REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	19.Jun.06	THESIS

4. TITLE AND SUBTITLE
RECOVERING SAMPLE DIVERSITY IN RAO-BLACKWELLIZED PARTICLE FILTERS FOR SIMULTANEOUS LOCALIZATION AND MAPPING.

5. FUNDING NUMBERS

6. AUTHOR(S)
2D LT ANDERSON ANDREW D

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

8. PERFORMING ORGANIZATION REPORT NUMBER
CI04-1787

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
THE DEPARTMENT OF THE AIR FORCE
AFIT/CIA, BLDG 125
2950 P STREET
WPAFB OH 45433

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Unlimited distribution
In Accordance With AFI 35-205/AFIT Sup 1

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

14. SUBJECT TERMS	15. NUMBER OF PAGES
	111
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

Recovering Sample Diversity in Rao-Blackwellized Particle Filters for Simultaneous Localization and Mapping

by
2d Lt Andrew D. Anderson, USAF
B.S. Astronautical Engineering
United States Air Force Academy, 2004

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of
Master of Science in Aeronautics and Astronautics
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 2006

Author .....

Department of Aeronautics and Astronautics
May 26, 2006

Certified by .....

Don Gustafson, Ph.D.
Distinguished Member of the Technical Staff
The Charles Stark Draper Laboratory, Inc.
Thesis Supervisor

Certified by .....

John Deyst, Ph.D.
Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by .....

Jaime Peraire
Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Students

Recovering Sample Diversity in Rao-Blackwellized Particle Filters for Simultaneous Localization and Mapping

by

2d Lt Andrew D. Anderson, USAF

Submitted to the Department of Aeronautics and Astronautics
on May 26, 2006, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

This thesis considers possible solutions to sample impoverishment, a well-known failure mode of the Rao-Blackwellized particle filter (RBPF) in simultaneous localization and mapping (SLAM) situations that arises when precise feature measurements yield a limited perceptual distribution relative to a motion-based proposal distribution. One set of solutions propagates particles according to a more advanced proposal distribution that includes measurement information. Other methods recover lost sample diversity by resampling particles according to a continuous distribution formed by regularization kernels.

Several advanced proposals and kernel shaping regularization methods are considered based on the RBPF and tested in a Monte Carlo simulation involving an agent traveling in an environment and observing uncertain landmarks. RMS error of range-bearing feature measurements was reduced to evaluate performance during proposal-perceptual distribution mismatch. A severe loss in accuracy due to sample impoverishment is seen in the standard RBPF at a measurement range RMS error of 0.001 m in a 10 m $\times$ 10 m environment. Results reveal a robust and accurate solution to sample impoverishment in an RBPF with an added fixed-variance regularization algorithm. This algorithm produced an average 0.05 m improvement in agent pose CEP over standard FastSLAM 1.0 and a 0.1 m improvement over an RBPF that includes feature observations in formulation of a proposal distribution.

This algorithm is then evaluated in an actual SLAM environment with data from a Swiss Ranger LIDAR measurement device and compared alongside an extended Kalman filter (EKF) based SLAM algorithm. Pose error is immediately recovered in cases of a 1.4 m error in initial agent uncertainty using the improved FastSLAM algorithm, and it continues to maintain an average 0.75 m improvement over an EKF in pose CEP through the scenario.

Thesis Supervisor: Don Gustafson, Ph.D.
Title: Distinguished Member of the Technical Staff

The Charles Stark Draper Laboratory, Inc.

Thesis Supervisor: John Deyst, Ph.D.
Title: Professor of Aeronautics and Astronautics

Acknowledgments

This research project would not have been possible without the generous support of Draper Laboratories. The opportunity to explore a broad range of research projects at Draper and interact with the talented engineers at the helm of these projects has made this two year fellowship an unforgettable experience. I am grateful to Don Gustafson for his guidance and insight into the SLAM problem, his natural teaching ability, engineering expertise, and also his willingness to let me run with an idea. Don guided me through an intense year of research and his support throughout this thesis was a blessing. I would also like to thank Dale Landis for leading me through most of the first year. Dale is not only a brilliant and dedicated engineer but an enjoyable person to talk with, about practically any subject. The members of the SLAM IR&D team, under the leadership of Marc McConley, provided valuable insight and ideas throughout the course of this project, and I am thankful for their support: Paul Debitetto, Scott Rasmussen, Pete Lommel, and Greg Andrews. Greg not only developed the EKF code and feature extraction algorithm for the Swiss Ranger, but went out of his way to provide additional MATLAB assistance. This thesis would not have been possible without his effort. I would also like to thank a number of Draper fellows and engineers for their friendship during this experience: Corban Bryant, Drew Barker, Jon Beaton, Matt Hale, Garrett Teahan, Susan Kim, Phil Springmann and Ellis King. Finally, to my family and to my lovely bride, I thank you most of all for believing in me and encouraging me throughout this endeavor.

This thesis was prepared at The Charles Stark Draper Laboratory, Inc., under Internal Research and Development Project 20318-001, GCC DLF Support.

Publication of this thesis does not constitute approval by Draper or the United States Air Force of the findings or conclusions contained herein. It is published for the exchange and stimulation of ideas.

Andrew D. Anderson — 26 May 2006 (Date)

ASSIGNMENT

Draper Laboratory Report Number T-1559

In consideration for the research opportunity and permission to prepare my thesis by and at The Charles Stark Draper Laboratory, Inc., I hereby assign my copyright of the thesis to The Charles Stark Draper Laboratory, Inc., Cambridge, Massachusetts.

Andrew D. Anderson Date

Contents

1 Introduction **17**

1.1 Joint Estimation 17

1.2 SLAM Applications 18

1.3 Bayesian Estimation 22

1.4 EKF SLAM 22

1.5 Particle Filter SLAM 23

1.6 Particle Filter Limitations 25

1.7 Thesis overview 27

2 Simultaneous Localization and Mapping **29**

2.1 SLAM Fundamentals 30

2.1.1 Extended Kalman Filter SLAM 31

2.1.2 Limits of EKF SLAM 32

2.2 The Particle Filter 34

2.2.1 Particle Filters for Agent Pose Tracking 34

2.2.2 Resampling 35

2.3 Particle Filter SLAM 36

2.3.1 Importance Weight Calculation 38

2.3.2 Data Association 39

2.3.3 Preventing Filter Degeneracy 40

2.4 FastSLAM vs. EKF SLAM 40

2.5 FastSLAM Challenges 41

2.5.1 Effects of Sample Impoverishment 42

2.5.2 Overcoming Sample Impoverishment 43

3 Recovering Sample Diversity **45**

3.1 Sample Impoverishment Revisited 45

3.2 Alternative Proposal Distributions for Position Tracking 47

3.2.1 Auxiliary Particle Filter 47

3.2.2 Local-Linearized Particle Filter 48

3.2.3 Mixture Monte Carlo Localization 48

3.3 Alternative Proposal Distributions for SLAM 49

3.4 Regularization 52

3.4.1 Regularized Particle Filter 53

3.4.2 Markov Chain Monte Carlo Criterion 54

3.5 Other Regularization Approaches 55

3.5.1 Fixed-Gaussian Regularization 55

3.5.2 Adaptive Regularization 56

3.5.3 Other Adaptive Regularization Techniques 57

3.5.4 Process Noise 58

4 Simulated Results for Sample Diversity Recovery Methods **59**

4.1 Assumptions and Simulation Setup 60

4.1.1 Development of the SLAM Environment 60

4.1.2 Robot Motion 61

4.1.3 Simulated Measurements 63

4.1.4 Performance Metrics 63

4.2 SLAM Posterior Estimation 63

4.2.1 Sample Impoverishment and Particle Drift 64

4.3 SLAM Algorithms 67

4.3.1 FastSLAM 1.0 67

4.3.2 FastSLAM 2.0 68

4.3.3 Auxiliary Particle Filter 68

4.3.4 Regularization Algorithms 68

4.4 Filter Accuracy and Diversity Analysis 69
4.4.1 FastSLAM 1.0 with Regularization 70
4.4.2 FastSLAM 2.0 Accuracy and Diversity Analysis 73
4.5 Monte Carlo SLAM Performance Analysis 75
4.5.1 Filter Performance Results 75
4.6 Regularization Performance Results 78
4.6.1 SpreadX Parameter Selection 78
4.7 Markov Chain Monte Carlo (MCMC) Analysis 81
4.8 Performance Summary for Filter/Regularization Marriages 82
4.9 Summary 84

5 Experimental Results for SLAM Algorithms **87**
5.0.1 Swiss Ranger Feature Observations 88
5.0.2 Algorithm Specifics 89
5.0.3 Initial Estimates and Anchor Features 89
5.0.4 Performance Metrics 90
5.1 Experimental Results 90
5.1.1 Scenario One: Position Tracking and Feature Mapping 90
5.1.2 Scenario Two: Localization 93
5.1.3 Scenario Three: Localization and Mapping 94
5.2 Summary 97

6 Conclusions and Future Work **103**
6.1 Research Conclusions 103
6.2 Future Work 104

List of Figures

1-1 Typical SLAM Scenario, Pose and Landmark Uncertainty Correlation 19

1-2 Typical SLAM Scenario, Correct Association of a Measurement with a Previously Observed Landmark 20

1-3 Matching Proposal and Posterior Distributions 25

1-4 Mismatch Between Proposal and Posterior Distributions 27

2-1 SIR Particle Filter 36

2-2 FastSLAM Factorization 37

2-3 Mismatch Between Proposal and Target Distributions 42

2-4 Incorrect Data Association and False Landmark Initialization 44

3-1 FastSLAM 2.0 Proposal Distribution 52

3-2 Gaussian Regularization 56

3-3 Adaptive Kernel Regularization 57

4-1 SLAM Environment 61

4-2 True Agent Path 62

4-3 SLAM Posterior Estimation 65

4-4 Sample Impoverishment and Particle Drift 66

4-5 End Result of Sample Impoverishment and Particle Drift 67

4-6 FastSLAM 1.0: CEP and Diversity 71

4-7 FastSLAM 1.0 + Regularization: CEP and Diversity 72

4-8 FastSLAM 2.0: CEP and Diversity 74

4-9 FastSLAM 2.0 Posterior Estimation 76

4-10 Monte Carlo Analysis: Filter Types 77
4-11 Auxiliary Particle Filter: CEP and Diversity 78
4-12 Monte Carlo Analysis: Regularization Algorithms 80

5-1 Environment Truth, Initial Estimate for Experimental Scenario One . 91
5-2 Final Posterior Estimate, Scenario One 92
5-3 Position CEP Time History for SLAM Scenario One 93
5-4 Initial Posterior Estimate, Including Anchor Features, for SLAM Scenario Two 95
5-5 Final Posterior Estimate, Scenario Two 96
5-6 Position CEP Time History for SLAM Scenario Two 97
5-7 Environment Truth, Initial Estimate for Experimental Scenario Three 99
5-8 Final Posterior Estimate, Scenario Three 100
5-9 Position CEP Time History for SLAM Scenario Three 101

List of Tables

4.1 Initial Pose and Anchor Feature Uncertainty 60

4.2 Agent Motion Model Parameters 62

4.3 Agent Measurement Model 63

4.4 Summary of Regularization Methods 69

4.5 SpreadX Regularization Parameter Determination 79

4.6 MCMC Criterion Performance Analysis 82

4.7 CEP Results for Filter-Regularization Marriages 83

5.1 Motion Model Parameters for Swiss Ranger SLAM Environment . . . 88

5.2 Swiss Ranger/Feature Extraction Measurement Specifications 89

Chapter 1

Introduction

A complicated but increasingly relevant scenario in robotic navigation and exploration involves an agent traveling without the aid of an absolute positioning system or an accurate map of the environment. To produce a globally consistent map, an agent must gather information about its surroundings through relative observations of local features. By combining these measurements with a correct notion its position and heading at the time of each observation, it can create a proper spatial model of the environment [46]. In a related manner, an agent can use relative observations of features in the environment to infer position and heading, but only when measurements are correctly associated with entities stored in an accurate *a priori* map. When neither the agent path nor the environment map are provided and must instead be estimated jointly, a unique correlation develops and errors in each state are linked [32].

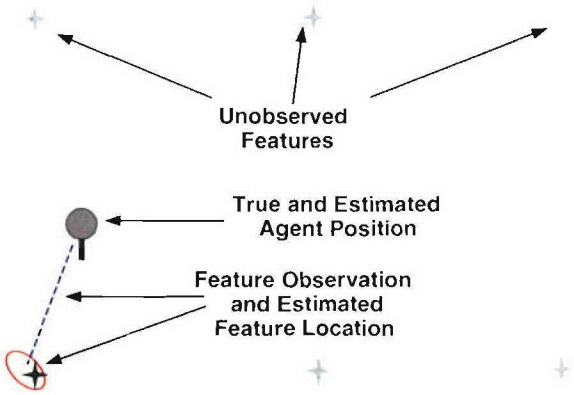
1.1 Joint Estimation

In a conventional mapping situation with an accurate position estimate at all times, a robot will measure the location of different features as it travels through the environment, storing the positions of these landmarks in an estimated map of an area. Since the true path is known, measurements between one state and another are statistically independent. Making more measurements of a state, such as a landmark position,

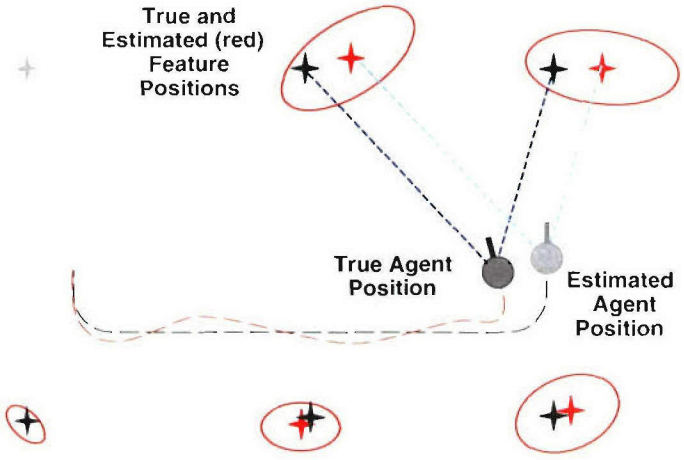
will only provide a better estimate of the state and will not affect the knowledge of any other state. When the true path of the agent is unknown and must be estimated along with each landmark, all states in the estimation problem become statistically dependent. Any error in the robot pose estimate at the time relative measurement is processed will have a systematic effect on the accuracy of the landmark estimate [46]. If pose error is not mitigated with the measurement of a well-localized landmark or an absolute position reference, this systematic error in the map will build over time as control errors accumulate, making it difficult or impossible for the agent to produce a consistent map. An illustration of this dilemma is shown in figure 1-1. An agent starts from a well-localized position and measures a feature in the environment. At this point, all other features are unknown. With an accurate estimate of the agent pose at the time of the first relative observation, there is little doubt in the location of the landmark. Over time, robot control errors lead to an increased uncertainty in agent pose. Statistical dependence inherent to the joint estimation problem leads to increased uncertainty of future landmark positions, denoted by larger red ellipses. Both the problem and approach involving joint estimation of agent pose and local map are referred to as Simultaneous Localization and Mapping (SLAM) [11, 32] or Concurrent Mapping and Localization (CML) [46, 45]. SLAM estimation algorithms take advantage of this statistical correlation between pose and landmark uncertainty. When the agent observes and correctly identifies a previously mapped feature, shown in figure 1-2, the agent position error is corrected. Because of the statistical correlation between agent pose and landmark position, the uncertainties of all other estimated landmarks are also reduced.

1.2 SLAM Applications

In many navigation situations, a full SLAM solution, both agent pose information and local landmark positions, may not be necessary. Obvious circumstances include many aerospace or open-field environments with unobstructed access to signals from Global Positioning System (GPS) satellites, views of stars contained in a star tracker



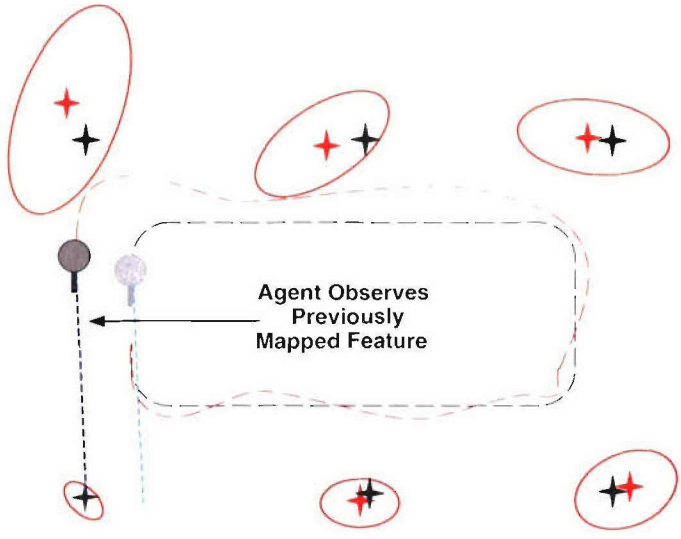
(a)



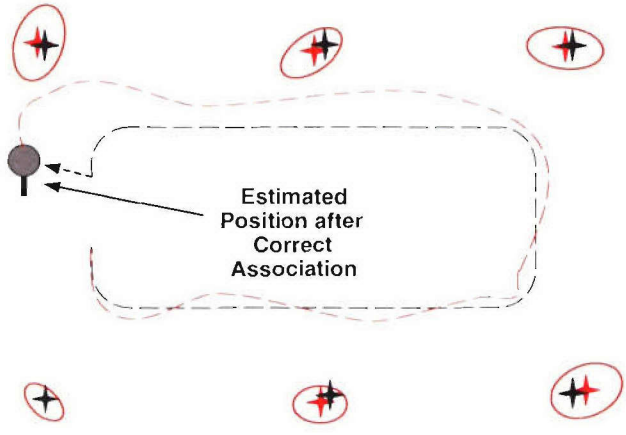
(b)

Figure 1-1: A typical SLAM scenario with accurate initial agent position estimate in (a). Over time agent uncertainty increases, leading to further uncertainty in feature positions as they are observed (b).

database, or any other absolute referencing system [11]. With the increasing availability of high-resolution satellite imagery, detailed maps of observable features can be provided as an additional absolute reference. However, the benefits of SLAM algorithms extend beyond the strict pose-and-map estimation explained previously. Consider a robot, vehicle, or even a human traveling through an urban environment, equipped with inertial measurements from accelerometers and gyroscopes (INS), and



(a)



(b)

Figure 1-2: The agent makes one full loop and returns to a previously mapped feature (a). If a correct assocation is made between a measurement and a previously observed landmark, overall uncertainty in the agent pose and map are reduced (b).

a suite of other measurement devices designed to augment inertial measurements: Doppler radar, wheel encoders, GPS, or an image-based pseudo-inertial measurement system. Of all the measurements in this system, only GPS provides an absolute position reference, and in an urban canyon this signal could be intermittent, reflected

by buildings (multipath), or completely blocked. During GPS outages from blocked signals, a vehicle interested in self-localization must continue to navigate using only relative measurements of rotation, velocity and acceleration from the other devices [11]. Errors in these measurements will quickly propagate over time since parameters must be integrated to determine the agent pose estimate. Unless an additional constraint is added to certain parameters of the navigation filter, error will grow without bound until another absolute position fix is obtained. In this scenario, a SLAM-based approach may help preserve the integrity of the navigation system during long periods without GPS measurements. Mapping local features with accurate GPS-determined path information will create a database of landmarks that can be used to maintain an acceptable pose estimate when GPS signals are blocked by natural or manmade obstructions [11]. Indoor environments, on the other hand, completely prevent GPS signal positioning, and in some cases an accurate floor plan or map of the building may not be available. Navigation in unfamiliar buildings with the contemporary measurement suite described above would be next to impossible unless a map of features is estimated along with the agent pose [1, 32].

Robotic platforms are now sent to the frontiers of exploration as advancements in structural technologies and design permit robust operation in austere environments. These areas, where prior maps are too difficult, costly, or dangerous to procure, present some of the most promising areas for the implementation of SLAM-based algorithms [32]. Particular target environments for SLAM approaches include undersea autonomous vehicles, robotic exploration of mines [30], and autonomous navigation on extraterrestrial planets. In general, independent localization and mapping is a necessary prerequisite to completely autonomous operation of mobile robotics in any situation. Currently, many advanced estimation algorithms exploit the flexibility provided by an implementation of a SLAM approach [47].

1.3 Bayesian Estimation

The most widespread and successful branch of SLAM estimation algorithms employ probabilistic techniques, meaning that they estimate a posterior probability distribution over all possible maps and all possible poses [39, 46]. Each agent control or environmental observation can be thought of as a probabilistic constraint [32]. This implies that the set of all possible agent poses at any time is reduced as more information is obtained about either the robot's motion or its surroundings. In the limit of an infinite amount of such information, the set of all possible SLAM posteriors converges to one agent pose and one map. Bayes' theorem is a recursive formula that incorporates sensor and control information to adjust the posterior probability distribution, accounting for any measurements that are available at a given time [14, 32, 39]. In this respect, Bayesian estimation in its purest form is a flexible estimation architecture that can update an estimate with any information that can be mathematically related to the posterior. Additionally, the recursive nature of Bayesian estimators is ideal for online applications. Since the agent pose and map estimates evolve from the posterior at the previous time step, all other past estimates can be forgotten. Finally, Bayes' filter can be used to estimate a state of any size, restricted only by the computational limitations of the navigation computer. Unfortunately, the estimation integral forming the basis of the Bayes' filter cannot be computed in closed form [14, 32]. Many Bayesian algorithms solve this by restricting the form of the posterior, motion model or measurement model. Others employ alternative sampling techniques to approximate the Bayesian posterior without making these limitations. Two popular SLAM algorithms that typify each approach are the extended Kalman filter (EKF) and the particle filter.

1.4 EKF SLAM

The Kalman filter is an optimal Bayesian estimator that operates under the strict assumptions of a Gaussian posterior probability distribution and linear motion and

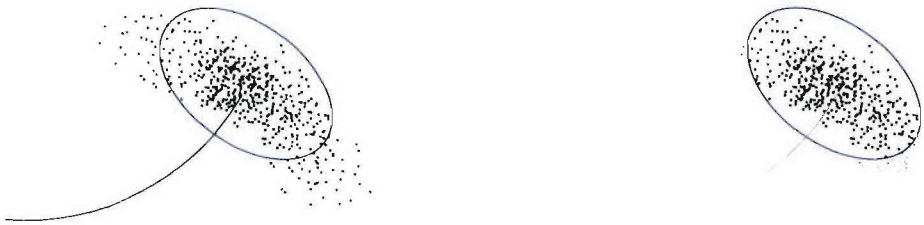
measurement models [32, 39]. Linearization of nonlinear motion and measurement models results in the extended Kalman filter, an analytical approximation of Bayes' filter. The recursive solution provided by the extended Kalman filter is sufficient if the posterior probability distribution for SLAM states can be adequately characterized by the uni-modal Gaussian parameters of mean and covariance [39, 47]. Uncertainty in an EKF SLAM algorithm is stored in a covariance matrix, with not only individual state uncertainty but also correlations of uncertainties between states. Unfortunately, in many scenarios a simple Gaussian distribution does not adequately encapsulate the full posterior probability distribution. Indoor navigation environments provide constraints in the form of physical obstructions or walls. Pure Gaussian uncertainty implies a small chance that the agent could be inside the wall, or outside the building in mid-air. In addition, Gaussian uncertainty carries only one mean, or most likely estimate, for a particular state. In many cases, such as the global localization problem, there is an equally likely chance that the robot could be at many points in the environment, and each of these points must be given equal consideration until more information is gathered [45].

1.5 Particle Filter SLAM

The particle filter is an approach to the nonlinear estimation problem that represents posterior probability with a large number of discrete, evenly weighted samples [14, 32, 39]. In the SLAM case, each sample is a hypothesis of the posterior (an agent pose and a corresponding set of landmarks) that is propagated according to a motion model and then weighted based on how well the hypothesis agrees with a target distribution [32]. The target distribution, in most formulations, is directly related to feature observations [39]. Successful particle filtering algorithms typically draw a new set of particles after weights have been assigned. In particle filters, uncertainty of the state is stored in dispersion of these uniformly weighted samples; a broader spread implies a more uncertain estimate. Consequently, multi-modal distributions from state constraints or nonlinear propagation can be approximated easily. An

example of particle filter propagation is shown in figure 1-3. In this example, a large number of particles are drawn from the prior agent pose according to a probabilistic motion model representing uncertainty in agent movement. A feature observation isolates the pose hypotheses that agree, and these particles are given larger weights. After resampling, most of these particles will be duplicated, whereas particles outside the blue ellipse will likely be eliminated. Surviving particles are then propagated according to agent control information at the next time step, and the process repeats.

