks considered by instantiating a single new track for each measurement received. Each of these new tracks will have a number of possible initial states equal to the number of possible vehicle states at the time of track instantiation. While the initial measurement proceeds identically to the obvious brute force implementation,

$${}_{\text{cl}}w_1 = 2^n, \tag{5.10}$$

the growth of possible vehicle states is significantly reduced,

$${}_{\text{cl}}w_{k+1} = {}_{\text{cl}}w_k \left(3\right)^n. \tag{5.11}$$

Restricted

Finding a new feature where a known feature exists and missing the known feature is a much less likely occurrence than matching the known feature. Using this reasoning, we can restrict hypothesis formation in the following way. If a measurement matches a single feature and that feature is matched only by the measurement in question, assume that the match hypothesis is correct and do not enumerate a miss hypothesis for the feature and spurious and new feature hypotheses for the measurement. In other cases, where some ambiguity regarding the data association is readily apparent, proceed with the full enumeration. For this example, the probability of detection and probability of match given detection ensure that at most a single feature track will be proposed for each feature. The initial step is the same,

$$_{\mathrm{re}}w_1 = 2^n, \tag{5.12}$$

but further complexity growth is eliminated for the case posed here,

$$_{\mathrm{re}}w_{k+1} = {}_{\mathrm{re}}w_k. \tag{5.13}$$

This method of restricting possibility enumeration for unambiguous matches is used by for integrated mapping and navigation. Any exceptions are explicitly noted.

5.3.3 Delayed track initiation

While the restricted hypothesis track initiation strategy addresses the growth of complexity once match hypotheses become available, it does little to address the complexity inherent in the initial step. A delayed track initiation scheme is used so that independent decisions about track instantiation can be made independently (i.e. without the combinatorial effect that occurs when the vehicle track depends on the track initiation reasoning process). When a measurement may be spurious or representa-

tive of a new feature (e.g. under the restrictive hypothesis enumeration technique, in ambiguous data association situations), a single hypothesis encompassing these possibilities is made for the purposes of possibility enumeration. The measurement and possible vehicle states are then combined to form a possible track initiation estimate. If three mutually-gating possible track initiation estimates are formed within five time cycles, a new feature track based on these three track initiation estimated is instantiated. This technique suboptimally combines the different possible vehicle states to form the track initiation estimate. However, vehicle state estimates are weighted by their likelihood, and, in any case, the variety of possible vehicle states should not vary too greatly, so this is a fairly good approximation. For this example, there is no problem with conflating possible vehicle states. Throughout time, there is a single possible vehicle state,

$$_{\text{de}}w_k = 1 \quad \forall\, k. \tag{5.14}$$

n vehicle tracks are instantiated after the third time cycle, but the feature track initiation decisions are separated from the vehicle track. The restricted hypothesis technique described above maintains a constant level of complexity once the feature tracks are instantiated. This is a case where screening enhances the decision-making process even when teleological techniques (based on global posteriors) fail.

5.3.4 Comparison of techniques

Figure 5-1 shows the number of possible vehicle states proposed by the four track initiation techniques described above of for the case of three features ($n = 3$). For comparison, the total number of states in an equivalent MHT formulation (that is, without vehicle uncertainty) with no model-based possibility restriction is also shown. Table 5.2 lists the number of states generated by each approach in this example.

Track initiation is, of course, not the whole answer to complexity management.

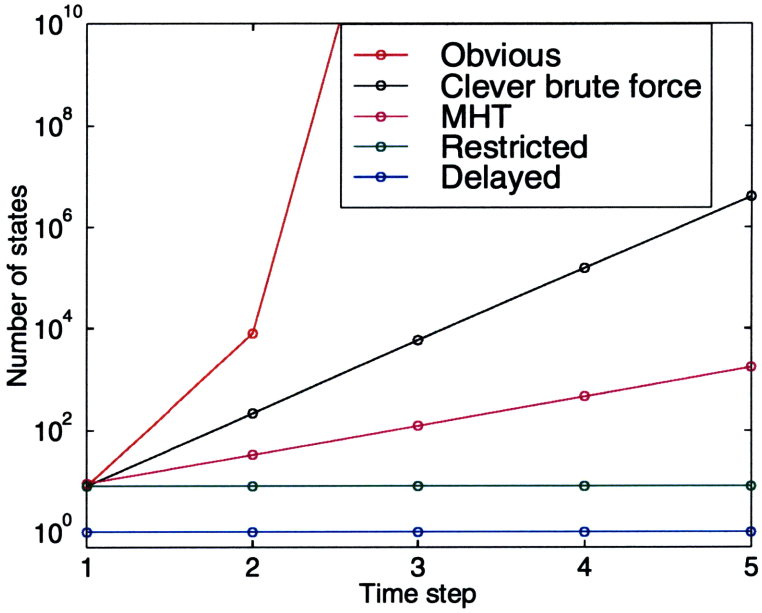


Figure 5-1: Comparison of track initiation techniques.

Table 5.2: Comparison of track initiation techniques.

Technique	$k = 1$	2	3	4	5
obvious	8	8000	4×10^{15}	2×10^{62}	2×10^{249}
clever brute force	8	216	5832	157464	4×10^{6}
restricted	8	8	8	8	8
delayed	1	1	1	1	1

The IMAN implementation makes use of both the delayed track initiation technique and the restricted hypothesis formation scheme detailed above, and yet, as will be shown in Chapter 7, there are a significant number of cases when computational complexity overwhelms the algorithm. The track initiation techniques developed for IMAN have significantly extended its operability (from about 3 time cycles to tens of cycles). Robustification of the algorithm to additional sources of complexity growth remains a research topic. Because of the introduction of vehicle uncertainty, ambiguous cases tend to cause doubly exponential increases in complexity (as with the obvious track initiation implementation considered above). This order of complexity is simply too great to handle through likelihood-based possibility rejection (without resorting to a zero-backscan algorithm, which is a poor alternative). The answer lies in model-based possibility restriction.

5.4 Track interaction

The primary remaining cause of complexity growth is track interaction. Track interaction occurs when measurements gate with more than one feature. The resulting complexity quickly overwhelms the IMAN algorithm, even with aggressive pruning. While fully addressing this problem is beyond the scope of this thesis, a number of potential approaches are worth observing, both to motivate research on this issue and to demonstrate the reasoning that goes into model-based possibility restriction. We suggest two possible approaches to track interaction ambiguity: default assumptions and interactive gating. Default assumptions are not currently implemented but offer a number of attractive possibilities. The default hypotheses would be to assume that a feature is missed or that a measurement is spurious. When complexity becomes an issue, the default hypotheses can be asserted for some or all of the unresolved decisions. This would also improve the robustness of the algorithm in terms of system consistency. The default hypothesis could provide a fallback position if things don't

work out for some reason (such as occurs in state-based pruning). Implementation of such a strategy is non-trivial, as pruning of default strategies would need to be handled in a special way. Additionally, some monitoring should be available to indicate how often the default hypotheses were enforced. Overuse of this strategy would indicate a more fundamental problem which needs to be overcome.

Iterative gating is the process of iteratively reducing the gating threshold (up to a point) until a single match hypothesis is obtained. This could reduce complexity considerably, particularly when observing relatively new features (which as a consequence have larger uncertainties, and hence larger gates). Again, implementation is not trivial, as the ability to make tentative possibility enumerations would be required.

5.5 Summary

In this chapter, we have explored complexity management for integrated mapping and navigation. We began by discussing the need for complexity management in any non-trivial discrete estimation problem. The method by which continuous estimation avoids the issue of complexity, assuming the sufficiency of a Gaussian representation, was noted. The basis for managing complexity was discussed and two types of complexity management were identified: (1) likelihood-based possibility rejection, or pruning, and (2) model-based possibility restriction, or screening. The structure of the IMAN algorithm was revisited. Pruning, being grounded in the teleological concept of posterior likelihood, is appropriately applied to assignments, which contain information about the global impact of dependencies. State-based pruning can lead to system inconsistency. On the other hand, states and hypotheses are the appropriate substrates for screening due to their proximal involvement with the prior models for feature behavior and sensor measurement. Next, pruning was discussed. Several techniques for pruning in multiple-hypothesis tracking were described. The application

of these techniques to the structurally different problem of IMAN was discussed and n-backscan assignment pruning was identified as the most useful pruning strategy for integrated mapping and navigation. Next, screening was considered in more depth. Track ordinality was identified as the dominant term in algorithmic complexity. A number of track initiation techniques were developed and compared. The integrated mapping and navigation algorithm uses a delayed track initiation technique along with a restricted hypothesis generation scheme. Finally, the outstanding problem of track interaction was considered, and several possible approaches were discussed. In the next chapter, we begin the performance analysis of the algorithms developed in this thesis by considering the constraints of the concurrent mapping and localization problem.

Chapter 6

Analysis of Concurrent Mapping and Localization

In this chapter, we explore the role of uncertainty in the problem of concurrent mapping and localization. As we saw in Chapter 2, existing CML implementations have been to some degree *ad hoc*. This is in part because a full characterization of the sources of error and their interactions has not been available. Here, we examine how model uncertainty, navigation uncertainty, and data association uncertainty structure the problem of concurrent mapping and localization. We start with a consideration of how model uncertainty and errors in the navigation system affect the navigational performance of the vehicle during dead reckoning. This illustrates (1) how vehicle error grows as the vehicle becomes lost and (2) what factors dominate the growth of vehicle uncertainty. Next, we consider environmental features as sources of information. The amount of information provided by measurements is quantified and examined within a common operational paradigm. The potential benefits of using features to aid in navigation are explored, and the impact of such measurements on the structure of the vehicle position error is examined. Finally, we discuss the effects of data association errors and other errors in discrete estimation. The types of data association errors and their effects are cataloged. The impact of clutter and failed

detections on feature estimation is considered. The spatial distinguishability of features and interaction between feature models is considered for the case of point-like features. We conclude the chapter with a summary of major points regarding how the major sources of error structure the problem of CML and what methods can be used to recover from these uncertainties.

6.1 Vehicle position error growth

Vehicle navigation is difficult primarily due to the growth of position error and uncertainty. In this section, we examine the mechanisms of error growth. The primary factors affecting this error growth are (1) model uncertainty, which is modeled by process noise in the vehicle dynamic model, and (2) navigational system uncertainty, that is, uncertainty that arises from navigational system sensors, such as dead-reckoning sensors and inertial navigation systems. Through this development, the nature of the problem should become clearer, as will the potential impact of measurement information from environmental features.

The growth of position error is often characterized for a given navigational system in terms of percentage of distance traveled. Although this is a somewhat simplistic interpretation, it is a fairly accurate description of error growth for a dead reckoning system, that is, a system without access to external position references (either absolute or relative). We examine how taking measurements alters this process, regardless of the actual measurement information obtained. Throughout this development, we consider straight-line transiting of the vehicle, a common operational paradigm [45]. The vehicle travels along a straight path, as shown in Figure 6-1. The direction of vehicle motion is referred to as the pathwise coordinate. The transverse coordinate measures distances orthogonal to the vehicle path. For clarity, a two-dimensional dynamic model is used. Extension to three dimensions is straightforward.

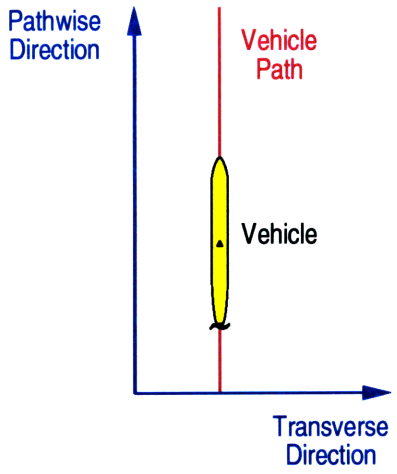


Figure 6-1: Transiting operational paradigm. The vehicle moves in a straight line. The direction of vehicle motion is the pathwise direction. The orthogonal coordinate is the transverse direction.

6.1.1 Model uncertainty

First, let us examine the case where the navigational system used by the vehicle is exact. In this case, the only source of uncertainty is the process noise of the dynamic model, which represents model uncertainty. Vehicle dynamics are based on a known speed u and heading ψ. The vehicle travels in a straight line with coordinates in the pathwise direction ${}^{p}x$ and the transverse direction ${}^{t}x$:

$$\begin{bmatrix} {}^{p}x_{k+1} \\ {}^{t}x_{k+1} \end{bmatrix} = \begin{bmatrix} {}^{p}x_{k} + u\cos\psi \\ {}^{t}x_{k} + u\sin\psi \end{bmatrix} + w_{k}, \tag{6.1}$$

where w_k is an additive zero-mean white Gaussian noise process. The Jacobian of the dynamic function with respect to the vehicle state is the identity matrix,

$$F = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{6.2}$$

The covariance of the vehicle state estimate is

$$P_k = \begin{bmatrix} {}^{p}P_k & {}^{pt}P_k \\ {}^{pt}P_k & {}^{t}P_k \end{bmatrix}; \tag{6.3}$$

the covariance of the noise process is

$$Q = \begin{bmatrix} {}^{p}\sigma^2 & 0 \\ 0 & {}^{t}\sigma^2 \end{bmatrix} \tag{6.4}$$

and is independent of the initial vehicle state. The error covariance grows according to the dynamic model, that is,

$$P_{k+1} = FP_kF^T + Q = P_k + Q. \tag{6.5}$$

If we assume that the vehicle begins with an accurate estimate of its location, this provides a view of the process of getting lost. The additive process noise accumulates, dominating the vehicle uncertainty. With perfect initial knowledge, the error covariance grows as follows:

$$P_{k+1} = \begin{bmatrix} k\,{}^{p}\sigma^2 & 0 \\ 0 & k\,{}^{t}\sigma^2 \end{bmatrix}. \tag{6.6}$$

The diagonal elements of the error covariance grow linearly in time (and distance). If each time cycle lasts for T seconds, we can define a time

$$t = kT. \tag{6.7}$$

The vehicle travels a distance

$$s = ukT \tag{6.8}$$

in k cycles. The vehicle positional uncertainty in the pathwise ${}^{p}_{x}\sigma$ and transverse ${}^{t}_{x}\sigma$ directions grow with the square root of time or distance traveled.

$$ {}^{p}_{x}\sigma = \frac{{}^{p}\sigma}{T^{1/2}}\sqrt{t} = \frac{{}^{p}\sigma}{(uT)^{1/2}}\sqrt{s}, \tag{6.9} $$

and

$$ {}^{t}_{x}\sigma = \frac{{}^{t}\sigma}{T^{1/2}}\sqrt{t} = \frac{{}^{t}\sigma}{(uT)^{1/2}}\sqrt{s}. \tag{6.10} $$

