Observable Autonomous SLAM in 2D Dynamic Environments

by

Daanish Khan, B. Eng.

A thesis submitted to the Faculty of Graduate and Postdoctoral Affairs

in partial fulfillment of the requirements for the degree of
Master of Applied Science in Electrical and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering (OCIECE)

Department of Systems and Computer Engineering

Carleton University

Ottawa, Ontario, Canada, K1S 5B6

September 2011

Your file Votre référence
ISBN: 978-0-494-83035-2
Our file Notre référence
ISBN: 978-0-494-83035-2

The undersigned recommend to
the Faculty of Graduate and Postdoctoral Affairs
acceptance of the thesis

Observable Autonomous Simultaneous Localization and Mapping (SLAM) in 2D Dynamic Environments

submitted by

Daanish Khan, B.Eng.

in partial fulfillment of the requirements for

the degree of Master of Applied Science in Electrical and Computer Engineering

_____

Chair, Howard Schwartz, Department of Systems and Computer Engineering

_____

Thesis Supervisor, Victor Aitken, Department of Systems and Computer Engineering

Carleton University
September 2011

Abstract

Simultaneous Localization and Mapping (SLAM) involves an unmanned vehicle system roaming through an unknown environment with the task of constructing a map of the environment and locating the vehicle within this map. The Autonomous SLAM problem has the objective of estimating the environment and vehicle pose relative to a world-centric reference frame when only vehicle-centric measurements are available and no prior information is available concerning the environment. The autonomous SLAM problem as formulated is not observable, and current methods to enforce observability require prior world-centric measurements for two stationary landmarks. This thesis proposes a locally observable solution to the autonomous SLAM problem without the need for prior measurements. Two stationary landmarks are needed in the environment to enforce three observability constraints that define the 2D world-centric reference frame. The thesis examines the implications on observability and accuracy when re-selection of these two landmarks is needed due to landmark motion or in order to roam through large environments. In dynamic environments landmarks may transition between stationary and moving. Two Extended Kalman Filters are operated in parallel, one for vehicle pose and stationary landmarks, and a second for moving landmarks. Hypothesis testing is used to determine when transitions between the filters are necessary. Simulation results demonstrate important properties of the proposed methods in representative environments.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Simultaneous Localization and Mapping

Simultaneous Localization and Mapping (SLAM) is something people do subconsciously each day without noticing. It is, however, a difficult task in robotics and represents a major hurdle in creating mobile robots that are truly autonomous [13]. At its core, SLAM is simply the process of answering the two questions "what's around me?" and "where am I?", but the answer to these questions gives rise to many difficulties. Historically in autonomous robotics, localization and mapping were done separately and were heavily relied upon in tasks such as industrial automation [45]. Robots often operated in structured environments with a priori maps, and a 'hard-engineering' approach was taken to develop more accurate sensors and robots [45]. With advances in robotic technologies and the desire to use robots in increasingly unstructured environments with cheaper and more readily available sensors, the need for SLAM has become prominent.

Robots typically collect information from their sensors and make control input decisions based upon information given by these sensors and an internal control strategy. Given this framework for a robot, consider what happens as this robot is placed in an unknown environment with no a priori map. To solve the SLAM problem [13], the robot must incrementally build a model of the environment, based on sensor measurements, and also place itself where it believes its position is in the

environment.

	Known Map	Unknown Map
Known Position	**No Problem**	**Mapping Problem**
Unknown Position	**Localization Problem**	**SLAM Problem**

Figure 1.1: How SLAM relates to Mapping and Localization

As shown in Figure 1.1, SLAM consists of 2 parts: localization and mapping. The mapping component of SLAM is the process of using on-board sensors to deduce information about the state of the environment that surrounds the robot. Historically, in the field of geography and cartography, mapping was done by hand using precise measuring instruments, but recently radar and photographic mapping techniques have become prevalent. A popular way of mapping is by converting an environment into a fine grid. For example, in a geographical context, this grid is known as longitude and latitude. In probabilistic robotics, this grid is often local to the robot (not an absolute reference frame) and is known as an occupancy grid.

Localization is the process of placing an object within an environmental map. Traditionally, mobile robots have relied on the use of an a priori map that is given to the robot prior to entering the area. A robot can use this map and its sensors to estimate where it is within the environment. In industrial settings, it was possible to structure environments (by paving the ground or building walls) and, further, it was possible to create a highly accurate map of a shop floor. For this reason, for autonomous robots, only the localization problem was relevant. To localize, robots could be given a path to follow using a buried wire with an induction coil on the robot, or a path could be made with beams of light which the robot could detect and localize itself. This system is highly economical as it requires simple sensors and simple processing units on board the robot. A popular method for localization is called Grid Localization. The current standard for localization is a technique known as Monte Carlo Localization (MCL). This algorithm is capable of localizing a robot given a static known environment (only the robot is in motion) and has been shown

to be able to localize with very few samples [49] [48] [60].

Concurrent Mapping and Localization (CML) was what SLAM was called during the early 1990s [50]. At that time research shifted to solving both problems, localization and mapping, simultaneously and autonomously. In the autonomous SLAM problem formulated in the literature, measurements of the environment are only available to the robot from vehicle-mounted sensors. For example, a laser or other vision system mounted on the robot can provide range and bearing measurements of environmental landmarks relative to the vehicle. Speed or intertial navigation sensors may also be available on the robot. However, absolute robot pose measurements relative to a pre-defined world reference system (such as might be provided by GPS) and absolute world-centric measurements of landmark positions are not included in the autonomous SLAM problem. All measurements are therefore vehicle-centric. The objective of autonomous SLAM is then to construct a world-centric map of the environment and to maintain an estimate of vehicle pose and motion within this world-centric map.

The SLAM problem is generally posed as a state estimation problem for dynamic systems. When vehicle-relative position measurements for each landmark are available, the Data Association problem must first be solved to assign each measured landmark to its corresponding landmark information in the environmental state retained by the robot. In dynamic environments, landmarks may be stationary or moving, or may transition from stationary to moving or from moving to stationary. SLAM for dynamic environments must therefore include capabilities in motion detection and tracking of moving landmarks relative to the stationary landmarks that define the environmental map. The robot pose and motion state and the environmental map state together form the complete state for the dynamic system that is updated with each measurement set received.

A dynamic system is said to be "observable" if it is possible to estimate the complete state of the system from available measurements over time. Concepts of observability take on different forms depending on whether the system is linear, nonlinear, and/or stochastic. The autonomous SLAM problem as posed is not completely observable. As a result, bounded state estimation errors cannot be guar-

anteed. This is a serious issue that has largely been neglected in the literature. The following subsections review the current literature in these methods and issues of the SLAM problem.

1.2 SLAM Literature Review

Many different types of solutions to the SLAM problem have been proposed, each with their own goals (speed, scalability, statistical distributions [53] and more). The Kalman Filter [15], or the Information form of the Kalman Filter [16] [43], are state estimation algorithms that can be performed recursively online while the vehicle is in motion. Other probabilistic methods, such as Particle Filtering and Expectation Maximization, can sometimes be performed online but are generally more computationally intensive compared to the Kalman approach. There also exist hybrid methods that apply different techniques to the localization and mapping components of the SLAM problem. These include the Unscented Kalman Filter and FastSLAM methods, each of which have underlying components from both of the Kalman Filtering and Particle filtering based algorithms.

1.2.1 Kalman Filtering Based Methods

Kalman Filter-Based methods for SLAM generally involve increasing the size of the state vector to include the state of the vehicle and the state of the environment (in this case, positions of the landmarks). Kalman Filtering methods all stem from the Linear Kalman Filter (LKF) [40] [27] [58]. For linear Gaussian systems, the LKF is an optimal unbiased minimum variance estimator [30] [19]. Mobile robots and the environmental measurements provided by on-board sensors are typically non-linear systems and may only approximately satisfy Gaussian assumptions. Introducing non-linearities eliminates the optimality of the LKF. The most popular way of performing SLAM with the Kalman approach is with the Extended Kalman Filter (EKF) that gives different qualities of estimates under different circumstances. The EKF linearizes the non-linear model about the current state estimate thereby giving a linear system for use in a LKF. In the SLAM literature, the EKF is considered to

be the reference to which any new algorithm is compared.

The EKF was first proposed as a solution for SLAM in [46] where the general theory for applying the EKF to a system with point features for landmarks was developed. In [46] the SLAM problem was analyzed in terms of deducing spatial relations between landmarks and the robot. The ideas presented in [46] were implemented and tested on real robots in [14] [12]. The computational complexity of the EKF is an issue when the size of a map becomes large. In [14], the issue of time-complexity of large-scale EKF-SLAM is explained. The realization is made that EKF-SLAM as traditionally described cannot accommodate environments with many features. To address this issue, in [10], the Bounded Region filter is created that modifies the EKF-SLAM framework to approximate a state estimate and covariance with two boundary values in which the true state must lie. In this way, the computational complexity of EKF-SLAM can be reduced. An alternative to the bounded region filter is proposed in [39] where the landmark locations are estimated relative to the position and orientation of the robot. This transformation method allows the landmark estimates to become independent of each other thereby reducing the computational complexity of the problem. As noted in [14], however, both of these methods will result in filters that take longer to arrive at a consistent result or, in the worst case, create a divergent solution.

There exist many variants of the EKF each of which incorporate different optimizations such as the Robust Extended Kalman Filter (REKF) [26], the Iterated Extended Kalman Filter (IEKF) [7] and the Compressed Extended Kalman Filter (CEKF) [54].

The dual of the Kalman filter is known as the Information Filter [36]. The information filter in theory provides the same state estimate as the LKF, but stores the information (state and covariance) in a different way that may improve performance in some cases. The way that the information is stored makes the correction phase of the filter simpler at the expense of added complexity during the prediction phase. This can be advantageous in multi-robot situations where information is added from each robot into the filter.

Since the EIF is the dual of the EKF, the use of EIF-SLAM is not explored inde-

pendently of EKF-SLAM in the literature. A key difference, however, between the EKF and the EIF is that the EIF maintains the inverse of the EKF Covariance Matrix. This inverse, known as the information matrix, is generally almost sparse, and exploiting the sparseness of this matrix to achieve greater computational efficiency (as compared to the EKF) is the primary focus of the literature [52] [43] [16] [17]. Through sparsification techniques developed by Thrun [52] [53] [49], it is possible to enforce sparsity in the information matrix thereby reducing computational complexity of the filter. This is known as the Sparse Extended Information Filter (SEIF). The SEIF exploits sparsity of the information matrix and uses an alternate formulation for the EIF equations that execute in constant time at the expense of accuracy of the filter. The contribution of [52] is that the amount of information lost by making approximations for constant time SLAM is minimal. Another approach to the sparsification procedure for the EIF is known as the Exactly Sparse Extended Information Filter (ESEIF) [55]. This algorithm operates by removing the robot from the EIF filter at each timestep (referred to as 'kidnapping') and creating a completely new information matrix using that timestep's measurements. In [17], a practical implementation of the EIF-SLAM algorithm with sparsification was implemented and used for exploration and mapping of the RMS Titanic on the bottom of the ocean. This work was carried out using an autonomous robot that relied on the SEIF SLAM algorithm, successfully demonstrating the algorithm in practice.