The Rao-Blackwellized particle filter (RBPF) is a specific type of particle filter that, in the context of SLAM, updates pose information with a particle filter and landmark information with a number of low-dimensional EKFs. The distinct advantage of the RBPF over standard particle filters is that it scales well to mapping problems of high-dimensionality [32]. It does this by marginalizing the posterior and eliminating cross-correlations between landmarks [4]. Since each sample in the particle filter is an estimate of the true position, landmarks measurements become conditionally independent. Advantages of the RBPF SLAM concoction include the ability to represent an arbitrarily complex posterior distribution of the agent pose, as well as many independent estimates of an environment map. As mentioned previously, this property may be particularly useful in cases of indoor localization and mapping. Additionally, the RBPF, as with other particle filters, converges to the optimal Bayesian posterior in the limit of infinite particles [39]. As computational power increases, estimators based on particle filtering will only improve their characterization of the posterior. Unlike the basic EKF approach, the computational complexity of the RBPF scales linearly with the dimension of the state, allowing favorable application to online SLAM scenarios [31]. Most importantly, the application of RBPF based SLAM algorithms has demonstrated solutions to two previously unsolvable problems in robot localization: global localization, and the kidnapped robot problem [47]. Both problems take advantage of the multiple hypothesis nature of the RBPF to determine true position under initial global uncertainty.



(a) (b)

Figure 1-3: Samples drawn from a probabilistic motion model (a) with a blue ellipse representing the measurement. In this case, the proposal and target distributions match well and particle diversity is preserved in (b).

1.6 Particle Filter Limitations

Despite the advantages that the RBPF brings to SLAM, it also brings certain complications that are currently difficult to overcome. A particular failure mode for the particle filter occurs when the proposal distribution (in most cases characterized by the motion model) and the target distribution (from a feature observation) are mismatched, usually from an accurate sensor measurement. This scenario is becoming increasingly relevant as current trends in inertial systems produce smaller, chip-based accelerometers and gyros [11]. Though small and dependable, these systems are often plagued with errors, including bias, scale factor, and random walk processes. At the same time, measurement devices have only become more accurate and precise, especially ranging systems based on Lidar- Light Detection and Ranging. Moreover, it is generally more feasible to implement accurate sensor technologies than to fully predict the motion characteristics of a complex robotic platform especially as it travels through an uncertain environment [11].

A particle filter will incorporate accurate sensor information into the SLAM posterior estimate by reproducing the particles that correspond to the measurement and eliminating others. In effect, a particle filter continually builds and trims a set of individual estimates of the true agent trajectory [32]. With an infinite number of particles, this trimming of conflicting possibilities for agent position would favorably

resolve the estimated state. In this case, an extremely accurate feature observation device would be ideal, since, hypothetically, there are still an infinite number of particles preserved in the process. Since practical implementations are restricted to a finite number of particles, this trimming reduces the number of discrete possibilities as particles are relocated to these few unique points. Coupling a noisy motion model with an accurate measurement device will only reduce the number of unique points that align with the target distribution. If the diversity lost in this process is not recovered, particles could eventually coalesce to one single trajectory. Since uncertainty is stored in the dispersion of the particle cloud, the filter is assuming perfect knowledge of the true state, which is obviously untrue. This failure mode of particle filters, also known as sample impoverishment or particle depletion, can lead to particle drift, incorrect associations between measurements and landmarks, false landmark creation, and a general loss of pose and map accuracy [16, 21, 32, 39, 41, 43].

In figure 1-4, particles are drawn according to a probabilistic motion model as in figure 1-3. With a precise measurement, the size of the ellipse representing the target distribution is reduced. Only a few discrete points now match the highly selective target criterion. During resampling, most points will stack to these few locations. In a way, the particle filter has prevented degeneracy by relocating and sharpening the area of interest. While it is true that the resulting particle cloud will encircle the most likely pose of the agent, a particle filter estimates a state with discrete samples, not a continuous distribution. A finite number of samples means that there will always be unsampled "gaps" in filter coverage. In a strict probabilistic sense, a finite sample set also implies that the agent will not coincide exactly with one of these discrete samples [32]. In order to ensure that the filter continues to converge to an accurate representation of the agent state, an adequate level of sample diversity, or unique filter samples after resampling, must be maintained. If not, the filter is prone to the many side effects of sample impoverishment listed previously that will cause the estimate to diverge from the true posterior.

(a) (b)

Figure 1-4: Mismatch between proposal and target distributions, a consequence of accurate measurements. Only a few unique particles are resampled (b).

1.7 Thesis overview

It is the aim of this document to explore alternative methods for recovering lost sample diversity in Rao-Blackwellized particle filters and to analyze the effects of increased diversity on the overall posterior accuracy of the algorithm. A literature review provides two possible solutions.

The first approach seeks to prevent sample diversity by adopting a more advanced proposal distribution than that provided by only the agent motion model. It is the hope that by incorporating measurement information in proposal calculation, more particles will propagate to favorable regions for resampling based on the target distribution. There are several documented attempts at using measurement information to influence particle propagation, with only limited information relating to their performance in a strict SLAM scenario [30, 36, 37, 39].

The second attempt focuses on regaining lost diversity during resampling by drawing samples from a more continuous distribution. Instead of stacking on discrete points that receive high weights, particle locations are adjusted or "regularized" according to an additional draw from a regularization kernel. As a result, regions of the target distribution are more evenly populated with unique pose estimates that fill in the unsampled "gaps" before propagation.

This document presents the results of a research effort to characterize the role of sample diversity in overall RBPF accuracy in SLAM scenarios. In addition, alter-

native proposal distributions are combined with regularization methods to explore the performance of each combination and to find a robust and accurate solution for particle depletion.

Chapter 2

Simultaneous Localization and Mapping

The Simultaneous Localization and Mapping problem considers a robot moving through an unknown environment. In the most basic example, a robot executes controls and makes observations about the relative positions of local features, both of which are corrupted by noise. Were an accurate, detailed map of the environment available, the problem reduces to determining the true path by observing the relative positions of features [46]. Conversely, if the true position of the robot is known through GPS or some other means, a map of the observed environment could be deduced using these relative measurements [45]. The process of recovering both the robot path and the environment map from limited or no initial information becomes much more difficult. Pose uncertainty introduces systematic errors that contribute to the uncertainty of landmark positions mapped with robot observations [32]. Successful attempts at this problem have taken advantage of this correlation between pose and landmark uncertainty by estimating both states simultaneously [12, 39]. Accurate knowledge about the position of a landmark will reduce both pose uncertainty and the uncertainty of other landmarks [32].

2.1 SLAM Fundamentals

The goal of SLAM is to recover an estimate of the most recent robot pose, s_t and the locations of local landmarks, Θ, given the set of control and measurement information, $u^t = \{u_0, u_1, \ldots u_t\}$, and $z^t = \{z_0, z_1, \ldots z_t\}$ respectively. This SLAM posterior state is represented probabilistically as:

$$p(s_t, \Theta | z^t, u^t) \tag{2.1}$$

To develop a recursive, optimal estimator for this problem, the posterior distribution is modeled as a partially observable Markov chain [39, 13]. Under this assumption, the present state is dependent only on the previous state; all other past and future states are conditionally independent [46]. Expanding this posterior using Bayes' Theorem yields:

$$p(s_t, \Theta | z_t, u_t) = \eta p(z_t | s_t, \Theta) p(s_t, \Theta | z_{t-1}, u_t) \tag{2.2}$$

where η is a proportionality constant. Through a simple derivation, a recursive formula is developed that infers the SLAM posterior at any time t given knowledge of the state at time $t-1$. This elegant and widely used recursion is known as the Bayes' Filter [40]:

$$p(s_t, \Theta | z^t, u^t) = \eta p(z_t | s_t, \Theta) \int p(s_t | s_{t-1}, u_t) p(s_{t-1}, \Theta | z_{t-1}, u_{t-1}) ds_{t-1} \tag{2.3}$$

Under the Bayes' filter, the *a priori* distribution at time t-1 evolves according to a motion model, also known as a transitional density [32, 39, 45]:

$$p(s_t | s_{t-1}, u_t) \tag{2.4}$$

The observation model that relates incoming measurements to the evolved state is given by:

$$p(z_t | s_t, \Theta) \tag{2.5}$$

Though difficult or impossible to compute in closed-form, equation 2.3 can be approximated by restricting the form of the SLAM posterior to a Gaussian probability density function (pdf). When the motion and observation models can be regarded as linear functions of the current state with with only uncorrelated, zero-mean white noise, this recursion for the optimal Bayesian posterior becomes the Kalman filter [3, 18, 25, 12]. Linearization of non-linear motion and measurement models forms the basis of the extended Kalman filter, an analytic approximation of the optimal filter for non-linear situations [39].

2.1.1 Extended Kalman Filter SLAM

The EKF represents the SLAM posterior distribution as a high-dimensional multivariate Gaussian parameterized by a mean μ_t and covariance Σ_t for each state. The mean posterior is the state vector in equation 2.8 and contains agent pose information (2-d or 3-d position and heading) and the mean position estimate for each mapped landmark. State covariances and pairwise correlations between states are stored in the filter covariance matrix, equation 2.9.

$$p(s_t, \Theta | u_t, z_t) = N(x_t; \mu_t, \Sigma_t) \tag{2.6}$$

$$x_t = \{s_t, \theta_{1t}, \ldots, \theta_{Nt}\} \tag{2.7}$$

$$\mu_t = \{\mu_{s_t}, \mu_{\theta_{1t}}, \ldots, \mu_{\theta_{Nt}}\} \tag{2.8}$$

$$\Sigma_t = \begin{bmatrix} \Sigma_{s_t} & \Sigma_{s_t\theta_1} & \cdots & \Sigma_{s_t\theta_N} \\ \Sigma_{\theta_1 s_t} & \Sigma_{\theta_1} & \Sigma_{\theta_1\theta_2} & \\ \vdots & \Sigma_{\theta_2\theta_1} & \ddots & \\ \Sigma_{\theta_N s_t} & & & \Sigma_{\theta_N} \end{bmatrix} \tag{2.9}$$

The first step in evaluating the SLAM posterior within an EKF at any time t is to propagate the mean agent state at the previous time step according to the non-linear motion model $h(\mu_{t-1}, u_t)$, and propagate the covariance using the linearized motion model F_x and noise covariance of the motion model P_t. The Jacobian of the

non-linear measurement model $g(x_t, n_t)$, where x_t is the agent orientation and n_t is a data association between the measurement measurement and a landmark, is then evaluated at the state estimate μ_t^-. The remaining equations 2.14 - 2.17 involve the calculation of a Kalman gain K_t and the application of this gain to the updated mean in equation 2.16 and the updated covariance in equation 2.17.

$$\mu_t^- = h(\mu_{t-1}, u_t) \tag{2.10}$$

$$F_x = \nabla_{x_{t-1}} h(\mu_{t-1}, u_t)|_{x_{t-1}=\mu_{t-1}} \tag{2.11}$$

$$\Sigma_t^- = F_x \Sigma_{t-1} F_x^t + P_t \tag{2.12}$$

$$G_x = \nabla_{x_t} g(x_t, n_t)|_{x_t=\mu_t^-} \tag{2.13}$$

$$Z_t = G_x \Sigma_t^- G_x^T + R_t \qquad \hat{z}_{n_t} = g(\mu_t^-, n_t) \tag{2.14}$$

$$K_t = \Sigma_t^- G_x^T Z_t^{-1} \tag{2.15}$$

$$\mu_t = \mu_t^- + K_t(z_t - \hat{z}_{n_t}) \tag{2.16}$$

$$\Sigma_t = (I - K_t G_t)\Sigma_t^- \tag{2.17}$$

A thorough derivation of the EKF SLAM algorithm is found in [5, 42]. The EKF algorithm is explored more generally in [3, 18, 25, 26].

2.1.2 Limits of EKF SLAM

One disadvantage of the basic EKF when applied to online SLAM situations is the quadratic complexity of the update equations. In a planar scenario with a three-state representation of agent pose, the SLAM state vector is of dimension $2N + 3$, where N is the number of landmarks stored in the filter map. Equation 2.17, the covariance update, requires an inner product calculation that will grow on the order of $(2N + 3)^2$ as more features are mapped. Hence, many online applications with detailed maps of millions of features either avoid the basic EKF algorithm or employ alternative schemes to reduce this complexity growth. A number of solutions break a global feature map into smaller submaps [10, 22]. Updates to features in the global map

are delayed while the agent remains within the vicinity of a submap. Since features at opposite ends of a large environment will have little or no correlation, covariance matrices for high-dimensional maps are often sparse. Measurement updates to this covariance matrix can be processed more efficiently by taking advantage of this sparsity and ignoring correlations between distant features [2].

Another drawback of the basic EKF SLAM algorithm is single-hypotheses data association. Data association is a decision-making process in which an incoming measurement is either matched with an existing landmark in the filter map or deemed a new feature. This decision is often non-trivial in SLAM situations, where pose and landmark uncertainty and measurement noise can all contribute to data association ambiguity [5]. In the basic EKF architecture, the filter must pick one association for a measurement, typically with a maximum likelihood heuristic, and the effects of an incorrect decision can never be undone. Alternative data association methods for EKF SLAM have been evaluated, with the more robust techniques reducing the chance of association errors [5]. Still, the inevitability of incorrect associations, especially in a SLAM environment where associations are unknown, poses a threat to EKF stability and accuracy [31, 32]. Multiple hypothesis tracking (MHT) presents a more flexible method that has the effect of delayed decision making [38]. In ambiguous association situations, where multiple valid interpretations exist, new hypotheses are created and maintained alongside the original estimate. Typically, these extra hypotheses must be trimmed after future observations to keep the number of unique hypotheses from growing without bound.

MHT methods are also essential in order for EKF-based algorithms to solve the global localization and kidnapped robot problems. In the former, a robot must use an accurate map of the environment to localize with global initial uncertainty. This problem has significant application to indoor autonomous navigation. The latter problem is the case when a well-localized robot is teleported unknowingly to a different region of the map. Both scenarios require the filter to simultaneously consider many different posterior hypotheses, giving each equal weight until many observations favor a single posterior over all others.

2.2 The Particle Filter

Successful attempts at solving for the SLAM posterior without restraining its form to a Gaussian distribution employ a more recent estimation tool known as the particle filter. Belonging to a class of Sequential Monte-Carlo (SMC) methods originating in the 1950s, the particle filter has recently enjoyed attention as advancements in applied statistics and computer processing speeds have prompted its application to a broad range of estimation problems [8, 24, 39]. Improvements to the basic SMC techniques by Gordon et. al., Kitagawa, and Liu and Chen in the mid-to-late 1990s have produced recursive Bayesian estimators with established theoretical convergence that are no longer bound to the Gaussian assumption of the Kalman filter and its derivatives [20, 27, 28].

2.2.1 Particle Filters for Agent Pose Tracking

The particle filter addresses the difficulty of computing a non-Gaussian posterior distribution from (2.3) in closed form by approximating this density with a large number of discrete, random samples [13, 39]. Briefly ignoring the entire SLAM posterior and focusing solely on tracking the posterior distribution of the robot pose, we begin with the Bayes' filter recursion:

$$p(s_t|z^t, u^t) = \eta p(z_t|s_t) \int p(s_t|s_{t-1}, u_t) p(s_{t-1}|z_{t-1}, u_{t-1}) ds_{t-1} \tag{2.18}$$

An optimal formulation would sample particles directly from $p(s_t|u_t, z_t)$ to approximate the pose posterior. However, having removed the Gaussian assumption, this target distribution may be difficult or practically impossible to draw from directly [39]. Instead, particles are drawn from a simpler proposal distribution $q(s_t)$ according to an SMC technique known as importance sampling [20, 39]. Weights are then assigned to the particles such that:

$$p(s_t|z_t, u_t) = \sum_{i=1}^{M} q(s^i) w(s^i) \tag{2.19}$$

where w is a set of importance weights given by the ratio of the target (posterior) distribution to the proposal distribution:

$$\tilde{w}(s^i) = \frac{p(s_t|z^t, u^t)}{q(s_t)} \tag{2.20}$$

and then normalized according to:

$$w(s^i) = \frac{\tilde{w}(s^i)}{\sum_{j=1}^{M} \tilde{w}(s^j)} \tag{2.21}$$

where M is the total number of particles used to represent the distribution. Using the agent motion model $p(s_t|s_{t-1}, u_t)$ as the proposal distribution, the assigned weighting factor conveniently becomes:

$$\tilde{w}(s^i) = p(z_t|s_t) \tag{2.22}$$

which in most applications is the agent observation or perceptual likelihood [32, 45, 39]. For a detailed derivation see [32]. Applying this principle to the recursive Bayesian framework results in sequential importance sampling, where particle weights are updated at each time step. The algorithm begins as each particle from an initial distribution $p(s_0)$ is propagated according to the agent motion model, producing a proposal distribution. Weights are then assigned to each particle based on the agent observation likelihood at that discrete point in the state space, and the process repeats.

2.2.2 Resampling

Over time, only a relatively small portion of particles in the state space will continue to receive significant weights. In a localization scenario, these particles would most likely represent the true pose of the agent. To reallocate computational resources and obtain a more detailed distribution, resampling is necessary. By drawing a new particle set (with replacement) from the previous set, with probabilities proportional to assigned weights, particles will converge to regions of the state space with high likelihoods. Initially proposed by Gordon et al., this resampling technique, known

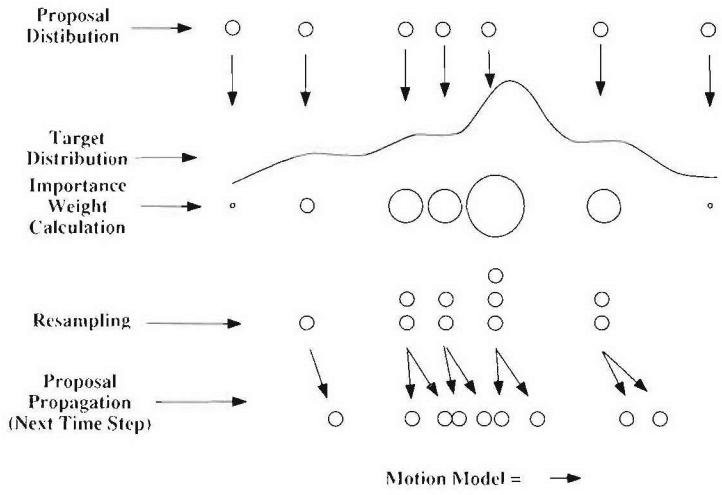


Figure 2-1: The basic Particle Filter uses discrete points and SMC methods to approximate an evolving posterior distribution

as sample importance resampling (SIR) or Bootstrap filtering, produced the first effective particle filter [20, 29]. This recursion, depicted graphically in figure 2-1, will approach the optimal Bayesian posterior in the limit of infinite particles [8, 39].

2.3 Particle Filter SLAM

Despite their ability to track arbitrarily complex, multi-modal distributions, particle filters carry a pronounced computational encumbrance: the number of particles needed to track a belief scales exponentially with the dimension of the state. A SLAM posterior that includes hundreds of landmarks (each a dimension of the posterior) could require millions of particles to be tracked effectively [6, 9]. A recent innovation introduced by M. Montemerlo solves this burden by conditioning the SLAM posterior on the entire robot path instead of the current pose [31]. The basic premise is this: if the entire path of the robot were known, not just the current pose, a single landmark observation would not affect the location or uncertainty of any other landmark. Consequently, landmark measurements are conditionally independent. All landmark correlations are ignored and the SLAM posterior can be represented as the product

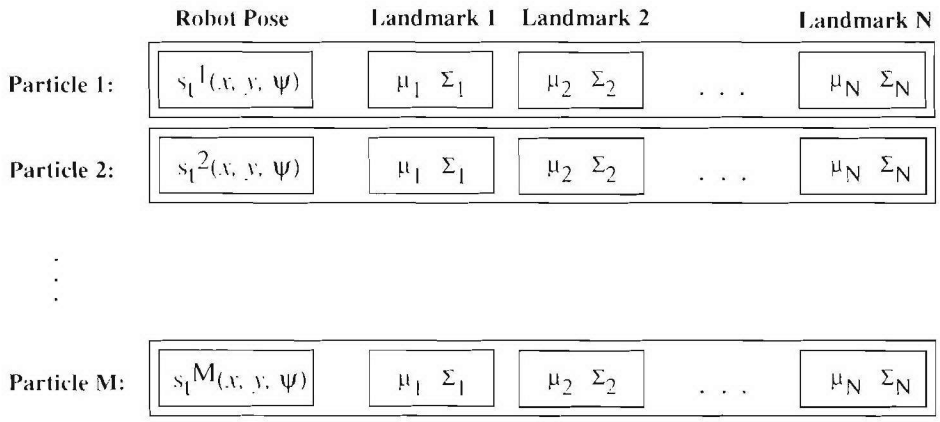


Figure 2-2: The factored SLAM posterior: each particle carries an agent pose estimate and a map of features [32].

of the path posterior and N independent landmark estimators:

$$p(s_t, \Theta | z_t, u_t) = \underbrace{p(s_t | z_t, u_t)}_{path\ posterior} \underbrace{\prod_{n=1}^{N} p(\theta_n | s_t, z_t, u_t)}_{landmark\ estimators} \tag{2.23}$$

Montemerlo also illustrates that all update equations for the filter will depend only on the most recent pose under the Markov property of the SLAM posterior. This factorization, illustrated in figure 2-2, forms a particle filter based on the sampling architecture of Rao-Blackwellization, where a small subset of variables are sampled (the agent pose information) and other marginals are calculated in closed form (landmark estimation parameters) [4]. The application of this principle to the position-tracking particle filter was introduced by Murphy and Russell [34]. Building on the structure of (2.23), Montemerlo develops an algorithm named FastSLAM 1.0 that represents the posterior with $N + 1$ filters, one for each term in (2.23). Each particle of the algorithm represents a different hypothesis of the SLAM posterior:

$$S_t^{[m]} = \langle s_t^{[m]}, \mu_{1,t}^{[m]}, \Sigma_{1,t}^{[m]}, \ldots, \mu_{N,t}^{[m]}, \Sigma_{N,t}^{[m]} \rangle \tag{2.24}$$

The bracketed notation represents the index of the particle. The agent pose information for each hypothesis $s_t^{[m]}$ is updated with the SIR method explained previously.

The rest of the SLAM posterior is maintained with independent Gaussian estimators representing the mean $\mu_{n,t}^{[m]}$ and covariance $\Sigma_{n,t}^{[m]}$ of each observed landmark. Given a two or three dimensional Cartesian space, these landmarks will be low-dimensional and fixed in size. Each particle carries its own set of landmark estimators. Taken in total, the particles form an array of M hypotheses that represent a discrete approximation to the optimal Bayesian SLAM posterior [32].

2.3.1 Importance Weight Calculation

As with a standard particle filter, particles in FastSLAM are drawn from the motion model to create a proposal distribution [32, 39, 41, 45]. The Gaussian landmark estimators are then updated for each particle using the agent observation model, the current measurement z_t, and standard EKF update equations. Assuming a planar SLAM scenario with an agent measuring range and bearing to nearby features, the observation function becomes:

$$g(s_t, \theta_{n_t}) = \begin{bmatrix} r(s_t, \theta_{n_t}) \\ \phi(s_t, \theta_{n_t}) \end{bmatrix} = \begin{bmatrix} \sqrt{(\theta_{n_t,x} - s_{t,x})^2 + (\theta_{n_t,y} - s_{t,y})^2} \\ \tan^{-1}(\frac{\theta_{n_t,y} - s_{t,y}}{\theta_{n_t,x} - s_{t,x}}) - s_{t,v} \end{bmatrix} \tag{2.25}$$

with the current agent pose and measured landmark represented by $s_t = \langle s_{t,x}, s_{t,y}, s_{t,v} \rangle$ and $\langle \theta_{n_t,x}, \theta_{n_t,y} \rangle$, respectively. The updated Gaussian parameters for the measured landmark are obtained by:

$$\hat{z}_t = g(s_t^{[m]}, \mu_{n_t,t-1}) \tag{2.26}$$

$$G_{\theta_{n_t}} = \nabla_{\theta_{n_t}} g(s_t, \theta_{n_t})\big|_{s_t = s_t^{[m]}; \theta_{n_t} = \mu_{n_t,t-1}^{[m]}} \tag{2.27}$$

$$Z_{n,t} = G_{\theta_{n_t}} \Sigma_{n_t,t-1}^{[m]} G_{\theta_{n_t}}^T + R_t \tag{2.28}$$

$$K_t = \Sigma_{n_t,t-1}^{[m]} G_{\theta_{n_t}}^T Z_{n,t}^{-1} \tag{2.29}$$

$$\mu_{n_t,t}^{[m]} = \mu_{n_t,t-1}^{[m]} + K_t(z_t - \hat{z}_t) \tag{2.30}$$

$$\Sigma_{n_t,t}^{[m]} = (I - K_t G_{\theta_{n_t}}) \Sigma_{n_t,t-1}^{[m]} \tag{2.31}$$

2.3.2 Data Association

An important hurdle for any SLAM algorithm is data association. Since a problem requirement includes mapping new features, the algorithm must decide first if the measurement corresponds to a new landmark. If not, it must decide on a per-particle basis which of the N known landmarks stored within the particle is most likely to have produced this observation. After this decision is made, EKF equations update the mean and covariance for the identified landmark. Assuming first that knowledge of data associations are known, the observation likelihood can be computed in close form. It is derived from the innovation, or difference between the actual measurement and the predicted measurement, given the current agent pose and the landmark estimation parameters [31]. Since the landmark estimator is an EKF, the sequence of innovations will be Gaussian and the observation likelihood is:

$$p(z_t|s_t^{[m]}) = \frac{1}{\sqrt{|2\pi Z_{n_t,t}|}} \exp\{-\frac{1}{2}(z_t - \hat{z}_{n_t,t})^T [Z_{n_t}]^{-1}(z_t - \hat{z}_{n_t,t})\} \tag{2.32}$$

By computing this likelihood for each landmark within the particle, we can obtain the maximum likelihood estimator for this measurement by simply selecting the landmark with the highest likelihood:

$$\hat{n}_t = \arg\max_{n_t} p(z_t|n_t, s_t) \tag{2.33}$$

The estimator parameters for this landmark are then updated within the particle. The observation likelihood for the maximum likelihood estimator, given by (2.32), also becomes the particle importance weight for resampling, $w_t^{[m]}$. If the likelihood for each landmark falls below a threshold, a new landmark is created and initialized as follows:

$$\mu_{n_t,t}^{[m]} = g^{-1}(s_t^{[m]}, z_t) \tag{2.34}$$

$$\Sigma_{n_t,t}^{[m]} = \left(G_{\theta_{n_t,t}}^T R^{-1} G_{\theta_{n_t,t}}\right) \tag{2.35}$$

The importance weight for this particle is a pre-defined likelihood threshold, p_0. This process is repeated until each particle has been assigned a weight. Weights are then normalized and new particles are drawn as in SIR.