6.1.2 Navigation system uncertainty

Now we extend this simple model to include uncertainty in the dead-reckoning sensors. Instead of exact navigational information, the vehicle merely has uncertain estimates of speed ${}^{u}x_k$ and heading ${}^{\psi}x_k$. The vehicle dynamic projection equation becomes

$$ \begin{bmatrix} {}^{p}x_{k+1} \\ {}^{t}x_{k+1} \\ {}^{u}x_{k+1} \\ {}^{\psi}x_{k+1} \end{bmatrix} = \begin{bmatrix} {}^{p}x_k + {}^{u}x_k \cos {}^{\psi}x_k \\ {}^{t}x_k + {}^{u}x_k \sin {}^{\psi}x_k \\ {}^{u}x_k \\ {}^{\psi}x_k \end{bmatrix} + w_k, \tag{6.11} $$

with a Jacobian with respect to this state given by

$$ {}_{x}F_k = \begin{bmatrix} 1 & 0 & \cos {}^{\psi}x_k & -{}^{u}x_k \sin {}^{\psi}x_k \\ 0 & 1 & \sin {}^{\psi}x_k & {}^{u}x_k \cos {}^{\psi}x_k \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_{x=\hat{x}_k} . \tag{6.12} $$

Note that the vehicle dynamics are now nonlinear, so the dynamic matrix ${}_{x}F_k$ depends on the vehicle state. By definition, the vehicle heading with respect to the pathwise

direction is nominally zero, giving

$$ {}_xF_k = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & {}^u x_k \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_{x=\hat{x}_k} . \tag{6.13} $$

The process noise is a zero-mean white Gaussian noise process with covariance

$$ Q = \begin{bmatrix} {}^p\sigma^2 & 0 & 0 & \\ 0 & {}^t\sigma^2 & 0 & 0 \\ 0 & 0 & {}^u\sigma^2 & 0 \\ 0 & 0 & 0 & {}^\psi\sigma^2 \end{bmatrix} . \tag{6.14} $$

The error covariance matrix is defined as

$$ P_k = \begin{bmatrix} {}^pP_k & {}^{pt}P_k & {}^{pu}P_k & {}^{p\psi}P_k \\ {}^{pt}P_k & {}^tP_k & {}^{tu}P_k & {}^{t\psi}P_k \\ {}^{pu}P_k & {}^{tu}P_k & {}^uP_k & {}^{u\psi}P_k \\ {}^{p\psi}P_k & {}^{t\psi}P_k & {}^{u\psi}P_k & {}^\psi P_k \end{bmatrix} . \tag{6.15} $$

Then the Kalman projection step yields

$$P_{k+1} = \begin{bmatrix} {}^{p}P_k + 2\,{}^{pu}P_k + {}^{u}P_k + {}^{p}\sigma^2 \\ {}^{pt}P_k + {}^{tu}P_k + {}^{u}x_k\left({}^{p\psi}P_k + {}^{u\psi}P_k\right) \\ {}^{pu}P_k + {}^{u}P_k \\ {}^{p\psi}P_k + {}^{u\psi}P_k \end{bmatrix}$$

$$\begin{bmatrix} {}^{pt}P_k + {}^{tu}P_k + {}^{u}x_k\left({}^{p\psi}P_k + {}^{u\psi}P_k\right) & {}^{pu}P_k + {}^{u}P_k & {}^{p\psi}P_k + {}^{u\psi}P_k \\ {}^{t}P_k + 2\,{}^{u}x_k\,{}^{t\psi}P_k + {}^{u}x_k^2\,{}^{\psi}P_k + {}^{t}\sigma^2 & {}^{tu}P_k + {}^{u}x_k\,{}^{u\psi}P_k & {}^{t\psi}P_k + {}^{u}x_k\,{}^{\psi}P_k \\ {}^{tu}P_k + {}^{u}x_k\,{}^{u\psi}P_k & {}^{u}P_k + {}^{u}\sigma^2 & {}^{u\psi}P_k \\ {}^{t\psi}P_k + {}^{u}x_k\,{}^{\psi}P_k & {}^{u\psi}P_k & {}^{\psi}P_k + {}^{\psi}\sigma^2 \end{bmatrix}. \quad (6.16)$$

If we assume that the vehicle begins with perfect navigational information, that is,

$$P_0 = 0, \quad (6.17)$$

then this simplifies to

$$P_{k+1} = \begin{bmatrix} {}^{p}P_k + 2^{pu}P_k + {}^{u}P_k + {}^{p}\sigma^2 & 0 \\ 0 & {}^{t}P_k + 2\,{}^{u}x_k\,{}^{t\psi}P_k + {}^{u}x_k^2\,{}^{\psi}P_k + {}^{t}\sigma^2 \\ {}^{pu}P_k + {}^{u}P_k & 0 \\ 0 & {}^{t\psi}P_k + {}^{u}x_k\,{}^{\psi}P_k \end{bmatrix}$$

$$\begin{bmatrix} {}^{pu}P_k + {}^{u}P_k & 0 \\ 0 & {}^{t\psi}P_k + {}^{u}x_k\,{}^{\psi}P_k \\ {}^{u}P_k + {}^{u}\sigma^2 & 0 \\ 0 & {}^{\psi}P_k + {}^{\psi}\sigma^2 \end{bmatrix}. \quad (6.18)$$

In general, a navigational system provides estimates of the vehicle speed and

heading. This estimation process reaches a steady state, at which point we have ${}^{u}P = {}^{u}\sigma_{ss}^2$ and ${}^{\psi}P = {}^{\psi}\sigma_{ss}^2$. Equation 6.18 indicates that the cross-correlations terms ${}^{pu}P$ and ${}^{t\psi}P$ integrate this steady-state uncertainty over time. The main directional uncertainties ${}^{p}P$ and ${}^{t}P$ have terms which integrate these cross-correlations over time, leading to terms which grow quadratically with time. Substituting time $t = kT$, assuming a steady state speed estimate u_{ss}, and ignoring initial uncertainty, we have

$$P_t = \begin{bmatrix} \frac{\left({}^{p}\sigma^2 + {}^{u}\sigma_{ss}^2\right)}{T} t + \frac{{}^{u}\sigma_{ss}^2}{T^2} t^2 & 0 \\ 0 & \frac{\left({}^{t}\sigma^2 + u_{ss}^2\, {}^{\psi}\sigma_{ss}^2\right)}{T} t + \frac{u_{ss}^2\, {}^{\psi}\sigma_{ss}^2}{T^2} t^2 \\ \frac{{}^{u}\sigma_{ss}^2}{T} t & 0 \\ 0 & \frac{u_{ss}\, {}^{\psi}\sigma_{ss}^2}{T} t \end{bmatrix}$$

$$\begin{bmatrix} \frac{{}^{u}\sigma_{ss}^2}{T} t & 0 \\ 0 & \frac{u^{ss}\, {}^{\psi}\sigma_{ss}^2}{T} t \\ {}^{u}\sigma_{ss}^2 & 0 \\ 0 & {}^{\psi}\sigma_{ss}^2 \end{bmatrix}. \quad (6.19)$$

The position coordinate covariances clearly becomes dominated by the quadratic term as time increases, leading to a linear growth rate for the RMS position error. An additional point to note is that it is the speed estimate uncertainty which leads to error growth in the pathwise direction and the heading estimate uncertainty which leads to error growth in the transverse direction. Consider a noise level defined by

$${}^{u}\sigma_{ss} = u_{ss}\, {}^{\psi}\sigma_{ss} = {}^{dr}\sigma, \quad (6.20)$$

and

$${}^{p}\sigma = {}^{t}\sigma = {}^{m}\sigma. \quad (6.21)$$

Table 6.1: Vehicle position uncertainty growth parameters. These parameters were chosen so that the growth of vehicle position uncertainty in the pathwise and transverse directions is equal. Note that the steady-state heading error is more reasonable for typical navigational systems than the steady-state speed estimate. This is in line with the experience that pathwise error growth is usually greater than error growth in the transverse direction.

Parameter	Symbol	Value
pathwise process noise variance	${}^{p}\sigma$	71 cm
transverse process noise variance	${}^{t}\sigma$	71 cm
steady-state speed	u_{ss}	3 $\frac{\text{m}}{\text{s}}$
steady-state speed estimate variance	${}^{u}\sigma_{ss}$	5.25 $\frac{\text{cm}}{\text{s}}$
steady-state heading estimate variance	${}^{\psi}\sigma_{ss}$	1^{o}

The error growth for the vehicle position coordinates then becomes

$$\frac{{}^{m}\sigma^2}{T}\,t + \frac{{}^{n}\sigma^2}{T}\left(t + \frac{1}{T}\,t^2\right), \tag{6.22}$$

where ${}^{m}\sigma^2$ denotes the level of model uncertainty and ${}^{n}\sigma^2$ denotes the level of navigational system uncertainty.

6.1.3 Quantifying position error growth

Now we are in a position to consider the actual growth of error in the vehicle position estimate, comparing the two cases derived above: (1) the use of exact navigational information and (2) the use of uncertain navigational estimates. This comparison uses the parameters listed in Table 6.1. The navigational system noise levels were chosen so that pathwise and transverse error growth are identical at the nominal vehicle speed. The evolution of the RMS vehicle position estimate error ($\sqrt{{}_{p}P}$ and $\sqrt{{}_{t}P}$) over a ten minute interval is shown in Figure 6-2. At first, the navigational system uncertainty has little effect and error growth is dominated by model uncertainty in the form of the model process noise. In this region, the position error grows with the square root of time or distance traveled. With increasing time, however, the

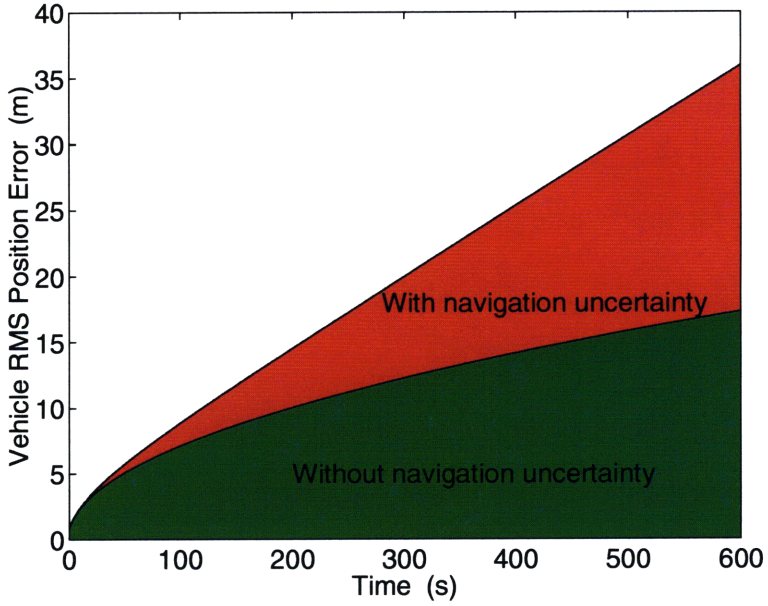


Figure 6-2: Growth of RMS vehicle position error. The lower region shows error growth in the absence of navigational system uncertainty and exhibits error growth with the square root of time or distance traveled. The upper region accounts for navigational system uncertainties and approaches linear error growth with time or distance traveled.

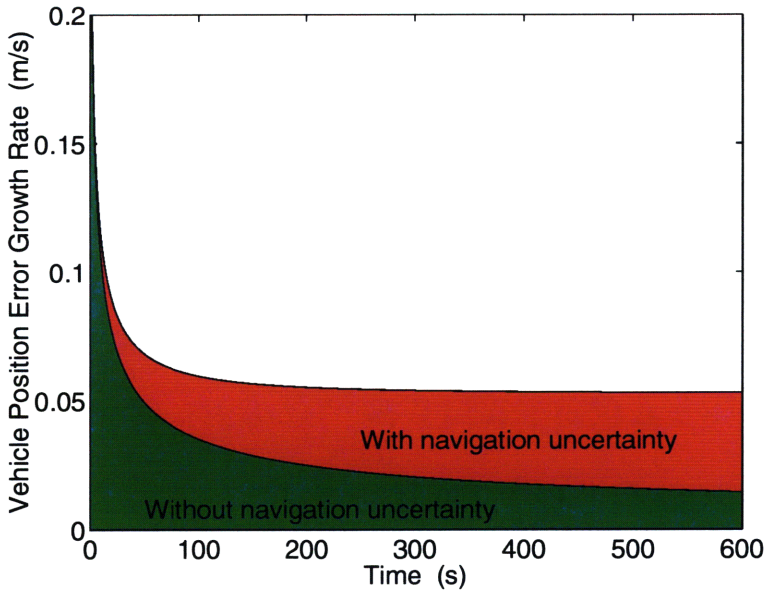


Figure 6-3: Vehicle position error growth rate

navigational system uncertainty comes to dominate error growth. In this region, vehicle position error grows linearly with time or distance traveled. To emphasize this, Figure 6-3 shows the rate of RMS position error growth (i.e. the derivative of the curves in Figure 6-2). In the model-uncertainty-dominated regime, the error growth is strictly decreasing, while in the navigational-uncertainty-dominated regime this growth approaches a constant. This change in regime comes about as the vehicle position estimate becomes highly correlated with the vehicle navigational estimates provided by the navigational system. The change of regime happens quickly for dead-reckoned systems. However, external position measurements, whether absolute or relative, have the effect of reducing the correlation between the vehicle position estimate and the navigational system.

6.2 Features as sources of information

Now that we have a better appreciation for the kinds of errors incurred during navigation, we turn to the measurements. Environmental features can provide measurements

to improve the vehicle position estimate. Since the measurements we consider are relative measurements, they can never reduce the global vehicle position uncertainty. However, they can bound vehicle error growth, both by decoupling the position estimate from navigational system estimates and by providing information about the vehicle's relative position through time. The goals of this section are to quantify the information extracted from relative measurements and to identify the utility of individual features within a basic operational paradigm.

6.2.1 Measurement information

The quantification of measurement information is usually accomplished by determining the Cramér-Rao bound on the estimate variance [51, 71, 80]. There are a number of reasons why this approach is inappropriate in the case of CML[1]. First, the Cramér-Rao bound only holds for linear measurements [51]. In the nonlinear case, the Cramér-Rao bound asymptotically approaches the actual covariance bound as the number of measurements becomes large. In the case of concurrent mapping and localization, the relative measurements typically consist of range and bearing.[2] Additionally, features often are visible for a limited time only, with the result that few measurements are available. Those measurements are also typically representative of different linearization points, as well. For example, if the vehicle is traveling at 6 knots, has a forward sensor range of 300 m, and can interrogate its surroundings using a high-resolution forward-looking array once every 5 seconds, then a feature will be measured at most 20 times before passing from view. During that time, the feature will traverse the entire half-angle of the sonar view. Thus the Cramér-Rao

[1] An alternative quantification of the evidential worth of information sources, the predictive ability measure, has been developed by Chong [19]. This technique focuses more on the worth of information sources in correctly identifying discrete events.