1.2.2 Particle Filtering and Expectation Maximization Methods

In contrast to Kalman based methods, particle filtering and expectation maximization methods have been researched to solve SLAM. Expectation Maximization (EM) consists of two steps, an expectation step (E-Step) and a maximization step (M-Step) [49], which, when combined, create a new map with the idea that each map will be more accurate than the previous. The goal of the expectation maximization algorithm is to compute the Maximum Likelihood (ML) estimate of a parameter. Unlike the Kalman filtering based methods, the EM algorithm is not an online algorithm in that it must be solved with all the data gathered by the robot over its

entire motion [49] [60].

Particle Filtering is similar to Monte Carlo Methods for performing SLAM. Similar to the Kalman Filter, the Particle Filtering (PF) methods use the Prediction/Correction approach. To perform the particle filter, a set of 'particles' are created. Each of these particles represent an independent belief about the state of the SLAM system. Each of these particles are updated according to the process and measurement models provided (as with the Kalman Fitler). A probabilistic weighting is created that weights each particle according to the strength of the belief contained within each particle. Many different strategies can be employed to resample the particles to choose the most likely outcome and deduce the weighting of each particle. In SLAM, a popular resampling algorithm is the *select with replacement* algorithm which probabilistically selects a particle from the particle set, but replaces that particle once a selection has been made [32] [42].

In [42], different resampling methodologies are explored including 'near-zero weight', 'select with replacement', and 'effective sample size (ESS)'. Select with replacement will choose a subset of particles selected by repeated random picks where each particle is picked with probability equal to its weighting. The weighting in this case is an estimate of how well the belief contained within the particle matches the true state of the system. The near-zero weight approach to resampling will pass the previously mentioned weighting through a function that will either make several copies of the particle or resample the particle with a given probability. The draw back of this approach, as noted in [42], is that the number of particles does not remain fixed. The effective sample size (ESS) approach uses a formula to determine how many particles in a given set of particles contain beliefs about the state of the system that are relatively good. Once a determination is made that an unacceptably high number of particles are diverging from the true belief, another resampling strategy can be employed to resample the particles. The ESS strategy is beneficial as it limits the number of times the particles need to be resampled.

1.2.3 Hybrid Methods

Hybrid methods of solving SLAM combine aspects of particle filtering with Kalman methods that are capable of being solved online. The Unscented Kalman Filter (UKF) and FastSLAM are two examples of such methods.

The Unscented Kalman Filter

The Unscented Kalman Filter (UKF) [20] is the application of the Unscented Transform (UT) [51] [20] [25] to the linear Kalman Filter such that it can accommodate non-linear functions. The idea of the UT is to transform a random variable through a non-linear function rather than linearize a non-linear function about a certain point. The UKF aims to address the following shortcomings of the Extended Kalman Filter: the linearization errors can produce instability in the EKF, and the EKF requires the calculation of a Jacobian which may be non-trivial for some systems [25]. The UKF addresses the issue of linearization errors by not linearizing the system, and directly uses the non-linear functions for the dynamic and measurement models. The UKF then uses Sigma Points which are carefully chosen near the current sample mean to propagate through the non-linear models. The results for each of these points is then weighted according to a selected probability distribution to find the most likely next point. It is important to note that the non-linear functions could be of any form, they could even be a piece of computer code that is run with specific initial conditions. This note is useful in practical systems as the Jacobian for a complex piece of computer code or a large robot may be difficult to compute whereas the non-linear model or code is often available and may be evaluated quickly. The application of the unscented transform was explored in [5] where it is shown that the UKF better represents the confidence in a measurement and that the EKF may be overconfident in the covariance estimate for SLAM. In [23], the standard EKF formulation for SLAM is converted to a UKF formulation. This paper also concludes that the computational complexity of UKF-SLAM is the same as for EKF-SLAM.

FastSLAM

The problem with Kalman filtering based approaches to SLAM (including the UKF) is that they do not scale well [21]. Particle Filtering methods also do not scale well and, indeed, many particle filtering methods are not able to be implemented in real time. FastSLAM proposes a particle filtering based SLAM algorithm that is able to be performed online while the robot is in motion. FastSLAM decomposes the SLAM state estimation problem into a robot localization problem and many landmark location estimation problems [53] [33] [32]. The FastSLAM algorithm uses the particle filter to estimate the robot pose and then uses as many EKFs (or other Kalman-Based Filters) as there are landmarks to estimate the state of the environment. This decomposition is referred to as a Rao-Blackwellized particle filter [53].

In [53], the FastSLAM algorithm is introduced as a hybrid method combining the ability to model non-Gaussian distributions of a particle filter with the computational efficiency of the EKF based solutions. By exploiting the conditional independence of the SLAM problem, the Rao-Blackwellization method is used to decouple robot position estimation from landmark position estimation. In [28], the issue of multiple robots is considered, and strategies for optimal exploration patterns and combining multiple maps is explored in the context of FastSLAM.

1.2.4 Data Association

The data association problem involves determination of which sensor readings match which landmarks in the environment and making sure that these determinations or 'associations' stay correct across timesteps. In dynamic environments, this process may not be trivial as landmarks may move, landmarks may become occluded by other landmarks, and the error in sensor measurements may falsely indicate a reading as belonging to a different landmark. The literature is rich with research on the data association problem. The method of Nearest Neighbour is popular but there exist other methods such as the Joint Conditional Branch and Bound (JCBB) [37] and the Joint Probabilistic Data Association Filter (JPDAF) methods.

The simplest way of performing a nearest neighbour data association is to use the Euclidean distance between the internal position estimate of the landmark and the position measured by the sensor. By computing this distance for all combinations of landmark-measurement pairs, the smallest distance would often be the correct association. This approach, however, does not take into account the stochastic nature of the system. To incorporate the statistical information that is also estimated about each landmark, it is useful to use what is known as the *Mahalanobis Distance*. The Mahalanobis distance works well for environments that are sparsely populated with landmarks or for robots with highly accurate sensors [37] but this may not always be the case. The reason for this shortcoming is that the Mahalanobis distance only checks the statistical distance between a single landmark and a single observation. This is known as *individual compatibility*. In [37] it is shown that the errors in the measurements and therefore the Mahalanobis distances between landmarks are correlated and therefore a method of checking the entire map for statistical matches between observations and estimated landmark positions is preferred.

Another way of computing the data association for SLAM is by using the Joint Compatibility Branch and Bound (JCBB) solution [37]. This method addresses the individual compatibility problem inherent in Mahalanobis distance calculations. The JCBB algorithm performs a joint compatibility test to find the most likely entire map for the system. The process of discovering all the possible sensor measurement to landmark associations is an exponentially large problem so the JCBB algorithm makes use of the Branch-and-Bound technique used in Integer Programming to build a tree structure of possible map layouts, and then a bounding function to prune branches that are statistically unlikely, progressively getting to the likeliest solution.

Another popular way of performing Data Association is through the use of the Joint Probabilistic Data Association Filter (JPDAF) first proposed by Bar-Shalom [7]. This method is capable of tracking multiple moving objects by computing a Bayesian estimate of the correspondence between the features detected by sensor data. Individual Kalman filters are used to track each individual object (or landmark in the environment). In this way, since the objects are being tracked using

a predictor-corrector filter, their states can be predicted even if an object becomes occluded. Closely related to this method is the idea of Multiple Hypothesis Tracking (MHT) [7]. This method maintains large trees with each node being a Kalman Filter that independently keeps an estimate of a landmark's location along with a probability that that estimate is correct. The tree structure is then parsed to find the most likely data association. There have also been some efforts to apply particle filtering to JPDAF and MHT to support non-Gaussian distributions [11].

1.2.5 Motion Detection/Tracking for SLAM

Motion detection and tracking became an important problem when radar tracking for air vehicles was heavily computerized [41]. The need to identify and track moving targets was of interest for both military and commercial applications in aerospace and many different models and approaches were used. In SLAM, the existing frameworks commonly used, such as the EKF or FastSLAM, generally accommodate only static environments and any inconsistency in the static environment surrounding the robot will reduce the quality of the map and localization results. Motion tracking for dynamic environments with moving objects must be considered to make SLAM useful for implementation in everyday environments.

Motion detection is generally performed in SLAM by checking the map estimate to determine if the state of the map has changed. This can be accomplished by determining if a previously observed landmark has moved significantly from its estimated location [56] [31] or, alternatively, all free space in the robot's environment can be monitored to determine if any previously free space is occupied, possibly by a new landmark [56].

Some research on motion detection relies heavily on the previous research on data association. The JPDAF framework has been used extensively in multiple moving object detection as it provides a computationally efficient way of determining the most likely configuration of the moving objects in the robot's environment [59]. In [59], however, it is shown that for cluttered environments, the JPDAF algorithm may give degraded results.

Motion Tracking can be performed in many ways, but generally this problem can

be approached using the standard EKF formulation for landmark motion prediction. The issue with this type of tracking is that neither the process nor input models for the unknown landmark to be tracked are available. In many radar systems, target tracking is performed using an assumed model for generic motion of unknown objects [41]. Some work has been made on tracking humans using SLAM, however, due to the ability for humans to move in an extremely unpredictable fashion, the current state of the art relies on matching patterns of objects or landmarks within the robot's environment without actually predicting the state of the human [34]. Tracking highly manoeuvrable targets [59] may be difficult to perform in real-time. As a result, many target tracking algorithms found in literature are performed offline [59] [31] [41] [7].

In [56], two methods for performing SLAM in dynamic environments, the method of SLAM with Generic Objects (GO), and the method of SLAM with Detection and Tracking of Moving Objects (DATMO), are compared. SLAM with GO is shown to be computationally infeasible for large scale SLAM implementations whereas the SLAM with Moving Objects algorithm is shown to be feasible for implementation in SLAM with many features. SLAM with Moving Objects has been implemented on a real vehicle and demonstrated in a real-world environment [57]. The DATMO algorithm has been shown to be capable of detecting and tracking unknown moving objects (in this case people and cars).