2.3.3 Preventing Filter Degeneracy

It is important to note that resampling is not always necessary. It simply reduces filter degeneracy by trimming excess particles that have little relevance to the current measurement and reproducing particles in the area of interest for agent pose information. Some particle filtering approaches include a measure of degeneracy, M_{eff} defined as:

$$M_{eff} = \frac{1}{\sum_{i=1}^{M} \left(w^{(i)} \right)^2} \tag{2.36}$$

M_{eff}, the effective size of the particle set, is in some ways a measure of dispersion of the importance weights. If particles were drawn according to the true posterior, all samples would receive the same weight. As variance of the weights increases, M_{eff} will decrease. Theoretically, resampling particles only when M_{eff} falls below a defined threshold will decrease the chances of pruning possibly accurate trajectories from the filter [21].

2.4 FastSLAM vs. EKF SLAM

There are several well documented strengths of the FastSLAM architecture over standard EKF SLAM approaches. Most importantly, the Monte Carlo, particle-based architecture of FastSLAM allows the filter to track multiple hypotheses simultaneously at each measurement step. This helps solve data association ambiguity inherent in the SLAM problem that particularly plagues standard EKF approaches [7, 47]. A robot must decide whether a current measurement is coming from a new or previously mapped landmark, which can be difficult if features are relatively close together. If landmark measurements are incorrectly attributed, the EKF can diverge rapidly. Instead, FastSLAM assigns data associations on a per particle basis. An implicit result

is delayed decision making about the most likely measurement association. Particles with maps that closely agree with incoming data will survive resampling, while particles that disagree due to incorrect previous data associations are eventually eliminated. In the limit of infinite particles, all data association ambiguities are resolved and FastSLAM provides a full Bayesian solution to the SLAM problem [32]. FastSLAM is also a universal density approximator, meaning it can represent arbitrarily complex distributions of the agent pose. This can be particularly useful in modeling non-linear motion models and the uncertainty of an agent mapping a constrained environment [39]. Finally, the computational complexity of the basic FastSLAM algorithm is $O(M \cdot N)$, compared to $O(N^2)$ with a standard EKF approach. Montemerlo also introduces a version of FastSLAM with a computational complexity of $O(\log N)$ [32].

2.5 FastSLAM Challenges

Despite its advantages, FastSLAM does suffer drawbacks common to particle filters. There will always be unsampled gaps in the agent state space when using a finite number of particles. While resampling reduces filter degeneracy by concentrating particles in an area of interest in the state space, it cannot guarantee convergence. This is especially true if the proposal and target distributions (and the uncertainty in these distributions) are not well matched, as shown in figure 2-3. If the agent's sensor is very accurate relative to the motion model, the target distribution will be sharply peaked relative to a flat proposal distribution. In the worst case scenario, no particles receive non-negligible importance weights, preventing filter convergence to the true state. Another possibility is sample impoverishment (used synonymously with particle depletion), wherein a small percentage of particles from the proposal distribution are assigned non-negligible weights, causing significant duplication of a few unique hypotheses and large "stacks" of particles. Stochastic proposal propagation with the next agent control input may not adequately scatter the particles to recover lost diversity. Over time, this could result in particles drifting away from the true state.

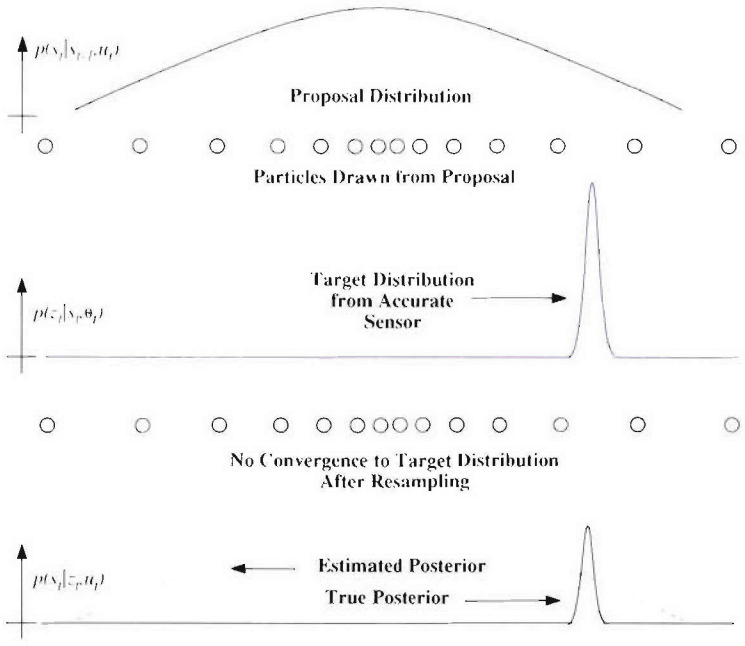


Figure 2-3: A noisy motion model creates a broad proposal distribution, a precise sensor measurement results in a narrow target distribution. Convergence of the particles to the true posterior is prevented since the narrow posterior occurs in an unsampled gap in the state space.

It also gives rise to a host of other issues that contribute to a loss of filter accuracy and stability.

2.5.1 Effects of Sample Impoverishment

In addition to particle drift, an obvious issue for all pose-tracking filters, sample impoverishment using FastSLAM is extremely dangerous because of the nature of uncertainty storage in Rao-Blackwellization. In EKF SLAM algorithms, new landmarks are initialized to include both the error characteristics of the measurement device and the uncertainty of the agent pose at the time of observation. In other words, an estimate of landmark position is only as good as the precision of the measurement and the knowledge of the agent state. Because the SLAM posterior measures landmark positions conditioned on an estimate of the robot path, each particle in the filter is considered an error-free hypothesis of the true pose. Agent pose uncertainty is stored in the dispersion of the particle cloud. As a result, each new feature in the landmark

array is initialized with uncertainty from measurement noise alone. Subsequent updates to the landmark estimation parameters in FastSLAM are also processed with EKF equations that include only the error model of the observation device.

This "false certainty" in landmark location greatly complicates the data association process. With an extremely accurate sensor, each particle in the filter has little error allowance when deciding on an association between the incoming measurement and one of its stored landmarks. Unless the measurement agrees exactly with a stored landmark, the particle receives a low weight. Without a diverse set of samples, only few of the particles will survive resampling and the overall uncertainty of each landmark will approach zero. Even if additional diversity is added as particles propagate, the precision of feature estimates will ensure that only few particles survive the next round of weighting and resampling. Over time, the pose estimate and all mapped landmarks will be overcome by the noisy motion model and diverge substantially from the true posterior [16]. As loops are closed and the agent returns to a previously mapped region of the environment, the skewed map and pose drift will lead the agent to believe that the previously observed feature is actually a new feature, hence the creation of false landmarks that further complicate data association in the future (figure 2-4).

2.5.2 Overcoming Sample Impoverishment

One way to overcome sample impoverishment, proposed by D. Fox et al. [17], is to use a sensor model that overestimates measurement noise. While this does tend to give more particles non-negligible weights and reduce particle depletion, it throws away valuable information from precise sensor measurements. Selective resampling based on a filter degeneracy estimate (M_{eff}) could delay the effects of sample impoverishment, as all trajectories are propagated and weighed until the degeneracy falls below a certain threshold. Some sources argue that in cases of extremely low measurement noise, the filter will degenerate quickly since only few (if any) of the particles will receive significant weights [13]. Degeneracy will only be further amplified if resampling is delayed, as the product of weights at each time step magnifies the dispersion

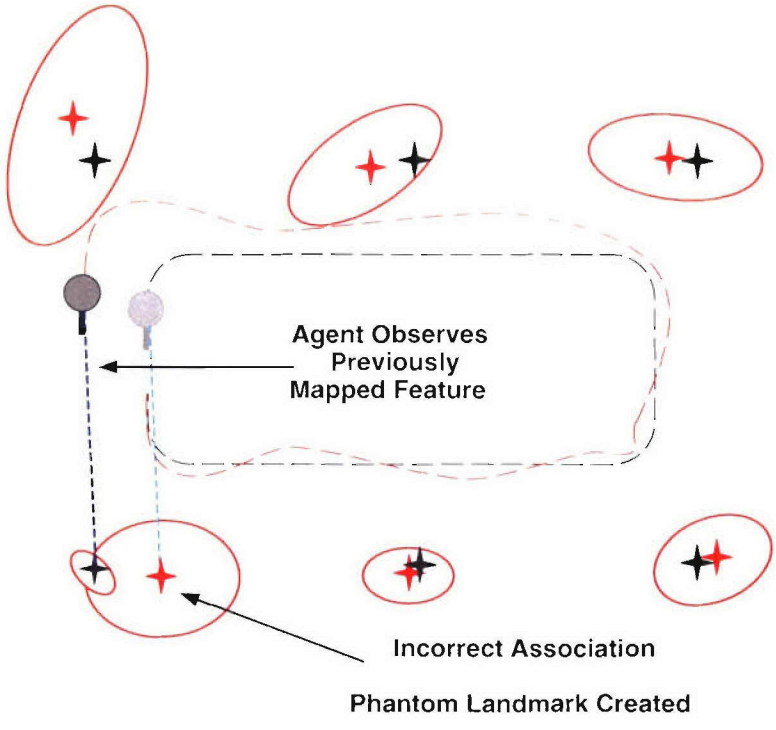


Figure 2-4: An incorrect data association with the current measurement and a previously mapped feature causes a new "false" landmark. Agent and landmark position uncertainty is not reduced as was the case in 1-2.

between particle weights [39, 13]. Principled approaches from Fox, Pitt and Shepard suggest changing the form of the proposal distribution altogether [17, 36, 37, 39, 45]. Other approaches focus instead on the resampling process, and propose a solution to the impoverishment involving regularization- a readjustment of the particles after the resampling step with the intent of introducing lost diversity into the posterior [6, 19, 39]. Both strategies have been evaluated in particle filters for tracking and localization applications with some success [23, 35, 36, 45]. The remainder of this thesis will involve a detailed investigation of solutions to sample impoverishment within the specific context of Rao-Blackwellized particle filters SLAM applications.

Chapter 3

Recovering Sample Diversity

Particle filters have gained recent attention in robotics research and have provided an alternative to the EKF with proven deftness in tackling more complicated navigation and mapping scenarios. The particle filter is not invincible, and several failure modes have already been well documented [16, 45, 32, 39, 47]. The increasing popularity of particle filters for non-linear position tracking applications has prompted the development of improvement strategies designed to answer some of the pitfalls associated with basic SMC filtering. This chapter begins by highlighting several of these early improvement strategies. With only recent research in the use of particle filters for SLAM environments, few methods exist for recovering sample diversity in situations prone to particle depletion, but this chapter outlines the most significant solutions to date. Additionally, new techniques are proposed that build upon the basic strategy of regularization, a common fix for particle filters in position tracking scenarios. No documented results on the application of regularization methods to RBPF SLAM algorithms were found.

3.1 Sample Impoverishment Revisited

In standard resampling, sample impoverishment arises when a small subset of particles receive high weights relative to the majority. These few particles are reproduced many times, and after resampling, the majority of the particles will occupy only a few

singular points within the state space. As one can imagine, these few singular points in the state space do not produce an accurate characterization of the true agent uncertainty. A better representation of agent uncertainty after an accurate sensor reading is a tight distribution of unique particles. Diversity is still maintained in the particle set because each occupies a different point in the state space. In the case of mobile robotics, this state space is easily visualized as a two or three dimensional Cartesian space. Maintaining appropriate sample diversity involves balancing a delicate relationship between the proposal and target distribution [17, 39]. The proposal must place an adequate number of particles in a favorable region of the state space in such a way that an acute target distribution can assign non-negligible weights to a large proportion of these particles. Maintaining this balance becomes more difficult as sensor accuracy increases and the target distribution becomes sharply peaked with respect to the proposal distribution. Solutions to sample impoverishment are based on implementing diversity recovery methods before or after resampling. The former approach seeks an improved proposal distribution that includes measurement information [30, 45]. As a result, particles would theoretically propagate to more favorable regions for resampling. The later group of solutions inject diversity into the posterior distribution after resampling to smooth the resulting density before the next propagation step [19]. Approaches vary in the rigor of their derivations and whether or not they demonstrate theoretical convergence. The more mathematically sound solutions improve sample diversity while maintaining an approximation to the optimal Bayesian posterior. Other more simple methods have also been introduced that fix sample impoverishment but do not necessarily guarantee convergence [39]. The next section will introduce several approaches found in literature and other intuitive methods developed over the course of this research project.

3.2 Alternative Proposal Distributions for Position Tracking

As mentioned in section 2.2.1, an optimal formulation would draw particles directly from the posterior distribution $p(s_t|u_t, z_t)$. Because this is difficult or impossible to implement for a complex distribution, the most recent observation is used to weight a proposal particle set according to the perceptual likelihood for a feature observation, thereby creating an approximation to the target distribution with a finite number of particles. Literature suggests that the relative mismatch between the proposal and target distribution affects the convergence of a particle filter to the true posterior [17]. Convergence is also prevented if the perceptual likelihood is extremely narrow, as would be the case with an accurate sensor measurement. Particles drawn from a proposal distribution that includes feature measurements would have a better chance of matching this narrow target density. More particles would therefore receive a non-negligible weight and survive resampling, increasing particle diversity and reducing the effects of sample impoverishment.

3.2.1 Auxiliary Particle Filter

The Auxiliary particle filter (APF) was introduced by Pitt and Shephard as one way to incorporate recent sensor measurements in the proposal distribution. A variant of standard SIR, the APF includes an additional sampling step at time $t-1$, using observation data at time t, before particles are propagated according to the motion model, $p(s_t|s_{t-1}, u_t)$. This "presampling" step selects particles that have a high likelihood of propagating to a favorable region of the state space, and only allows these particles to advance [37]. The algorithm begins at a time $t-1$ by propagating the previous posterior distribution to an auxiliary distribution using the motion model for the current time step. Next, importance weights are calculated and resampling proceeds as in SIR, but this time only the indices of particles are of interest. The selected particles are traced back to their location at the previous time step, before

motion model propagation. These parent particles are then propagated according to the motion model. Weights are calculated and particles are resampled producing the APF posterior distribution. The advantage of this scheme is that it only propagates particles that are more likely to end up in the regions of high-likelihood according to the recent sensor measurement.

3.2.2 Local-Linearized Particle Filter

Another way to incorporate recent sensor measurements in particle filters for tracking applications is to update the proposal density, before weighting and resampling, with sensor information via a bank of extended Kalman filters. This SMC variant is known as a Local-Linearized particle filter. A posterior density from the $t-1$ time step is first propagated according to the agent motion model. Mean and covariance parameters for this proposal distribution are updated on a per-particle basis with an EKF [39]. A sample is drawn from this updated proposal and an importance weight is calculated as before. This propagate-update-draw step is repeated for each particle. Montemerlo introduces a Rao-Blackwellized version of a Local-Linearized particle filter for SLAM purposes known as FastSLAM 2.0 [30].

3.2.3 Mixture Monte Carlo Localization

Extending the application of particle filters beyond position tracking to the more encompassing problem of mobile robot localization shows similar drawbacks from sample impoverishment. D. Fox et al. describe the effects of highly accurate sensor measurements coupled with a relatively noisy motion model and propose a solution that involves drawing from a more sophisticated proposal distribution [17]. A subset of the proposal distribution will be drawn from the motion model and another subset, approximately 10% of the particles, is drawn from the perceptual model $p(z_t|s_t)$. Importance factors are more difficult to calculate for particles drawn according to the latter distribution: the prior posterior belief must be transformed into a kd-tree in order to obtain an evolution of the perceptual density [33]. The importance

weight is proportional to this density tree and a constant factor, which is ignored since weights are normalized before resampling. Their results do show a significant improvement over standard particle filter performance in cases of low measurement noise, simply because a percentage of samples from the proposal distribution are drawn from this accurate perceptual density. While this technique works well for mobile robot localization and position tracking, it does not address specific challenges posed by SLAM [32]. In some cases, a partial map of local features may be available, but not in the strict SLAM problem. Without a priori map information it may not be possible draw particles from the perceptual likelihood. This particular algorithm could potentially be used to refine position uncertainty when preliminary landmark locations have been established by SLAM and after loop closures [43].

3.3 Alternative Proposal Distributions for SLAM

Montemerlo, in his development and evaluation of FastSLAM, also describes the effect of sample impoverishment on a Rao-Blackwellized particle filter; it also suffers a loss of diversity with greater measurement precision and a noisy agent motion model. He therefore develops an alternative proposal distribution that takes advantage of incoming measurements [32]. After a thorough and elegant derivation, Montemerlo arrives at a version of FastSLAM that updates the proposal distribution with measurement information via a a series of extended Kalman filters, one for each measurement within the observation set for a time step. This approach is similar to the Local-Linearized particle filter for position tracking applications explained previously. Montemerlo has also derived an expression for importance weights that considers not only the uncertainty in landmark positions and measurements, but also the uncertainty of the proposal distribution after measurement updates. More importantly, the algorithm incorporates previously unmapped landmarks, making it a complete approach to both subsets of the SLAM problem. Theoretical convergence is proven for the Linear-Gaussian SLAM scenario with one particle. This is a profound result because prior to FastSLAM 2.0, SLAM algorithm convergence was only proven for a full co-

variance matrix representation of the posterior with correlations between landmark estimates [47]. In experimental results, FastSLAM 2.0 provides a more accurate and diverse SLAM posterior and requires fewer particles to effectively track an agent pose than the original FastSLAM algorithm. The algorithm begins as particles are drawn from a previous time step posterior distribution according to a motion model, again characterized as a nonlinear function with zero-mean, uncorrelated process noise. This propagation yields an initial proposal density:

$$\hat{s}_t^{[m]} = h(s_{t-1}, u_t) \tag{3.1}$$

where $h(s_{t-1}, u_t)$ is a nonlinear function with noise covariance P_t. From that initial proposal draw, an expected observation is produced (per-particle) for each landmark according to the agent motion model:

$$\hat{z}_{t,n} = g(\hat{s}_t, \mu_{n,t-1}^{[m]}) \qquad n = 1, \ldots, N \tag{3.2}$$

As before, the measurement noise covariance matrix is given by R_t. After predicted measurements are calculated, an updated proposal distribution is calculated for each landmark using Kalman filter update equations:

$$G_{\theta,n} = \nabla_{\theta_n} g(s_t, \theta_n)\big|_{s_t=\hat{s}_t;\theta_n=\mu_{n_t,t-1}^{[m]}} \tag{3.3}$$

$$G_{s,n} = \nabla_{s_t} g(s_t, \theta_n)\big|_{s_t=\hat{s}_t;\theta_n=\mu_{n_t,t-1}^{[m]}} \tag{3.4}$$

$$Z_{t,n}^{[m]} = R_t + G_{\theta,n}\Sigma_{n,t-1}^{[m]} G_{\theta,n}^T \tag{3.5}$$

$$\Sigma_{s_t,n}^{[m]} = \left[G_{s,n}^T Z_{t,n}^{-1} G_{s,n} + P_t^{-1}\right]^{-1} \tag{3.6}$$

$$\mu_{s_t,n} = \hat{s}_t + \Sigma_{s_t,n}^{[m]} G_{s,n}^T Z_{t,n}^{[m]-1}(z_t - \hat{z}_{t,n}) \tag{3.7}$$

A particle for the proposal is then drawn from the resulting distribution that includes the most recent measurement:

$$s_{t,n}^{[m]} \sim N(s_t; \mu_{s_t,n}, \Sigma_{s_t,n}) \tag{3.8}$$

After a particle has been drawn from each of N different distributions, likelihood weights are calculated in the same fashion of FastSLAM 1.0. The drawn particle with the largest weight then becomes part of the proposal that will be resampled to approximate the posterior. This weight will not, however, be used as the importance weight for resampling. Since the proposal particles are drawn from a different distribution than the agent motion model, the importance weights for resampling must be calculated in a slightly different way:

$$L_t = G_{s,\hat{n}_t} P_t G_{s,\hat{n}_t}^T + G_{\theta,\hat{n}_t} \Sigma_{\hat{n}_t,t-1} G_{\theta,\hat{n}_t}^T + R_t \tag{3.9}$$

$$w_t^{[m]} = \frac{1}{\sqrt{|2\pi L_t|}} \exp\left\{-\frac{1}{2}(z_t - \hat{z}_{t,\hat{n}_t})^T L_t^{-1}(z_t - \hat{z}_{t,\hat{n}_t})\right\} \tag{3.10}$$

As before, the perceptual likelihood used to calculate resampling importance weights is a multi-variate Gaussian probability density function, only this time the normalizing measurement uncertainty L_t includes the contribution from the agent process noise. New landmarks are initiated, in the same way as FastSLAM 1.0, when all landmark likelihoods fall below a pre-defined threshold. Also in the case of a new landmark, poses from the proposal are drawn from the original distribution $\hat{s}_t$ excluding feature measurement information. When multiple measurements are considered at each time step, the algorithm becomes slightly more complicated. The proposal is updated iteratively, once for each measurement. A particle is drawn after each iteration according to (3.8) in order to update landmark estimator parameters. Particles for the proposal, however, are only sampled after all measurements have been processed. An illustration of this algorithm and its solution to the proposal-target mismatch from accurate sensor measurements is shown in figure 3-1. Though FastSLAM 2.0 grows at the similar favorable rate of $O(N \cdot M)$, it includes update equations for the proposal distribution and is therefore much more computationally expensive than FastSLAM 1.0.

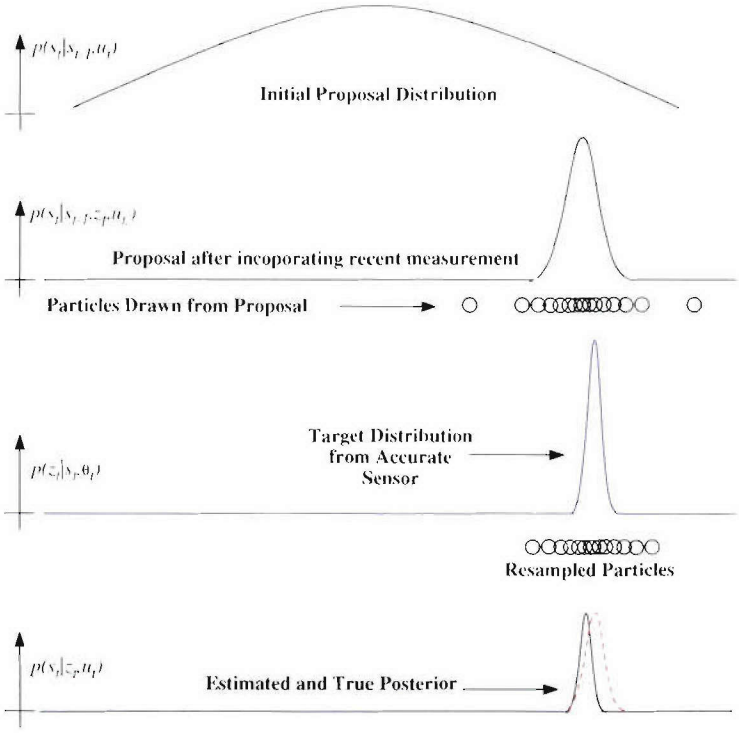


Figure 3-1: In FastSLAM 2.0, the proposal distribution incorporates recent measurements. Particles for importance weight calculations have a greater chance of receiving non-negligible weights in the case of an accurate sensor and a noisy motion model.