[2] Other measurements are of course possible, including, for example, waveform information. However, the basic information about the relative feature position in inherently represented within the egosphere of the vehicle. Thus, any estimate of feature position involves a transformation from cylindrical or spherical coordinates to Cartesian coordinates. This is a fundamental nonlinearity and should not be dismissed lightly.

bound can not be trusted to represent an accurate limit on estimate covariance. Second, the Cramér-Rao bound is not well-suited for use within a Bayesian framework. The joint probability could be used, but the identification of likelihood (and, more importantly, the separation of likelihood from prior) can be quite difficult [8, 61]. For these reasons, an accurate bound on estimate covariance is not readily computable.

In spite of this, we can use a similar formulation to quantify the amount of information present in measurements about the vehicle location. The measurement model provides a description of the conditional probability density for the measurements given the vehicle and feature states $p(z|x,\xi)$. Using the Fisher information provides an information matrix

$$I_z = -\mathrm{E}_z\left[\nabla^2 \ln p(z|x,\xi)\right], \tag{6.23}$$

where the Lagrangian is taken with respect to x and ξ and the expectation is over z. This provides a quantification of the information in the measurement itself, that is, without regard to any prior information, about the vehicle and feature states. Measurement models are discussed in detail in Sections 3.2.3 and 3.4.1. We assume a measurement model

$$z = h(x,\xi) + v, \tag{6.24}$$

where v is a zero-mean Gaussian white-noise process with covariance R. The Jacobian of the measurement function with respect to the vehicle and feature states is H. The conditional probability density is then normal with mean $h(x,\xi)$ and covariance R,

$$p(z|x,\xi) \sim N(h(x,\xi), R). \tag{6.25}$$

The Fisher information for this measurement is then

$$I_z = H^T R^{-1} H. \tag{6.26}$$

Recalling the measurement model definition from Chapter 3, and combining the vehicle and feature state into a single vector,

$$\begin{bmatrix} {}^n x & {}^e x & {}^\psi x & {}^n \xi & {}^e \xi \end{bmatrix}^T,$$

we specify the measurement Jacobian. To simplify the notation, we define

$$\Delta n = {}^n \xi - {}^n x, \tag{6.27}$$

$$\Delta e = {}^e \xi - {}^e x, \tag{6.28}$$

and

$$r = \sqrt{(\Delta n)^2 + (\Delta e)^2}. \tag{6.29}$$

The measurement Jacobian is

$$H = \begin{bmatrix} \frac{-\Delta n}{r} & \frac{-\Delta e}{r} & 0 & \frac{\Delta n}{r} & \frac{\Delta e}{r} \\ \frac{\Delta e}{r^2} & \frac{-\Delta n}{r^2} & -1 & \frac{-\Delta e}{r^2} & \frac{\Delta n}{r^2} \end{bmatrix}. \tag{6.30}$$

The inverse of the measurement noise covariance is denoted

$$R^{-1} = \begin{bmatrix} I_r & 0 \\ 0 & I_\phi \end{bmatrix}. \tag{6.31}$$

Table 6.2: Information matrix example parameters.

Parameter	Symbol	Value
range information	I_r	4 m^{-2}
bearing information	I_ϕ	1.56e6 rad^{-2}

The information matrix I_z is

$$I_z = \begin{bmatrix} A & B & -A \\ B^T & I_\phi & -B^T \\ -A^T & -B & A \end{bmatrix}, \tag{6.32}$$

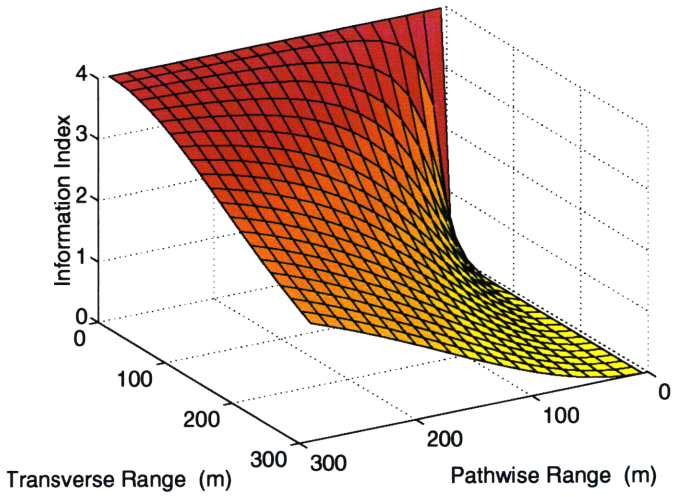
where

$$A = \begin{bmatrix} I_r \frac{(\Delta n)^2}{r^2} + I_\phi \frac{(\Delta e)^2}{r^4} & I_r \frac{\Delta e}{r^2} - I_\phi \frac{\Delta n\, \Delta e}{r^4} \\ I_r \frac{\Delta e}{r^2} - I_\phi \frac{\Delta n\, \Delta e}{r^4} & I_r \frac{(\Delta e)^2}{r^2} + I_\phi \frac{(\Delta n)^2}{r^4} \end{bmatrix} \tag{6.33}$$

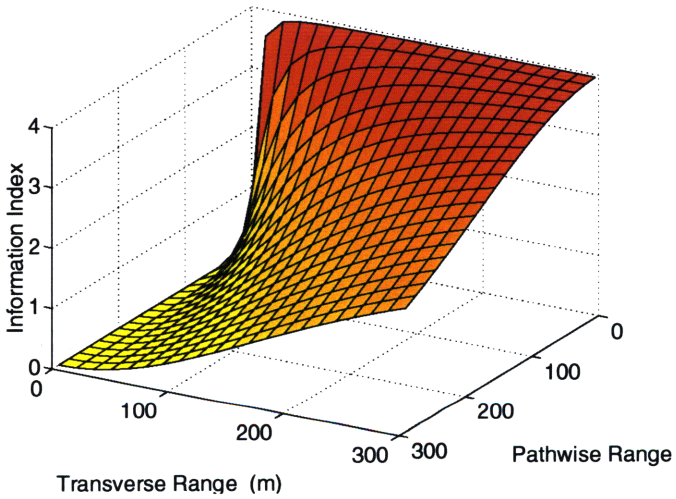
and

$$B = \begin{bmatrix} -I_\phi \frac{\Delta e}{r^2} \\ I_\phi \frac{\Delta n}{r^2} \end{bmatrix}. \tag{6.34}$$

Figure 6-4 shows the variation of the first two diagonal elements of this matrix, corresponding to the information about the north and east vehicle coordinates, as the feature location is varied. Note that measurements of features directly north of the vehicle provide information about the vehicle's north position but comparably little information about the east direction. These results are based on the parameters listed in Table 6.2.



(a) Vehicle north coordinate ${}^{n}x$



(b) Vehicle east coordinate ${}^{e}x$

Figure 6-4: Information obtained from relative measurements. (a) and (b) show the information matrix elements corresponding to the north and east vehicle coordinates, respectively, based on single measurements of the range and bearing to a feature. Because range is more accurate than bearing for typical sonar systems, measurements provide substantially more information about the direction in which they lie than the orthogonal direction.

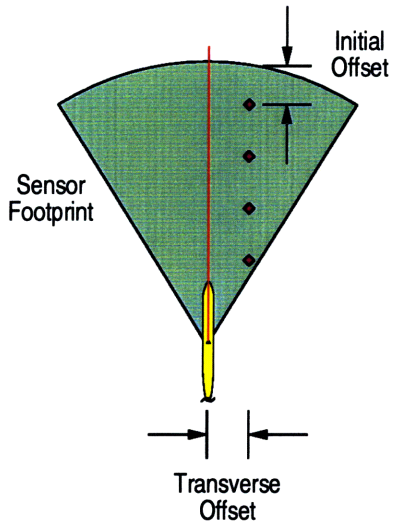


Figure 6-5: Feature measurement example setup. The vehicle transits in a straight line. Features appear to move through the sensor footprint at a constant transverse offset. Features are characterized by this transverse offset and their offset from the edge of the sensor range when they are first measured.

6.2.2 Feature information

While this information is of some use, a more interesting metric is obtained by summing the information matrix over multiple observations of a single feature. Using the parameters in Table 6.3, the sum of the information matrices of all of the measurements taken of a given feature while it is within the sonar footprint can be calculated. We assume that the vehicle is transiting in a straight line, so that measurements appear to approach the vehicle with a constant perpendicular distance between the feature and the vehicle path, as shown in Figure 6-5. The summed information matrix depends on the transverse distance of the feature from the path of the vehicle and the initial offset of the feature from the edge of the viewable range when it is first measured. To quantify this matrix more simply, we consider two norms for the vehicle portion of the matrix (i.e. the upper-left 3x3 block). First, we consider the determinant of the vehicle portion of the summed information matrix. This gives an estimate of the overall impact the feature can have on navigation performance.

Table 6.3: Feature information example parameters.

Parameter	Value
vehicle speed	3 $\frac{\text{m}}{\text{s}}$
sonar range	300 m
sonar view angle	$\pm 40^{\circ}$
sonar ping interval	5 s

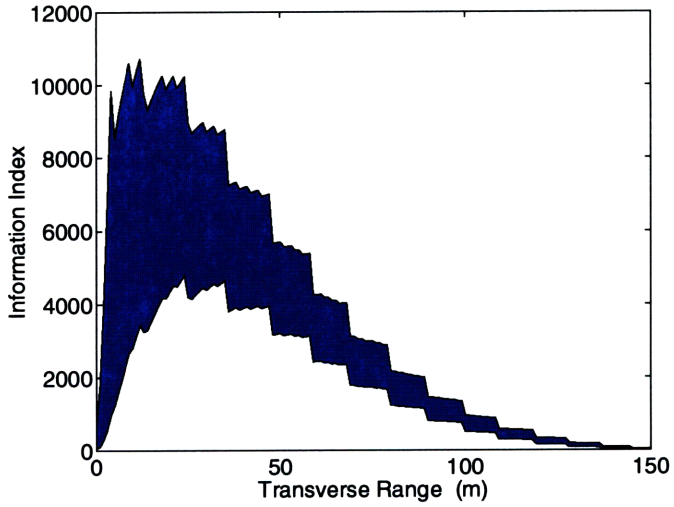
Second, we consider the trace of the block. This indicates the amount of information without penalizing features which fail to provide information about all vehicle states. These norms are shown in Figure 6-6. Note that features in the direct path of the vehicle are of less use because they can provide little information about the transverse vehicle coordinate. As the transverse distance to the vehicle path increases, the features are visible for less time, and therefore fewer measurements can be taken. The most useful measurements combine a long potential observation time with significant relative angular motion.

6.3 Data association errors

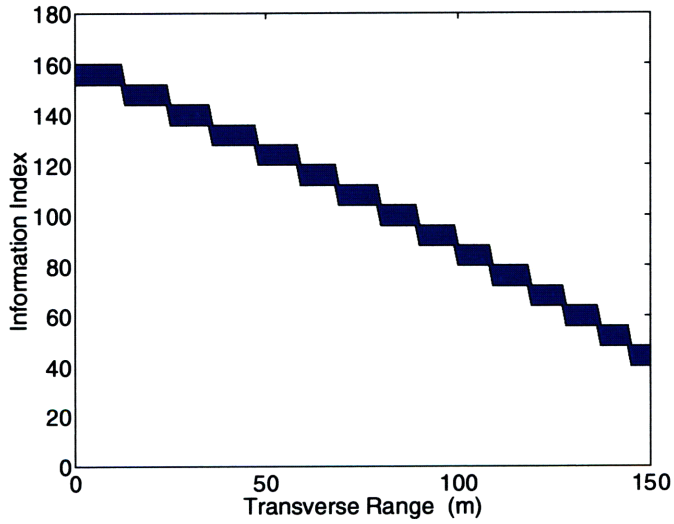
The effect of data association errors is quite difficult to analyze. Existing research [5, 57, 104] takes an empirical approach, demonstrating the relative efficacy of various algorithms. In this section, we describe the kinds of errors to which data association uncertainty may give rise and discuss the practical aspects of understanding the impact of data association errors.

6.3.1 The effects of event uncertainty

Algorithms which ignore event uncertainty have achieved a certain limited competence. Any algorithm which explicitly considers event uncertainty in a non-separable way (e.g. multiple-hypothesis tracking and integrated mapping and localization) takes on a great deal of complexity to unravel navigational events. The question arises



(a) Determinant of the summed vehicle information matrix.



(b) Trace of the summed vehicle information matrix.

Figure 6-6: Total information from features. The information matrix I_z is summed over all measurements taken of a given feature and the vehicle information block is extracted. The determinant and trace of this block are indicated. Plots show the minimum and maximum vehicle information norms versus transverse distance to the vehicle path for the full range of initial viewpoint offsets.

whether it is worthwhile to model event uncertainty explicitly and non-separably. Part of the answer lies in the demonstrated performance gains that multiple-hypothesis tracking has realized [17, 58]. The other part lies in a belief that such methods offer a better understanding of what is going wrong when data association errors occur. Because of this, sub-optimal implementations of algorithms which explicitly model event uncertainty tend to make better overall choices than alternative implementations, even when the amount of reasoning about possible events is reduced [29]. At any rate, reasoning about event uncertainty is surely the only way to sensibly make decisions about data association and other discrete portions of the hybrid estimation problem. It is a matter of exploiting the prior knowledge which is available and appropriately representing it in order to provide efficient yet effective algorithms.

There are many possible data association errors which can occur. They all amount to mistakenly identifying a measurement with an incorrect target. The basic types of data association error we consider are clutter and feature interaction. Clutter, also referred to as spurious measurements, are sonar returns which do not correspond to any modelable feature according to the measurement models being considered. In other words, they are returns that are not properly explainable given the prior models available to the algorithm. Feature interaction occurs when two or more modelable features are close together. Measurements from the features may be improperly associated, leading to a confusion between the features. Other data association errors are more typical in the target tracking problem and include splitting and merging targets and changing target behavior. Overall, data association errors have a greater impact for moving targets, but model uncertainty can give rise to complex data association problems as well.

6.3.2 Clutter

Clutter may be present due to a variety of causes. It may be due to real measurements which are not representable with the measurement models provided. It may

be discrete noise. Gating algorithms and conservative track initiation strategies can take care of most clutter. The difficult cases are when a spurious measurement occurs where a valid measurement is expected. In this case, the predominant effect is that the clutter can be ignored only if it is further from the expected measurement than a valid return. If it is closer to the expected return than the valid return (if one exists), the spurious measurement will most likely be chosen as the appropriate measurement to match with the feature in question. Clutter thus has the effect of occasionally giving the algorithm too much confidence in its estimate. When dense clutter is encountered, this can lead to filter divergence.