1.2.6 Observability in SLAM

A dynamic system is said to be "observable" if it is possible to estimate the complete state of the system from available measurements over time. Observability for stochastic systems generally implies that enough information about the system is available from the measurements to estimate the complete system state over time with bounded estimation errors. For example, a system must be observable to use the Kalman Filter for state estimation, otherwise the Kalman covariance matrix will never be reduced relative to initial estimates and in many cases will grow without bound. The autonomous SLAM problem, in which all measurements are vehicle-centric, is posed with the objective of estimating the vehicle pose and motion state

as well as the environmental state relative to a world-centric reference frame. A common approach is to assume that the vehicle reference frame is aligned with the world-centric frame at the initial time. Proposed solutions for SLAM then proceed with various forms of state estimation algorithms designed for the complete autonomous SLAM problem that is not completely observable. This means that the filter cannot accumulate information about the state of the system continuously and can never achieve a reliable solution.

The issue of observability of the SLAM problem has received very little attention in the literature. In [6], Cetto and Sanfeliu investigate the observability of a linear form of the SLAM problem in one and two dimensions. This analysis shows that the autonomous SLAM problem as posed is not completely observable. They propose, for the one-dimensional (monobot) case, that the problem can be made observable provided the world-centric coordinate of one landmark is known a priori. The position of this landmark is not estimated by the filter, but the sensor measurements for this landmark are used. Cetto and Sanfeliu [6] also consider the two-dimensional planar SLAM problem and conclude that an observable system can be created provided the world-centric coordinate of one landmark is known a priori, however, this conclusion for the two-dimensional case is not correct as pointed out by Lee, Wijesoma and Guzman [29].

Lee, Wijesoma and Guzman [29] provide a detailed analysis of observability for the non-linear continuous-time two-dimensional autonomous SLAM problem using differential geometric methods and the Lie Algebra. They also conclude that the autonomous SLAM problem is not observable. Furthermore, they show that a priori knowledge of the position of a single landmark in the two-dimensional case also results in an unobservable system. However, with further analysis they prove that the two-dimensional problem can be made observable if the world-centric positions of two landmarks are known a priori. This approach does not permit a solution of the autonomous SLAM problem because two world-centric coordinates for each of these "special" landmarks, and hence the distance between them, must be available in advance for the state estimation algorithm, and these two special landmarks must always be visible to the robot. One approach [1] for practical implementation

when this a priori information is not available is to decouple two landmarks from the state estimation algorithms once sufficient accuracy has been obtained, and thereafter treat these landmark coordinates as known. However, using this approach, the SLAM problem begins in an unobservable configuration and hence estimation accuracy for these two landmarks is not guaranteed to improve relative to initial estimates.

Apart from fixing landmarks, there exist other methods that attempt to correct for the unobservability of autonomous SLAM. In the context of EKF-SLAM, there is a technique called Loop Closure. This technique involves the case where the robot revisits a certain location. This method is explored in [38] and it is discovered that when the vehicle revisits a location, the new estimate is not only capable of improving the estimate for the revisited landmark but also for reducing the uncertainty in all previously observed landmarks.

1.3 Thesis Statement

SLAM is a very important problem in mobile robotics as a solution would allow for truly autonomous systems. There have been many proposed solutions for SLAM that are all subject to advantages and disadvantages. This thesis focusses on the two-dimensional planar autonomous SLAM problem for dynamic environments and a solution based on the Extended Kalman Filter. All measurements are vehicle-centric, and no a priori environmental information is available to the robot. Landmark position measurements relative to the vehicle are provided in the form of range and bearing as could be made available by laser or vision systems mounted on the robot. The Data Association problem is not investigated in this thesis. The vehicle pose and motion state and the environmental state are to be estimated relative to a world-centric reference frame that is not specified in advance, but must be defined by the estimation algorithm. There are several problems to be overcome in order to achieve these goals:

1. The autonomous SLAM problem must be formulated to be completely observable;

2. Estimation accuracy for both the vehicle and environmental state must be maintained while the vehicle may roam through large dynamic environments;

3. The solution must be capable of distinguishing between moving and stationary landmarks, and must have the capability to detect a transition from moving to stationary, or stationary to moving;

4. Moving landmarks must be tracked in position and motion relative to the defined world-centric reference system;

5. Observability must be maintained in the dynamic environment.

Contributions of the thesis work are as follows.

1. An observable formulation of the autonomous SLAM problem is defined. The proposed method is related to the recommendations of Lee, Wijesoma and Guzman [29], but does not rely on a priori information or external sensors. The proposed approach is based on original methods developed in Aitken's Ph.D. dissertation [2], also summarized in [4], which considered the problem of estimating motion and structure of objects in three-dimensional space based on feature position measurements from multiple-camera image sequences. In that work, Aitken encountered a similar requirement to formulate an observable problem and to completely define a reference frame based on visible feature points. For the two-dimensional SLAM problem, this approach requires the selection of two "special" stationary landmarks in the environment that are treated differently from the remaining landmarks. The world-centric frame is then defined as a landmark-centric frame based on these two special landmarks.

2. An analysis of state estimation accuracy is conducted for representative SLAM problems. The analysis is based on Fisher's Information theory and the Cramer-Rao Lower Bounds (CRLB) [47] for the covariance of state estimates as developed for discrete-time nonlinear systems in Appendix C of [2]. In particular it is demonstrated that the estimation accuracy degrades with increasing distance of the vehicle from the special landmarks of Item 1 above due to sensor uncertainty.

3. A re-selection strategy is developed for the special landmarks of Item 1 to provide an autonomous mothod to maintain observability in the system. Re-selection of special landmarks, however, has implicatons on observability and accuracy. Two special stationary landmarks are retained at all times, but the system can autonomously re-select these special landmarks in order to roam through a large environment. The proposed re-selection method does not rely on a priori information or external sensors, but does require a re-organizaton of the state vector and covariance matrix.

4. Methods are developed to distinguish between moving and stationary landmarks, and to detect a transition from moving to stationary, or stationary to moving. Two state estimation algorithms are run in parallel: one for the vehicle pose and positions of stationary landmarks, and a second for position and motion of moving landmarks. Moving landmarks are modeled with Singer's approach [44] [2]. Methods adopted from target tracking and data association are integrated to detect transitions between stationary or moving. Such transitions require re-organizaton of the state vectors and covariance matrices of the two filters.

5. Simulation results are given for typical vehicle and sensor models in dynamic environments to demonstrate performance of the proposed methods.

1.4 Thesis Overview

This chapter introduced the concept of SLAM and some of the challenges associated with solving the SLAM problem. A brief history of SLAM was given and some of the classical methods along with some contemporary methods were reviewed. The remainder of this thesis is organized as follows:

- **Chapter 2** provides theoretical background information needed for remaining Chapters of the thesis. Mathematical notation is introduced to clearly define the autonomous SLAM problem. A review of the Kalman and Extended Kalman filter is given, concepts of observability for linear and non-linear

discrete-time systems are reviewed, Fisher's Information and the Cramer-Rao lower bounds are reviewed, and a framework for landmark motion detection based on hypothesis testing is developed. Model parameters are also given for simulation studies conducted in Chapters 3 and 4.

- **Chapter 3** focusses first on the observability problem for autonomous SLAM. Previously proposed methods to implement SLAM algorithms for the unobservable problem are examined using the Cramer Rao Bounds and simulation of the Extended Kalman filter for representative geometries. An observable formulation of the autonomous SLAM problem is proposed and analysed in simulation. Accuracy of the proposed method is investigated for large environments. The required steps for addition or deletion of landmarks for the two state estimation filters is given. Finally, issues of observability and accuracy are investigated when re-selection of special landmarks may be needed.

- **Chapter 4** provides exploratory simulation results to demonstrate expected performance of the proposed methods. Static environments are treated initially without special landmark reselection. Bias introduced into the filters through re-selection of special landmarks is investigated. Final simulations demonstrate the feasibility of operating the two state estimation algorithms in parallel with the ability to detect motion in dynamic environments and re-structure the filters in response to the changing environment.

- **Chapter 5** provides conclusions of the thesis work and lists suggestions for future research.

Chapter 2

Theoretical Background

A fully autonomous robot is one that must be able to perform SLAM. The ability to solve SLAM online in real-time will allow robots to function for longer periods of time without human control. The SLAM problem is composed of multiple sub-problems that must be individually considered before a complete solution can be formulated. This Chapter provides background information needed for further development of the proposed solution and execution of the simulation studies to evaluate performance. Notation is introduced in Section 2.1 for the process and measurement models of the autonomous SLAM problem commonly treated in the literature. A review of the Kalman and Extended Kalman Filters is given in Section 2.2, concepts of observability are reviewed in Section 2.3, and the Cramer-Rao Lower Bounds are introduced in Section 2.4. Hypothesis testing within the Kalman Filtering framework is reviewed in Section 2.5 to prepare for SLAM in dynamic environments where an ability to detect transitions between moving and stationary landmarks is required.

2.1 SLAM Problem Definition

The SLAM problem involves estimating the state of a robot in an environment as well as the map of the environment. This problem is illustrated in Figure 2.1 for the two-dimensional planar problem considered in this thesis. A robot is placed

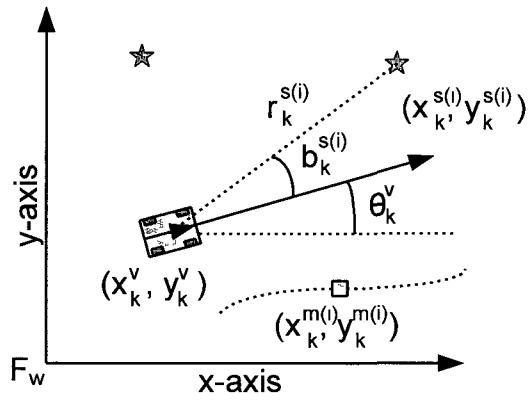

Figure 2.1: An overview of a generic SLAM problem.

in an unknown environment and is equipped with sensor systems that can provide measurements of landmark positions relative to the vehicle. The scope of this thesis does not include capabilities to detect landmarks in the raw sensor data, nor does it include the data association problem. Instead, it is assumed that vehicle-relative landmark positions are available for the SLAM state estimation algorithm at each timestep. In the autonomous SLAM problem, the objective is to employ only vehicle-centric measurements of landmark positions in order to estimate both the vehicle pose and motion and the landmark positions relative to a defined world-centric reference frame F_w. Landmarks may be stationary or moving and may transition between moving and stationary. For moving landmarks, the state estimation algorithm must estimate their position and motion relative to F_w. This thesis employs the Extended Kalman Filter (EKF) to solve the state estimation problem. It is first necessary, therefore, to develop models for vehicle dynamics, landmark measurements, and landmark motion.