3.4 Regularization

Another class of improvement strategies for the sample impoverishment problem focuses specifically on recovering diversity after the resampling step. A severely impoverished posterior will most likely consist of a few discrete points with many particles "stacked" at these points. Regularization methods attempt to create a more diverse posterior density approximation by relocating the particles in stacks to a more continuous distribution [1, 19, 35]. An easy way to regularize would be to simply draw a new set of particles about the wide-sense mean and covariance of the distribution. However, this approach would not preserve the possibly non-linear and multi-modal characteristics of the distribution and would thus negate the advantages of using a particle filter in the first place. Consequently, designing a regularization scheme that introduces an appropriate amount of diversity, while preserving the complex nature of a distribution, can be difficult. Most particle filter regularization schemes in literature

approach this difficulty by representing a continuous distribution for particle adjustment by a series of Epanechnikov or Gaussian kernels, centered at points in the state space occupied by resampled particles [1, 39]. The parameters of these individual kernels can be manipulated so that the kernel set approximates an arbitrarily complex posterior. Particle state adjustments are drawn from individual kernels and then added to resampled particles. The next few section describe possible kernel shaping methods with slight variations that can alter the effect of particle regularization.

3.4.1 Regularized Particle Filter

The original Regularized Particle Filter (RPF) was designed by S. Godsill and T. Clapp. It is essentially a standard SIR filter with a regularization step included after resampling. During regularization, particles are adjusted according the continuous approximation:

$$p(s_t|z_t, u_t) \approx \sum_{i=1}^{M} w_t^i \frac{1}{h^{n_s}} K\frac{(s_t - s_t^i)}{h} \tag{3.11}$$

where $K(\cdot)$ is a rescaled kernel density and h is the kernel bandwidth, a scalar specific to the kernel that also depends on the number of particles in the filter. The kernel that minimizes the mean integrated square error between the true posterior density and the regularized version in (3.11) is the Epanechnikov kernel. Practically, this kernel is difficult to implement, and the Gaussian kernel is normally used as a computationally efficient substitute. The optimal bandwidth is then given by:

$$h_{opt} = (\frac{4}{n_s + 2})^{\frac{1}{n_s+4}} N^{-\frac{1}{n_s+4}} \tag{3.12}$$

where n_s is the dimension of the agent state vector, s_t. Before resampling, the empirical covariance, $\mathbf{A}_t$, is calculated from the proposal distribution. The empirical covariance can be thought of as a weighted proposal covariance that accounts for the uncertainty stored in particle dispersion. After resampling, particles are adjusted according to:

$$s_t^{i*} = s_t^i + h_{opt}\mathbf{D}_t\epsilon^i \tag{3.13}$$

where

$$\mathbf{D}_t\mathbf{D}_t^T = \mathbf{A}_t \qquad \text{and} \qquad \epsilon^i \sim N(\epsilon; 0, 1) \tag{3.14}$$

Computationally, the RPF differs from the standard SIR filter only in M additional draws from a Gaussian kernel and the formulation of the empirical covariance matrix before resampling. These steps have a minimal effect on overall processing time [19]. Despite a rigorous derivation, regularizing particles according to the RPF does not necessarily guarantee asymptotic convergence to the optimal Bayesian posterior. This is a common theoretical drawback of almost every regularization scheme. The RPF has improved performance in tracking applications, but no literature results were found that describe its application to the SLAM problem. Another advantage of the RPF is that by setting the kernel adjustment proportional to the empirical covariance, the RPF avoids "particle shock" that can occur when a relatively broad distribution converges quickly to a more precise distribution. Instead, a limit is placed on the convergence speed of a particle cloud, maintaining diversity along with greater precision.

3.4.2 Markov Chain Monte Carlo Criterion

The Markov Chain Monte Carlo step is a regularization criterion designed to ensure that any regularization of resampled particles asymptotically approach the Bayesian posterior in the limit of infinite particles [39]. The idea behind the scheme is that a particle s_t^i can be regularized, or moved to a new state s_t^{i*}, only if $u \leq \alpha$, where $u \sim U[0, 1]$ and α is the acceptance probability derived from the Metropolis-Hastings algorithm:

$$\alpha = \min\left\{1, \frac{p(z_t|s_t^{i*})p(s_t^{i*}|s_{t-1}^i, u_t)}{p(z_t|s_t^i)p(s_t^i|s_{t-1}^i, u_t)}\right\} \tag{3.15}$$

Put simply, the particle can be adjusted according to a regularization scheme only if its intended move will place it in a "more likely" region of the state space, as determined by the pre-move and post-move proposal and perceptual densities. While in literature the MCMC move step is used in context with the mathematically derived regularization scheme of the RPF, it is important to note that this criterion can be

applied to any regularization algorithm to ensure that asymptotic convergence to the Bayesian posterior is maintained.

3.5 Other Regularization Approaches

Although the literature search conducted for this thesis did not produce experimental results on the use of regularization for SLAM purposes, a closer look at the Regularized particle filter shows that the idea can easily be extended to SLAM in a Cartesian environment. The RPF algorithm uses a kernel (Epanechnikov or Gaussian) to locally spread particles about the discrete stacks often produced after resampling. The variance, or bandwidth as it is referred to in literature, is a product of the root of the empirical covariance matrix of the particles before resampling. The intuitive methods introduced below include a Gaussian kernel, similar to the computationally inexpensive version of the RPF, but the variances are calculated in differently in order to shape the kernel for a possibly better sample diversity.

3.5.1 Fixed-Gaussian Regularization

A simple version of regularization would involve creating a series of fixed-variance Gaussian kernels after resampling, as shown in figure 3-2. Each particle would then be adjusted within the Cartesian space according to an individual draw from these kernels:

$$s_t^{i*} = s_t^i + \lambda_t \epsilon^i \qquad \text{where} \qquad \epsilon^i \sim N(\epsilon; 0, 1) \tag{3.16}$$

Though this method introduces diversity to the posterior by sampling from a continuous distribution, the probability density will become uni-modal as the spreading parameter λ_t increases. Consequently, a balance must be maintained by spreading the particles with enough variance to introduce a proper amount of diversity, while at the same time keeping this variance small enough to preserve the possible multi-modal characteristics of the distribution. The proper λ_t will need to be determined empirically, and will likely differ in every situation. It will also be the lower limit of

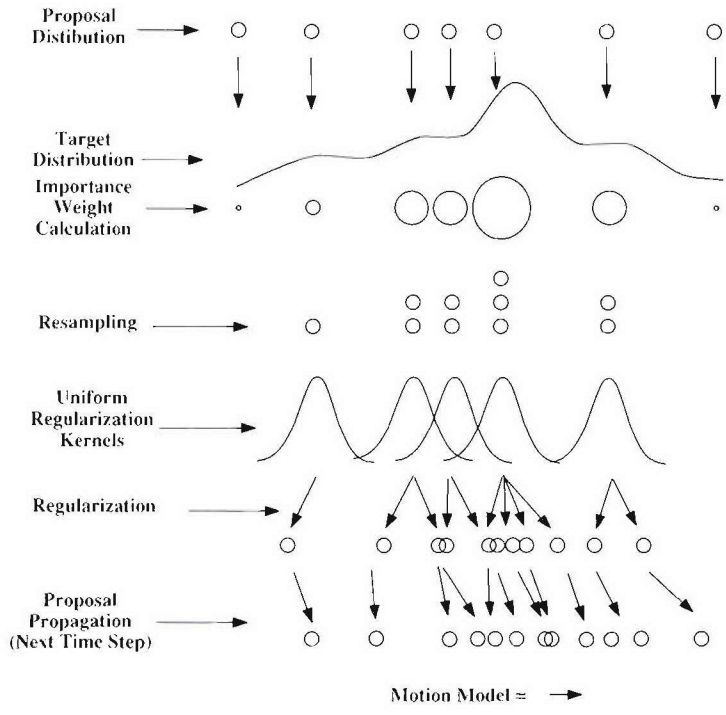


Figure 3-2: Particles are regularized after resampling according to a set of kernels generated at the resampled points.

the wide-sense variance of the posterior distribution. Factors affecting the optimal spreading parameter will include the initial uncertainty of the agent position and the accuracy of the sensor. If the mean-square error of the sensor is less than the variance of this regularization kernel, precious sensor information is lost.

3.5.2 Adaptive Regularization

One advantage of the RPF over a simple, fixed-variance Gaussian particle adjustment is that the variance of the kernel changes according to characteristics of the weighted proposal distribution. In this respect, properties of the regularization kernels can change over time, but there is only one kernel "shape" per time step. A further level of adaptation can also be formed by basing the standard deviation of the regularization kernel on the proportion of particles that are resampled to a particular state.

$$s_t^{i*} = s_t^i + \lambda_t \epsilon^i \qquad \text{where} \qquad \epsilon^i \sim N(\epsilon; 0, 1) \tag{3.17}$$

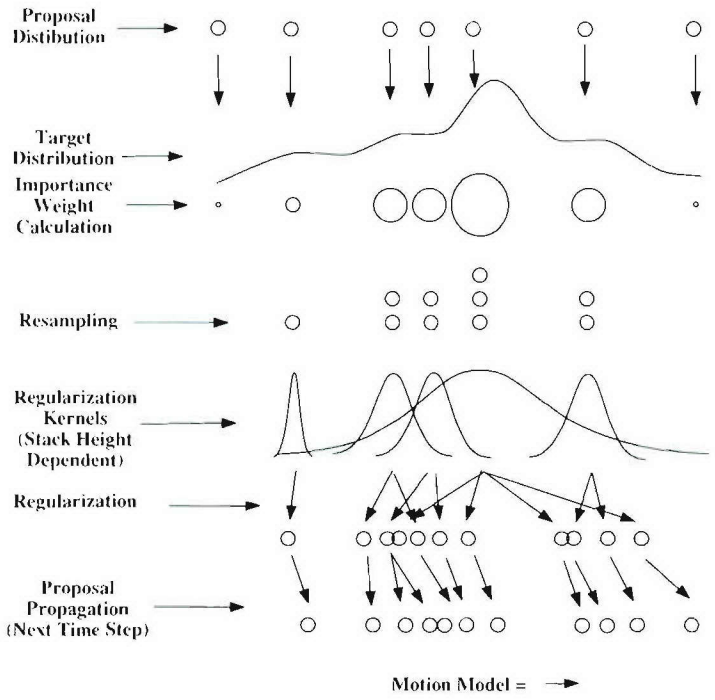


Figure 3-3: Particles are regularized by kernels with adaptive variances. The spreading radius is proportional to the height of a particle "stack."

$$\lambda_t^i = \frac{\tilde{m}_{s_t^i}}{M} \tag{3.18}$$

where $\tilde{m}_{s_t^i}$ is the number of duplicated particles at a particular point in the state space. This method is pictured in figure 3-3. Theoretically, it would generate a larger spreading radius about particle locations that received high likelihoods and were thus largely reproduced in resampling. It will produce the largest variance, and thus have the greatest potential of recovering diversity, in cases with a sharply peaked perceptual density relative to the proposal distribution. As stated earlier, this is the case most vulnerable to sample impoverishment.

3.5.3 Other Adaptive Regularization Techniques

A third intuitive regularization attempt combines some of the properties of the mathematically derived RPF with the above method of adapting the kernel based on the resampled particle stack height. Introducing the λ_t^i parameter into the standard RPF

equation yields the following regularization scheme:

$$s_t^{i*} = s_t^i + \lambda_t^i h_{opt} \mathbf{D}_t \epsilon^i \tag{3.19}$$

In theory, this move will reshape the optimal kernel bandwidth introduced in RPF regularization based on particle stack height, increasing density in cases prone to sample impoverishment.

3.5.4 Process Noise

A thorough look at regularization and an understanding of SMC proposal propagation leads to the awareness that the propagation of the particles according to a stochastic agent motion model is itself a form of regularization similar to a Gaussian kernel used above. Given this, it should lead to the question of whether or not regularization is needed in the first place. Perhaps an over-estimation of the agent process noise would suffice. It is true that basic regularization using a fixed-variance Gaussian kernel is equivalent to propagation in some cases [47]. Additionally, a more advanced model that accurately characterizes the stochastic properties of the agent motion will produce proposal distributions with a higher likelihood of matching the target distribution. Over-estimation of the agent process, while it would introduce more diversity, would be a less desirable solution to the problem for the same reason as an over-estimation of sensor noise. Valuable information regarding the true propagation characteristics of the agent would be thrown away. Additionally, this approach would further mismatch the relative noise of the motion model and the perceptual model, leading to a severe decrease in diversity after resampling. Regularization techniques with adaptive-variance kernels ensure that a proper amount of diversity is introduced at specific regions of the posterior density. Without regularization, all particles would be propagated in the same fashion and valuable information about irregularity of the distribution could be lost.

Chapter 4

Simulated Results for Sample Diversity Recovery Methods

The previous chapter presented several approaches designed to improve particle filter SLAM performance in scenarios prone to particle depletion. One set of methods focused on the proposal distribution, before resampling, by drawing a more optimal set of particles for importance weight calculation. Other techniques adjusted particle locations after resampling with a set of regularization kernels that approximated a continuous distribution. This section presents experimental results showing the relative strengths and weaknesses of many of the ideas introduced in the previous section. The goal of this analysis was to use a simulated SLAM environment to

1. demonstrate particle filter SLAM performance at different measurement noise levels and show the effect of sample impoverishment on filter accuracy and stability.

2. thoroughly evaluate particle filter enhancements designed to recover sample diversity in depleted scenarios and improve the overall accuracy of the SLAM filter.

Three Rao-Blackwellized particle filters were developed, based on the FastSLAM algorithm presented by Montemerlo [32]. In addition, four regularization methods were coded. Each strategy was tested independently to characterize its performance

in different SLAM environments. Marriages between the filters and regularization methods were also tested to determine if their combined effect provides even greater filter accuracy.

4.1 Assumptions and Simulation Setup

The basic SLAM scenario modeled for this simulation consisted of a robot agent traveling around a small, elliptical track. At each time step the agent advanced according to a motion model that included a control input and noise from the motion error model. The robot then received simulated measurements from each landmark within its field of view. Only six landmarks existed in this 10 m $\times$ 10 m environment, and each landmark was uniformly spaced around the commanded path of the robot.

Figure 4-1 shows the simulation environment, the agent initial position, and the commanded path, as well as all landmarks that the robot encountered as it traveled. The robot was initialized with an *a priori* estimate of its pose and the location of three anchor features. The first task of the agent was to localize using relative measurements to these anchor features. As the simulation progressed, it would encounter three new features that it must map. With two full loops around the track, the robot would encounter previously mapped landmarks.

$\sigma_{s_0.x}$	$\sigma_{s_0.y}$	$\sigma_{s_0.v}$	$\sigma_{\theta_0.x}$	$\sigma_{\theta_0.y}$
1.5 m	1.5 m	0.0349 rad	0.3 m	0.3 m

Table 4.1: Initial RMS uncertainty of agent pose $\langle x, y, v \rangle$ and anchor feature location $\langle x, y \rangle$ for simulations.

4.1.1 Development of the SLAM Environment

This particular environment configuration was chosen in order to test several essential abilities of a successful SLAM estimation routine. Setting a higher uncertainty in the initial agent pose and providing anchor features required the filter to localize and improve its initial pose estimate. As it encountered new features, it needed

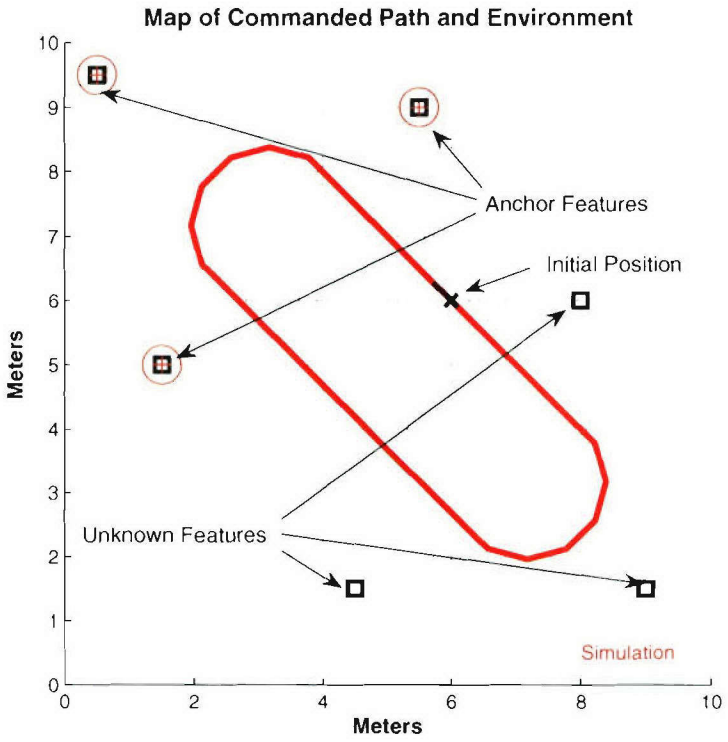


Figure 4-1: SLAM environment used for simulations with robot path and initial position shown.

to first recognize these landmarks as previously unmapped, and then augment its map accordingly. Finally, the agent was required to close an observation loop by measuring previously mapped landmarks. This is often the most difficult task of any SLAM algorithm, especially in cases of motion noise and accurate measurements. As discussed in section 2.5.1, algorithms in this situation tend to produce badly skewed maps with many additional phantom landmarks.

4.1.2 Robot Motion

Though its commanded angular and tangential velocities would realize two rotations about the track, kinematic errors in the agent motion model altered the true path significantly. A stochastic, four parameter motion model was used to represent slip scale factor and skid errors encountered in most wheel-based robots. The parameter values are listed in table 4.2, with the tangential and angular velocities at each time

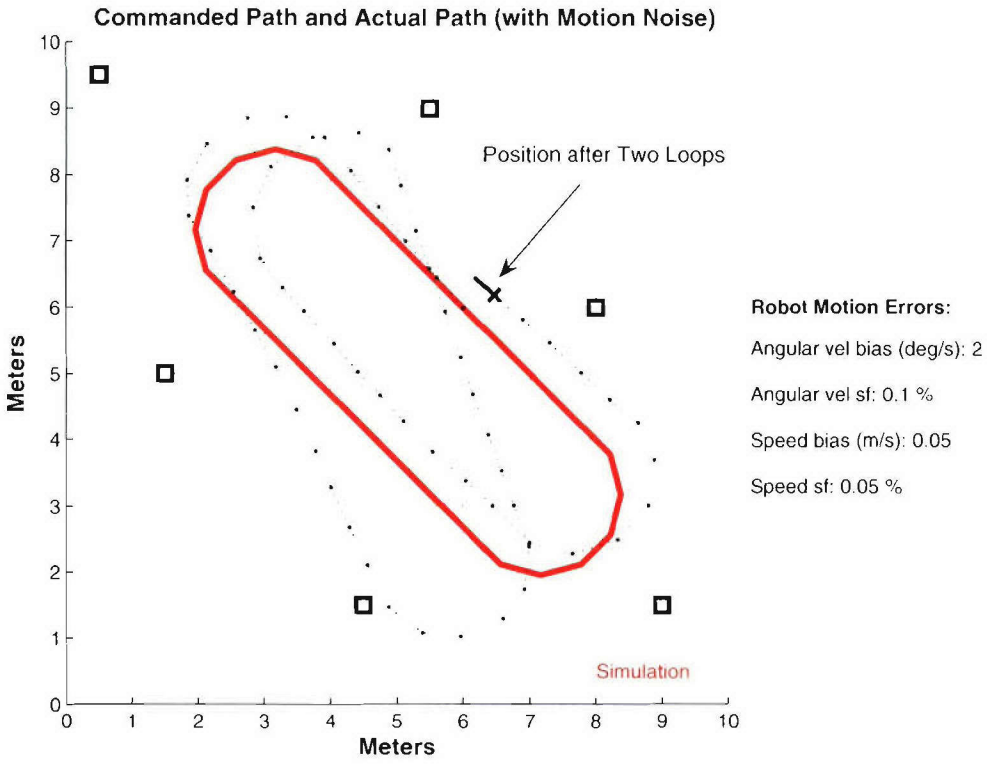


Figure 4-2: True agent path from one realization of the stochastic motion model.

point drawn according to:

$$v_t' \sim N(v; v_t, \alpha_{slip}|v_t| + \alpha_{skid}) \tag{4.1}$$

$$\omega_t' \sim N(\omega; \omega_t, \beta_{slip}|\omega_t| + \beta_{skid}) \tag{4.2}$$

v_t	ω_t	α_{slip}	α_{skid}	β_{slip}	β_{skid}
0.3068 m/s	0.0-0.5236 rad/s	0.05	0.05 m/s	0.1	0.0349 rad/s

Table 4.2: Agent motion model parameters, including commanded translational v_t and rotational ω_t velocities and skip and skid errors used for simulations.

One realization of this stochastic motion model was used as the true agent position for every simulation, shown in figure 4-2, in order to compare filter performance for equivalent scenarios.

4.1.3 Simulated Measurements

Characteristics of the agent measurement model are listed in table 4.3. It was assumed that each feature observation yielded a range-bearing measurement pair. Because the agent received simulated measurements from every landmark within its field of view, each SLAM algorithm needed the capability to process multiple measurements in a single time step. While easily incorporated in the standard FastSLAM 1.0, this particular enhancement is only briefly addressed by Montemerlo in his development of FastSLAM 2.0. Incorporating sensor measurements in proposal calculation is not a trivial task when measurements from new landmarks must be considered. While most measurement model characteristics were remain fixed throughout this analysis, the range uncertainty, considered the independent variable for most trials, was manipulated in order to evaluate filter performance. Lowering this RMS value from 1.0 m to 0.001 m would reveal how each algorithm responds as the measurement noise is reduced and a proposal-perceptual distribution mismatch is encountered.

Field of View	Maximum Range	RMS Bearing Uncertainty	RMS Range Uncertainty
3.142 rad	7.0 m	0.0175 rad	0.001-1.0 m

Table 4.3: Measurement model specifications and uncertainties

4.1.4 Performance Metrics

The primary metric for filter accuracy was the circular error probable (CEP) of the agent x-y pose location. Though agent heading error and landmark position error were not directly measured by this metric, the correlated nature of the SLAM problem infers that errors in these unmeasured parameters would contribute to the pose CEP.

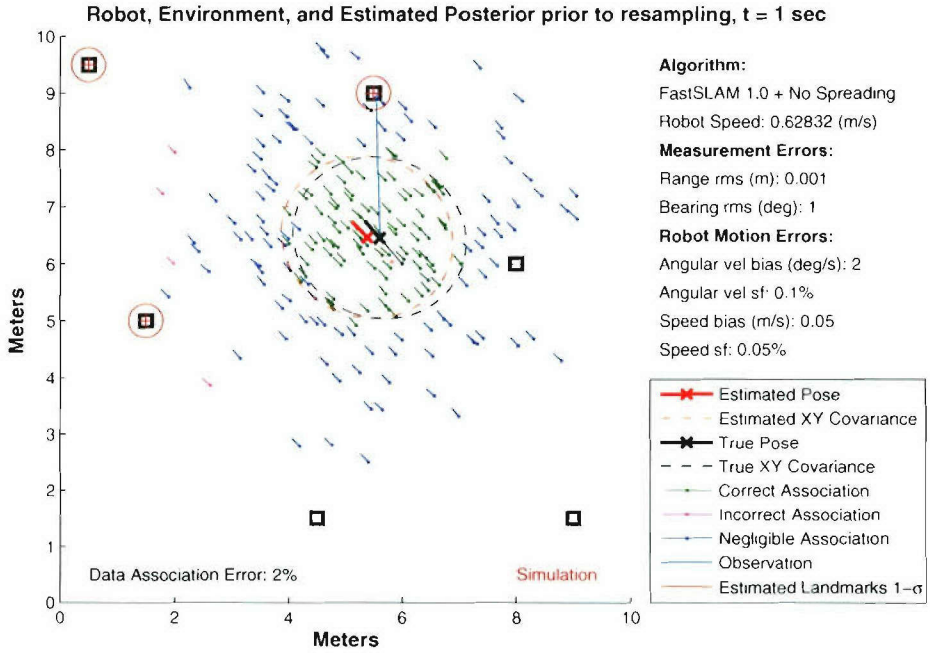
4.2 SLAM Posterior Estimation

Figure 4-3 illustrates a single run through a SLAM scenario. With a large initial uncertainty represented by a large spread of particles, the first task of the filter

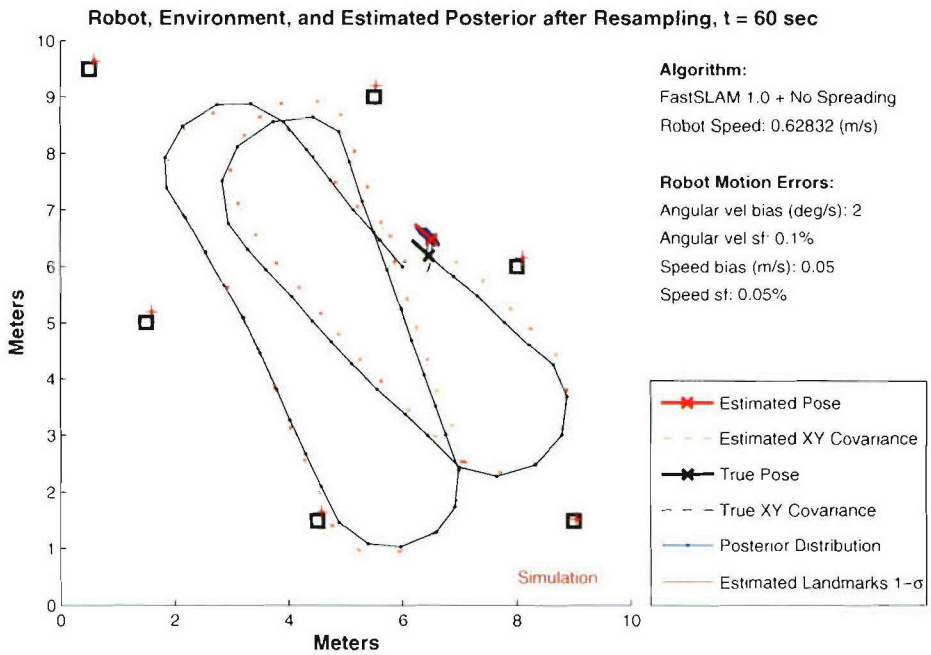
was to decrease pose uncertainty using measurements from anchor features. Each particle was propagated according to the stochastic motion model and measurements were used to weight each particle according to the maximum likelihood heuristic as outlined in section 2.3.1. Particles were then resampled according to weights, reducing the overall uncertainty of the filter. The posterior distribution at the end of the 60 second simulation is also shown 4-3(b). Notice that new landmarks have appropriately been added to map and the filter has tracked the pose of the robot with reasonable accuracy despite noisy kinematics.