There are a number of ways to address clutter. The first is to ignore possible data association errors due to clutter. If the clutter is not too dense, the feature probability of detection is not too low, and there is enough process noise to compensate for erroneous associations, this strategy can be effective. If clutter is non-uniformly dense, features in cluttered regions can simply be ignored in favor of less ambiguous information sources. Finally, if a given sonar return has a great deal of clutter (either overall or in the region of a particular expected measurement), the algorithm can assume all returns (or all returns in that area) to be spurious for that time cycle. By essentially throwing out the ambiguous data, features can be tracked without too much additional complexity or probability of data association error, provided enough unambiguous measurements are available. Throughout this thesis, no special steps are taken to ensure that clutter is rejected.

6.3.3 Feature interaction

Feature interaction in similar to clutter in effect. The real problem arises when valid measurements appear closer to the expected measurements of opposing features than to their respective expected measurements. There is, of course, a limit of distinguishability for features which depends on the measurement noise, the feature process noise, and the model uncertainty. Without special precautions, this situation can cause a

rapid explosion in complexity. The obvious answer is to ignore features which are unresolvable. This, however, requires a method of reasoning about the resolvability and distinguishability of features. In this thesis, no special steps are taken to reduce the impact of feature interaction.

6.4 Summary

In this chapter, we have examined the problem of concurrent mapping and localization to determine how different types of uncertainty interact and affect the problem structure. We have examined the growth of vehicle position uncertainty and identified two regimes of error growth. In the first, which is dominated by model uncertainty, vehicle error increases with the square root of distance traveled. The second regime, which is dominated by navigational system uncertainties, involves vehicle errors that are linear with distance traveled. This linear growth range is the limiting factor in dead-reckoned missions; it arises because of the ever-increasing coupling between navigation estimates and position estimates. External position information has the potential to reduce this coupling to recover a more acceptable error growth regime. Additionally, measurements provide real information about the vehicle position through time. We have quantified (1) the information content of individual measurements and (2) the summed information content of features. Two factors affect the usefulness of features during transiting operation: (1) the transverse distance from the vehicle path and (2) the range of the relative bearing while in view. The most useful features for this operational paradigm were 10 to 40 m from the vehicle path. Such features proscribe significant paths relative to the vehicle egosphere and remain within the viewable area for many measurements. Finally, the errors which may arise from data association uncertainty were described. In the next chapter, we analyze the performance of the integrated mapping and navigation algorithm.

Chapter 7

Performance Analysis of IMAN

In this chapter, we analyze the level of performance achieved by the integrated mapping and navigation algorithm. We begin with a summary of the testing conditions. Monte Carlo simulation is used to explore IMAN performance. Environmental conditions, point-like features, and a number of performance metrics are discussed. Dead reckoning and an augmented stochastic mapping algorithm developed by H. J. S. Feder (both with and without cross-model correlations) provide contrasting algorithms. Results are presented which show the effects of clutter density, feature separation, and feature ordinality on IMAN performance. Several issues regarding IMAN performance are then discussed. The estimator performance and convergence are considered, and algorithmic complexity is assessed. A number of failure modes are also examined. The loss of valid tracks is considered, along with causes, impact, and potential improvements. The completion rate of the algorithm is discussed, and the role of complexity in overwhelming the algorithm is described. Potential improvements, both to improve handling of ambiguous situations and to recover from complexity-based failure, are considered. The phenomenon of map slip, in which sudden global errors are developed while relative mapping performance remains good, is discussed. The impact of feature ordinality and environmental conditions on map slip is highlighted, and possible improvements to address the problem of map slip are

considered. Finally, a summary of the IMAN performance analysis is provided.

7.1 Testing IMAN performance

There are two primary purposes for performance analysis testing of the current implementation of the integrated mapping and navigation algorithm: (1) assessment of the validity of IMAN as an approach to concurrent mapping and localization and (2) identification of the limitations of this implementation. The focus points for this research, as stated in Chapter 1, are enhancing the estimation and decision-making ability of the estimation process and managing algorithmic complexity. Because of this, there are two factors to consider in evaluating the success of this enterprise. First, does the algorithm, complexity issues aside, provide improved estimation as a result of the consideration of multiple hypotheses within an non-separable estimation framework? Second, have the methods for complexity management developed in this thesis improved the robustness of the algorithm in ambiguous situations? The answer to both of these questions is yes, and although the current implementation is by no means the last word on IMAN, it provides a valid and useful basis for further research.

7.1.1 Monte Carlo simulation

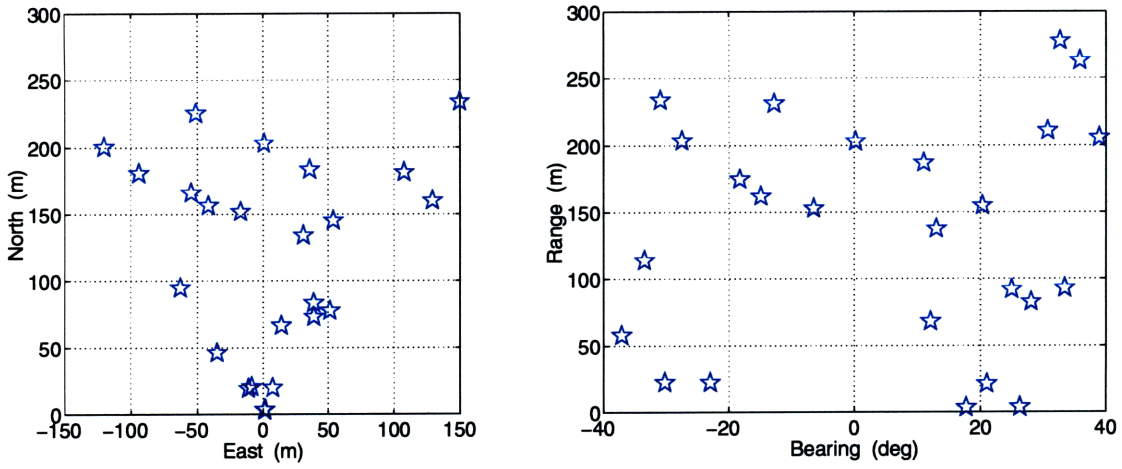
The complexity of the concurrent mapping and localization problem precludes a closed-form analysis of performance. Monte Carlo testing [85] is the use of the statistics of many simulations to characterize the range of performance achieved over the full space of possible mission realizations. Ten different mission scenarios were selected to aid in examining IMAN performance. One hundred simulations were run for each of these scenarios. Each mission consists of a thirty second stream of measurements. The performance metrics presented are the population statistics of these sets of simulations.

7.1.2 Environmental conditions

While the prior models considered are fairly simple, realistic environmental contexts were desired. There are a number of environmental factors that are not explicitly modelable using the prior models, but which must be resolved by the estimation process. First, there is the presence of clutter, or spurious measurements. Clutter arises in real data sets for a variety of reasons. For example, clutter may be caused by unmodeled feature types, multipath reflections, channel variations, and electronic sensor noise. In all of these cases, the spurious returns provided by clutter can not aid in the estimation process given the set of prior models used. To simulate clutter, a Poisson arrival rate is assumed over the footprint of the sensor, which has a range of 300 meters and a forward view angle of $\pm 40^o$, as shown in Figure 7-1. This arrival rate is characterized by a single parameter, λ, the expected number of spurious measurements within the viewable region. Once the number of spurious returns is determined, actual measurements are calculated. The clutter returns are uniformly distributed over the viewable region in measurement space (i.e. range and bearing). Note that these assumptions result in a relatively higher density of perceived targets near the vehicle in Cartesian space. This is an appropriate characterization of clutter that arises from electronic noise in the sensor itself, but does not necessarily reflect the spatial distribution of unmodelable features or multipath.[1]

To address the possibility of model errors in the vehicle dynamic model, an unknown cross-current is simulated. While this current is not extreme, it results in a vehicle drift in the transverse direction, as illustrated in Figure 7-2. This drift was unobservable from the dead reckoning measurements provided to the vehicle. While dead reckoning systems can be made robust to modeling errors by increasing the assumed process noise of the vehicle, they are incapable of resolving systematic error growth resulting from such unmodeled effects.

[1]Note that actual feature positions are more properly modeled fractally than by the assumption of a uniform distribution. Also, some causes of clutter, such as multipath and channel variation, are



(a) Cartesian space

(b) Measurement space

Figure 7-1: Clutter and the sonar footprint. The forward-looking sonar [67] has a range of 300 meters and a view angle of $\pm 40^o$. Twenty five clutter returns are shown. Clutter is uniformly distributed in measurement space, which results in a greater density near the vehicle in Cartesian space.

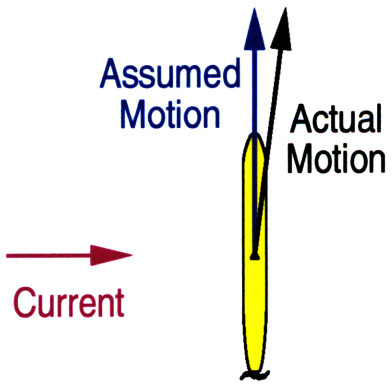


Figure 7-2: Current-induced vehicle drift. An unknown cross-current flows east, resulting in a transverse drift by the vehicle, which is assumed to be heading north. This drift is not observable from dead reckoning sensors, that is, measurements of speed, depth, and orientation.

7.1.3 Features

The primary task in defining the environmental context is of course the specification of features. For this performance analysis, actual point-like features are assumed to be present in the environment. As discussed in Section 3.1.3, this assumption is not necessary for the IMAN algorithm to remain valid; however, it does make the representation of the environment (external to the algorithm) simpler and more compact. The only concession to the stochastic nature of actual environmental objects is the inclusion of a substantial amount of process noise in the feature model. Actual feature representation, that is, external to the algorithm, is assumed to be exact, although measurement noise is also included. The measurement models used as prior models are assumed to be accurate for the purposes of random measurement generation.

To explore the role of clutter, a single feature is used in five different scenarios. While this results in a strictly unobservable global system, it allows the consideration of clutter independent of additional factors, such as track interaction. Two features are used to explore the effects of feature separation on estimation performance. These features are initially located at identical ranges from the vehicle. This complicates the problem by maximizing their possible interaction, as bearing is much more uncertain than range. Finally, a set of simulations were run with eight features. While real data often contains more features than this, we feel that these experiments illustrate the range of performance of the IMAN algorithm and identify its current limitations.

7.1.4 Performance metrics

Three performance metrics provide the primary characterization of algorithm performance: completion rate, global error, and relative error. Completion rate ${}^{c}\varepsilon$ is the percentage of simulations which run to completion. During some runs, the decision process becomes too complex for the integrated mapping and navigation algorithm.

---

dependent on vehicle and feature state.

A **complexity fault** is a failure to complete a simulation due to decision-making complexity. When a **complexity fault** occurs, the number of state estimates being considered rapidly increases until available memory is exhausted.

The completion rate is a measure of the robustness of the algorithm to the complexity of a given scenario. In general, complexity increases as ambiguous situations become more common, or likely.

Global error ${}^{g}\varepsilon$ is the mean-square error of the mapping and navigation portions of the estimation process (errors in the navigational system are not included). At each step, the vehicle has a number of feature tracks and a vehicle track. For each track, the most likely updated state estimate is chosen as the track state estimate. A simple nearest-neighbor procedure is used to match proposed tracks to actual targets.[2] For each actual feature or vehicle track ζ, the actual object location $\bar{x}$ is subtracted from the most likely estimate $\check{x}$ to determine an error vector

$$ {}_{\zeta}e = {}_{\zeta}\bar{x} - {}_{\zeta}\check{x}, \tag{7.1} $$

where x is used to denote a vehicle or feature state and ζ is an index over all tracks that have been matched to actual objects, i.e. the vehicle and actual features in the environment. Any additional tracks proposed by the algorithm are ignored. The set of all these error vectors is denoted

$$ E = \{{}_{\zeta}e\}_{\zeta} . \tag{7.2} $$

[2]Remember that the vehicle has no knowledge about what or how many features there actually are. Tracks that do not correspond to any actual feature are sometimes generated.

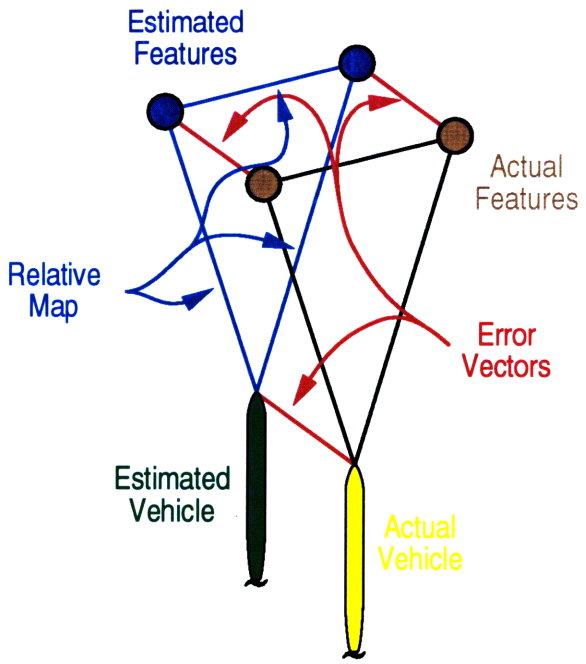


Figure 7-3: Global and relative error metrics.

The global error is then defined as

$$^{g}\varepsilon = \sum_{\zeta} {}_{\zeta}e^{T}\,{}_{\zeta}e \tag{7.3}$$

The relative error $^{\mathcal{L}}\varepsilon$ represents the success of the algorithm at estimating the relative positions of the vehicle and features. If the navigational objects are considered pairwise, the components of the relative error are the squared differences between the estimated and actual vectors connecting each pair, as shown in Figure 7-3. In terms of the errors themselves, the relative error can be represented as

$$^{\mathcal{L}}\varepsilon = \frac{N(\zeta)\;{}^{g}\varepsilon - \sum_{\zeta_1}\sum_{\zeta_2}\;{}_{\zeta_1}e^{T}\,{}_{\zeta_2}e}{N(\zeta)-1}, \tag{7.4}$$

where $N(\zeta)$ is the number of navigational objects, that is, the number of elements in the set E.

7.2 Alternative algorithms

In order to provide a benchmark for IMAN performance, additional navigational algorithms are also considered in these performance analysis simulations. While these algorithms do not capture the complexity of the hybrid estimation problem to the extent that IMAN does, they provide a sanity check regarding the validity of IMAN and the effects of taking this more complex approach to concurrent mapping and localization.