2.1.1 Vehicle Process Model

Unicycle Model

In the SLAM literature, a simple unicycle model is often used to represent vehicle motion in order to test new algorithms and demonstrate aspects of SLAM where the vehicle model is not critical. The unicycle model includes non-linear kinematics for vehicle motion, but does not model vehicle inertia nor complete dynamics. The unicycle model has a single constraint in that the vehicle translational velocity is directed along the vehicle heading (in the direction the vehicle is currently facing). Rotational motion is not constrained. This model is representative of a skid-steered vehicle in which individual control is available for each of the left and right wheel speeds.

The pose of the vehicle is represented as

$$\mathbf{p}_t^v = \begin{bmatrix} x_t^v \\ y_t^v \\ \theta_t^v \end{bmatrix}, \tag{2.1}$$

where (x_t^v, y_t^v) represents the continuous-time (subscript t) Cartesian coordinates of the vehicle position relative to the world-centric frame F_w, and θ_t^v represents the vehicle heading relative to the x-axis of F_w as illustrated in Figure 2.1. The continuous-time vehicle state vector for the unicycle model is defined as

$$\mathbf{x}_t^v = \mathbf{p}_t^v. \tag{2.2}$$

The simplified motion model for the unicycle is given as [9] [35] [24] [22] [8]

$$\dot{\mathbf{x}}_t^v = \begin{bmatrix} \dot{x}_t^v \\ \dot{y}_t^v \\ \dot{\theta}_t^v \end{bmatrix} = \begin{bmatrix} \cos\theta_t^v & 0 \\ \sin\theta_t^v & 0 \\ 0 & 1 \end{bmatrix} \mathbf{u}_t, \tag{2.3}$$

with the known system input being

$$\mathbf{u}_t = \begin{bmatrix} v_t^d \\ \omega_t^d \end{bmatrix}, \tag{2.4}$$

where v_t^d is the desired forward speed and ω_t^d is the desired angular velocity.

The continuous-time model in (2.3) is commonly cast into discrete-time form using Euler's discretization procedure [18] assuming the sample period is sufficiently small. Euler's discretization is written as

$$\dot{\mathbf{x}}(t) \approx \frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{T}, \tag{2.5}$$

where $t = kT$, T is the sample period, and k is the current timestep. Using this approach, the discrete-time model for the unicycle is

$$\mathbf{x}_{k+1}^v = \mathbf{f}(\mathbf{x}_k^v, \mathbf{u}_k) + \mathbf{q}_k^v, \tag{2.6}$$

where $\mathbf{q}_k^v$ represents noise and modelling errors, and

$$\mathbf{f}(\mathbf{x}_k^v, \mathbf{u_k}) = \mathbf{I}_3\mathbf{x}_k^v + T \begin{bmatrix} \cos\theta_k^v & 0 \\ \sin\theta_k^v & 0 \\ 0 & 1 \end{bmatrix} \mathbf{u}_k \tag{2.7}$$

where $\mathbf{I}_3$ is the 3x3 identity matrix. The discrete-time unicycle model is intended for use in the Extended Kalman filter where it is assumed that $\mathbf{q}_k^v$ is zero-mean Gaussian with covariance matrix

$$\mathbf{Q}^v = \begin{bmatrix} {\sigma_{x_k^v}}^2 & 0 & 0 \\ 0 & {\sigma_{y_k^v}}^2 & 0 \\ 0 & 0 & {\sigma_{\theta_k^v}}^2 \end{bmatrix}. \tag{2.8}$$

The Jacobian matrix

$$\mathbf{F}_k^v = \frac{\partial f(\mathbf{x}_k^v, \mathbf{u}_k)}{\partial \mathbf{x}_k^v} = \begin{bmatrix} 1 & 0 & -Tv_k^d \sin\theta_k^v \\ 0 & 1 & Tv_k^d \cos\theta_k^v \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.9}$$

is also required in the EKF for time propagation of the Kalman covariance matrix.

Extended Unicycle Model

The unicycle model represents simple non-linear vehicle motion, but does not represent a real vehicle with inertia. This Section extends the simple unicycle state vector to include vehicle speed, v_t^v, and turning rate, ω_t^v, in order to more accurately represent an unmanned vehicle. In this way, estimation of vehicle speed and turning rate can be included in the SLAM problem. In applications, measurements of vehicle speed and turning rate may be available from on-board sensors and can easily be incorporated into the measurement models of the filters. Such measurements are not strictly required because if the vehicle pose is observable, then v_t^v and ω_t^v are also observable under the following models. Simulation studies in Chapters 3 and 4 use the simple unicycle model for simulations. The extended unicycle model is included in the formulation of the problem in order to facilitate future research.

The state vector for the extended model is expanded to include the vehicle pose $\mathbf{p}_t^v$ as well as vehicle speed v_t^v and turning rate ω_t^v,

$$\mathbf{x}_t^v = \begin{bmatrix} \mathbf{p}_t^v \\ v_t^v \\ \omega_t^v \end{bmatrix}. \tag{2.10}$$

The non-linear continuous-time process model then becomes

$$\dot{\mathbf{x}}_t^v = \begin{bmatrix} \dot{x}_t^v \\ \dot{y}_t^v \\ \dot{\theta}_t^v \\ \dot{v}_t^v \\ \dot{\omega}_t^v \end{bmatrix} = \begin{bmatrix} v_t^v \cos\theta_t^v \\ v_t^v \sin\theta_t^v \\ \omega_t^v \\ -\alpha_v v_t^v + b_v u_t^v \\ -\alpha_\omega \omega_t^v + b_\omega u_t^\omega \end{bmatrix}, \tag{2.11}$$

where $\alpha_v > 0$ and $\alpha_\omega > 0$, and these as well as b_v and b_ω are modelling parameters. The system input is

$$\mathbf{u}_t = \begin{bmatrix} u_t^v \\ u_t^\omega \end{bmatrix}, \tag{2.12}$$

and a simple control law can be defined as

$$\mathbf{u}_t = \begin{bmatrix} \hat{v}_t^v - v_t^d \\ \hat{\theta}_t^v - \theta_t^d \end{bmatrix}, \tag{2.13}$$

where $\hat{v}_t^v$ and $\hat{\theta}_t^v$ are the current speed and heading estimates, and v_t^d and θ_v^d are the desired speed and heading, respectively.

Using Euler's discretization procedure the discrete-time model is

$$\mathbf{x}_{k+1}^v = \mathbf{f}(\mathbf{x}_k^v, \mathbf{u}_k) + \mathbf{q}_k^v, \tag{2.14}$$

where $\mathbf{q}_k^v$ represents noise and modelling errors, and

$$\mathbf{f}(\mathbf{x}_k^v, \mathbf{u_k}) = \begin{bmatrix} 1 & 0 & 0 & T\cos\theta_k^v & 0 \\ 0 & 1 & 0 & T\sin\theta_k^v & 0 \\ 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 1 - T\alpha_v & 0 \\ 0 & 0 & 0 & 0 & 1 - T\alpha_\omega \end{bmatrix} \mathbf{x}_k^v + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ Tb_v & 0 \\ 0 & Tb_\omega \end{bmatrix} \mathbf{u}_k. \tag{2.15}$$

For implementation of the EKF, the noise $\mathbf{q}_k^v$ is assumed to be zero-mean Gaus-

sian with covariance matrix

$$\mathbf{Q}^v = \begin{bmatrix} {\sigma_{x_k^v}}^2 & 0 & 0 & 0 & 0 \\ 0 & {\sigma_{y_k^v}}^2 & 0 & 0 & 0 \\ 0 & 0 & {\sigma_{\theta_k^v}}^2 & 0 & 0 \\ 0 & 0 & 0 & {\sigma_{v_k^v}}^2 & 0 \\ 0 & 0 & 0 & 0 & {\sigma_{\omega_k^v}}^2 \end{bmatrix}. \tag{2.16}$$

The Jacobian of the process model needed for time propagation of the Kalman covariance is

$$\mathbf{F}_k^v = \frac{\partial f(\mathbf{x}_k^v, \mathbf{u}_k)}{\partial \mathbf{x}_k^v} = \begin{bmatrix} 1 & 0 & -Tv_k^v \sin\theta_k^v & T\cos\theta_k^v & 0 \\ 0 & 1 & Tv_k^v \cos\theta_k^v & T\sin\theta_k^v & 0 \\ 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 1 - T\alpha_v & 0 \\ 0 & 0 & 0 & 0 & 1 - T\alpha_\omega \end{bmatrix}. \tag{2.17}$$

2.1.2 Landmark Process Models

The environment is modelled in terms of landmark positions relative to the world-centric reference frame F_w. For N_k^M landmarks currently modelled in the system at time step k, each landmark position, $\mathbf{p}_k^{M(i)}$, $i = 1, 2, \ldots, N_k^M$, is written as

$$\mathbf{p}_k^{M(i)} = \left[x_k^{M(i)}, y_k^{M(i)}\right]^T, \tag{2.18}$$

where superscript 'T' represents transposition. The environmental map is captured in the vector

$$\mathbf{M}_k = \begin{bmatrix} \mathbf{p}_k^{M(1)} \\ \vdots \\ \mathbf{p}_k^{M(N_k^M)} \end{bmatrix}. \tag{2.19}$$

Stationary landmarks are required to fix the world-centric frame, developed in Chapter 3, to create an observable system. Dynamic SLAM, treated in Chapter 4, also includes landmarks that are in motion and may transition between stationary and

moving.

With N_k^s stationary landmarks and N_k^m moving landmarks at time step k, $N_k^M = N_k^s + N_k^m$, the environmental map $\mathbf{M}_k$ is partitioned as

$$\mathbf{M}_k = \begin{bmatrix} \mathbf{p}_k^s \\ \mathbf{p}_k^m \end{bmatrix}, \tag{2.20}$$

where the stationary landmark positions are

$$\mathbf{p}_k^s = \begin{bmatrix} \mathbf{p}_k^{s(1)^T} & \cdots & \mathbf{p}_k^{s(N_k^s)^T} \end{bmatrix}^T, \tag{2.21}$$

and the moving landmark positions are

$$\mathbf{p}_k^m = \begin{bmatrix} \mathbf{p}_k^{m(1)^T} & \cdots & \mathbf{p}_k^{m(N_k^m)^T} \end{bmatrix}^T. \tag{2.22}$$

Stationary Landmarks

The motion model for each stationary landmark is linear,

$$\mathbf{p}_{k+1}^{s(i)} = \mathbf{I}_2 \mathbf{p}_k^{s(i)} + \mathbf{q}_k^{s(i)}, \tag{2.23}$$

where $\mathbf{I}_n$ is the $n \times n$ identity matrix, and $\mathbf{q}_k^{s(i)}$ is zero-mean Gaussian with constant covariance matrix

$$\mathbf{Q}^{s(i)} = \begin{bmatrix} \sigma_{x^{s(i)}}^2 & 0 \\ 0 & \sigma_{y^{s(i)}}^2 \end{bmatrix}. \tag{2.24}$$

In practice the covariance $\mathbf{Q}^{s(i)}$ models the confidence that this landmark is stationary. Maintaining some doubt (nonsingular $\mathbf{Q}^{s(i)}$) in the stationary model is needed to detect a transition from stationary to moving.