4.2.1 Sample Impoverishment and Particle Drift

The difficulty of capturing an evolving posterior distribution using a SMC method with a finite number of particles became apparent when measurement noise was reduced without a corresponding drop in kinematic noise. This mismatch created an environment prone to cases of particle depletion. Notice in figure 4-4(a) that within a few seconds of initialization, the maximum likelihood heuristic with a highly accurate sensor has assigned non-negligible weights to only a small portion of particles. Consequently, samples were "stacked" at these points during resampling. Instead of a smooth posterior representing the actual uncertainty of the agent, the distribution was reduced to only a few discrete hypotheses. At this point the wide sense mean of this depleted posterior still provided an accurate estimate of the true position. Over time, however, sever particle drift was evident (figure 4-4(b)). The filter then had little chance of recovering to a reasonable accuracy. At the end of 60 seconds, filter pose error was less than one meter, primarily due to the fact that the robot control commands in a noise-free realization traced two loops. A look at estimated landmark locations shows the correlation between pose error and landmark error, as accurate measurements coupled with pose inaccuracy produced many false landmarks. Though Montemerlo discusses the use of negative information to eliminate phantom landmarks [32], the implementation of this feature proved more difficult in practice and was therefore not included in the algorithms for this analysis.

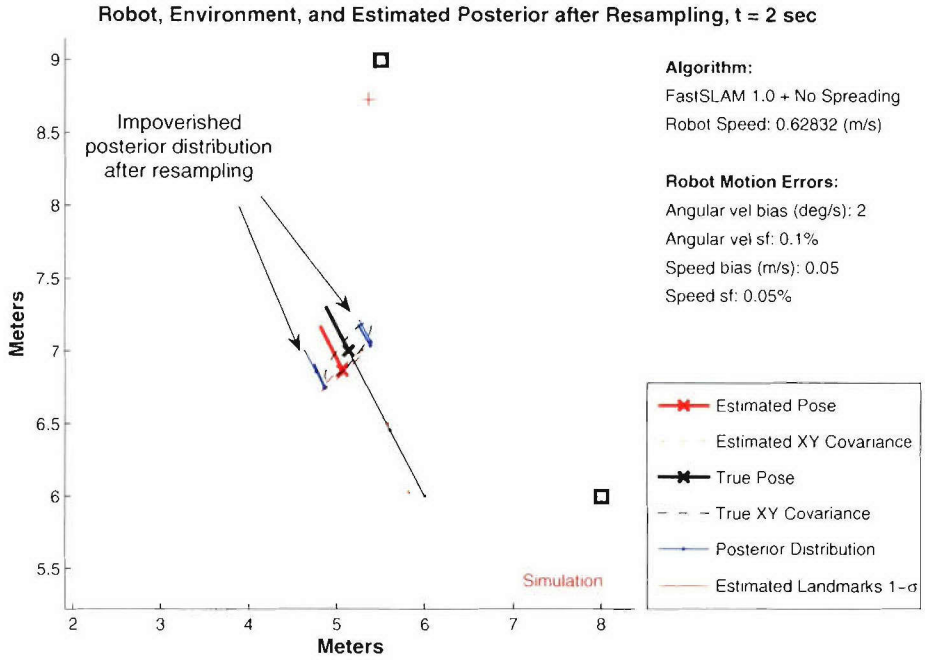


(a)

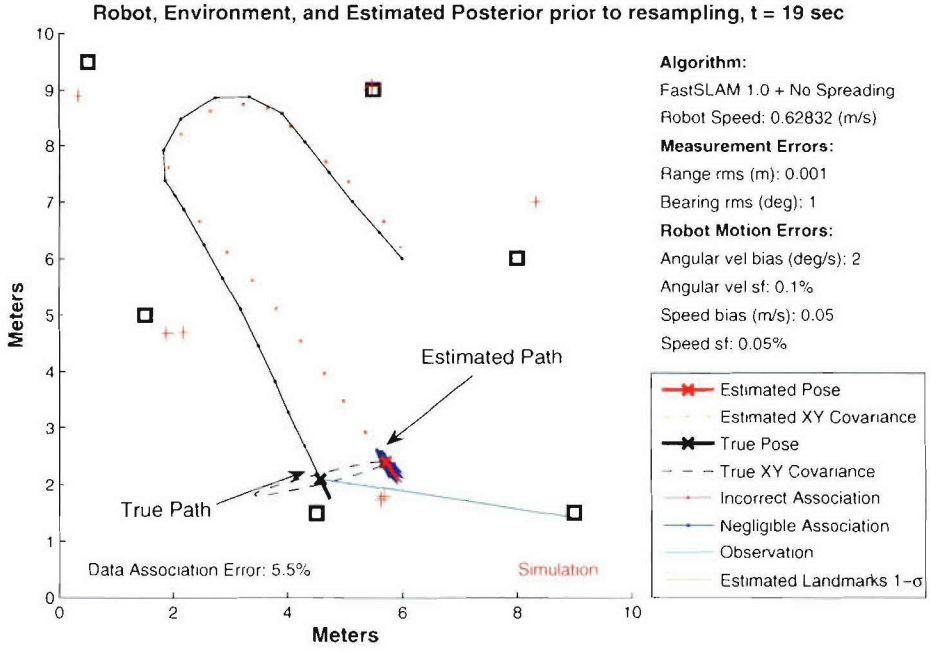


(b)

Figure 4-3: A typical SLAM scenario showing initial uncertainty (a) and the estimated posterior after 60 seconds (b).



(a)



(b)

Figure 4-4: An impoverished posterior (a) leading to particle drift as the simulation progresses (b). The end of the simulation is shown in figure 4-5.

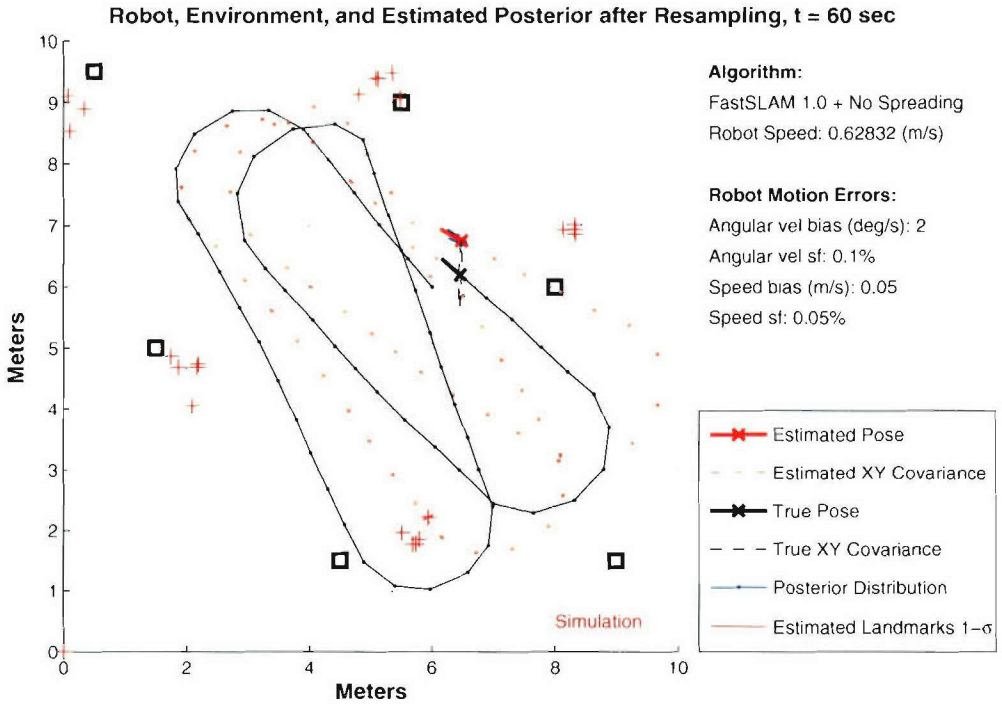


Figure 4-5: At the end of the scenario, particle drift has lead to severe inaccuracies in both the pose and feature estimates. Many spurious landmarks were created.

4.3 SLAM Algorithms

4.3.1 FastSLAM 1.0

The first of the three filters used in this analysis was FastSLAM 1.0. Since it is basically the basic Rao-Blackwellized particle filter, it formed the backbone of the other two filter methods. It this set of tests, it also served as a benchmark against which the performance all other filters and regularization algorithms were measured. It is designed to operate with unknown data association and is therefore ideally suited to the SLAM problem. As with the more advanced filters used in this analysis, 200 particles were used to estimate the SLAM posterior. The importance threshold for new landmarks, in this filter and the others, was 10^{-6}. Because it uses only motion model information to propagate the proposal distribution and not information from recent sensor measurements, it can in some cases be the most sensitive to a motion-sensor accuracy mismatch.

4.3.2 FastSLAM 2.0

FastSLAM 2.0, Montemerlo's more advanced particle filter that includes recent measurement information in the proposal distribution, was coded as a second filter for this analysis. Propagation of the proposal distribution at each step began with a draw from the motion model using a pre-defined initial covariance. This mean and covariance were then updated using EKF equations and the current measurement. Data association in this case was more difficult because the algorithm needed to associate a measurement with a known or new landmark before proposal update. The advantages of this filter are described in literature to outweigh this computational burden, as Montemerlo proves one-particle convergence in a Linear-Gaussian SLAM estimation scenario. Based on literature results, FastSLAM 2.0 was expected to perform best without additional regularization after resampling.

4.3.3 Auxiliary Particle Filter

Using FastSLAM 1.0 as a basis, a Rao-Blackwellized Auxiliary particle filter was developed as another example of measurement influence on the proposal distribution. It is similar in every respect to FastSLAM 1.0, except an additional resampling step was added consistent with the Auxiliary Particle Filter algorithm [39]. Currently used only in position tracking scenarios, it was coded to evaluate whether or not the additional resampling step improves sample diversity and accuracy in a SLAM environment. It incorporated multiple measurements per time step in the same fashion as FastSLAM 1.0, and used the combination of these measurements in shaping the propagation of the proposal.

4.3.4 Regularization Algorithms

Table 4.4 lists the regularization methods coded for analysis and briefly describes the properties of the kernels used for particle adjustment in each one. It also mentions the Markov Chain Monte Carlo criterion, which can supplement any of the four spreading algorithms.

Regularzation Method (Algorithm Pseudoname)	Description
SpreadX	Fixed Gaussian regularization. Particles at each resampled point are adjusted using a fixed-variance Gaussian kernel. Ref. section 3.5.1
SpreadX2	Gaussian regularization with the standard deviation of the kernel dependent on the number of particles sampled at that point and a fixed parameter. Ref. section 3.5.2
SpreadX3	Gaussian regularization with the standard deviation of the kernel dependent on the number of particles sampled at that point and the empirical covariance matrix of the particle distribution before resampling. Ref. section 3.5.3
RPF	(Regularized Particle Filter) Gaussian regularization dependent on a fixed, derived parameter and the empirical covariance matrix of the particle distribution prior to resampling. Ref. section 3.4.1
MCMC	Markov Chain Monte Carlo criterion. Can supplement any above regularization method. Ensures that regularized particles asymptotically approach the optimal Bayesian posterior distribution. Ref. section 3.4.2

Table 4.4: Summary of regularization methods tested in simulations

4.4 Filter Accuracy and Diversity Analysis

Each filter was tested with a singular run through the SLAM scenario: the CEP of the filter position estimate was extracted at each second. Additionally the number of unique particle states after resampling was recorded to provide a measure of diversity at that time step. The purpose of this test was to show performance of the filters in a controlled scenario and to observe the relationship between the diversity of the particle filter posterior and the CEP.

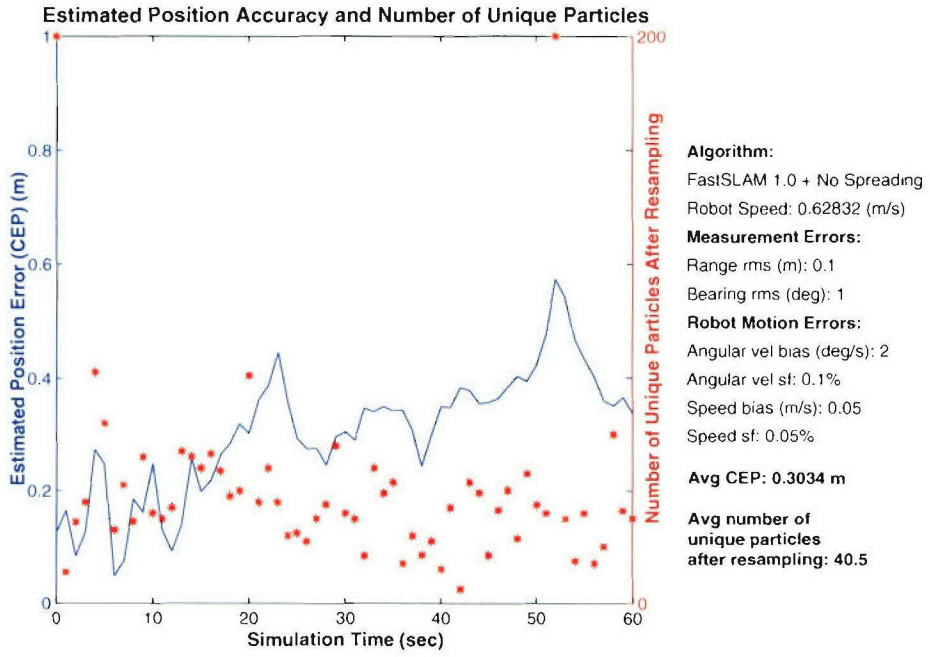
Figure 4-6(a) shows FastSLAM 1.0 baseline performance at 0.1 m range measurement RMS error, with an average CEP of 0.3034 m and an average number of unique particles of around 40 after each resampling step. Figure 4-6(b) shows the same filter and scenario, only this time with a mismatch in the relative accuracy of agent motion and feature observations. There is a pronounced difference in both the accuracy of

the filter and the number of unique particles. An average of 10.5 distinct states in the posterior implies a significant depletion in state space coverage. The average CEP for this test was 0.4284, but more important conclusions can be drawn by evaluating the CEP time history for the run. At points the CEP drops to around 0.1 m, however the long-period variations between 0.07 m and 0.85 m show that the filter is not tracking the true position at all, but instead propagating the posterior according to the motion model. Feature observations did little to affect the position estimate as it sometimes wandered close to the true position but then drifted away. If the accuracy of the agent heading estimate were captured, it would likely reflect a difference in the true and estimated agent heading that reflect poor state estimate despite close proximity.

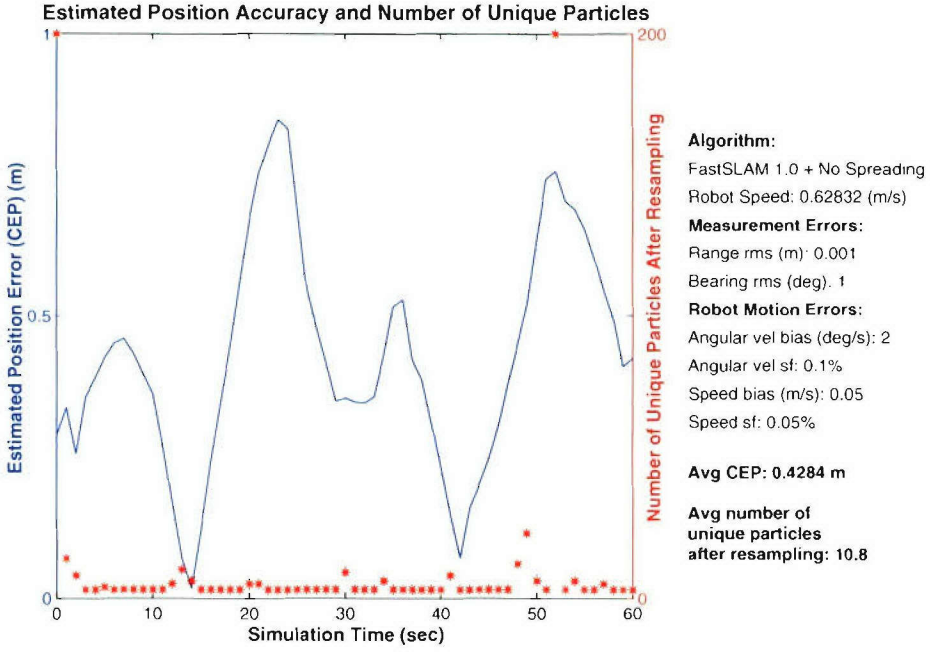
Though some diversity would be recovered as particles propagated according to a stochastic representation of uncertainty in the motion model, it is reasonable to assume that this propagation may not have sufficiently recovered the loss in state space coverage caused by resampling, which caused the particle drift, a loss in accuracy, and CEP instability.

4.4.1 FastSLAM 1.0 with Regularization

Figure 4-7 shows the result of a simple regularization method added to the FastSLAM 1.0 algorithm. The average number of particles remained at 200 for the duration of the scenario since SpreadX readjusted every particle according to a set of kernels. There was a considerable drop in position error at the 0.1 m RMS measurement noise level, though it is unclear whether the increase in particle diversity was the sole cause of the increased estimation accuracy. There was only a slight reduction in average CEP gained at the 0.001 m RMS measurement noise level when a regularization method is added. More importantly, short period adjustments in the CEP reveal a sensitivity to measurement information not evident in the CEP results for FastSLAM 1.0 without regularization.

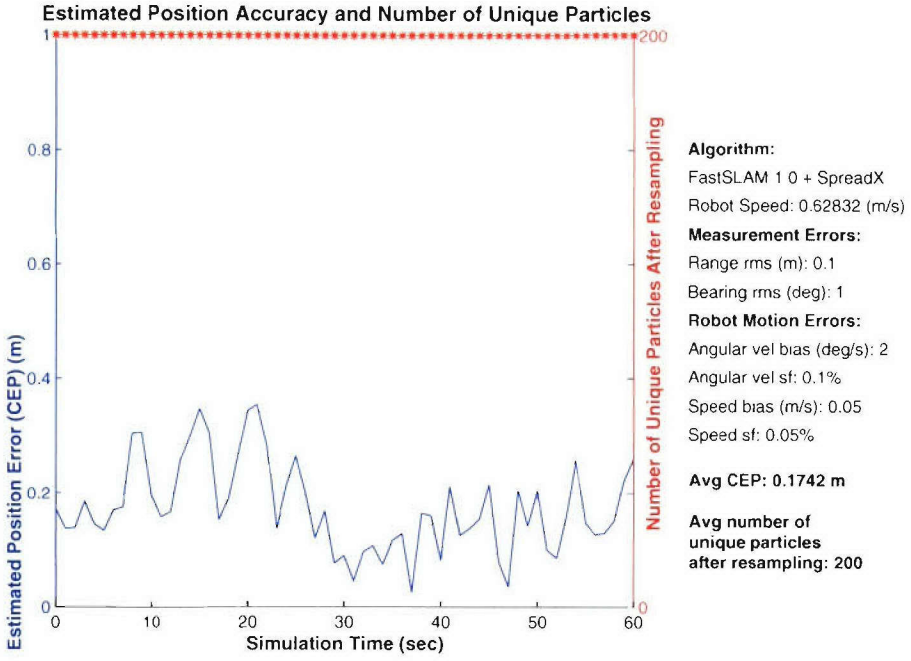


(a)

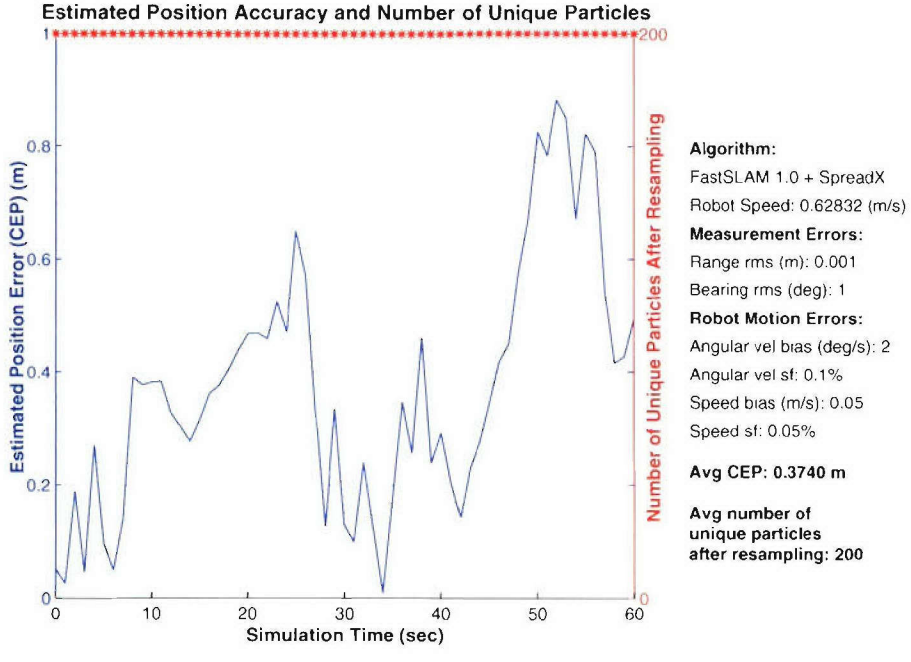


(b)

Figure 4-6: FastSLAM 1.0: CEP and diversity for single run at range measurement RMS error of 0.1 m (a) and 0.001 m (b).



(a)

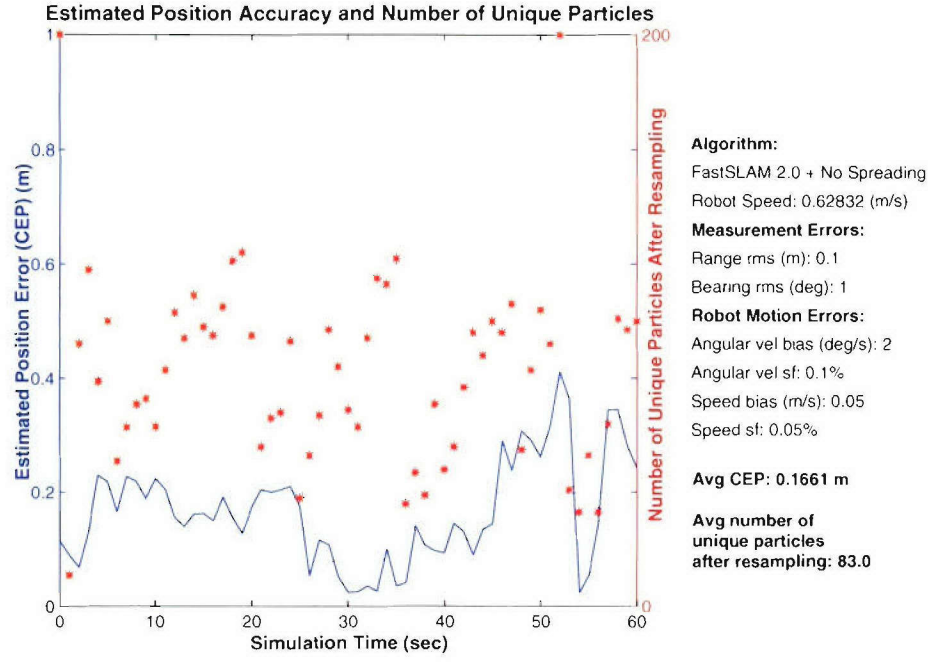


(b)

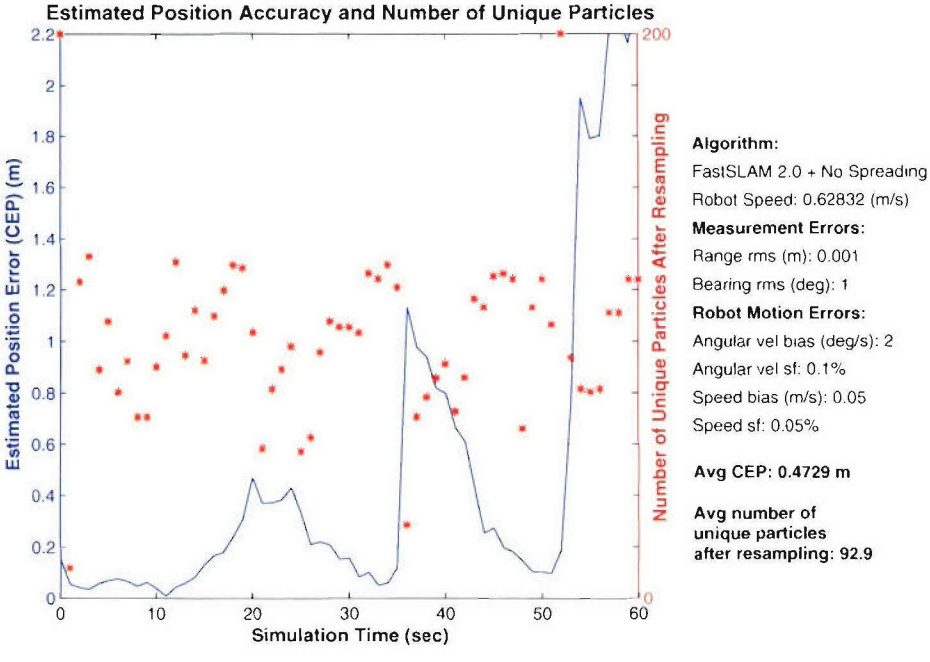
Figure 4-7: FastSLAM 1.0 with Regularization: CEP and diversity for single run with range measurement RMS error of 0.1 m (a) and 0.001 m (b). Regularization provided a slight improvement in CEP at both measurement noise levels.