7.2.1 Dead reckoning

Dead reckoning is navigation using only the navigational system estimates of vehicle kinematics to estimate vehicle position. No external feature measurements are incorporated. Dead reckoning measurement updates are performed in the same way (and using the same models) as for IMAN and the other algorithms. Because of the simulated current, dead reckoning performs consistently worse than the other algorithms. Also, since relative measurements are not incorporated, there is little difference in dead reckoning performance over the scenarios tested. A representative comparison of IMAN and dead reckoning is shown in Figure 7-4. The dead reckoning algorithm fails to recover the drift of the vehicle in the cross-current. While the dead-reckoned estimate is robust to this drift, this robustness comes at the price of extremely uncertain estimates.

7.2.2 Augmented stochastic mapping

As discussed in Chapter 2, stochastic mapping is unable to handle data association uncertainty due to the assumption that feature matching is provided along with measurements. This is particularly relevant when considering clutter, as no ready basis exists for explaining such a phenomenon within traditional stochastic mapping. An augmented stochastic mapping algorithm developed by H. J. S. Feder which is ca-

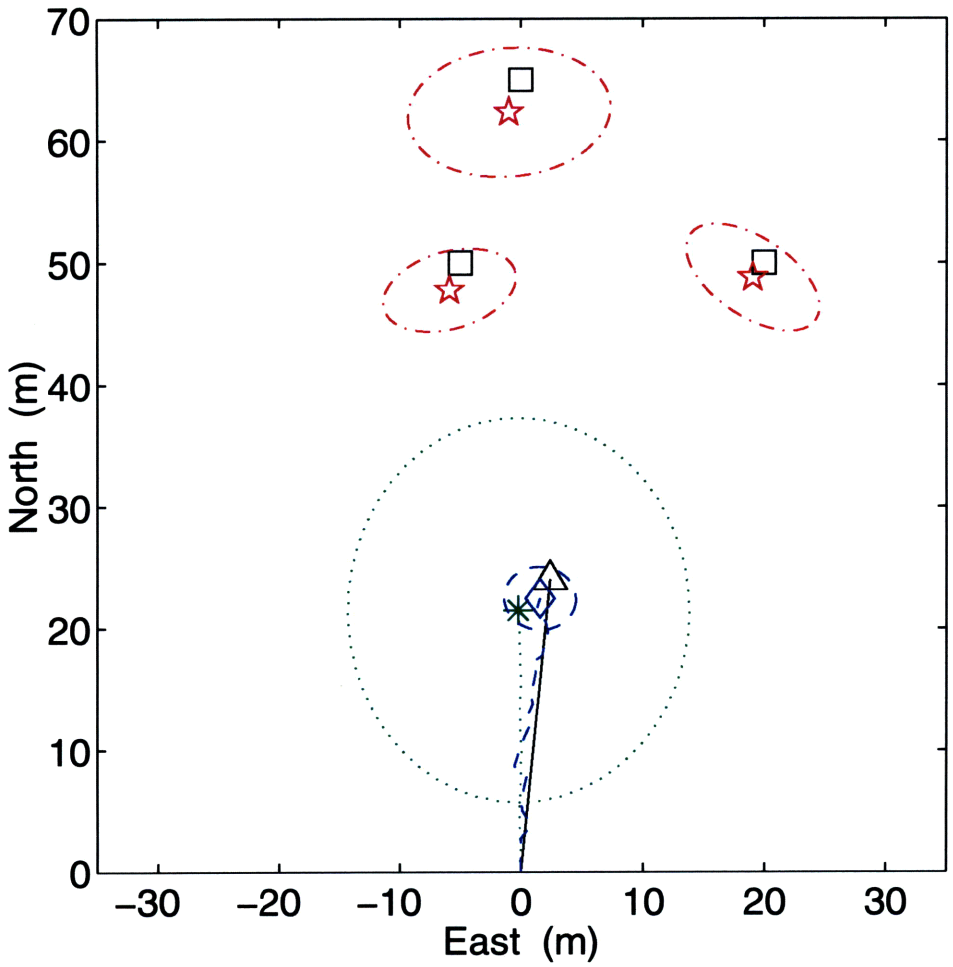


Figure 7-4: Comparison of IMAN and dead reckoning. The actual vehicle track, shown by a solid line and ending at a triangle, starts at (0,0). The vehicle is assumed to head directly north, but is pushed east by a cross-current. Three features, indicated by squares, are located at (50,-5), (50,20), and (65,0). The dead-reckoned vehicle path is shown by a dotted line, with the final position estimate marked with an asterisk. The three-sigma highest density region (HDR) is also shown by a dotted line. The IMAN estimated vehicle path is shown by a dashed line. The final estimate is indicated by a diamond, with the three-sigma HDR also shown by a dashed line. The final IMAN feature estimates are marked by stars. Their three-sigma HDRs are shown by dashed-dotted lines. The IMAN algorithm is able to estimate vehicle drift with the aid of sonar measurements. IMAN also provides a more accurate error estimate. Accuracy in calculating the estimate error is important when making decisions based on the quality of the vehicle navigation estimate.

$$\begin{bmatrix} P_v & P_{v1} & \cdots & P_{vn} \\ P_{v1}^T & P_1 & \cdots & P_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ P_{vn}^T & P_{1n}^T & \cdots & P_n \end{bmatrix} \qquad \begin{bmatrix} P_v & 0 & \cdots & 0 \\ 0 & P_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & P_n \end{bmatrix}$$

(a) ASM1 covariance matrix (b) ASM2 covariance matrix

Figure 7-5: Augmented stochastic mapping covariance matrices. The ASM1 algorithm retains all cross-model correlations. ASM2 discards the cross-model correlations, as does IMAN.

pable of operation in clutter is used as an alternative algorithm for comparison with IMAN. Data association is resolved using a nearest-neighbor algorithm [5]. Thus discrete decisions are completed instantaneously from the point of view of the continuous estimation problem; delayed decision-making is not possible. This separable approach to hybrid estimation is similar to, but less robust than, the probabilistic data association filter developed by Bar-Shalom [5, 18].

Two versions of the augmented stochastic mapping algorithm are considered. The first, ASM1, provides the complete estimate, including a full account of the cross-model correlations, which are ignored by IMAN. The second version, ASM2, drops the cross-model correlations. The result is a block diagonal covariance matrix for the state estimates. Figure 7-5 illustrates the covariance matrices used in these two augmented stochastic mapping algorithms.

7.3 Results

Figure 7-6 shows a representative run, illustrating some common aspects of the experimental simulations. Identical actual vehicle dynamics and environmental effects are present in all simulated runs. The differences between runs are the number of features present, the actual feature locations and the measurement values received. The

Table 7.1: Parameters characterizing dead reckoning measurement errors.

Parameter	Value
Depth measurement variance	25 cm^2
Speed measurement variance	0.25 $\frac{m^2}{s^2}$
Pitch measurement variance	0.26 deg^2
Yaw measurement variance	0.26 deg^2

Table 7.2: Parameters characterizing sonar measurement errors.

Parameter	Value
Probability of detection	0.9
Probability of false alarm	0.3
Range measurement variance	0.5 m^2
Bearing measurement variance	2.6 deg^2

vehicle is subject to an *a priori* unknown cross-current which results in a 10% transverse drift rate. The vehicle assumes that it is heading directly north in a straight line. Uncertain measurements of vehicle depth, speed, pitch, and yaw are available at each time cycle. Table 7.1 lists the noise parameters for these dead reckoning measurements. Sonar measurements provide the relative range and bearing to apparent targets. The noise parameters used to model the sonar measurement process are listed in Table 7.2. The algorithms use accurate measurement models of these processes. The prior models for the vehicle and point-like features, as developed in Chapter 3, are used to model vehicle and feature tracks. The parameters used to specify these models are shown in Tables 7.3 and 7.4. In all runs, the vehicle starts at the origin and proceeds to the north with a speed of 1 $\frac{m}{s}$. The cross-current causes an easterly drift of 0.1 $\frac{m}{s}$. The vehicle processes measurement data at one second intervals. Runs continue for thirty seconds unless the algorithm is overwhelmed by complexity.

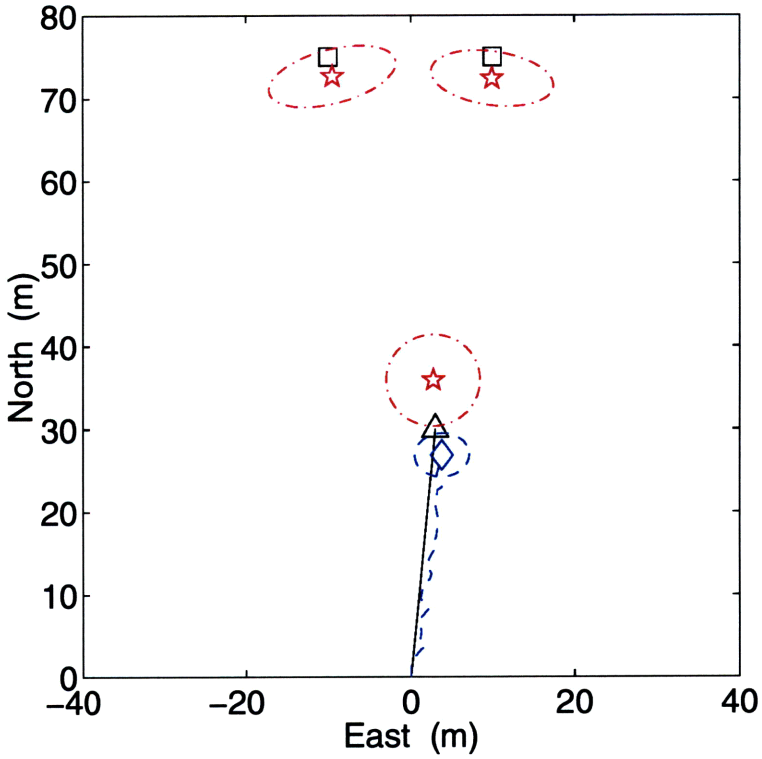


Figure 7-6: An example IMAN simulation. The actual vehicle is indicated by a triangle; its path is a solid line. The vehicle is assumed to be heading directly north, but drifts to the east in the presence of an unknown current. Two features, indicated by squares, are present at (75,±10). The estimated vehicle is indicated by a diamond; its path is a dashed line. The three-sigma error ellipse for the vehicle is also shown with a dashed line. Each proposed feature is marked by a star. The three-sigma error ellipses for the feature estimates are shown by dashed-dotted lines. Note that in this case, a false feature is being considered, due to the presence of clutter. While the relative map is quite accurate in this example, there is some map slip; note that the actual vehicle falls outside of the estimated vehicle highest density region.

Table 7.3: Process noise values for the vehicle model.

Parameter	Value
North coordinate process noise variance	0.5 m^2
East coordinate process noise variance	0.5 m^2
Depth process noise variance	0.0025 m^2
Speed process noise variance	0.01 $\frac{m^2}{s^2}$
Pitch process noise variance	2.6 deg^2
Yaw process noise variance	2.6 deg^2

Table 7.4: Process noise values for the point-like feature model.

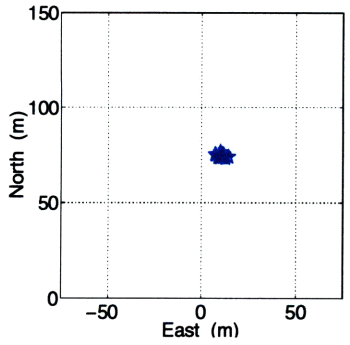
Component	Value
North coordinate process noise variance	2 m^2
East coordinate process noise variance	2 m^2

7.3.1 Clutter

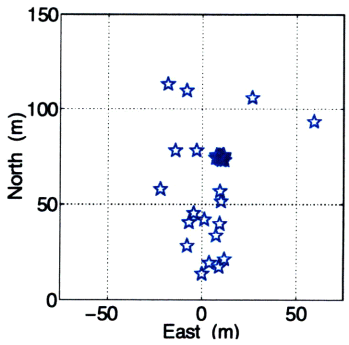
The effect of clutter on navigation and mapping is explored using a single feature and varying clutter densities. The single feature is located at (75,10) for all of the clutter scenarios. The expected number of spurious returns λ varies between 0 and 4. Spurious returns are uniformly distributed in measurement space. Figure 7-7 illustrates the amount of clutter for each scenario. Apparent measurement locations are shown based on the actual vehicle position. Measurement noise and vehicle navigational uncertainty provide additional ambiguity.

Completion rates for the various algorithms are shown in Figure 7-8. While IMAN has no problem completing the runs in the clutter-free case, significant failure rates are found as clutter density is increased. In the densest clutter considered, IMAN is able to complete only 34% of the runs. This is due to increased situational ambiguity when spurious returns gate for several time cycles in a row.

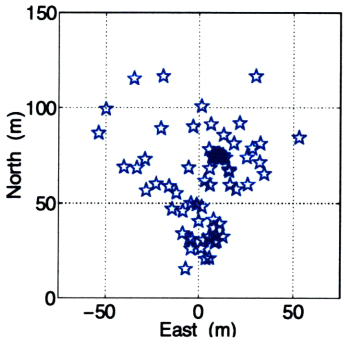
Figure 7-9 presents the global and relative errors for the IMAN and augmented stochastic mapping algorithms. When complexity faults do not occur, the IMAN algorithm provides a global mapping performance which is comparable to the augmented mapping algorithms. The relative mapping performance of IMAN is significantly better than that of the augmented stochastic mapping algorithms. Divergence of the estimate is not a problem despite the unobservability of the system. By ignoring cross-model correlation, IMAN allows map slip, but retains accuracy in estimating the relative vehicle and feature positions.



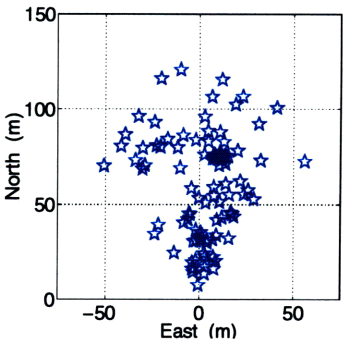
(a) $\lambda = 0$



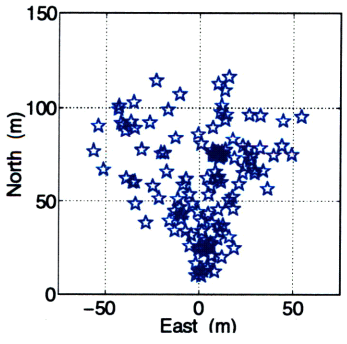
(b) $\lambda = 1$



(c) $\lambda = 2$



(d) $\lambda = 3$



(e) $\lambda = 4$

Figure 7-7: Examples of clutter density. The expected number of spurious returns per cycle λ is varied. A single feature at (75,10) is measured each cycle with a probability of detection $P_D = 0.9$. Apparent measurement locations are shown based on the actual vehicle position through thirty measurement cycles.