The overall process model for stationary landmarks is defined with

$$\mathbf{x}_k^s = \mathbf{p}_k^s, \tag{2.25}$$

that has dynamics

$$\mathbf{x}_{k+1}^s = \mathbf{F}_k^s \mathbf{x}_k^s + \mathbf{q}_k^s, \tag{2.26}$$

where $\mathbf{F}_k^s = \mathbf{I}_{N_k^s}$, and $\mathbf{q}_k^s$ is the concatenation of the noise vectors $\mathbf{q}_k^{s(\imath)}$ resulting in the diagonal covariance matrix

$$\mathbf{Q}_k^s = \text{diag}\left[\mathbf{Q}^{s(1)}, \ldots, \mathbf{Q}^{s(N_k^s)}\right]. \tag{2.27}$$

Note that although each $\mathbf{Q}^{s(\imath)}$ is assumed to be constant over time, the covariance $\mathbf{Q}_k^s$ may change dimensions as landmarks are added or deleted from the stationary model.

Moving Landmarks

Singer's [44] exponentially correlated acceleration model for target tracking is used in this thesis to model the motion of moving landmarks. This model is extensively used in radar tracking of unknown objects [41] [7]. A key challenge in tracking motion of unknown objects is that both their process and control models are completely unknown. In the autonomous SLAM problem, the only information about moving landmarks are the vehicle-centric measurements of the landmark positions at each sample time. Singer's model is expressed in 2D form for the purposes of this thesis using landmark position, velocity and acceleration. For the ith moving landmark, $i = 1, \ldots, N_k^m$, define

$$\mathbf{x}_k^{m(\imath)} = \begin{bmatrix} \mathbf{p}_k^{m(\imath)} \\ \dot{\mathbf{p}}_k^{m(\imath)} \\ \ddot{\mathbf{p}}_k^{m(\imath)} \end{bmatrix}. \tag{2.28}$$

Singer's discrete-time process model for a moving landmark has the linear form

$$\mathbf{x}_{k+1}^{m(\imath)} = \mathbf{F}^{m(\imath)}\mathbf{x}_k^{m(\imath)} + \mathbf{W}\mathbf{q}_k^{m(\imath)}, \tag{2.29}$$

in which the state transition matrix is the exponential

$$\mathbf{F}_k^{m(\imath)} = \mathrm{e}^{\mathbf{A}T}, \tag{2.30}$$

where T is the sample period and, with $\mathbf{0}_{n\times m}$ representing an $n \times m$ matrix of zeros,

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_{4\times 2} & \mathbf{I}_4 \\ \mathbf{0}_{2\times 4} & -\alpha_m \mathbf{I}_2 \end{bmatrix}. \tag{2.31}$$

In (2.29), the two-dimensional zero-mean Gaussian noise $\mathbf{q}_k^{m(i)}$ is assumed to enter equally with variance ${\sigma_m}^2$ into the acceleration channels with

$$\mathbf{W} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{2.32}$$

The noise in the model is the product $\mathbf{W}\mathbf{q}_k^{m(i)}$ that has zero mean and an approximate covariance matrix [2]

$$\mathbf{Q}^{m(i)} = 2\alpha_m {\sigma_m}^2 T \mathbf{W}\mathbf{W}^T. \tag{2.33}$$

Note that superscript 'T' still denotes transposition. In the covariance (2.33), the user-defined parameter $\alpha_m \geq 0$ is the reciprocal of the acceleration time constant. A larger value of α_m is used for targets with unpredictable changes in acceleration, whereas a lower value of α_m is used for more smooth landmark motion.

The overall process model for moving landmarks is constructed with the state vector

$$\mathbf{x}_k^m = \begin{bmatrix} \mathbf{x}_k^{m(1)} \\ \vdots \\ \mathbf{x}_k^{m(N_k^m)} \end{bmatrix}. \tag{2.34}$$

The dynamics are written as

$$\mathbf{x}_{k+1}^m = \mathbf{F}_k^m \mathbf{x}_k^m + \mathbf{q}_k^m, \tag{2.35}$$

where

$$\mathbf{F}_k^m = \text{diag}\left[\mathbf{F}_k^{m(1)}, \cdots, \mathbf{F}_k^{m(N_k^m)}\right], \tag{2.36}$$

and $\mathbf{q}_k^m$ is the concatenation of the noise vectors $\mathbf{q}_k^{m(\imath)}$ resulting in the diagonal covariance matrix

$$\mathbf{Q}_k^m = \text{diag}\left[\mathbf{Q}^{m(1)}, \cdots, \mathbf{Q}^{m(N_k^m)}\right]. \tag{2.37}$$

Note that although each $\mathbf{Q}^{m(\imath)}$ is assumed to be constant over time, the covariance $\mathbf{Q}_k^m$ may change dimensions as landmarks are added or deleted from the moving model.

2.1.3 Measurement Model

In the autonomous SLAM problem, all measurements of landmark positions are vehicle-centric. In this thesis, measurements are modelled in terms of range and bearing of the landmark relative to the vehicle as illustrated in Figure 2.1. At each sample period, and for the ith landmark, $i = 1, \ldots, N_k^M$, (including both stationary and moving landmarks) define

$$\Delta x_k^{M(\imath)} = x_k^{M(\imath)} - x_k^v, \tag{2.38}$$

$$\Delta y_k^{M(\imath)} = y_k^{M(\imath)} - y_k^v. \tag{2.39}$$

The range is

$$r_k^{M(\imath)} = \sqrt{\Delta x_k^{M(\imath)^2} + \Delta y_k^{M(\imath)^2}} \tag{2.40}$$

and the bearing is

$$b_k^{M(\imath)} = \arctan\left(\frac{\Delta y_k^{M(\imath)}}{\Delta x_k^{M(\imath)}}\right) - \theta_k^v. \tag{2.41}$$

The measurements for the ith stationary landmark $i = 1, \ldots, N_k^s$, can then be written as

$$\mathbf{z}_k^{s(\imath)} = \mathbf{h}^{s(\imath)}(\mathbf{x}_k^v, \mathbf{x}_k^{s(\imath)}) + \mathbf{w}_k^{s(\imath)}, \tag{2.42}$$

where

$$\mathbf{h}^{s(i)}(\mathbf{x}_k^v, \mathbf{x}_k^{s(i)}) = \begin{bmatrix} r_k^{s(i)} \\ b_k^{s(i)} \end{bmatrix}, \tag{2.43}$$

and the noise process $\mathbf{w}_k^{s(i)}$ is assumed to be zero-mean Gaussian with covariance matrix

$$\mathbf{R}^{s(i)} = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_b^2 \end{bmatrix}. \tag{2.44}$$

Similarly, for the ith moving landmark, $i = 1, \ldots, N_k^m$, the measurements can be written as

$$\mathbf{z}_k^{m(i)} = \mathbf{h}^{m(i)}(\mathbf{x}_k^v, \mathbf{x}_k^{m(i)}) + \mathbf{w}_k^{m(i)}, \tag{2.45}$$

where

$$\mathbf{h}^{m(i)}(\mathbf{x}_k^v, \mathbf{x}_k^{m(i)}) = \begin{bmatrix} r_k^{m(i)} \\ b_k^{m(i)} \end{bmatrix}, \tag{2.46}$$

and the noise process $\mathbf{w}_k^{m(i)}$ is assumed to be zero-mean Gaussian with covariance matrix

$$\mathbf{R}^{m(i)} = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_b^2 \end{bmatrix}. \tag{2.47}$$

The difference between $\mathbf{h}^{s(i)}(\mathbf{x}_k^v, \mathbf{x}_k^{s(i)})$ in (2.43) and $\mathbf{h}^{m(i)}(\mathbf{x}_k^v, \mathbf{x}_k^{m(i)})$ in (2.46) is in the dimension of the second argument; each $\mathbf{x}_k^{s(i)}$ is two-dimensional, whereas each $\mathbf{x}_k^{m(i)}$ is six-dimensional. Hence the Jacobian matrices of these functions, needed for the EKF implementation, have different dimensions.

The complete measurement model for stationary landmarks can now be written as

$$\mathbf{z}_k^s = \mathbf{h}^s(\mathbf{x}_k^v, \mathbf{x}_k^s) + \mathbf{w}_k^s, \tag{2.48}$$

where

$$\mathbf{z}_k^s = \begin{bmatrix} \mathbf{z}_k^{s(1)} \\ \vdots \\ \mathbf{z}_k^{s(N_k^s)} \end{bmatrix}, \tag{2.49}$$

$$\mathbf{h}^s(\mathbf{x}_k^v, \mathbf{x}_k^s) = \begin{bmatrix} \mathbf{h}^{s(1)}(\mathbf{x}_k^v, \mathbf{x}_k^{s(1)}) \\ \vdots \\ \mathbf{h}^{s(N_k^s)}(\mathbf{x}_k^v, \mathbf{x}_k^{s(N_k^s)}) \end{bmatrix}, \tag{2.50}$$

and $\mathbf{w}_k^s$ is the concatenation of the noise vectors $\mathbf{w}_k^{s(i)}$ resulting in the diagonal covariance matrix

$$\mathbf{R}_k^s = \text{diag}\left[\mathbf{R}^{s(1)}, \cdots, \mathbf{R}^{s(N_k^s)}\right]. \tag{2.51}$$

For moving landmarks the complete measurement model is

$$\mathbf{z}_k^m = \mathbf{h}^m(\mathbf{x}_k^v, \mathbf{x}_k^m) + \mathbf{w}_k^m, \tag{2.52}$$

where

$$\mathbf{z}_k^m = \begin{bmatrix} \mathbf{z}_k^{m(1)} \\ \vdots \\ \mathbf{z}_k^{m(N_k^m)} \end{bmatrix}, \tag{2.53}$$

$$\mathbf{h}^m(\mathbf{x}_k^v, \mathbf{x}_k^m) = \begin{bmatrix} \mathbf{h}^{m(1)}(\mathbf{x}_k^v, \mathbf{x}_k^{m(1)}) \\ \vdots \\ \mathbf{h}^{m(N_k^s)}(\mathbf{x}_k^v, \mathbf{x}_k^{m(N_k^m)}) \end{bmatrix}, \tag{2.54}$$

and $\mathbf{w}_k^m$ is the concatenation of the noise vectors $\mathbf{w}_k^{m(i)}$ resulting in the diagonal covariance matrix

$$\mathbf{R}_k^m = \text{diag}\left[\mathbf{R}^{m(1)}, \cdots, \mathbf{R}^{m(N_k^m)}\right]. \tag{2.55}$$