4.4.2 FastSLAM 2.0 Accuracy and Diversity Analysis

Switching algorithms to FastSLAM 2.0 demonstrated the effects of a more advanced proposal distribution on particle diversity and filter accuracy, with results shown in figure 4-8. At the 0.1 m RMS range measurement noise level, FastSLAM 2.0 without regularization yielded a noticeable improvement in accuracy from FastSLAM 1.0, and an additional increase in average number of particles after resampling. This was expected, as literature testifies to an improved posterior estimate over the standard Rao-Blackwellized particle filter. However, testing the algorithm again at the 0.001 m RMS measurement level resulted in a disturbing loss of accuracy near the end of the run. Since pronounced position tracking losses happened suddenly, a reasonable explanation could involve poor propagation effects at this measurement noise level. In original FastSLAM, position tracking errors happened gradually, as a result of drift. Conversely, errors in FastSLAM 2.0 happen suddenly, likely not from a gradual propagation away from the true mean but an erroneous propagation altogether. Since measurement information was included in proposal calculation, the algorithm could have made large data association errors that led to rapid deviation from the prior estimate. Figure 4-9 shows the FastSLAM 2.0 posterior after the 60 second scenario, including mapped landmarks, at two levels of range measurement noise. FastSLAM 2.0 experienced a breakdown in overall posterior accuracy at the end of the scenario, figure 4-9(b), as some particles made correct data associations and remained close to the true agent position and other particles incorrectly associated measurements to false landmarks. As is evidenced in this more detailed view of the filter posterior estimate, FastSLAM 2.0 may have had a reasonable chance of recovering to an accurate posterior estimate at a later point in time, but only if more correct data associations were made and erroneous hypotheses eliminated. Because of the delayed decision making inherent in the FastSLAM structure, a 60 second SLAM scenario may not encapsulate a longer term robustness that could emerge with more loops around the environment. Nevertheless, it is important to characterize the true performance of this filter since some engineering applications may call for proven stability over



(a)



(b)

Figure 4-8: FastSLAM 2.0: CEP and diversity for single run at range measurement RMS error of 0.1 m (a) and 0.001 m (b).

possible long term accuracy.

4.5 Monte Carlo SLAM Performance Analysis

In order to better characterize the average behavior of the filters and regularization methods coded for the analysis, 100 Monte Carlo runs were performed for each filter and regularization method. Each of the three filters were tested first without regularization, then in a marriage with each of the four regularization methods, giving a total of 15 possible combinations. In addition, each of these combinations was tested at 15 different range measurement RMS values, from 0.001 to 1.0 m, focusing on the following performance trends:

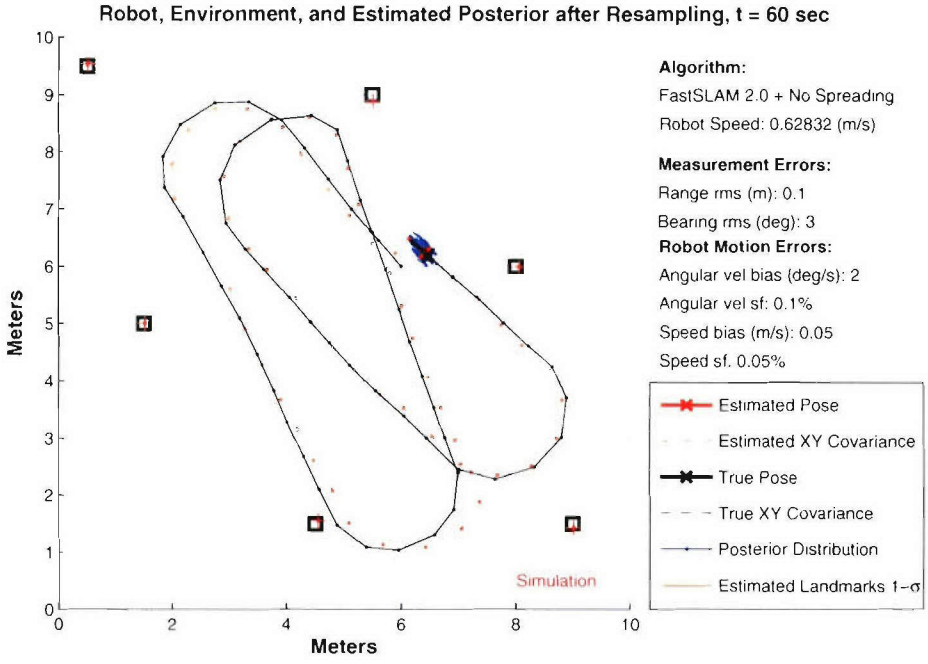
1. The overall effect of regularization and the average performance of each regularization method.

2. A comparison of the accuracy provided by the three filter types

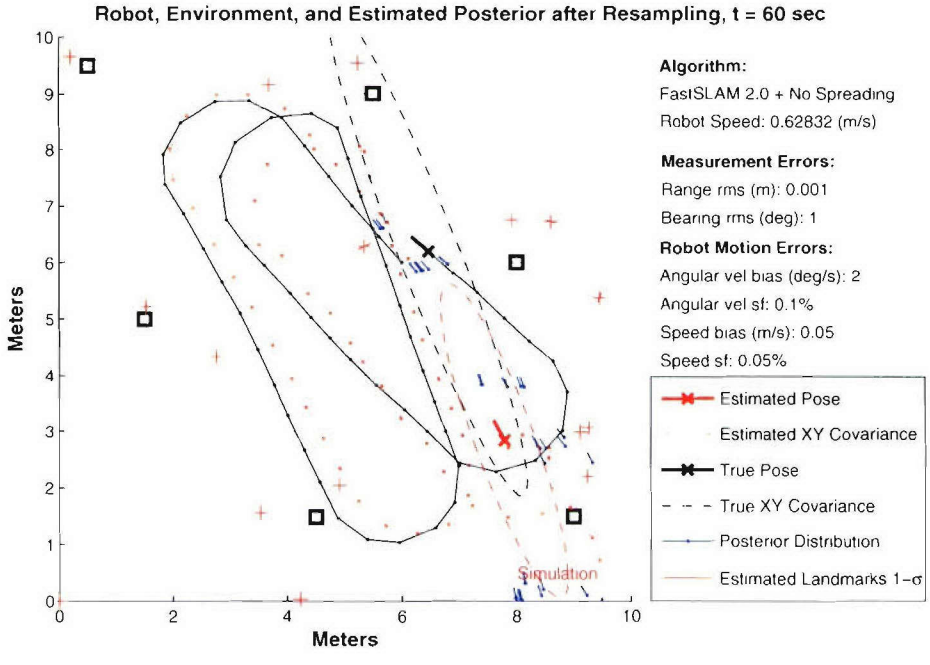
3. The trend in agent position accuracy for each filter-regularization combination as measurement noise is reduced to the point where particle depletion occurs.

4.5.1 Filter Performance Results

The first set of Monte Carlo runs compared the performance of the three filters coded for this exercise, absent of any regularization algorithms. Results are given in figure 4-10. As range measurement RMS error is reduced, average CEP for each approach drops as expected but then increases dramatically at the lowest RMS error levels. Even FastSLAM 2.0, though it maintained accuracy to a lower RMS level than the other filters, met a point at which the the motion-sensor mismatch causes the adverse propagation and instability mentioned earlier. One surprising result is that the Auxiliary particle filter performed much worse in this scenario than the original FastSLAM algorithm. It seems that this method was far more susceptible to the effects of particle drift despite the inclusion of an additional resampling step. One possible

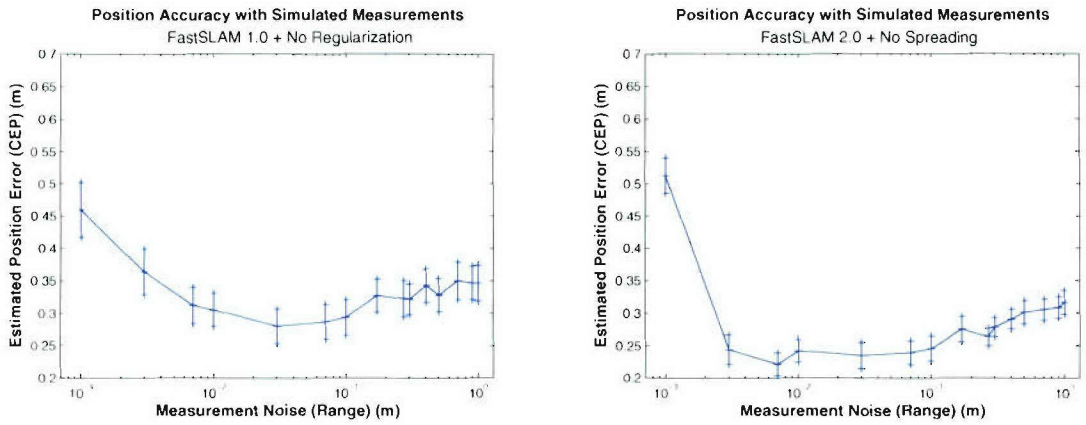


(a)



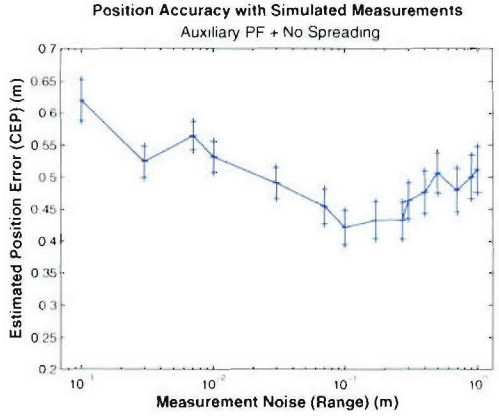
(b)

Figure 4-9: SLAM posterior estimation with FastSLAM 2.0. At very low measurement noise levels, substantial errors in the estimated posterior are noted (b).



(a)

(b)



(c)

Figure 4-10: Average filter CEP and 95% confidence intervals for various RMS measurement noise levels.

explanation is that the resampling step before particle propagation, though it may in some cases increase the chances of particles propagating to a more favorable region for importance weight calculation, reduced particle diversity to an even greater degree than in a standard particle filter. Evaluating an isolated case, shown in figure 4-11, reveals that the additional resampling step did in fact produce an accurate position estimate for the first half of the simulation. Eventually, however, this filter drifted substantially from the true position. It could be that, as in FastSLAM 2.0, using measurement information in particle propagation results in a proposal distribution that is

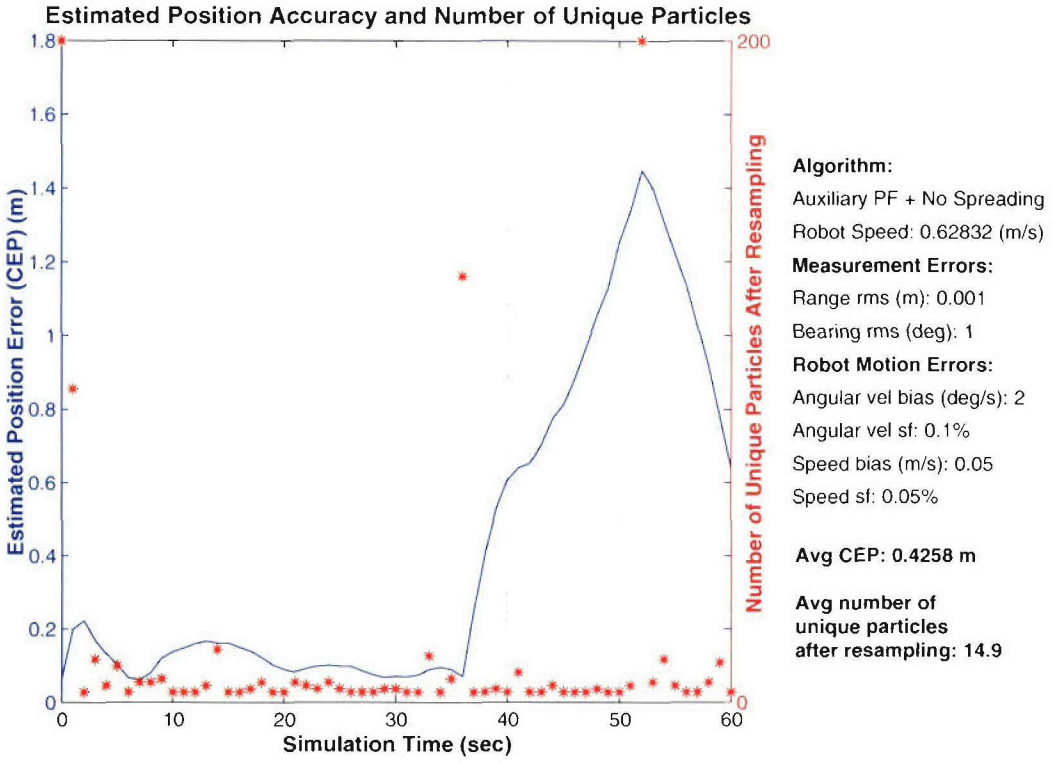


Figure 4-11: Auxiliary Particle Filter: CEP and diversity for single run. Range measurement RMS error of 0.001 m.

more susceptible to the adverse effects of data association errors. Unlike FastSLAM 2.0, however, particles in an APF are not directly drawn the proposal that includes these association errors, letting the multiple hypothesis property eventually eliminate bad particles. Instead, the Auxiliary particle filter uses measurement information to trim the size of the proposal; thus the effects of data association errors as they shape the filter proposal at each time step can never be undone.

4.6 Regularization Performance Results

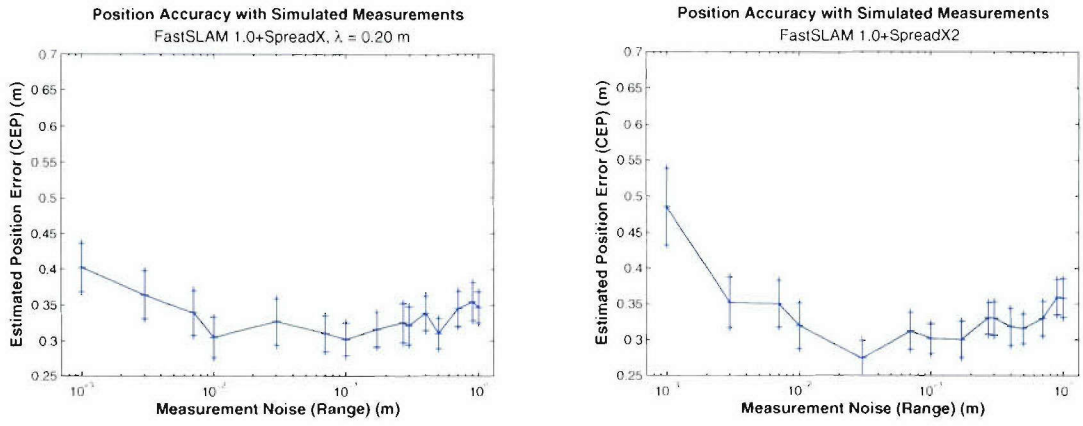
4.6.1 SpreadX Parameter Selection

Since the SpreadX regularization algorithm utilized an empirically chosen parameter for kernel generation, a set of Monte Carlo runs was performed to determine the optimal parameter for this SLAM scenario. Table 4.5 shows average CEP for parameter

Regularization Parameter, λ (m)	Range RMS Error, (m)			
	0.001	0.01	0.1	1.0
0.05	0.5008	0.3110	0.2884	0.3786
0.10	0.4288	0.3023	0.3022	0.3710
0.15	0.4314	0.3182	0.2948	0.3467
0.20	**0.4024**	**0.2990**	**0.2927**	**0.3408**
0.25	0.4141	0.3040	0.3016	0.3517
0.30	0.4296	0.3054	0.3009	0.3413
0.35	0.4686	0.3281	0.3062	0.3447
0.40	0.4624	0.3341	0.3210	0.3576
0.45	0.4445	0.3104	0.3104	0.3502
0.50	0.4410	0.3567	0.3143	0.3625

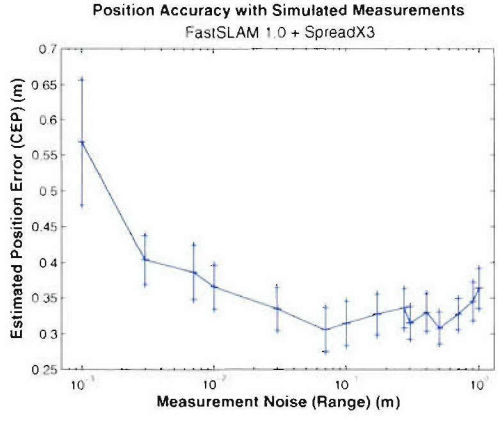
Table 4.5: SpreadX regularization parameter determination, using average CEP (m) of 100 Monte Carlo runs. Results show $\lambda = 0.20$ m to be the optimal value for this simulation.

values between 0.05 m to 0.5 m, with the best performance at each noise level from $\lambda = 0.2$ m. This regularization parameter value was subsequently chosen for each implementation of SpreadX for this analysis. Figure 4-12 shows the result of a Monte Carlo performance analysis using FastSLAM 1.0 and each regularization method from table 4.4. Despite the variety of approaches employed by the different regularization algorithms, the only method with an improvement in performance at the lowest measurement noise level was the SpreadX algorithm, using a fixed-Gaussian kernel. All other methods produced mean CEP values comparable to or greater than the basic FastSLAM filter at each isolated measurement noise level. This is initially surprising, considering that SpreadX is the most basic of all tested regularization methods and does not involve a complicated, derived parameter for kernel generation. The backbone of the SpreadX algorithm, and possibly the reason that it fared well in this analysis, is an empirically chosen constant regularization kernel. By testing many possible values and arriving at an optimal spreading parameter for this scenario, SpreadX introduced a proper amount of diversity for this environment configuration, measurement model and agent motion characteristics. Also, SpreadX injects a guaranteed amount of diversity into the posterior distribution, while in other algorithms the amount of diversity is variable and may not be sufficient in some cases. For instance, SpreadX2 adjusts resampled states with kernels that depend on the "stack

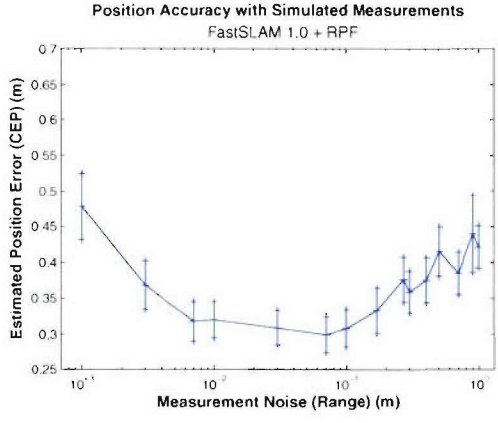


(a)

(b)



(c)



(d)

Figure 4-12: FastSLAM 1.0 with four different regularization algorithms. Average CEP and 95% confidence intervals for various RMS measurement noise levels.

height" of particle sets. In the situation where measurements are not associated with any previously mapped landmarks, all particles receive equal weights, and the effect of the regularization algorithm is minimized. Incidentally, in cases where no existing landmarks are observed and new features are being mapped, more diversity should be recovered through regularization in order to keep the uncertainty of these new landmark positions high. This would give the estimation routine a greater chance at closing loops and recognizing previously mapped features. It is highly likely that the filter position estimate could drift away from the true position before mapping new features. With an accurate sensor, these new features are initialized with undue position certainty unless a reasonable amount of diversity is kept. SpreadX maintains a larger amount of diversity in this situation than the other methods since the particle cloud is not allowed to converge below a certain RMS distance. Nonetheless, regularization itself does not appear to increase filter accuracy in any case but the most severe mismatch between motion and sensor noise levels. Average CEP values for FastSLAM 1.0 without regularization are, for the most part, better without regularization at every other measurement noise level.

4.7 Markov Chain Monte Carlo (MCMC) Analysis

Since the Markov Chain Monte Carlo acceptance criterion can supplement any regularization algorithm, it was appropriate to test this algorithm with each of the four spreading methods. Because of its solid theoretical foundations, it was expected that the inclusion of this criterion in any state adjustment would only improve the average CEP of the filter position estimate. Table 4.7 summarizes the results of a Monte Carlo analysis with average CEP at four different measurement RMS error levels. Performance was improved for SpreadX2, SpreadX3, and the RPF method, but only at the lowest and highest measurement RMS levels. No improvement in SpreadX performance was offered by the MCMC step. Again, this result is surprising given

Regularization Algorithm	Range RMS Error, (m)			
	0.001	0.01	0.1	1.0
SpreadX	**0.4024**	**0.3040**	0.3016	**0.3467**
SpreadX + MCMC	0.4586	0.4011	0.2968	0.3744
SpreadX2	0.4852	**0.3197**	0.3015	0.3576
SpreadX2 + MCMC	**0.4395**	0.4197	0.3022	**0.3285**
SpreadX3	0.5682	**0.3651**	0.3144	0.3633
SpreadX3 + MCMC	**0.4756**	0.4526	0.3170	**0.3396**
RPF	0.4778	0.3191	0.3070	0.4214
RPF + MCMC	**0.4265**	0.3254	0.3082	**0.3671**

Table 4.6: MCMC criterion performance analysis for FastSLAM 1.0 and various regularization methods. Average CEP values (m).

the fact that the Markov Chain Monte Carlo algorithm theoretically provides for the convergence of a regularization method to the optimal Bayesian posterior. As in the case with the more advanced regularization methods, it could be hypothesized that the MCMC criterion restricts the recovery of diversity in the filter position estimate after resampling, as particles are only moved provided they meet a strict selection criterion that involves a current feature measurement.

MCMC restricts particle adjustment based on the current measurement, thus one possibility is that incorrect data association between measurements and features could keep the regularization method from recovering particle diversity lost in resampling. As in the case where incorrect data association adversely affected proposal propagation in FastSLAM 2.0 and the Auxiliary particle filter, using measurement information to restrict regularization could potentially weaken the filter in cases of particle drift where estimated feature locations are in fact erroneous.

4.8 Performance Summary for Filter/Regularization Marriages

Table 4.7 shows comprehensive results of Monte Carlo analysis of each algorithm developed for this exercise, including marriages between filters and regularization algorithms. FastSLAM 2.0 without regularization produced optimal filter CEP per-

Measurement Range RMS Error = 0.001 m					
Filter Type	No Regularization	SpreadX	SpreadX2	SpreadX3	RPF
FastSLAM 1.0	0.4596	**0.4024**	0.4852	0.5682	0.4778
FastSLAM 2.0	0.5123	0.5852	0.5196	0.5832	0.8589
Auxiliary PF	0.6199	0.7474	0.5812	0.5930	0.6150

Measurement Range RMS Error = 0.01 m					
Filter Type	No Regularization	SpreadX	SpreadX2	SpreadX3	RPF
FastSLAM 1.0	0.3050	0.3040	0.3197	0.3651	0.3191
FastSLAM 2.0	**0.2419**	0.2570	0.2621	0.2533	0.2653
Auxiliary PF	0.5315	0.5467	0.4714	0.5289	0.5183

Measurement Range RMS Error = 0.1 m					
Filter Type	No Regularization	SpreadX	SpreadX2	SpreadX3	RPF
FastSLAM 1.0	0.2939	0.3016	0.3015	0.3144	0.3070
FastSLAM 2.0	**0.2452**	0.2882	0.2506	0.2598	0.2958
Auxiliary PF	0.4214	0.4321	0.3970	0.4087	0.3993

Measurement Range RMS Error = 1.0 m					
Filter Type	No Regularization	SpreadX	SpreadX2	SpreadX3	RPF
FastSLAM 1.0	0.3463	0.3467	0.3576	0.3633	0.4214
FastSLAM 2.0	0.3161	0.3864	0.3146	0.3153	0.6807
Auxiliary PF	0.5116	0.6642	0.4886	0.5208	0.4888

Table 4.7: Average CEP values (m) for filter-regularization marriages. Bold values indicate significant results.

formance at all measurement noise levels except the extreme cases of 0.001 m and 1.0 m RMS range error. In these situations, other algorithms produced better results. As explained previously, FastSLAM 1.0 with the simple spreading routine, SpreadX, provided a much lower CEP than any other algorithm. At 1.0 m RMS range error, FastSLAM 2.0 with SpreadX3 provided the best average CEP, but the 95% confidence intervals for this result do not support significance for this conclusion, thus the result is not bolded in the table.