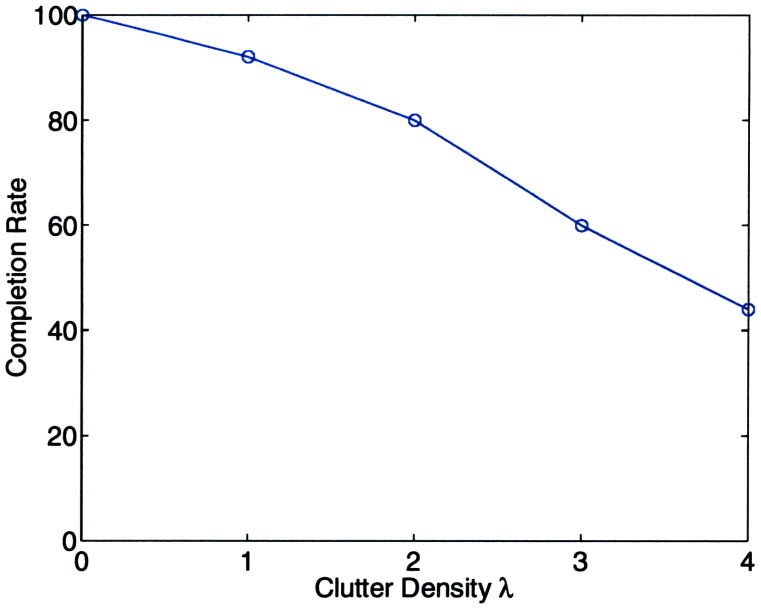
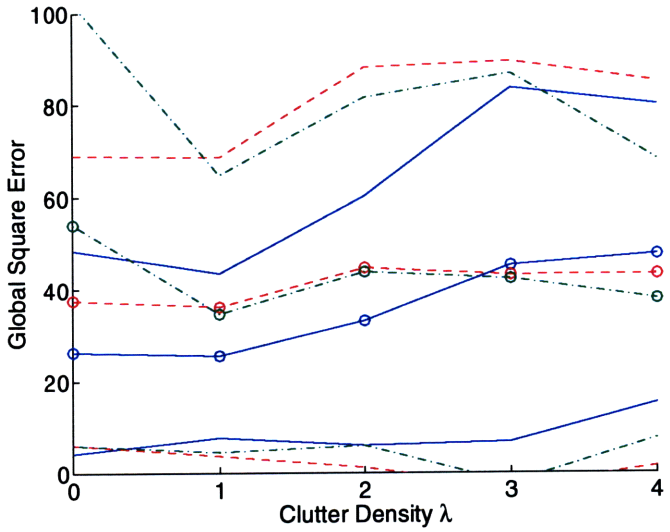


Figure 7-8: The effect of clutter density on completion rate. Clutter density is characterized by the expected number of spurious returns during each scan λ. While the IMAN algorithm has no problem in the clutter-free case, higher clutter densities lead to a large number of complexity faults.

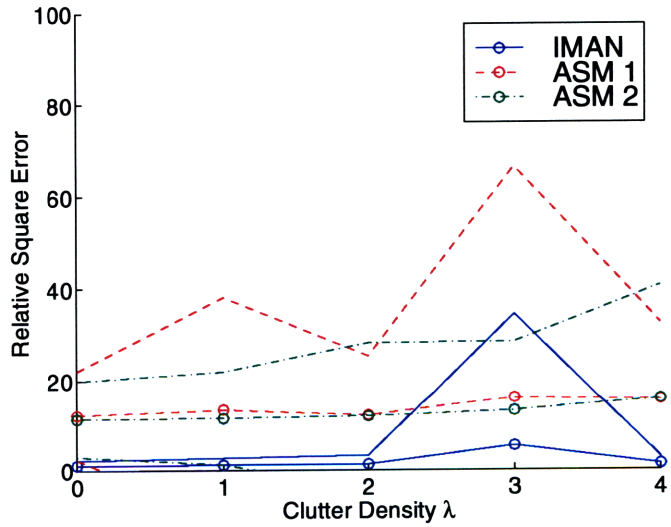
7.3.2 Feature separation

To explore the effects of track interaction, we consider the effect of feature separation on navigation and mapping performance. Two features, at $(75, \pm\frac{\rho}{2})$, are used; the feature separation ρ is varied between 10 and 40 meters. Clutter is also added, with an arrival rate $\lambda = 2$.

The completion rates for IMAN in these four scenarios are shown in Figure 7-10. IMAN had significant difficulties overcoming the complexity of reasoning about feature separation. A completion rate of 0% is obtained for feature separation of 10 meters. As ρ is increased to 40 meters, a 75% completion rate is obtained. The difficulties of separating feature tracks are complicated by the fact that the initial ranges are identical and clutter is present. The significant failures of IMAN in this case underscore the need for improved robustness, for example through the implementation of default hypothesis and recovery from complexity faults.



(a) Global error



(b) Relative error

Figure 7-9: The effect of clutter of navigation and mapping performance. The global mapping performance of IMAN is comparable to the performance of the augmented stochastic mapping algorithms when complexity faults do not occur. IMAN outperforms the ASM algorithms in relative mapping. Note that the relative error for IMAN is considerably smaller than the global error as a result of ignoring cross-model correlations.

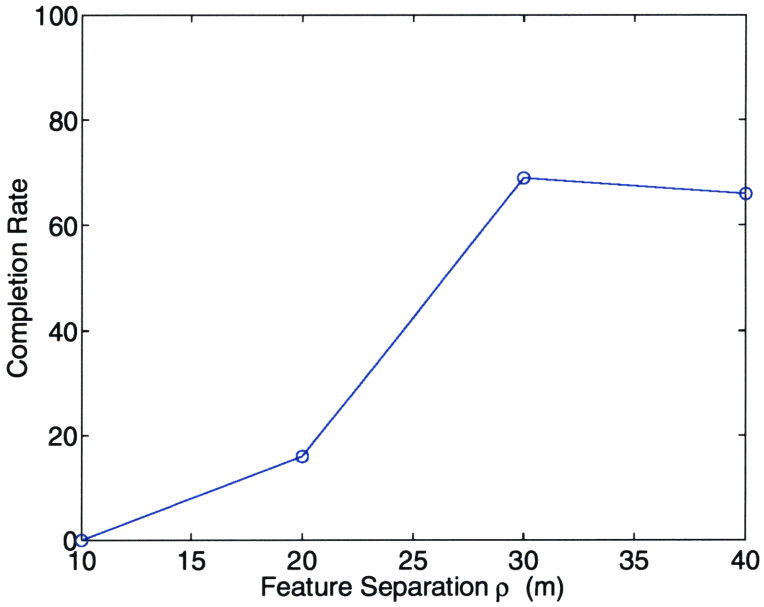


Figure 7-10: The effect of feature separation on completion rate. Track interaction remains a research issue for IMAN and is the largest factor in precluding a robust implementation. When features are 10 meters apart, the IMAN algorithm cannot disambiguate the tracks. As feature separation increases, IMAN fares better. These scenarios included a clutter arrival rate of $\lambda = 2$. The baseline performance for this clutter density is reached with a feature separation of 30 meters, indicating that at this distance, features are essentially distinct to the algorithm.

Table 7.5: Results of the eight-feature experiment.

Algorithm	Global Error	Relative Error	Completion Rate
IMAN	13.51	1.56	15
ASM with correlations	20162.79	22144.06	100
ASM without correlations	22663.18	24964.86	17

Figure 7-11 shows the global and relative error metrics for the feature separation scenarios. When complexity faults do not cause failure, IMAN demonstrates considerable performance gains in both global and relative mapping. Delayed decision-making and the consideration of discrete events allows better estimation choices to be made.

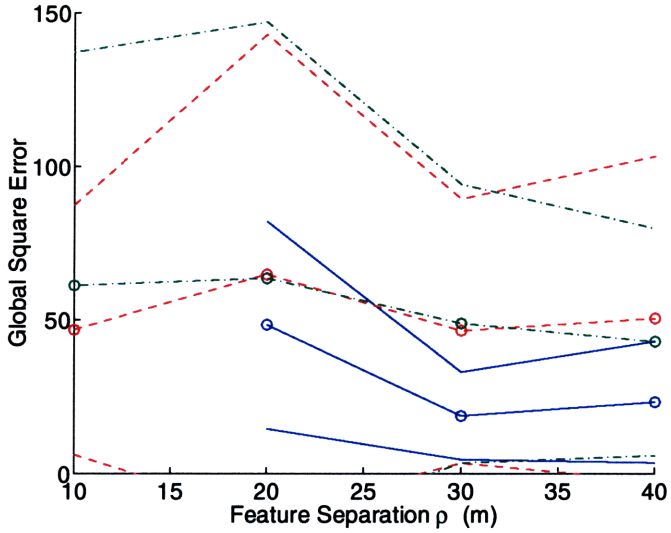
7.3.3 Feature ordinality

Performance in the presence of more features is illustrated by a scenario with eight features located at (75, ±20), (100, ±20), (125, ±20), and (125, ±20). No clutter is present in these simulations. Figure 7-12 shows measurements and IMAN performance for a representative simulation. Performance metrics for the various algorithms are listed in Table 7.5. The low completion rate for the ASM2 algorithm is due to track loss due to filter divergence, rather than complexity.

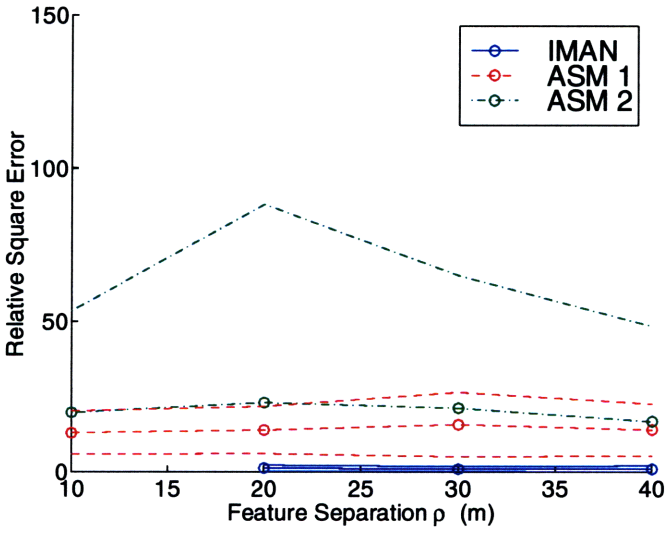
7.4 Discussion

7.4.1 Estimator performance

Estimator performance for the IMAN algorithm is quite promising. In situations where complexity faults do not cause failure, IMAN outperforms the augmented stochastic mapping algorithms. Divergence of the estimator has not been as issue, but could become a difficulty for longer missions. This could be addressed by refinements to the filtering process [99, 84, 27, 39, 28, 91], for example, incorporating higher-order filters, iterated Kalman filtering, or bootstrap filtering.

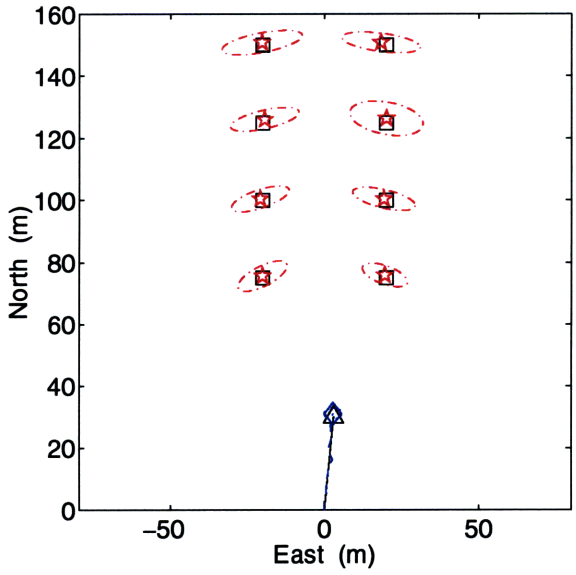


(a) Global error

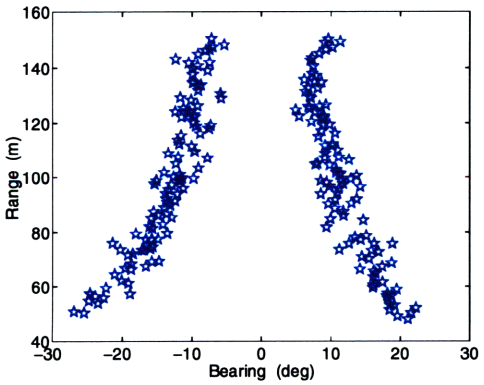


(b) Relative error

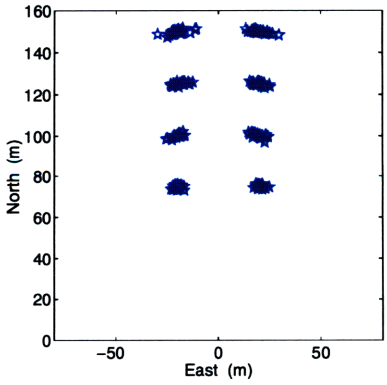
Figure 7-11: The effect of feature separation on mapping and navigation performance. While IMAN is unable to complete highly ambiguous measurement sets, its performance for completed sets shows substantial gains. IMAN outperforms the augmented stochastic mapping algorithms in both global and relative mapping.



(a) Navigation and mapping performance. The actual vehicle is marked by a triangle; its path is shown by a solid line. The estimated vehicle is marked by a diamond. Its path and three-sigma error ellipse are shown by a dashed line. Eight actual features are present and are marked by squares. Feature estimates are marked by stars; their three-sigma error ellipses are shown by dashed-dotted lines.



(b) Apparent measurement locations in measurement space.



(c) Apparent measurement locations in Cartesian space.

Figure 7-12: Representative results with eight features.

7.4.2 Track loss

Track loss has not been a problem for IMAN in these experiments. The few completion failures that occurred for the augmented stochastic mapping algorithms are due to a combination of initial vehicle uncertainty and track loss (or divergence). As longer missions are considered, this may become an issue. However, the relative error performance of IMAN suggests that track loss is less of a problem when relative mapping schemes are utilized. The role of map representation and the relationship between global and relative mapping need to be more fully considered to address this question.

7.4.3 Complexity faults

Complexity faults are a significant remaining problem, as can be seen by the completion rates (the overall completion rate was 53% over 1000 missions). The major cause for this is track interaction and clutter density. Suggestions for addressing this problem are discussed in Chapter 5. In spite of the complexity problems associated with hybrid estimation, we feel that IMAN represents the correct approach for CML. The performance when complexity is adequately managed, and, in particular, the relative mapping performance, bear this out. The significant intrinsic complexities engendered by the ambiguity of concurrent mapping and localization cannot be effectively overcome using simplistic techniques. While managing complexity will remain a major concern, aggressive confrontation of this problem from a determinedly decision-theoretic standpoint is the key to realizing true performance gains and robust operation.