Implementation of the EKF requires computation of the Jacobian matrices for the measurement models. These matrices can be assembled based on the following submatrices. For the stationary landmarks, define

$$\mathbf{H}_k^{v(i)} = \frac{\partial \mathbf{h}^{s(i)}(\mathbf{x}_k^v, \mathbf{x}_k^{s(i)})}{\partial \mathbf{x}_k^v}. \tag{2.56}$$

For the simple unicycle model,

$$\mathbf{H}_k^{v(i)} = \begin{bmatrix} \frac{-\Delta x_k^{s(i)}}{r_k^{s(i)}} & \frac{-\Delta y_k^{s(i)}}{r_k^{s(i)}} & 0 \\ \frac{\Delta y_k^{s(i)}}{r_k^{s(i)^2}} & \frac{-\Delta x_k^{s(i)}}{r_k^{s(i)^2}} & -1 \end{bmatrix}, \tag{2.57}$$

and for the extended unicycle model,

$$\mathbf{H}_k^{v(i)} = \begin{bmatrix} \frac{-\Delta x_k^{s(i)}}{r_k^{s(i)}} & \frac{-\Delta y_k^{s(i)}}{r_k^{s(i)}} & 0 & 0 & 0 \\ \frac{\Delta y_k^{s(i)}}{r_k^{s(i)^2}} & \frac{-\Delta x_k^{s(i)}}{r_k^{s(i)^2}} & -1 & 0 & 0 \end{bmatrix}. \tag{2.58}$$

Differentiation with respect to the stationary landmark positions is also needed,

$$\mathbf{H}_k^{s(i)} = \frac{\partial \mathbf{h}^{s(i)}(\mathbf{x}_k^v, \mathbf{x}_k^{s(i)})}{\partial \mathbf{x}_k^{s(i)}} = \begin{bmatrix} \frac{\Delta x_k^{s(i)}}{r_k^{s(i)}} & \frac{\Delta y_k^{s(i)}}{r_k^{s(i)}} \\ \frac{-\Delta y_k^{s(i)}}{r_k^{s(i)^2}} & \frac{\Delta x_k^{s(i)}}{r_k^{s(i)^2}} \end{bmatrix}. \tag{2.59}$$

The required Jacobian for the vehicle and stationary landmark state vectors is then assembled as follows:

$$\mathbf{H}_k = \begin{bmatrix} \mathbf{H}_k^{v(1)} & \mathbf{H}_k^{s(1)} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{H}_k^{v(2)} & \mathbf{0} & \mathbf{H}_k^{s(2)} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & & \ddots & \vdots & \vdots \\ \mathbf{H}_k^{v(N_k^s)} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{H}_k^{s(N_k^s)} \end{bmatrix}. \tag{2.60}$$

For moving landmarks, define

$$\mathbf{H}_k^{m(i)} = \frac{\partial \mathbf{h}^{m(i)}(\mathbf{x}_k^v, \mathbf{x}_k^{m(i)})}{\partial \mathbf{x}_k^{m(i)}} = \begin{bmatrix} \frac{\Delta x_k^{m(i)}}{r_k^{m(i)}} & \frac{\Delta y_k^{m(i)}}{r_k^{m(i)}} & 0 & 0 & 0 & 0 \\ \frac{-\Delta y_k^{m(i)}}{r_k^{m(i)^2}} & \frac{\Delta x_k^{m(i)}}{r_k^{m(i)^2}} & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{2.61}$$

The required Jacobian for the moving landmark state vector is then block diagonal,

$$\mathbf{H}_k^m = \text{diag}\left[\mathbf{H}_k^{m(1)}, \mathbf{H}_k^{m(2)}, \cdots, \mathbf{H}_k^{m(N_k^m)}\right]. \tag{2.62}$$

2.1.4 Autonomous SLAM in Dynamic Environments

The models developed in the above subsections are used to define the SLAM problem in mathematical terms. The complete problem for autonomous SLAM in dynamic environments can now be posed. This thesis approaches the dynamic SLAM problem by decoupling the state estimation algorithms into two Extended Kalman Filters that operate in parallel.

Define the SLAM filter state vector to combine the vehicle state and the stationary landmark state,

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_k^v \\ \mathbf{x}_k^s \end{bmatrix}. \tag{2.63}$$

The vehicle process model is generally nonlinear,

$$\mathbf{x}_{k+1}^v = \mathbf{f}(\mathbf{x}_k^v, \mathbf{u}_k) + \mathbf{q}_k^v, \tag{2.64}$$

and the stationary landmark process model is linear

$$\mathbf{x}_{k+1}^s = \mathbf{F}_k^s \mathbf{x}_k^s + \mathbf{q}_k^s. \tag{2.65}$$

Moving landmarks are modelled with Singer's method to give linear dynamics,

$$\mathbf{x}_{k+1}^m = \mathbf{F}_k^m \mathbf{x}_k^m + \mathbf{q}_k^m. \tag{2.66}$$

Measurements provided by on-board sensors and processing relative to stationary landmarks are

$$\mathbf{z}_k^s = \mathbf{h}^s(\mathbf{x}_k^v, \mathbf{x}_k^s) + \mathbf{w}_k^s, \tag{2.67}$$

and measurements relative to moving landmarks are

$$\mathbf{z}_k^m = \mathbf{h}^m(\mathbf{x}_k^v, \mathbf{x}_k^m) + \mathbf{w}_k^m. \tag{2.68}$$

The dynamic SLAM problem is to estimate the state vectors $\mathbf{x}_k^v$, $\mathbf{x}_k^s$, and $\mathbf{x}_k^m$ based on the measurements $\mathbf{z}_k^s$ and $\mathbf{z}_k^m$. The parallel solution structure used in this thesis includes the SLAM filter and the Moving Landmark Filter (MLF). The SLAM filter

is used to estimate $\mathbf{x}_k$, the vehicle pose and motion and the positions of stationary landmarks, based solely on the measurements $\mathbf{z}_k^s$. The MLF is implemented to estimate $\mathbf{x}_k^m$ to recover the position and motion of moving landmarks based on the measurements $\mathbf{z}_k^m$. These filters are coupled in the sense that the current estimate of vehicle pose and motion from the SLAM filter is employed in the Moving Landmark filter as described further in Chapter 4. Landmarks may also transition between filters in response to the dynamic environment.

2.2 The Kalman Filter

Linear systems are fundamental in describing how many different types of processes behave when a control input is given. Accurate control of a linear system allows engineers to build the types of complex machinery found in modern society. Many control techniques require that an estimate of the complete system state is available. Observers (or estimators) are used to predict the complete state of a system from available measurements when the full state of the system cannot be estimated directly with each measurement event [18]. The Kalman Filter was created as an optimal, unbiased, minimum-variance observer for linear Gaussian systems [27]. Under linear Gaussian assumptions, the probability distribution of the system state can be fully represented by a mean state estimate and its covariance. The Gaussian assumption has been shown to be valid in many cases whereas linear and unbiased assumptions can pose problems with many real world systems. The Kalman Filter can be thought of as a recursive, predictor-corrector algorithm that computes an improved a posteriori estimate using measurements to feedback into an a priori estimate that is based on the process model.

The Kalman Filter has two phases: the time update (prediction or projection) phase and the measurement correction phase. The time update phase is responsible for propagating the changes given by the input and process model and reflecting those changes in the system. After the time update phase, the system reflects the 'ideal' state of the system, as if the model describing the system and the input are perfect. In practice, this is not sufficient as the process model and input model are

not fully accurate and do not accommodate small non-linearities or random errors. Given that there is always noise added to the system, if only time update phases were performed, the error in the system may increase unbounded. This case is known as 'dead-reckoning'. To limit the increase in the error and get a better estimate for the state of the system, a correction phase is performed. The correction phase is designed to integrate sensory information into the state estimate. The key to this filter, however, is deciding how to weight the information given by the sensor and how to weight the information given by the process and input models. If a designer has prior knowledge that their process model is inaccurate, it may be wise to place more trust in the sensor model, and vice versa. In practice, parameters within the filter can be tuned thereby allowing for differing amounts of "trust" to be placed in each of the process and sensor models.

The Linear/Gaussian assumptions that form the basis for the Kalman Filter sometimes do not accurately reflect reality in a system. Specifically when it comes to wheeled mobile robots, the linear assumption fails as standard wheels are highly non-linear in that they can only travel along the direction the wheel is pointing. To address this shortcoming, the classical approach is to use the Extended Kalman Filter. This filter allows for the state estimation of non-linear systems at the expense of the optimality of the filter. The EKF is explored in more detail later in this section.

2.2.1 The Discrete-Time Kalman Filter

Since mobile robots generally have computer systems on board, and computer systems are discrete time systems, it is useful to consider the discrete time case for the Kalman Filter. The continuous time filter exists and is closely related to the discrete time version. The Kalman Filter aims to find an optimal observer for a linear system of the form:

$$\mathbf{x}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1} + \mathbf{q}_{k-1}, \tag{2.69}$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{w}_k. \tag{2.70}$$

Equation (2.69) represents the system dynamics where k is the timestep, $\mathbf{x}_k$ is the system state, $\mathbf{u}_k$ is the known input to the system, and $\mathbf{q}_k$ represents a zero-mean white Gaussian noise process with covariance $\mathbf{Q}_k$,

$$E[\mathbf{q}_k] = 0, \tag{2.71}$$

and

$$\mathbf{Q}_k = E[\mathbf{q}_k \mathbf{q}_k^T]. \tag{2.72}$$

Note that superscript T denotes transposition, and the notation $E[\cdot]$ represents expectation. Equation (2.70) models the system measurements, $\mathbf{z}_k$, and $\mathbf{w}_k$ represents a zero-mean white Gaussian noise process with covariance $\mathbf{R}_k$,

$$E[\mathbf{w}_k] = 0, \tag{2.73}$$

and

$$\mathbf{R}_k = E[\mathbf{w}_k \mathbf{w}_k^T]. \tag{2.74}$$

It is also assumed that processes $\mathbf{q}_k$ and $\mathbf{w}_k$ are uncorrelated.

The objective of the Kalman Filter is to maintain an estimate, $\hat{\mathbf{x}}_k$, of the true state $\mathbf{x}_k$, and also to maintain the covariance of this estimate, $\mathbf{P}_k$. The filter is initialized with

$$\hat{\mathbf{x}}_0^+ = E[\mathbf{x}_0], \tag{2.75}$$

and

$$\mathbf{P}_0^+ = E[\hat{\mathbf{x}}_0^+ \hat{\mathbf{x}}_0^{+T}]. \tag{2.76}$$

Special notaton is introduced to identify a priori and a posteriori state and covariance values:

- $\hat{\mathbf{x}}_k^-$ is the a priori state estimate at timestep k;
- $\hat{\mathbf{x}}_k^+$ is the a posteriori state estimate at timestep k;

- $\mathbf{P}_k^-$ is the a priori covariance matrix at timestep k; and
- $\mathbf{P}_k^+$ is the a posteriori covariance matrix at timestep k.