4.9 Summary

It would be interesting to perform this analysis of SLAM algorithms with a more diverse set of performance metrics than simply the CEP of the position estimate at each time. A lot could be inferred by evaluating the error in landmark positions at certain key moments, such as when the filter maps a previously undetected feature. Of course, altering the SLAM environment by changing the positions and relative spacing of landmarks, giving the agent more or fewer anchor features, or simply changing the commanded robot path could have a profound impact on the performance of each filter and regularization scheme.

In some ways it is relatively difficult to improve upon the accuracy of a SLAM filter once landmarks have been mapped with errors. As landmarks are mapped, the filter will default to these locations when performing data association. If the filter is already experiencing particle drift, landmarks will be placed in badly skewed positions and loop closure will become difficult.

By recovering sample diversity through particle readjustment, regularization can add additional uncertainty in the estimate of agent pose location, especially when particle adjustments are based on a fixed kernel. The results of the Monte Carlo runs suggest that this could be one way to keep the filter from locking on an erroneous heading and creating a skewed map. The effects of the motion-measurement accuracy mismatch scenarios reveal an interesting performance paradox for particle filters when applied to the SLAM problem. Observations from unseen or newly mapped features are processed with the same amount of sensor accuracy as well-established and accurately marked features. However, if the estimation algorithm places full faith in precise measurements from new features, it will eventually experience a loss in diversity. While in some cases, such as the global localization problem where a well established and complete map is known *a priori*, a loss in diversity could signal convergence to an accurate position estimate. If the filter is tasked with localization and mapping, this loss in diversity is detrimental and will eventually lead to the phenomena experienced in this analysis.

Recovering sample diversity through regularization, though it may not be the most optimal or principled prescription, could help maintain a necessary uncertainty in the locations of new landmarks until successive measurements or observations of well established features are obtained. One possible improvement on the standard FastSLAM 1.0 algorithm not studied in this analysis would be to initialize new landmark locations with a larger degree of uncertainty than just the measurement noise covariance matrix. Keeping new landmarks more uncertain and then gradually fixing their position as further observations are processed could be one way to keep the estimated posterior resistant to the effects of particle drift.

Chapter 5

Experimental Results for SLAM Algorithms

In order to gain a better understanding of the performance of the FastSLAM algorithm with improved regularization, several runs with measurement information from a real-world SLAM scenario were performed. In this analysis, only one FastSLAM filter type and regularization combination was used in a side-by-side comparison with an EKF SLAM algorithm. The environment for this comparison consisted of five box-shaped objects surrounding a 5.1-meter straight-line path. A cart carrying measurement equipment was pulled along this path, pausing every 0.3068 meters for a sensor measurement. At the end of the path, the cart was rotated 90 degrees, with measurements taken every 30 degrees. This particular path and set of measurement points provided for simple calculation of true positions as a reference for filter comparison. In total, this set of measurement and motion data would simulate an 18-second scenario with the agent advancing for 15 seconds then performing a 90 degree right turn over the final 3 seconds. Several assumptions were made to give the scenario a more realistic quality. First, a motion model was developed that would match characteristics of a robot advancing along this straight-line path with relatively noisy odometry information. This motion model would be used in the propagation step for both the EKF and particle filter algorithms, producing a similar situation to position tracking in the presence of actual motion noise. Motion model parameters used for

both algorithms are listed below for both the translation and rotation phases. Error parameters and are given in table 5.1.

Agent Motion Model

Translation Phase: $(t = 1 - 15)$

$$v_t' \sim N(v; v_t, \alpha_{skid})$$
$$\omega_t' \sim N(\omega; 0, \beta_{skid})$$

Rotation Phase: $(t = 16 - 18)$

$$v_t' \sim N(v; 0, \alpha_{skid})$$
$$\omega_t' \sim N(\omega; \omega_t, \beta_{skid})$$

v_t	ω_t	α_{skid}	β_{skid}
0.3068 m/s	0.5236 rad/s	0.03 m/s	0.0524 rad/s

Table 5.1: Motion model parameters for FastSLAM-EKF comparison. Only two motion error parameters are used: skid errors in tangential and rotational velocity.

5.0.1 Swiss Ranger Feature Observations

Range and bearing measurements were collected by processing data from the CSEM Swiss Ranger 3000, a LIDAR imaging system that provided a high-resolution, three-dimensional representation of the environment. Raw outputs from the Swiss Ranger included the accurate ranges for every pixel within the field of view. By processing these ranges using a median filter and searching for large gradients in the range pattern, individual features were identified and translated into a range, bearing and elevation relative to the agent. Since in this scenario pose and landmark locations were tracked in only two dimensions, measurements were projected onto a planar environment. Features in this case were the edges of objects, since these locations produced the range differences identified by the gradient-based feature extraction technique. For more information on the Swiss Ranger imaging system see [44]. Specifics of the measurement noise model for the Swiss Ranger and feature extraction are listed in table 5.2. The effective uncertainties for range and bearing listed are for the entire feature extraction process. They do not represent the noise characteristics of the Swiss Ranger alone.

2D Effective Field of View [44]	Non-ambiguity Range [44]	Effective Bearing Uncertainty, RMS	Effective Range Uncertainty, RMS
47.5 deg	7.5 m	1.0 deg*	0.05 m*

Table 5.2: Effective measurement model specifications and uncertainties for the feature observation system, including Swiss Ranger, and feature extraction. *Values are approximate.

5.0.2 Algorithm Specifics

The EKF SLAM algorithm used in this analysis was developed according to the basic framework described in section 2.1.1, with a $2N+3$ posterior state vector. It utilized the maximum likelihood data association heuristic with a fixed likelihood threshold for new landmark initialization. To prevent false landmark initialization from spurious measurements, a feature had to be observed twice before incorporation into the EKF map. The only modification to the FastSLAM algorithm outlined in the previous chapter was the addition of the SpreadX regularization method (for a description see table 4.4). This particular filter and regularization combination was chosen because it demonstrated both accuracy and robustness with simulated data, outperforming all other combinations in situations prone to particle depletion. An initial particle distribution was drawn according to the Gaussian parameters representing the same *a priori* mean and covariance as in the EKF.

5.0.3 Initial Estimates and Anchor Features

The agent starts from the same point for every test, with filter *a priori* estimates changed for comparison of filter qualities. Anchor features are occasionally included in a priori posterior estimates, the locations of which were determined after processing the measurements in order to place them in favorable locations for recognition by the filter as it processes measurements. Each anchor feature given was assumed an initial uncertainty RMS of 0.3 m in both x and y directions.

5.0.4 Performance Metrics

The primary metric for this evaluation was position CEP with respect to the true agent position, but observation of the entire posterior is also plotted and will be helpful in understanding the strengths and drawbacks of each filter and why a certain filter performed as it did for each situation. Several important observations were deduced visually as the SLAM posterior estimate from each filter type was superimposed on the true environment. In particular, this bird's-eye view of both filter estimate and truth helps identify situations where data association errors were made and how these errors effect the processing of subsequent feature observations.

5.1 Experimental Results

5.1.1 Scenario One: Position Tracking and Feature Mapping

Figure 5-1 shows the environment used for all tests, as well as the initial posterior estimate and uncertainty for the first experiment. The initial pose estimate reflected only a slight inaccuracy in the *a priori* notion of agent position, with the true position still well within the uncertainty ellipse (1σ). No anchor features are provided for this first assessment, which tested pose tracking and feature mapping abilities given a well-localized initial estimate. Figures 5-2(a) and 5-2(b) illustrate the end result posterior estimate after processing all motion and measurement information for the scenario. Notice that the path of the dead-reckoning estimate from propagation of odometry information has deviated significantly from the true path, while both filter estimates have maintained a reasonably accurate position estimate by mapping observed features and then adjusting a motion-based estimate by subsequent measurements of this map. In addition, CEP time history is shown for both filters and the dead-reckoning estimate from one realization of the stochastic motion (figure 5-3). Visual inspection of the estimated posterior in figure 5-2(b) shows a slightly skewed map, a probable cause of the slightly inferior CEP performance of FastSLAM in figure 5-3. Examination of the landmark covariances for both methods shows notably larger ellipses

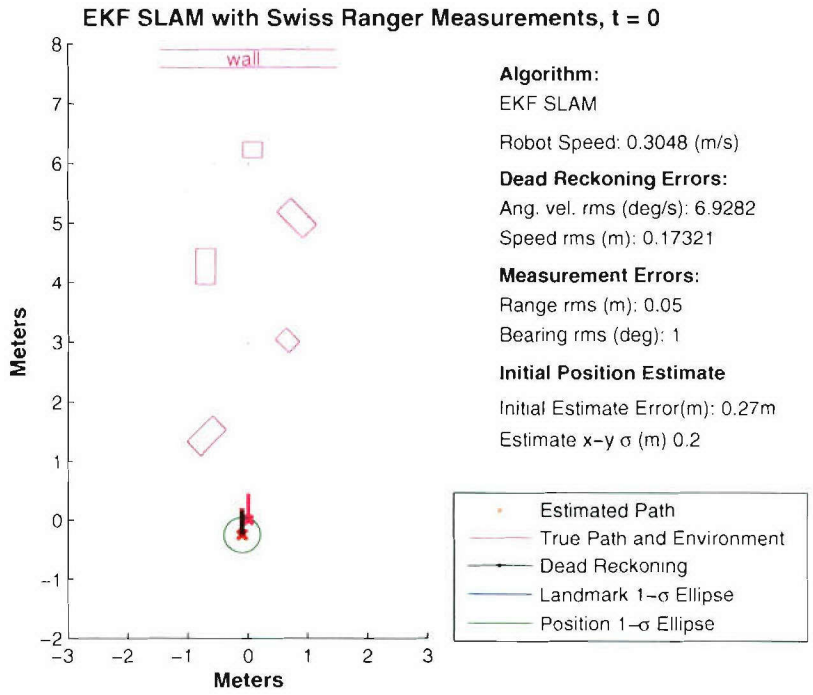
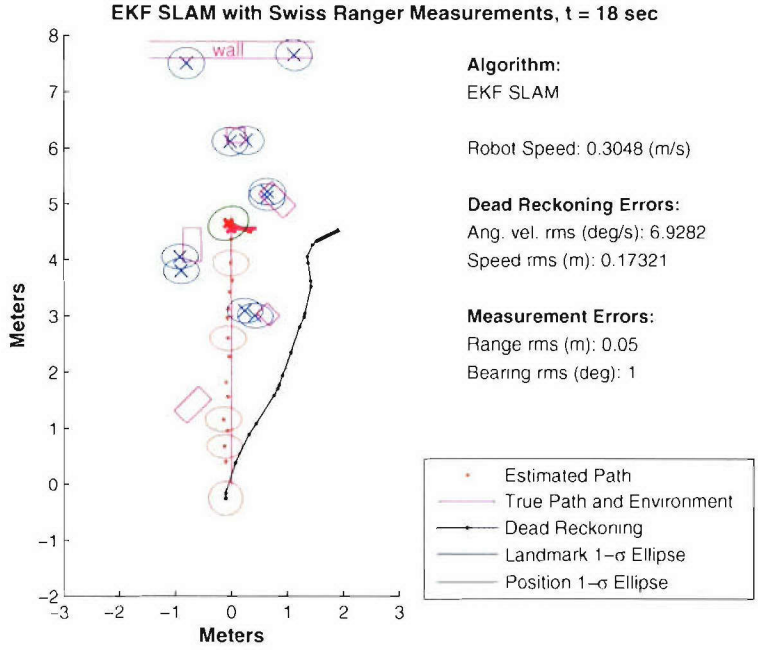
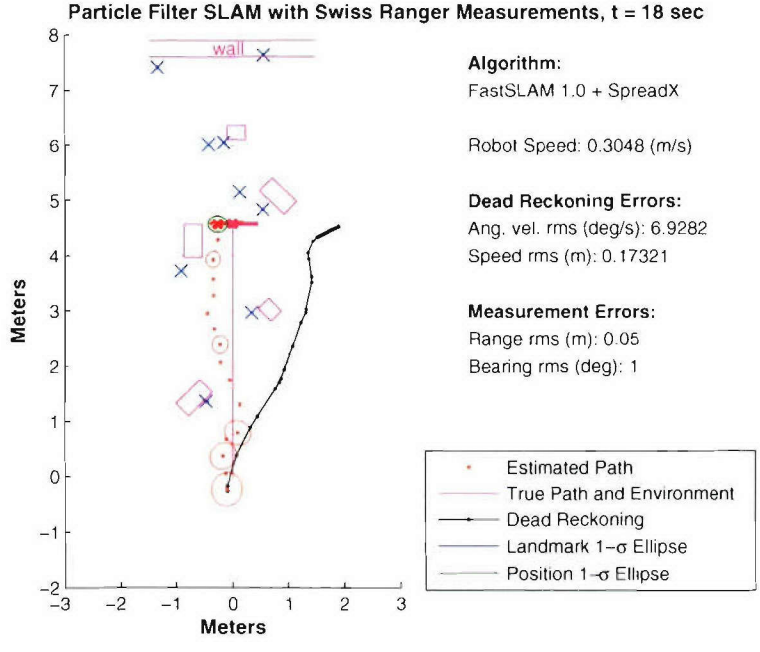


Figure 5-1: Environment truth and initial pose estimate for SLAM scenario. No anchor features, accurate initial estimate.

for EKF landmarks than with FastSLAM. The fundamentals of the SLAM problem, as mentioned in section 2.1, state that landmark and pose uncertainties are firmly linked. That is, uncertainty in the pose location at the time the landmark must be included in landmark estimate covariance. Red ellipses indicate historical covariance ellipses at each position. With accurate measurements from the Swiss Ranger, the sizes of feature uncertainty ellipses in figure 5-2(a) are approximately the same size as the agent pose uncertainty at the time they were mapped. Little or no additional uncertainty is added by measurement noise. The tendency of FastSLAM to produce a false certainty in landmark positions, as mentioned in section 2.5.1 is noticed in this scenario (figure 5-2(b)). Landmark uncertainty was reduced to virtually zero at the end of the test.



(a)



(b)

Figure 5-2: Final posterior estimate for EKF (a) and FastSLAM (b) after 18-second scenario. No anchor features, accurate initial estimate.

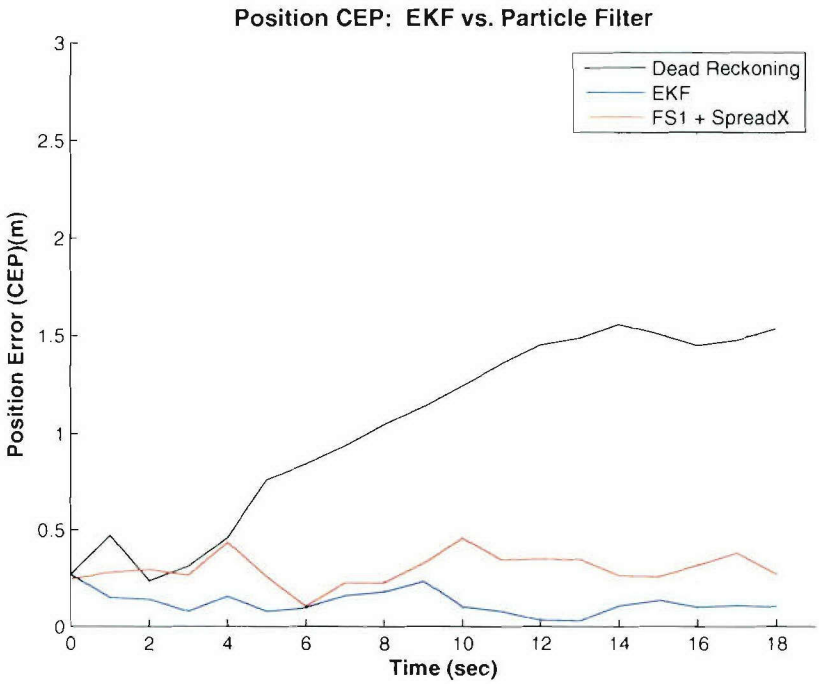


Figure 5-3: Agent position CEP time history for dead reckoning estimate and both filter estimates. No anchor features, accurate initial estimate.

5.1.2 Scenario Two: Localization

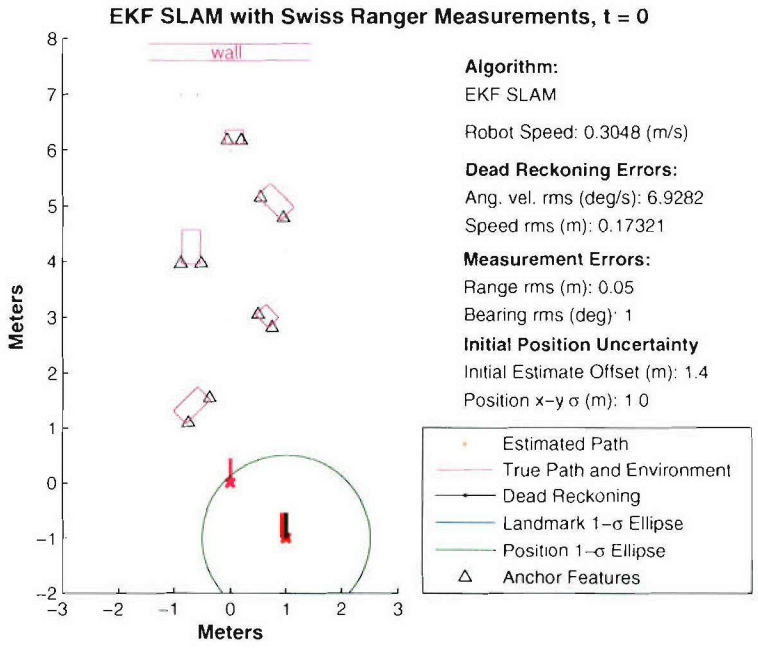
The ability of each filter to localize given an accurate an accurate *a priori* map was assessed by providing each filter with a full set of anchor features. The initial pose estimate, however, was offset significant distance from the true location. Uncertainty in this estimate was set at 1 m (1σ) to include the true initial position within the covariance bounds. Figure 5-4 shows both the initial environment and the posterior estimate after the first measurement. While individual particles are not illustrated in the figures, it is clear that particles in the initial dispersion located near the true estimate made correct data associations with stored landmarks and were weighted highly. The path estimate reflects a dramatic shift in the mean after this first re-sampling step as particles are repopulated to these few discrete points. FastSLAM responded quickly to correct data associations and recovered from poor initial estimate.

Figure 5-5 shows the estimated posteriors for the test, and the EKF was in this

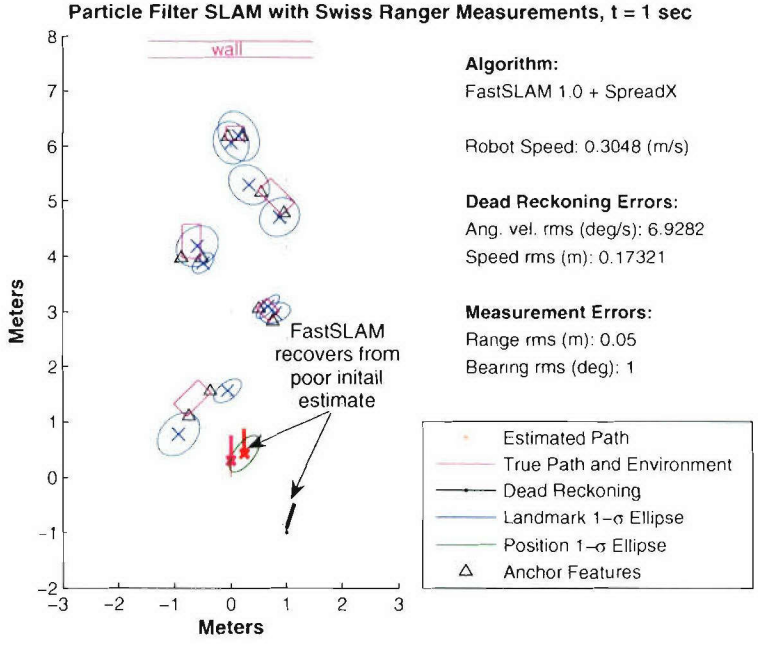
case unable to recover from this poor initial estimate. While uncertainty decreased over the length of the scenario, it does not appear that the EKF would have converged, even with a longer scenario. The EKF made a critical data association error early in the experiment. Red arrows describe this association, as feature clusters clearly correspond to other true landmarks. The EKF also created additional features with measurements that differed from any stored landmarks. Other anchor features were never associated with measurements, as their covariance ellipses reflect the initial uncertainty of 0.3 m. CEP time history for this scenario is shown in figure 5-9. The fundamental drawback of the EKF in this scenario was its inability to track multiple hypotheses of its location. Instead, it created a map that corresponded with the poor initial position estimate and a data association error early in the test. These experimental results are consistent with other sources that testify to the strengths of particle filters in tackling problems of global localization based on an accurate a priori map [17, 45].

5.1.3 Scenario Three: Localization and Mapping

The third and final test was designed to stress the ability of each filter to both localize based on anchor feature observations and then proceed to map the remaining features in the environment. Only three anchor features were given, and once again a poor initial position estimate was provided (see figure 5-7). Results in this scenario are similar to the previous experiment. Once again the EKF algorithm built a map consistent with a poor initial position. It was able to maintain a proper heading estimate despite motion noise, but the map estimate is significantly shifted from the true feature positions (figure 5-8(a). Again, the rigid relationship between pose and landmark uncertainty is evident. All mapped landmarks reflect an uncertainty similar to the one measurement that was associated with a known anchor feature, albeit incorrectly. After that point, there is no apparent additional convergence, of either the pose or subsequent mapped landmarks.

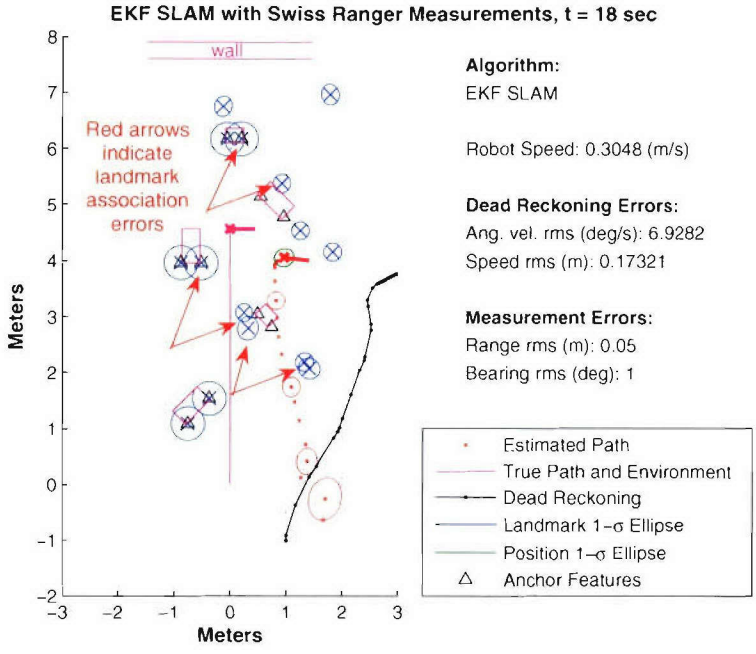


(a)

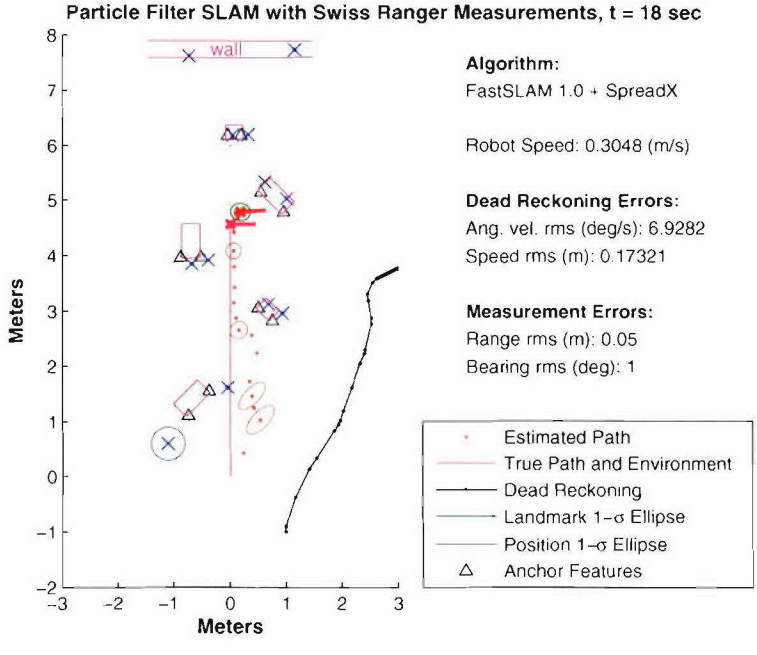


(b)

Figure 5-4: Initial posterior estimate with poor initial estimate and three anchor features (a). Position estimate recovery for FastSLAM after first measurements (b).