7.4.4 Map slip

Map slip is a significant and poorly understood phenomenon in concurrent mapping and localization. Manifesting particularly when few features are available, it is related

to the observability of the system. The interdependence of observability, modeling flexibility, and model uncertainty is poorly understood, perhaps because of the success of Kalman filtering in many situations despite its extensive assumptions. In order to develop more effective hybrid estimation algorithms, a deeper understanding of these issues is essential. The amount of map slip increases if cross-model correlations are ignored, but this assumption does not necessarily impair relative mapping performance. Practical methods for addressing map slip depend on an improved understanding of the relationship between global and relative mapping and the map representation problem in general.

7.5 Summary

In this chapter, we have examined the performance of the integrated mapping and navigation algorithm and contrasted it with dead reckoning and augmented stochastic mapping approaches. Monte Carlo simulation was used to capture the range of performance for a variety of scenarios. Global error, relative error, and completion rate were used to evaluate the estimation and decision-making performance and complexity management capabilities of the algorithms. The effects of clutter density, feature interaction, and feature ordinality on navigation and mapping performance were considered. Finally, a number of observations regarding these results were made. In the next chapter, we draw conclusions about the work in this thesis and consider future research directions.

Chapter 8

Conclusions and Future Research

We began this thesis with the recognition that the current state of the art in estimation is insufficient to encompass the problem of concurrent mapping and localization. There is a demonstrable need, however, to incorporate environmental information more effectively as an aid to navigation and mapping. We have identified data association uncertainty, navigational uncertainty, and prior model uncertainty as the key issues shaping the problem of CML. Existing approaches to concurrent mapping and localization have failed to account for these uncertainties and so are prevented from effectively reasoning about navigational events such as measurements and changes in model class. Integrated mapping and navigation is an approach to the problem of concurrent mapping and localization that is capable of addressing all of these uncertainties. The goals of this dissertation have been (1) to develop a unified theory for integrated mapping and navigation, (2) to formalize an IMAN algorithm, (3) to produce an operational codebase that can process realistic data sets of limited duration, and (4) to provide an analysis of both the performance limitations imposed by the problem of CML itself and the actual abilities of the algorithm as implemented. In this chapter, we summarize the thesis and provide a roadmap for further development in the areas of concurrent mapping and localization, reasoning about measurement phenomenology, and integrating high-level vehicle control with data collection and

assimilation. We begin by discussing the contributions made in this thesis toward the theory of integrated mapping and navigation, the implementation of IMAN, and the analysis of CML and IMAN. The overall impact of this research is briefly considered. A number of future research directions are discussed, including the long-term research potential of integrated mapping and navigation. Finally, we briefly summarize the most cogent points from this dissertation.

8.1 Contributions

Integrated mapping and navigation involves a fundamental departure from existing concurrent mapping and localization techniques. We feel that this generalization of the problem is essential in the development of improved mapping and navigation algorithms. As a result of this, the theoretical development and formalization of IMAN represent the most important contribution from this work. Additional contributions have been made (theoretically, implementationally, and analytically), but these will necessarily be superseded as further research and development ensues. The state of the art in concurrent mapping and localization has reached a point where recent advances have been increasingly *ad hoc*. The paradigm shift embodied in posing the question of integrated mapping and localization will, we hope, provide a unified general framework allowing a more structured development of the field.

8.1.1 Theory

The major contributions of this thesis are theoretical. Here we summarize a number of the developments made during the course of this research.

Problem formalization

The problem of integrated mapping and navigation is fundamentally more complex that stochastic mapping and multiple-hypothesis tracking approaches because there

may be multiple possible vehicle states at any given time. This changes the structure of the problem and demands novel techniques for keeping track of vehicle and feature estimates and their interdependencies. We have formalized the statement of IMAN and identified the structural artifacts engendered by the consideration of multiple vehicle estimates.

Probabilistic basis for IMAN

We have developed the theory for integrated mapping and navigation within a Bayesian probabilistic framework. The effects of suboptimal estimation (due to, for example, nonlinearities, non-Gaussianity, and pruning) have been identified. The formulation of state, hypothesis, and assignment likelihoods has been developed.

Delayed track initiation

A delayed track initiation technique was developed. This isolates reasoning about track initiation from the main decision-based tree growth of the algorithm, significantly reducing the computational burden of considering track instantiation. The probabilistic basis of this (suboptimal) approach has also been developed.

n-backscan assignment pruning

The application of pruning techniques, which are quite common in multiple-hypothesis tracking, to IMAN has been considered. The impact of structural differences on the robustness of these techniques was identified, and the n-backscan pruning algorithm was successfully adapted for assignment-based pruning of hypotheses.

Specific prior models

We developed two concrete ontological models to enable tracking survey-class AUVs and point-like environmental features. Measurement models for dead reckoning and sonar interrogation of point-like features were also developed.

8.1.2 Implementation

Implementation of the theory of integrated mapping and navigation is an essential step is assessing the viability of such an approach. Also, structuring and programming such a complex framework is no trivial feat. In practice, the implementation has evolved alongside the theory, allowing a synergy between the object-oriented analysis of the problem domain and the appropriate representation of theoretical elements. An extensive codebase ($\approx$ 20,000 lines) has been developed that implements a superset of the concepts addressed in this thesis. This implementation has been designed for maintainability and extensibility, with possible future developments in mind at each step.

8.1.3 Analysis

Finally, a series of Monte Carlo simulations were performed to analyze the performance of the integrated mapping and navigation algorithm. The performance analysis is intended to aid in understanding the limits of the concurrent mapping and localization problem itself and to assert the validity of IMAN as an approach to CML. The present implementation of IMAN is necessarily preliminary; further developments will rapidly eclipse the performance level shown herein. The essential point is an understanding of (1) what the current implementation has accomplished and (2) why it has accomplished that. Additionally, an understanding of the limitations of the current algorithm is useful in considering the future development needs for integrated mapping and navigation.

Navigational uncertainty and error growth

The connection between navigational uncertainty and error growth while dead reckoning was explored. Two regimes of error growth were identified, corresponding to dominance of model uncertainty and navigational system uncertainty, respectively.

The potential for reducing the greater error growth rates caused by correlations between the navigational system and the position estimate was noted.

Information content of features

The Fisher information was used to assess the worth of point-like features as sources of information for a common survey-class operational regime. While the Fisher information does not bound estimate uncertainty, it does provide a useful quantification of environmental information.

IMAN performance

The performance of IMAN was considered through Monte Carlo simulation of a number of simulated missions. Global and relative errors and completion rate were used to assess the performance of IMAN and two augmented stochastic mapping algorithms. IMAN displayed superior performance for those missions in which complexity-based failure did not occur, demonstrating the negative impact of data association error when the discrete estimation problem is oversimplified.

Map slip

Cross-model correlations are ignored in IMAN. This simplification increases the chance of map slip, or the sudden increase in global error without relative error divergence. However, the effectiveness of IMAN in minimizing relative error demands a reconsideration of the map representation problem.

Remaining complexity issues

Track interaction has been identified as the most important remaining factor in complexity-based failure of the IMAN algorithm. We have proposed a number of potential solutions to address this problem.

8.2 Impact

In many ways, our conception of the navigation problem has remained fairly static since the early 1960's, when much of the seminal work on Kalman filtering was done. To be sure, many refinements have been developed, and our understanding of recursive estimation has expanded tremendously. However, the inability to reason about prior models themselves from within the estimation process has necessarily limited the technologies which have been developed. Robust control was perhaps the first application to systematically consider what happens when the prior models used for filtering are incorrect or imprecise. We feel that the further development of navigation technologies depends on a shift of focus to incorporate not only the processing of information within a set model framework, but also deliberation about the model framework itself.

During this same period, a widening gulf has developed in the artificial intelligence community, resulting in often completely independent focuses on high-level planning and reasoning on the one hand and local reactivity and robustness to realistic environments on the other. These two goals indeed often seems at odds. Responsiveness and robustness often require truncated decision-making and the acceptance of broad assumptions. The reasoning process is highly subjective and often must rely on heuristic evaluations rather than provable optimality. Navigation has most often been concerned with the low-level, data-focused, and deliberation-deficient side of this dialectic. The time has come to bridge the gap between experience and understanding. This thesis is a first step in this direction, an integration of the data provided by sensors and the prior understanding of context. This reconceptualization of navigation to include not only the logical interpretation of measurements, but also a deeper understanding of the events that transpire and the context in which they occur, has the potential to provide a powerful lever toward the development of navigation and mapping techniques that transcend the current paradigm and achieve truly intelligent behavior.

8.3 Future research directions

8.3.1 Theoretical extensions

A number of theoretical extensions were proposed in Chapters 3 and 5 to address remaining issues in the algorithmic complexity of IMAN. This included the concept of default hypotheses, which would allow graceful recovery from ambiguous situations. There is also a need for development of additional feature ontologies. Empirical consideration of sensor physics can be used to develop new classes of useful features. The process by which such feature ontologies can be efficiently developed remains an open question. Integration of the IMAN algorithm with higher-level vehicle control (or even low-level dynamic control) poses another theoretical challenge.

8.3.2 Implementation improvements

Several improvements could be made to the implementation. Some specific recommendations are provided in Chapter 4. Suggested improvements include an improved interface to aid in integration of IMAN with vehicle or user-interface code. The representation of resolved track in persistent form (i.e. the off-loading of these estimates to disk) could improve the memory requirements of the algorithm. The code itself is, of course, in an initial version as well. Iteration of the code design will undoubtedly remove any memory leaks present and improve efficiency. Specialized memory management may improve performance and increase robustness to complexity.

8.3.3 Further analysis

A number of additional investigations regarding the limitations of CML and the performance of IMAN are also recommended. A convincing theoretical analysis of the effects of data association does not exist and would greatly enhance our understanding of the problem domain. A better understanding of the effect of environmental

measurements on the correlation between the vehicle position estimate and the navigational system would improve our understanding of error growth limitations in CML, possibly leading to an analysis of conditions under which vehicle position error remains bounded. The role of map representation and map slip are related phenomena which are crucial in advancing our operational understanding of IMAN and its potential applications.

8.3.4 Long-range research potential

As stated above, the theory developed in this thesis has potential applications to wide variety of fields. Navigation, in general, can benefit by an increased understanding of non-separable hybrid estimation. This is particularly true when absolute positioning systems (such as GPS) are unavailable or unreliable, such as underwater, in space, underground, and on land in urban or cluttered environments. Further research directions include sharing map information, recovering correlation information, and multiple-vehicle implementations. As on-board computational abilities improve (and algorithmic complexity is reduced) adaptive feature ontology becomes a possibility. Also, interacting prior models can be considered in more depth. For example, by providing a more extensive channel (environmental) model, both feature and environmental models can be estimated, both parametrically and regarding navigational events. Finally, the basis of prior knowledge representation can be extended to consider target behavior in an increasingly deliberative manner. Situated navigation, in which the vehicle reacts to recognized behavior in targets, becomes a possibility.

8.4 Summary

We have developed a novel approach to concurrent mapping and localization which explicitly addresses the major sources of uncertainty in a generalized way. Significant structural differences from existing algorithms have required the development

of a more generalized theory for considering hybrid estimation in the presence of navigation uncertainty. The two primary focus points for this research have been (1) enhancing estimation and decision-making through the realistic representation of the major classes of uncertainty and (2) managing complexity. The most important contribution of this work has been the theoretical development and formalization of the problem of integrated mapping and navigation. Additional contributions have been made in implementation and an analysis of the limitations of CML and IMAN. IMAN provides improved vehicle and feature estimation when track interaction does not overwhelm the algorithm. While improved track initiation techniques have extended the operable range of IMAN by an order of magnitude, robustness to complexity remains a problem. Further research is expected to enable recovery from ambiguous situations such as those generated by track interaction. IMAN represents a valid and viable approach to concurrent mapping and localization that is still in the early stages of its development. The theoretical and practical advances realized in this work have broad potential applications to AUV navigation, robotics in general, and a wide range of non-separable hybrid estimation problems.

Bibliography

[1] B. Anderson and J. Moore. *Optimal Filtering.* Prentice-Hall, 1979.

[2] Anon. Comparison of two algorithms for determining ranked assignments with application to multitarget tracking and motion correspondence. *IEEE Trans. Aerospace and Electronic Systems*, 33(1):295–301, January 1997.

[3] M. Athans and C. B. Chang. Adaptive estimation and parameter identifcation using multiple model estimation algorithm. Technical Report 1976-28, MIT Lincoln Laboratories, June 1976.

[4] A. V. Balakrishnan. *Kalman Filtering Theory.* Optimization Software, Inc., 1987.

[5] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association.* Academic Press, 1988.

[6] J. E. Bares and D. S. Wettergreen. Lessons from the development and deployment of Dante II. In *International Conference on Field and Service Robotics*, pages 72–79, Canberra, Australia, 1997. Panther Publishing and Printing.

[7] A. Barker, D. Brown, and W. Martin. Static data association with a terrain-based prior density. *IEEE Trans. Systems, Man, and Cybernetics, Part C*, 1(28):151–157, February 1998.

[8] M. Bayarri, M. DeGroot, and J. Kadane. What is the likelihood function? In S. Gupta and J. Berger, editors, *Statisical Decision Theory and Related Topics IV Volume 1*, pages 3–28. Springer Verlag, 1988.

[9] Andrew A. Bennett. *Feature Relative Navigation for Autonomous Underwater Vehicles.* PhD thesis, MIT, 1997.

[10] S. Betge-Brezetz, P. Hebert, R. Chatila, and M. Devy. Uncertain map making in natural environments. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1048–1053, April 1996.

[11] M. Betke and L. Gurvits. Mobil robot localization using landmarks. *IEEE Trans. Robotics and Automation*, 13(2):251–263, April 1997.

[12] A. Bobick and J. Davis. Real-time recognition of activity using temporal templates. In *3rd IEEE Workshop on Applications of Computer Vision, WACV'96*, Sarasota, FL, USA, 1996. IEEE.

[13] W. L. Brogan. Algorithm for ranked assignments with applications to multiobject tracking. *J Guid Control Dyn*, 12(3):357–364, May-June 1989.

[14] D. K. Brown. Damn the mines! *Proceedings of the U. S. Naval Institute*, 118(3):45–50, March 1992.

[15] H. Bulata and M. Devy. Incremental construction of a landmark-based and topological model of indoor environments by a mobile robot. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1054–1060, April 1996.

[16] J. Castellanos, J. Tardos, and G. Schmidt. Building a map of the environment of a mobile robot: The importance of correlations. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1053–1059, April 1997.

[17] K. Chang, S. Mori, and C. Chong. Performance evaluation of a multiple hypothesis multitarget tracking algorithm. In *IEEE Int. Conference on Decision and Control (CDC)*, pages 2258–2263, 1990.