The Kalman Filter is implemented recursively due to the Markov assumption made in the underlying Bayes Filter. The state at any timestep is completely determined by the state at the previous timestep, the inputs to the system, and the sensor readings. This means that the full state history of the system does not need to be stored, thereby allowing the Kalman Filter to be run *online*. Each iteration of the filter is executed in two steps. In the first step, the state estimate and covariance are updated over one sample period based on the process model (2.69) to produce the a priori estimates. The second step is a measurement update of the state and covariance based on the measurement model (2.70)to give the a posteriori estimates.

Assume that $\hat{\mathbf{x}}_{k-1}^+$, $\mathbf{P}_{k-1}^+$, and the known input $\mathbf{u}_{k-1}$ are available. The time update equations are given by

$$\hat{\mathbf{x}}_k^- = \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1}^+ + \mathbf{G}_{k-1}\mathbf{u}_{k-1}, \tag{2.77}$$

and

$$\mathbf{P}_k^- = \mathbf{F}_{k-1}\mathbf{P}_{k-1}^+\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1}. \tag{2.78}$$

The a priori estimates $\hat{\mathbf{x}}_k^-$ and $\mathbf{P}_k^-$ are now available for the measurement update step once $\mathbf{z}_k$ is available,

$$\mathbf{v}_k = \mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-, \tag{2.79}$$

$$\mathbf{S}_k = \mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k, \tag{2.80}$$

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^T\mathbf{S}_k^{-1}, \tag{2.81}$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k\mathbf{v}_k, \tag{2.82}$$

and

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^-. \tag{2.83}$$

This generates the a posteriori estimates $\hat{\mathbf{x}}_k^+$ and $\mathbf{P}_k^+$. In equation (2.79), $\mathbf{v}_k$ is referred to as the *innovation* sequence. The covariance of $\mathbf{v}_k$ is $\mathbf{S}_k$ given in equation (2.80). The matrix $\mathbf{K}_k$ in (2.81) is the Kalman gain that weights the innovation in (2.82) in order to give the improved estimate $\hat{\mathbf{x}}_k^+$. The covariance $\mathbf{S}_k$ and the innovation $\mathbf{v}_k$ are used in Section 2.5 in order to detect landmarks that transition from stationary to moving.

2.2.2 The Discrete-Time Extended Kalman Filter

The Kalman Filter as described in Section 2.2.1 is designed specifically for linear systems. In practice, most systems modelled in the field of mobile robotics are non-linear. At the expense of optimality, it is possible to allow for the linear framework to accommodate non-linear systems through linearization of the system. The Extended Kalman Filter (EKF) is a method to apply the linear Kalman Filter algorithm to non-linear systems. The EKF linearizes the non-linear system about the most recent state estimate. The EKF cannot guarantee optimality or that the covariance $\mathbf{P}_k$ represents the true state covariance. A key assumption that must be made when the EKF is used is that the linearized model for the system reasonably represents the true behaviour of the system over short time intervals. The EKF approach can be successfully used for estimating the state of non-linear systems and represents the classical approach to solving the SLAM problem.

For a nonlinear system with process model

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{q}_{k-1}, \tag{2.84}$$

and measurement model

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{w}_k. \tag{2.85}$$

the EKF linearizes the process ($\mathbf{f}(.)$) and measurement ($\mathbf{h}(.)$) models about the most recent state estimate, and then proceeds with the linear form of the filter for

implementation. The linearization is achieved by defining

$$\mathbf{F}_{k-1} = \left.\frac{\partial \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1})}{\partial \mathbf{x}_{k-1}}\right|_{\mathbf{x}_{k-1}=\hat{\mathbf{x}}_{k-1}^{+}} \tag{2.86}$$

and

$$\mathbf{H}_k = \left.\frac{\partial \mathbf{h}(\mathbf{x}_k)}{\partial \mathbf{x}_k}\right|_{\mathbf{x}_k=\hat{\mathbf{x}}_k^{-}} \tag{2.87}$$

which are Jacobian matrices representing the linearized process and measurement models.

The time update phase of the EKF is carried out as

$$\hat{\mathbf{x}}_k^{-} = \mathbf{f}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}) \tag{2.88}$$

and

$$\mathbf{P}_k^{-} = \mathbf{F}_{k-1}\mathbf{P}_{k-1}^{+}\mathbf{F}_{k-1}^{T} + \mathbf{Q}_k. \tag{2.89}$$

The innovation calculation in (2.79) is replaced with

$$\mathbf{v}_k = \mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_k^{-}), \tag{2.90}$$

and the remainder of the measurement update step proceeds as in equations (2.80) through (2.83).

2.3 Observability

A dynamic system is said to be "observable" if it is possible to estimate the complete state of the system from available measurements over time. Concepts of observability take on different forms depending on whether the system is linear, nonlinear, and/or stochastic. An observer (or estimator) is a subsystem that is capable of estimating the complete state of a system based on the available measurements. The need for observers arises when the full state of a system cannot be measured directly, thus complicating feedback for control systems. The Kalman filter is an observer that must satisfy observability conditions in order to reliably estimate the state of

a system.

Consider a linear time invariant system with the state vector $\mathbf{x}_k \in \Re^n$ for which the process model is

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{G}\mathbf{u}_{k-1}, \tag{2.91}$$

and the measurement model, with $\mathbf{z} \in \Re^m$, is

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k. \tag{2.92}$$

This system is observable if and only if [18] the $nm \times n$ observability matrix given by

$$\mathbf{O} = \begin{bmatrix} H \\ HF \\ HF^2 \\ \vdots \\ HF^{n-1} \end{bmatrix} \tag{2.93}$$

has full rank, namely $\text{rank}(\mathbf{O}) = n$.

The SLAM problem as described in Section 2.1.4 is nonlinear. Observability for nonlinear discrete-time systems replaces matrix multiplication in (2.93) with functional composition and normally gives only local (in time and space) indications of whether a system is observable. A detailed analysis of observability for the non-linear continuous-time two-dimensional autonomous SLAM problem is given by Lee, Wijesoma and Guzman [29] using differential geometric methods and the Lie Algebra. However, it is usually the case that the SLAM problem is formulated in discrete-time.

For a nonlinear deterministic discrete-time system of the form

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{f}(\mathbf{x}_k) \\ \mathbf{z}_k &= \mathbf{h}(\mathbf{x}_k), \end{aligned} \tag{2.94}$$

define $\mathbf{f} \circ \mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{f}(\mathbf{x}))$ and

$$\begin{aligned} \mathbf{f}^0(\mathbf{x}) &= \mathbf{x} \\ \mathbf{f}^j(\mathbf{x}) &= \mathbf{f} \circ \mathbf{f}^{j-1}(\mathbf{x}). \end{aligned} \tag{2.95}$$

Following the review in [2], for any given initial state $\mathbf{x}$ this system produces an output sequence given by the ordered set

$$\mathcal{O}(\mathbf{x}) = \left\{ \mathbf{h}(\mathbf{x}), \mathbf{h} \circ \mathbf{f}(\mathbf{x}), \mathbf{h} \circ \mathbf{f}^2(\mathbf{x}), \ldots, \mathbf{h} \circ \mathbf{f}^k(\mathbf{x}), \ldots \right\}. \tag{2.96}$$

A pair of points $\mathbf{x}$ and $\bar{\mathbf{x}}$ are said to be *indistinguishable* if $\mathcal{O}(\mathbf{x}) = \mathcal{O}(\bar{\mathbf{x}})$. Define the set of points that are indistinguishable from $\bar{\mathbf{x}}$ as

$$\mathcal{I}(\bar{\mathbf{x}}) = \{ \mathbf{x} | \mathcal{O}(\mathbf{x}) = \mathcal{O}(\bar{\mathbf{x}}) \}. \tag{2.97}$$

The system is then said to be observable at $\bar{\mathbf{x}}$ if $\mathcal{I}(\bar{\mathbf{x}}) = \{\bar{\mathbf{x}}\}$, otherwise the system is not globally observable. If $\mathbf{x}$ is indistinguishable from a set of isolated points, $\bar{\mathbf{x}}^i$, $i = 1, 2, 3, \ldots$, then the system is locally observable about $\mathbf{x}$. The distance between $\mathbf{x}$ and the closest indistinguishable point provides a measure of the minimum size of the observable subspace.

In Chapter 3, an explanation is given based on this concept of indistinguishability to answer the question of why the autonomous SLAM problem is not observable. In particular, this explanation clearly shows that there are three degrees of freedom for the two-dimensional planar case. Additional analysis of observability and accuracy of the nonlinear discrete-time SLAM problem is conducted in this thesis using Fisher's Information Matrix and the Cramer Rao Lower Bounds (CRLBs).

2.4 Cramer-Rao Lower Bounds

The EKF represents a sub-optimal solution to a non-linear state estimation problem. It is desirable, however, to determine how well the EKF-based solution performs in comparison to the optimal performance possible. The Cramer-Rao Lower Bounds

(CRLBs) provide this performance assessment for state estimation techniques.

Consider a nonlinear discrete time state estimation problem with noise free process model

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k), \tag{2.98}$$

and noisy measurement model

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{w}_k, \tag{2.99}$$

where $\mathbf{w}_k$ is zero-mean temporally white Gaussian noise with covariance matrix $\mathbf{R}_k$. Assume further that an unbiased initial estimate $\hat{\mathbf{x}}_0$ of the state at time $k = 0$ is available with covariance matrix $\mathbf{P}_0$. The Cramer-Rao Lower Bounds provide an assessment of optimal accuracy in estimation of the state $\mathbf{x}_k$ based on the observations

$$\mathbf{Z}_k = \{\hat{\mathbf{x}}_0, \mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_k\}. \tag{2.100}$$

The Cramer-Rao results state that the covariance matrix $\mathbf{C}_k$ for an unbiased estimator $\hat{\mathbf{x}}_k$ of the true state $\mathbf{x}_k$ is bounded from below by the inverse of Fisher's Information Matrix, $\mathbf{J}_k$ [47] [2]. These lower bounds are expressed as

$$\mathbf{C}_k = \mathbf{E}[(\hat{\mathbf{x}}_k - \mathbf{x}_k)(\hat{\mathbf{x}}_k - \mathbf{x}_k)^T] \geq \mathbf{J}_k^{-1}, \tag{2.101}$$

where Fisher's Information Matrix is defined as the conditional expectation

$$\mathbf{J}_k = \mathbf{E}\left[(\frac{\partial \ln p(\mathbf{Z}_k|\mathbf{x}_k)}{\partial \mathbf{x}_k})^T (\frac{\partial \ln p(\mathbf{Z}_k|\mathbf{x}_k)}{\partial \mathbf{x}_k}) \middle| \mathbf{x}_k \right], \tag{2.102}$$

and $p(\mathbf{Z}_k|\mathbf{x}_k)$ is the joint conditional probability density function of $\mathbf{Z}_k$ given $\mathbf{x}_k$. The matrix inequality $\mathbf{C}_k \geq \mathbf{J}_k^{-1}$ is equivalent to stating that $(\mathbf{C}_k - \mathbf{J}_k^{-1})$ is positive semi-definite. Since the diagonal elements of a positive semi-definite matrix are non-negative, the diagonal elements of $\mathbf{J}_k^{-1}$ provide the estimate error variance lower bounds for the corresponding elements of $\hat{\mathbf{x}}_k$.