(a)



(b)

Figure 5-5: Final posterior estimate for EKF (a) and FastSLAM (b) after 18-second scenario. Red arrows show data association errors of feature clusters.

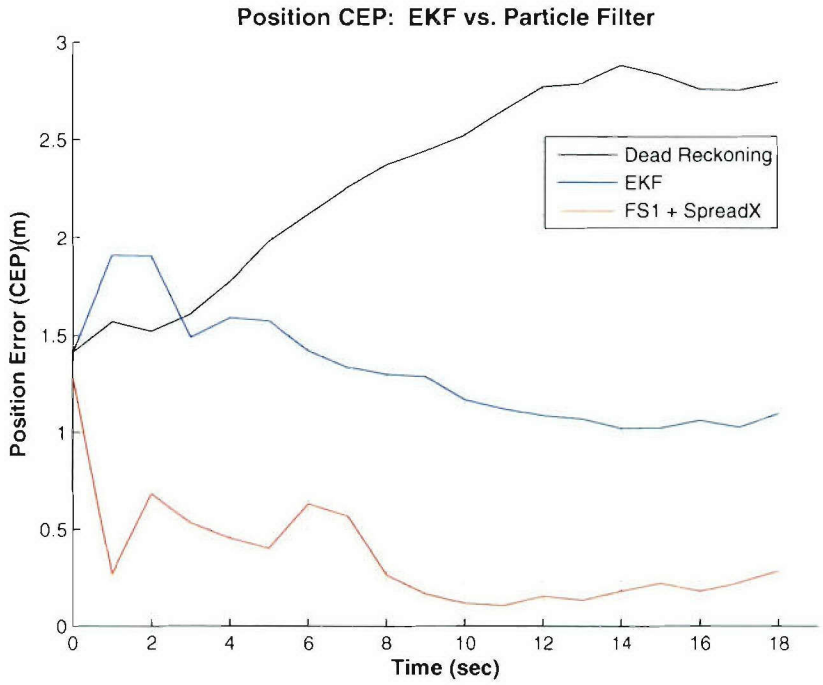


Figure 5-6: Agent position CEP time history for dead reckoning and filter estimates, poor initial pose estimate and accurate initial map estimate. EKF convergence is limited by initial pose inaccuracy.

5.2 Summary

The results of these experiments demonstrate that while the particle filter and EKF can provide similar robustness and accuracy in SLAM cases with little initial uncertainty, the particle filter approach clearly outperforms a single-hypothesis EKF in cases where the agent is initially poorly localized. While a comprehensive evaluation that would support the conclusive acceptance of a particular particle filter SLAM algorithm over EKF based algorithms would require results from a broad range of scenarios and Monte Carlo tests, the single data set and few variations in this test revealed several basic conclusions. First, there are cases where particle filter SLAM algorithms and EKF based algorithms yield comparable results, both in robustness and general accuracy. The EKF algorithm has the advantage of analytically approximating the optimal Bayesian posterior under the restrictive assumptions mentioned in chapter 2, whereas the particle filter is a sampling approach that only approximates

the optimal posterior when properly configured. Convergence of a sampling approach will depend on the number of samples used, a proper weighting heuristic, and an adequate model of motion and sensor characteristics, among other factors [8]. Additionally, the principles behind the basic EKF SLAM algorithm and its performance have been well documented. It is currently held as the "gold standard" approach to state estimation, with acceptable performance in certain SLAM situations [18, 32]. As evident from the results in this section, there are also some cases where the performance of the EKF SLAM algorithm breaks down and the FastSLAM algorithm maintains an accurate estimate of the robot pose and landmark locations. Finally, the improved robustness and accuracy of FastSLAM over the basic EKF algorithm in certain scenarios lies in its ability to track multiple hypotheses of the pose locations, landmark locations, and data associations between landmarks and measurements. A more involved performance analysis for each filter with Swiss Ranger data would have helped firm many of these conclusions. Future tests should involve the agent making a complete loop around the environment in order to test SLAM filter performance during loop closures.

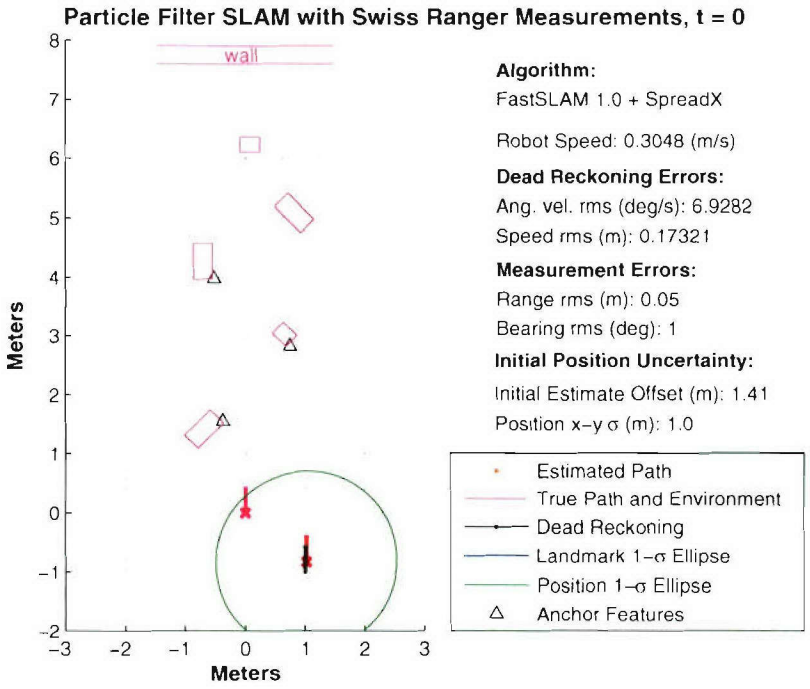
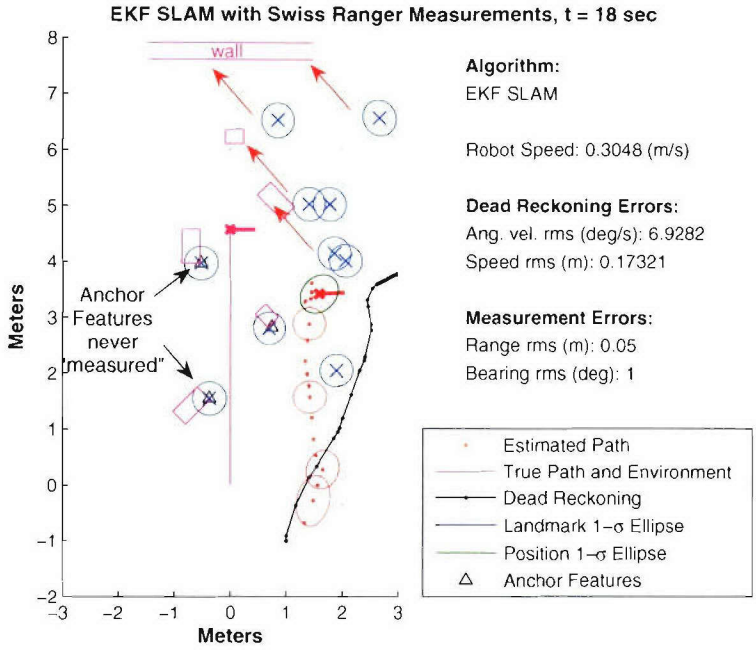
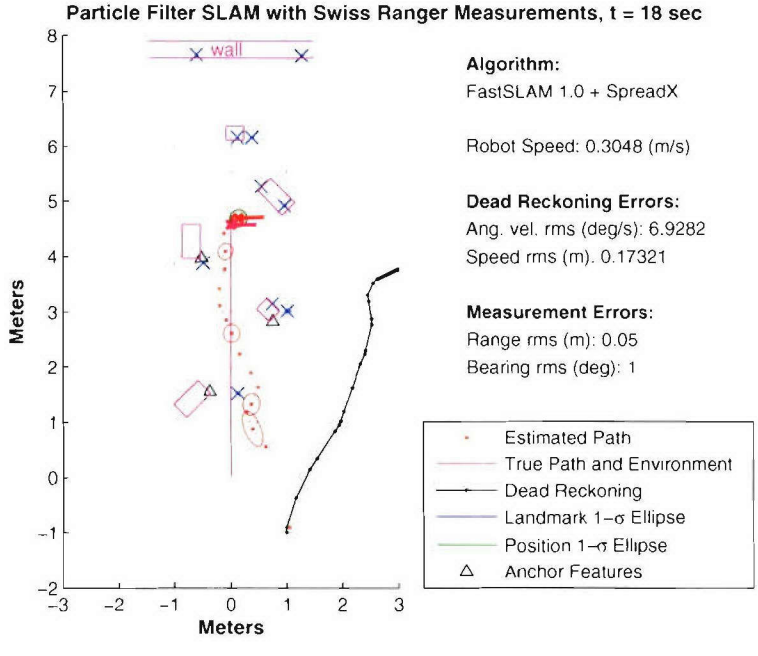


Figure 5-7: Environment truth and initial pose estimate for SLAM scenario, inaccurate initial estimate and partial map knowledge.



(a)



(b)

Figure 5-8: Final posterior estimate for EKF (a) and FastSLAM (b) after 18-second scenario. Red arrows show skewed map of EKF from data association errors.

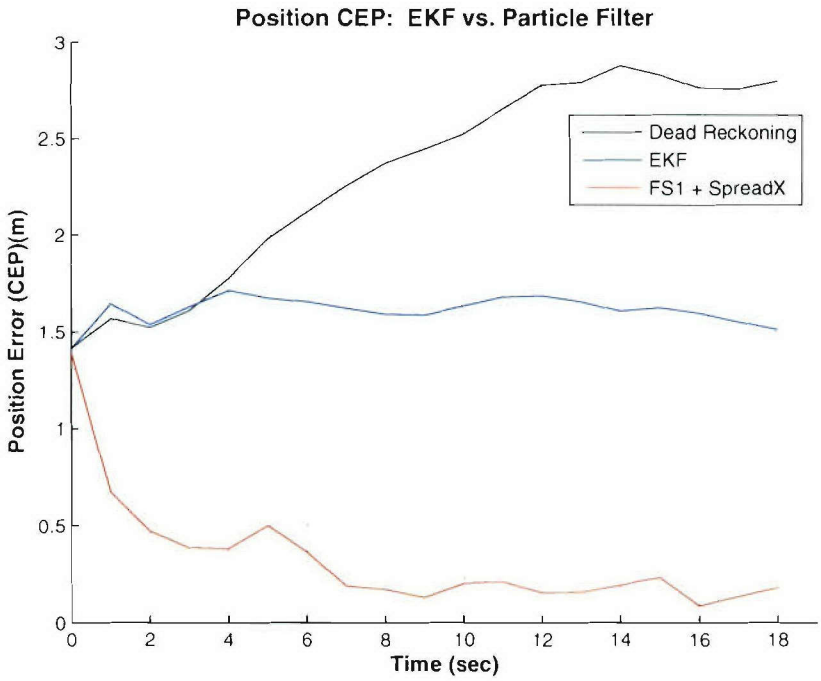


Figure 5-9: Agent position CEP time history for dead reckoning and filter estimates. poor initial pose estimate and three initial anchor features. Once again, EKF convergence is limited by initial pose inaccuracy.

Chapter 6

Conclusions and Future Work

6.1 Research Conclusions

Results from the Monte Carlo analysis of FastSLAM derivatives combined with various regularization techniques revealed a substantial improvement in sample diversity and accuracy with FastSLAM 2.0 in most situations. At the lowest measurement RMS values, where the proposal-perceptual mismatch is most severe, FastSLAM 2.0 was prone to disturbing losses in CEP pose accuracy. It was in this situation that FastSLAM 1.0 with the addition of a fixed-variance regularization algorithm, SpreadX, maintained a better estimate of agent pose. The addition of SpreadX provided a 0.05 m average CEP improvement over the standard FastSLAM algorithm, and a 0.1 m CEP improvement over FastSLAM 2.0. These conclusions support the adoption of FastSLAM 1.0 with an empirically derived, fixed-variance regularization algorithm over the more complicated FastSLAM 2.0 in SLAM situations where robustness of the filter in the presence of extremely low measurement noise is a primary performance requirement.

Comparing FastSLAM 1.0 + SpreadX with an extended Kalman filter in the same actual SLAM scenario with Swiss Ranger feature observations highlighted the ability of particle filter-based algorithms to recover from situations of high initial uncertainty. Starting from initial pose errors of 1.4 m, both filters processed feature observations with partial or complete *a priori* maps of the environment. The RBPF-

based algorithm recovered to within 0.4 m pose CEP by the end of the scenario, whereas the EKF maintained a pose error of at least 1.0 m. These results demonstrate the flexibility of an RBPF algorithm and its ability to recover accuracy despite initial error by efficiently tracking multiple agent pose hypotheses. This feature makes it an ideal algorithm for estimation in SLAM situations with of large or global initial uncertainty and a partial or complete initial map.

The experiments with Swiss Ranger measurements demonstrate the abilities that RBPF SLAM algorithms offer in an unknown environment where conventional localization methods such as GPS are unavailable. The statistical correlation between landmark location and pose estimates is clearly evident. When exploited, this correlation can provide a better solution than simply dead reckoning via odometry information. The drawbacks of the pose-landmark statistical relationship are also seen. In the particle filter, cases prone to sample impoverishment and spurious landmarks can produce maps that are locally accurate, but badly skewed and shifted from the true map. Cases with both simulated and real data illustrated this effect in varying degrees. In the EKF algorithm, tested in similar scenarios to the FastSLAM algorithm, this correlation between posterior states allowed the single hypothesis carried in the EKF mean to accept data association errors, preventing it from converging to the true agent location. The result was a shifted map, offset by the initial estimate error. It reflects that a single-hypothesis EKF algorithm in SLAM environments is limited in accuracy by its initial pose estimate.

6.2 Future Work

An encompassing goal of this research effort was to thoroughly evaluate the performance of RBPFs in SLAM environments, and experiment with solutions to a commonly accepted failure mode. Given their advantages over the EKF in several difficult localization problems and their alternative and descriptive representation of the posterior distribution, particle filters have the potential to become a powerful estimation technique. As evidenced by tests with Swiss Ranger data, they provide

similar performance to an EKF-based algorithm for pose tracking in a SLAM environment. Moreover, other tests revealed the strengths of FastSLAM in SLAM scenarios with poor initial estimates and high initial pose uncertainty and a partial environment map. The experimental results for alternative proposals and regularization techniques contained in this thesis do not provide a noteworthy case for FastSLAM 2.0, SpreadX, or any other algorithm as a definitive solution to sample impoverishment in particle filters. However, in light of the results from this limited survey of improvement strategies, it does seem reasonable to conclude that the recent experimental efforts aimed at solving the particle depletion problem are worthy causes that will hopefully, with more research, provide a highly advanced posterior estimation technique based on sequential Monte Carlo methods. Solving the sample impoverishment failure mode could greatly expand the number of solvable estimation scenarios and potentially yield a single robust filter with the architecture to enable autonomous vehicle operation in almost any unknown environment.

One of the surprising results from the simulation phase of this project was the fact that FastSLAM 2.0 was not, in some extreme cases, the ultimate answer to sample impoverishment. While the inclusion of feature observation information in proposal development does provide a marked increase in posterior accuracy at most measurement RMS levels, the most mismatched proposal-perceptual scenario revealed disturbing propagation effects that practically eliminated any posterior tracking ability. It appears as though FastSLAM 1.0, while not as advanced and not as accurate in all situations, provided the most robust proposal distribution. The particle diversity recovered by the addition of a simple regularization algorithm, such as SpreadX, can give this simple filter an increased sensitivity to feature observations and greater posterior accuracy with the most precise measurement device. Unfortunately, the SpreadX regularization approach used a fixed, empirically derived parameter and was therefore not as flexible to implement as other methods. A seemingly worthwhile endeavor could be to continue a regularization method research effort and develop other analytical solutions to match the accuracy of the empirical method. Perhaps an optimization formula using characteristics of the SLAM scenario could lead to a more

flexible regularization technique. Another consideration would be to include within the regularization kernel a parameter based on the residual between the estimated and observed feature measurements. In other words, spreading would be dependent upon how well measurements from the pose and landmark estimates match the actual observations, with less adjustment for particles that correctly predict the feature observation.

In order for FastSLAM to estimate effectively when measurement and motion noise are severely mismatched, there should be some way to incorporate more uncertainty in landmarks. One way, proposed by Montemerlo, is the incorporation of negative evidence to eliminate false landmarks; this was not used, but should be studied further in situations with low measurement noise. Also, landmarks should be initialized with more uncertainty than just what is represented in measurement noise. As was seen in several occasions, both in a simulated environment and with Swiss Ranger measurements, the Rao-Blackwellized particle filter, in a mapping environment, is prone to a false certainty in landmark position. This intensified data association errors through the creation of false landmarks. One way to incorporate this would be to include parameters that measure particle dispersion in the calculation of landmark covariance at initialization. Another way would be to include an ad hoc criteria for landmark initialization, namely that a particular landmark should be observed a defined number of times before it is incorporated into the filter. This feature was included in the EKF SLAM algorithm used with Swiss Ranger measurements in chapter 5.

Bibliography

[1] M. Sanjeev Arulampalam, Simon Maskell, Neil Gordon, Tim Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," in *IEEE Trans. on Signal Processing*, vol. 50, no. 2, February, 2002.

[2] T. Bailey, "Mobile robot localisation and mapping in extensive outdoor environments." PhD Thesis, University of Sydney, 2002.

[3] Robert G. Brown and Patrick Y. C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering.* 3rd ed. New York: Wiley, 1997.

[4] G. Castell and C. P. Robert, "Rao-Blackwellization of sampling schemes," *Biometrika*, vol. 83, 1996.

[5] A. J. Cooper, "A comparison of data association techniques for simultaneous localization and mapping." Master's Thesis, Massachusetts Institute of Technology, Cambridge, MA, 2005.

[6] John L. Crassidis, "An overview of particle filters with applications to aerospace systems." Technical presentation at C. S. Draper Laboratory, Cambridge, MA 2005.

[7] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering methods for practitioners," *IEEE Trans. Signal Processing*, vol. 50, March 2002.

[8] D. Crisan, "Particle filters - a theoretical perspective," in *Sequential Monte Carlo Methods in Practice.* A Doucet, N. de Freitas, and N. J. Gordon, eds. New York: Springer, 2001.

[9] F. Daum and J. Huang, "Curse of dimensionality and particle filters," in *Proc. IEEE Aerospace Conf.*, (Big Sky, MT), 2003.

[10] A. J. Davison, "Mobile robot navigation using active vision." PhD Thesis, University of Oxford, 1998.

[11] G. Dissanayake, S. Sukkarieh, E. Nebot, H. Durrant-Whyte, "The aiding of a low-cost strapdown inertial measurement unit using vehicle model constraints for land vehicle applications," in *IEEE Trans. on Robotics and Automation*, vol. 17, no. 5, October, 2001.

[12] Petar M. Djurić, "Sequential estimation of signals under model uncertainty," in *Sequential Monte Carlo Methods in Practice*, A Doucet, N. de Freitas, and N. J. Gordon, eds. New York: Springer, 2001.

[13] A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*. New York: Springer, 2001.

[14] A. Doucet, N. de Freitas, and N. Gordon, "An introduction to sequential Monte Carlo methods," in *Sequential Monte Carlo Methods in Practice*, New York: Springer, 2001.

[15] A. Doucet, N. de Freitas, K. Murphy, and S. Russell, "Rao-Blackwellized particle filtering for dynamic Bayesian networks," in *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, Stanford, 2000.

[16] Austin I. Eliazar and Ronald Parr, "Hierarchical linear/constant time SLAM using particle filters for dense maps." Department of Computer Science, Duke University, 2002.

[17] D. Fox, S. Thrun, W. Burgard, F. Dellaert, "Particle filters for mobile robot localization," in *Sequential Monte Carlo Methods in Practice*, A Doucet, N. de Freitas, and N. J. Gordon, eds. New York: Springer, 2001.

[18] Arthur Gelb, editor. *Applied Optimal Estimation.* Cambridge, MA: MIT Press, 1974.

[19] S. Godsill and T. Clapp, "Improvement strategies for Monte Carlo particle filters," in *Sequential Monte Carlo Methods in Practice.* A Doucet, N. de Freitas, and N. J. Gordon, eds. New York: Springer, 2001.

[20] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEEE Proc.-F*, vol. 140, no. 2, 1993.

[21] G. Grisetti, C. Ctachniss, W. Burgard, "Improving grid-based SLAM with Rao-blackwellized particle filters by adaptive proposals and selective resampling." Technical Paper. University of Freiburg, Germany, 2002.

[22] J. Guivant and E. Nebot, "Optimization of the simultaneous localization and map building algorithm for real time implimentation." in *IEEE Trans. on Robotics and Automation.* vol. 17, no. 3, 2001.

[23] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle filters for positioning, navigation and tracking." *IEEE Trans. Signal Processing.* vol. 50, February 2002.

[24] J. M. Hammersley and K. W. Morton, "Poor man's Monte Carlo," *Journal of the Royal Statistical Society B*, vol. 16, 1954.

[25] A. H. Jazwinski, *Stochastic Processes and Filtering Theory.* New York: Academic PRess, 1970.

[26] R. Kalman, "A new approach to linear filtering and prediction problems." *Trans. of the ASME-Journal of Basic Engineering*, vol. 82, series D, 1960.

[27] G. Kitagawa, "Monte Carlo filter and smoother for non-Gaussian non-linear state space models," *Journal of Computational and Graphical Statistics*, vol. 5, no. 1, 1996.

[28] J. S. Liu and R. Chen, "Sequential Monte Carlo methods for dynamical systems," *Journal of the American Statistical Association*, vol 93, 1998.

[29] W. Madow, "On the theory of systematic sampling, ii," *Annals of Mathematical Statistics*, 20, 1949.

[30] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in *Proc. of the Int. Conf. on Artificial Intelligence*, 2003.

[31] M. Montemero and S. Thrun, "Simultaneous localization and mapping with unknown data association using FastSLAM," in *IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, 2003.

[32] Michael Montemerlo, *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association*, PhD Thesis, Carnegie Mellon University, 2003.

[33] A. Moore, "An introductory tutorial on kd-trees," Technical Report N. 209, Cambridge University, 1991.

[34] K. Murphy, "Bayesian map learning in dynamic environments," in *Advances in Neural Information Processing Sytems*, Cambridge, MA: MIT Press, 1999.

[35] Christian Musso, Nadia Oudjane, Francois Le Gland, "Improving regularised particle filters," in *Sequential Monte Carlo Methods in Practice*, A Doucet, N. de Freitas, and N. J. Gordon, eds, New York: Springer, 2001.

[36] Michael K. Pitt, Neil Shephard, "Filtering via simulation: Auxiliary particle filters," *Journal of the American Statistical Association*, vol. 94, March 2001.

[37] Michaiel K. Pitt, Neil Shephard, "Auxiliary variable based particle filters," in *Sequential Monte Carlo Methods in Practice*, A Doucet, N. de Freitas, and N. J. Gordon, eds, New York: Springer, 2001.

[38] D. Reid. "An algorithm for tracking multiple targets." in *IEEE Trans. on Automatic Control.* vol. 24. no. 6. 1979.

[39] B. Ristic, S. Arulampalam, N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications.* Boston, MA: Artech House, 2004.

[40] D. B. Rubin, *Bayesian Statistics 3.* Oxford University Press, 1988.

[41] R. Sim, P. Elinas, M. Griffin, J. Little, "Vision-based SLAM using the Rao-blackwellized particle filter," in *Proc. of Reasoning with Uncertainty in Robotics,* Edinburgh, Scotland, 1995 .

[42] R. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty." Technical Report TR 4760 & 7239, SRI, 1985.

[43] C. Stachniss, G. Grisetti, W. Burgard, "Recovering particle diversity in a Rao-blackwellized particle filter for SLAM after actively closing loops." Technical Paper. University of Freiburg, Germany, 2005.

[44] "Swiss Ranger SR-300 Specifications," CSEM website, *http://www.swissranger.ch/specifications.php.*

[45] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo localization for mobile robots," in *Sequential Monte Carlo Methods in Practice.* A Doucet, N. de Freitas, and N. J. Gordon, eds. New York: Springer, 2001.

[46] S. Thrun. "Robotic mapping: A survey," in *Exploring Artificial Intelligence in the New Millenium.* Ed. G. Lakemeyer and B. Nebel. Morgan Kaufmasnn, 2002.

[47] S. Thrun, "Particle filters in robotics," *Proceedings of the 17th Annual Conference on Uncertainty in AI.* 2002.