[18] K-C. Chang, C-Y. Chong, and Y. Bar-shalom. Joint probabilistic data association in distributed sensor networks. *IEEE Trans. Automatic Control*, 31(10):889–897, October 1983.

[19] C. C. Chong and J. C. Jia. Classification of multi-source data using predictive ability measure. In *International Geoscience and Remote Sensing Symposium*, pages 180–182. IEEE, 1996.

[20] K. S. Chong and L. Kleeman. Sonar-based map building for a mobile robot. In *Proc. IEEE Int. Conf. Robotics and Automation*, 1997.

[21] E. Y. Chow and A. S. Willsky. Bayesian design of decision rules for failure detection. *IEEE Trans. Aerospace and Electronic Systems*, AES-20(6):761–774, november 1984.

[22] J. Collins and J. Uhlmann. Efficient gating in data association with multivariate Gaussian distributed states. *IEEE Trans. Aerospace and Electronic Systems*, 28(3):909–916, 1992.

[23] I. J. Cox and J. J. Leonard. Temporal integration of multiple sensor observations for dynamic world modeling: A multiple hypothesis approach. In *International Workshop on Information Processing for Autonomous Mobile Robots: Theory and Application*, Munich, Germany, 1991. Berlin: Springer-Verlag.

[24] 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.

[25] I. J. Cox and M. M. Miller. On finding ranked assignments with application to multi-target tracking and motion correspondence. *IEEE Trans. Aerospace and Electronic Systems*, 31(1):486–489, January 1995.

[26] T. Curtin, J. G. Bellingham, J. Catipovic, and D. Webb. Autonomous ocean sampling networks. *Oceanography*, 6(3):86–94, 1993.

[27] F. Daum. New exact nonlinear filters. In James C. Spall, editor, *Bayesian Analysis of Time Series and Dynamic Models*, pages 199–226. Marcel Dekker, Inc., 1988.

[28] A. C. Davison. *Bootstrap Methods and their Application.* Cambridge University Press, 1997.

[29] M. de Feo, A. Graziano, R. Miglioli, and A. Farina. IMMJPDA vs. MHT and Kalman filter with NN correlation: Performance comparison. *IEE Proceedings — Radar, Sonar Navigation*, 144(2):49–56, April 1997.

[30] D. E. Di Massa. *Terrain-Relative Navigation for Autonomous Underwater Vehicles.* PhD thesis, Massachusetts Institute of Technology, 1997.

[31] G. Dissanayake, M. Herbert, A. Stentz, and H. Durrant-Whyte. Map building and terrain-aided localization in an underground mine. In *International Conference on Field and Service Robotics*, pages 34–40, Canberra, Australia, 1997. Panther Publishing and Printing.

[32] O. Egeland, M. Dalsmo, and O. J. Sordalen. Feedback control of a nonholonomic underwater vehicle with a constant desired configuration. *International Journal of Robotics Research*, 15(1), February 1996.

[33] R. Fierro and F. L. Lewis. A framework for hybrid control design. *IEEE Trans. Systems, Man, and Cybernetics, Part C*, 27(6):776–772, November 1997.

[34] E. Fortin. Those damn mines. *Proceedings of the U. S. Naval Institute*, 118(6):30–34, July 1992.

[35] T. Fortmann, Y. Bar-shalom, M. Scherre, and S. Gelfand. Detection thresholds for tracking in clutter — a connection between estimation and signal processing. *IEEE Trans. Automatic Control*, 30(3):221–228, March 1985.

[36] T. I. Fossen and O.-E. Fjellstad. Robust adaptive control of underwater vehicles: A comparative study. *Identification and Control*, 17(1), January 1996.

[37] D. Fryxell, P. Oliveira, A. Pascoal, and C. Silvestre. Integrated approach to the design and analysis of navigation, guidance, and control systems for autonomous underwater vehicles. *IEE Computing and Control Division Colloquium on Control and Guidance of Remotely Operated Vehicles*, (124):1/1–1/3, June 1995.

[38] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns.* Addison-Wesley, 1995.

[39] A. C. Gelb. *Applied Optimal Estimation.* The MIT Press, 1973.

[40] S. Gelfand, T. Fortmann, and Y. Bar-shalom. Adaptive detection threshold optimization for tracking in clutter. *IEEE Trans. Aerospace and Electronic Systems*, 32(2):514–522, April 1996.

[41] 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.

[42] R. L. Greenspan. GPS and inertial navigation. In *Global Positioning System: Theory and Applications*, volume 2, chapter 7. American Institute of Aeronautics and Astronautics, 1996.

[43] H. Greiner, A. Shectman, C. Won, R. Elsley, and P. Beith. Autonomous legged underwater vehicles for near land warfare. In *IEEE Symposium on Autonomous Underwater Vehicle Technology*, Monterey, CA, USA, 1996. IEEE.

[44] J. Hallam. Resolving observer motion by object tracking. In *Proc. Int. Joint Conf. Artificial Intelligence*, 1983.

[45] A. Heathershaw, D. Waymount, R. Rogers, and D. Curtis. A surveying decision aid for optimum ocean sampling strategies. In *Undersea Defence Technology*, pages 15–21, 1997.

[46] P. Hebert, S. Betge-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.

[47] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte. A new approach for filtering nonlinear systems. In *American Control Conference*, pages 1628–1632, June 1995.

[48] R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. In H. W. Sorenson, editor, *Kalman Filtering: Theory and Application*, pages 34–47. IEEE Press, 1985.

[49] B. Kamgar-Parsi, L. Rosenblum, F. Pipitone, L. Davis, and J. Jones. Toward an automated system for a correctly registered bathymetric chart. *IEEE J. Ocean Engineering*, 14(4):314–325, October 1989.

[50] P. G. Kaminski, A. E. Bryson, and S. F. Schmidt. Discrete square root filtering: A survey of current techniques. In H. W. Sorenson, editor, *Kalman Filtering: Theory and Application*, pages 210–218. IEEE Press, 1985.

[51] S. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory.* Prentice Hall, 1993.

[52] P. Kelsen. Optimal parallel algorithm for maximal matching. *Information Processing Letters*, 52(4):223–228, November 1994.

[53] M. Khatib, B. Bouilly, T. Simeon, and R. Chatila. Indoor navigation with uncertainty using sensor-based motions. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 3379–3384, April 1997.

[54] C. Klaassen. *Statistical Performance of Location Estimators.* The Mathematical Centre, Amsterdam, 1980.

[55] R. Kulhavy. *Recursive Nonlinear Estimation.* Springer, 1996.

[56] Y. Kuniyoshi, M. Inaba, and H. Inoue. Learning by watching: Extracting reusable task knowledge from visual observation of human performance. *IEEE Transactions on Robotics and Automation*, 10(6), December 1994.

[57] 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.

[58] T. Kurien. Framework for integrated tracking and identification of multiple targets. In *The AIAA Digital Avionics Systems Conference*, October 1991.

[59] J. Lakos. *Large-Scale C++ Software Design.* Addison-Wesley, 1995.

[60] D. Lane, M. Chantler, and D. Dai. Robust tracking of multiple objects in sector-scan sonar image sequences using optical flow motion estimation. *IEEE J. Ocean Engineering*, 23(1):31–46, January 1998.

[61] P. M. Lee. *Bayesian Statistics: An Introduction.* Arnold, 1989.

[62] J. J. Leonard. *Directed Sonar Sensing for Mobile Robot Navigation.* PhD thesis, University of Oxford, 1990.

[63] J. J. Leonard, B. A. Moran, I. J. Cox, and M. L. Miller. Underwater sonar data fusion using an efficient multiple hypothesis algorithm. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 2995–3002, May 1995.

[64] X. Li and Y. Bar-Shalom. Detection threshold selection for tracking performance optimization. *IEEE Trans. Aerospace and Electronic Systems*, 30(3):742–749, July 1994.

[65] D. B. Marco, A. J. Healey, and R. B. McGhee. Autonomous underwater vehicles: Hybrid control of mission and motion. *Autonomous Robots*, 3(2-3):169–186, June-July 1996.

[66] Brenan J. McCarragher. Hybrid modeling and simulationfor the design of an advanced industrial robot controller. *IEEE Robotics and Automation Magazine*, 4(2):27–44, June 1997.

[67] M. Medeiros and R. Carpenter. High resolution array signal processing for AUVs. In *AUV 96*, pages 10–15, 1996.

[68] T. M. Melia. *Damn the Torpedoes: a short history of U.S. naval mine countermeasures.* Washington, D.C.: Naval Historical Center, Dept. of the Navy, 1991.

[69] J. M. Mendel. Computational requirements for a discrete Kalman filter. In H. W. Sorenson, editor, *Kalman Filtering: Theory and Application*, pages 219–229. IEEE Press, 1985.

[70] M. L. Miller, H. S. Stone, and I. J. Cox. Optimizing Murty's ranked assignment method. *IEEE Trans. Aerospace and Electronic Systems*, 33(3):851–862, July 1997.

[71] J. Minkoff. *Signals, Noise, and Active Sensors: Radar, Sonar, Laser Radar.* John Wiley and Sons, Inc., 1997.

[72] P. Moore and L. Bivens. The bottlenose dolphin: Nature's ATD in SWMCM autonomous sonar platform technology. In *Proceedings of the Autonomous Vehicles in Mine Countermeasures Symposium*, pages 6–63 – 6–67. Naval Postgraduate School, 1995.

[73] B. A. Moran. *Underwater Shape Reconstruction in Two Dimensions.* PhD thesis, Massachusetts Institute of Technology, 1994.

[74] 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.

[75] S. Mori, C. Chong, E. Tse, and R. Wishner. Tracking and classifying multiple targets without *a priori* identification. *IEEE Trans. on Automatic Control*, AC-31(5), May 1986.

[76] 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.

[77] 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.

[78] K. G. Murty. An algorithm for ranking all the assignments in order of increasing cost. *Operations Research*, 16:682–687, 1968.

[79] D. Musser and A. Saini. *Standard Template Library Tutorial and Reference Guide.* Addison Wesley, 1996.

[80] N. Nahi. *Estimation Theory and Applications.* John Wiley and Sons, 1969.

[81] X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. An approach to the description and analysis of hybrid systems. In R. Grossman, A. Nerode, A. Ravn, and H. Rischel, editors, *Hybrid Systems*, pages 149–178. Springer, 1993.

[82] Office of Naval Research. Very shallow water and surf zone MCM. http://www.onr.navy.mil/sci_tech/ocean/info/VSWSZMCM/, March 1998.

[83] L. E. Parker. Effect of action recognition and robot awareness in cooperative robotic teams. In *IEEE International Conference on Intelligent Robots and Systems*, Pittsburgh, PA, USA, 1995. IEEE.

[84] A. Pole, M. West, and P. J. Harrison. Nonnormal and nonlinear dynamic Bayesian modeling. In James C. Spall, editor, *Bayesian Analysis of Time Series and Dynamic Models*, pages 167–198. Marcel Dekker, Inc., 1988.

[85] W. Press, S. Teukosky, W. Vetterling, and B. Flannery. *Numerical Recipes in C.* Cambridge University Press, 1992.

[86] B. Quine, J. K. Uhlmann, and H. F. Durrant-Whyte. Implicit jacobians for linearized state estimation in nonlinear systems. In *American Control Conference*, pages 1645–1646, June 1995.

[87] D. B. Reid. An algorithm for tracking multiple targets. *IEEE Trans. on Automatic Control*, AC-24(6), Dec. 1979.

[88] 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.

[89] S. I. Sagatun and R. Johansson. Optimal and adaptive control of underwater vehicles. *Advanced Robotics*, 10(3):283–299, 1996.

[90] L. Schwartz. Nonlinear filtering and comparison with kalman filtering. In C. T. Leondes, editor, *Theory and Applications of Kalman Filtering*, pages 143–162. NATO Advisory Groupe for Aerospace Research and Development, 1970.

[91] J. Shao. *The Jackknife and Bootstrap.* Springer Verlag, 1995.

[92] R. D. Short. The modern intelligent mine and its implications for mine warfare vessels. In *International Conference on Mine Warfare Vessels and Systems 2*, London, 1989. The Royal Institution of Naval Architects.

[93] R. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56, 1987.

[94] R. Smith, M. Self, and P. Cheeseman. A stochastic map for uncertain spatial relationships. In *4th International Symposium on Robotics Research.* MIT Press, 1987.

[95] 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.

[96] T. Sobh and B. Benhabib. Discrete event and hybrid systems in robotics and automation: An overview. *IEEE Robotics and Automation Magazine*, 4(2):16–19, June 1997.

[97] H. W. Sorenson. Least-squares estimation: from Gauss to Kalman. *IEEE Spectrum*, 7(7):63–68, July 1970.

[98] H. W. Sorenson. Kalman filtering techniques. In H. W. Sorenson, editor, *Kalman Filtering: Theory and Application*, pages 90–126. IEEE Press, 1985.

[99] H. W. Sorenson. Recursive estimation for nonlinear dynamic systems. In James C. Spall, editor, *Bayesian Analysis of Time Series and Dynamic Models*, pages 127–166. Marcel Dekker, Inc., 1988.

[100] W. K. Stewart. *Multisensor Modeling Underwater with Uncertain Information.* PhD thesis, Massachusetts Institute of Technology, 1988.

[101] S. Sukkarieh, E. M. Nebot, and H. F. Durrant-Whyte. The GPS aiding of INS for land vehicle navigation. In *International Conference on Field and Service Robotics*, pages 278–285, Canberra, Australia, 1997. Panther Publishing and Printing.

[102] S. T. Tuohy. A simulation model for AUV navigation. In *Proceedings of the 1994 Symposium on Autonomous Underwater Vehicle Technology*, pages 470–478. The Oceanic Society of the IEEE, IEEE, July 1994.

[103] S. T. Tuohy, N. M. Patrikalakis, J. J. Leonard, J. G. Bellingham, and C. Chryssostomidis. AUV navigation using geophysical maps with uncertainty. In *Proc. Int. Symp. on Unmanned Untethered Submersible Technology*, pages 265–276, 1993.

[104] J. K. Uhlmann. *Dynamic Map Building and Localization: New Theoretical Foundations.* PhD thesis, University of Oxford, 1995.

[105] T. Uno. Algorithms for enumerating all parfect, maximum, and maximal matchings in bipartite graphs. *Lecture Notes in Computer Science*, 1350:92, 1997.

[106] P. Vuillod, L. Benini, and G. De Micheli. Generalized matching from theory to application. In *Proceedings of the 1997 IEEE/ACM International Conference on Computer-Aided Design, ICCAD*, pages 13–20, San Jose, CA, USA, November 1997.

[107] C. Yang, P. Bertrand, and M. Mariton. A failure detection method for systems with poorly known parameters. In *IEEE International Conference on Control and Applications (ICCON '89)*, Jerusalem, April 1989.