For many probability density functions, the calculation of $\mathbf{J}_k$ in (2.102) may be difficult. However, the density function $p(\mathbf{Z}_k|\mathbf{x}_k)$ has Gaussian form due to the

Gaussian assumption on the measurement noise $\mathbf{w}_k$. In this case the calculation of Fisher's Information Matrix is reduced to a simple recursive form (see Appendix C of [2]) with initialization

$$\mathbf{J}_0 = \mathbf{P}_0^{-1} + \mathbf{H}_0^T \mathbf{R}_0^{-1} \mathbf{H}_0, \tag{2.103}$$

and time updates

$$\mathbf{J}_{k+1} = \mathbf{F}_k^{-1^T} \mathbf{J}_k \mathbf{F}_k^{-1} + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k, \tag{2.104}$$

where

$$\mathbf{F}_k = \left. \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_k}, \tag{2.105}$$

and

$$\mathbf{H}_k = \left. \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_k}. \tag{2.106}$$

It is important to note that the Cramer-Rao Lower Bounds are primarily used to address the question, "If the system follows a known trajectory through the state space and produces the defined noisy measurements, what is the lower bound on the variance for the estimation error for each element of the state vector over this trajectory?" The key assumption is the "known trajectory" that results from the noise free process model in (2.98). This is why the Jacobian matrices in (2.105) and (2.106) are computed based on the true state $\mathbf{x}_k$ rather than the estimated state $\hat{\mathbf{x}}_k$ as in the EKF. Fisher's information matrix also has the property that in the absence of a priori information (i.e. when $\mathbf{P}_0^{-1} = \mathbf{0}$) if $\mathbf{J}_k$ remains singular, then the system has an unobservable subspace along this trajectory. The Cramer-Rao Bounds also permit one to analyze estimation accuracy in the presence of highly accurate a priori information (with $\mathbf{P}_0 \approx \mathbf{0}$). These properties of the Cramer-Rao Bounds are exploited in Chapter 3 to analyze observability and accuracy in state estimation for the autonomous SLAM problem.

2.5 Landmark Motion Detection

A solution to the SLAM problem requires the availability of vehicle-relative measurements of positions of stationary landmarks in the environment. In order to

implement SLAM for dynamic environments, the solution method must be able to recognize when a landmark is no longer stationary. It is also desireable to be able to detect when a moving landmark becomes stationary so that computational effort in the filtering algorithms may be reduced. In Chapter 4, the proposed SLAM solution is extended to dynamic environments using two Extended Kalman Filters that operate in parallel: the SLAM filter is used to estimate the vehicle pose and motion and the positions of stationary landmarks; and the Moving Landmark Filter (MLF) is implemented to estimate the position and motion of moving landmarks. Hypothesis testing with hysteresis is demonstrated in Chapter 4 as a method to determine when a stationary landmark starts moving and should therefore be removed from the SLAM filter and added to the MLF.

A simple method to detect when a stationary landmark starts to move is to compute the Cartesian distance between the estimates of a landmark's position between single or multiple timesteps and use a threshold test. However, this approach does not account for the stochastic nature of the estimation problem. In the literature for the data association problem or for adaptive Kalman filtering [7], such tests are conducted in a statistical way using the Mahalanobis Distance (or Statistical Distance) moved by the landmark. As illustrated in Figure 2.2, the set of points within a given Cartesian distance of a specified point defines a hyperspere in state space. In contrast, the Mahalanobis distance accounts for the covariance of the observed process and results in hyperellipsoids rather than hyperspheres.

Under the usual Gaussian assumptions for the linear Kalman filter reviewed in Section 2.2.1, the innovation sequence $\mathbf{v}_k$ in (2.79) (repeated here)

$$\mathbf{v}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-, \tag{2.107}$$

is zero mean Gaussian with covariance $\mathbf{S}_k$ given in (2.80). The measurements $\mathbf{z}_k$ in (2.79) are also Gaussian with mean equal to the predicted measurements,

$$\hat{\mathbf{z}}_k^- = \mathbf{H}_k \hat{\mathbf{x}}_k^-, \tag{2.108}$$

and the same covariance, $\mathbf{S}_k$, as the innovation. Although the complete measure-

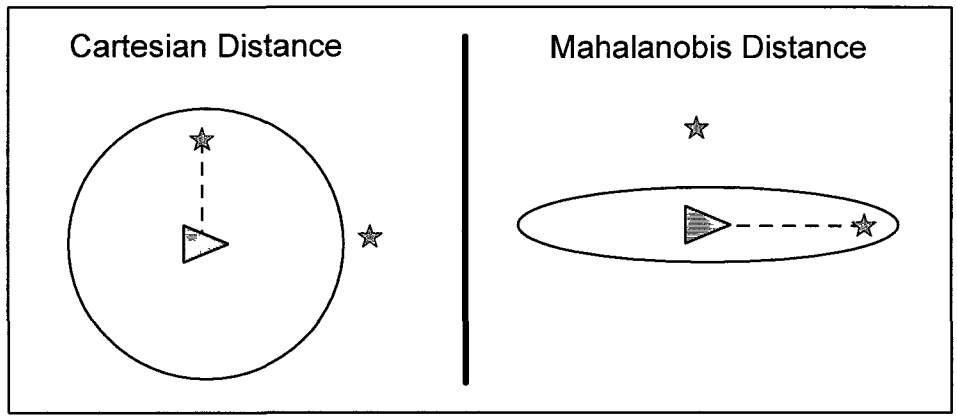

Figure 2.2: Cartesian vs. Mahalanobis Distance

ment set $\mathbf{z}_k$ is used to simplify notation in the following review, the applications of these methods in Chapter 4 extract measurements corresponding to single landmarks, $\mathbf{x}_k^{s(i)}$, along with the corresponding elements from $\hat{\mathbf{z}}_k^-$ and the corresponding submatrix of $\mathbf{S}_k$ in order to test each stationary landmark for the onset of motion.

The Mahalanobis norm represents the statistical distance between the measurements received at time k, $\mathbf{z}_k$, and the predicted measurements $\hat{\mathbf{z}}_k^-$ and is defined as

$$D(\mathbf{z}_k, \hat{\mathbf{z}}_k^-) = \sqrt{\mathbf{v}_k^T \mathbf{S}_k^{-1} \mathbf{v}_k}. \tag{2.109}$$

The set of points in measurement space defined as

$$V_k(\gamma) = \left\{ \mathbf{z}_k | D^2(\mathbf{z}_k, \hat{\mathbf{z}}_k^-) \leq \gamma \right\} \tag{2.110}$$

represents an ellipsoid of probability concentration. Under the Gaussian assumptions, the squared Mahalanobis distance follows a Chi-squared distribution with the number of degrees of freedom equal to the dimension of the measurement vector. In this thesis there are two degrees of freedom (x and y coordinates) when testing each landmark. The value of γ used in the tests is taken from a Chi-squared distribution table (for example from [7]) based on the user-defined confidence level α. For example, given two degrees of freedom, at a confidence level of 99% ($\alpha = 0.01$), the

value of γ from the Chi-squared table is 9.31.

The motion detection problem can now be formulated as a hypothesis test where the null hypothesis (H_0) is that the landmark is stationary and the alternative hypothesis (H_A) is that the landmark has started moving. When a measurement $\mathbf{z}_k$ is received, the squared Mahalanobis distance is tested against the value of γ. If $D^2(\mathbf{z}_k, \hat{\mathbf{z}}_k^-) \leq \gamma$, this indicates that the measurements have been drawn from a Gaussian distribution with mean $\hat{\mathbf{z}}_k^-$ and covariance $\mathbf{S}_k$ at the $100(1-\alpha)\%$ confidence level, and hence the null hypothesis H_0 should not be rejected. If, however, it is found that $D^2(\mathbf{z}_k, \hat{\mathbf{z}}_k^-) > \gamma$, then this result indicates that the measurements $\mathbf{z}_k$ belong to a population with larger covariance than represented by $\mathbf{S}_k$. In practice, this means that the observed process may no longer be represented accurately by the models that assume the landmark is stationary. In this case, the null hypothesis can be rejected in favour of the alternate hypothesis H_A at the $100(1-\alpha)\%$ confidence level.

The primary difficulty with this approach is that a measurement may result in rejection of H_0 due to reasons other than a landmark making a transition from stationary to moving. An outlier measurement can cause $D^2(\mathbf{z}_k, \hat{\mathbf{z}}_k^-) > \gamma$, for example. Moreover, the use of the Extended Kalman Filter for the nonlinear SLAM problem means that the Gaussian assumptions are only approximate. In practice the hypothesis test stated above can be augmented with hysteresis; the decision to transition a landmark from stationary to moving may only be made after $D^2(\mathbf{z}_k, \hat{\mathbf{z}}_k^-) > \gamma$ for more than a specified number of consecutive measurements.

In order to reduce computational complexity in tracking moving landmarks, it is desirable to detect when a moving landmark has become stationary and could therefore be removed from the MLF and added to the SLAM filter. A simple approach implemented in Chapter 4 is to monitor the velocity estimate of each moving landmark. If this velocity estimate falls below a user-defined threshold (v_{min}) for more that a consecutive user-defined number of timesteps, then the landmark is assumed to be stationary and can be moved to the SLAM filter.

2.6 Parameter Selection

A simulation environment was constructed using MatLab r2008b and executed on a Dell T3400 Workstation. In practice, parameter selection is a challenging task in system modeling and calibration. For the purposes of this thesis, following many simulation runs, parameters that appear in the above models were selected to be representative of typical vehicles and sensors, but some are set to emphasize noise in the system in order to clearly demonstrate performance in a relatively large noisy environment. All simulations were conducted with a sample period of

$$T = 0.05\,\text{sec.} \tag{2.111}$$

In the discrete-time formulation, process noise covariance depends on this sample period in that the covariance of discrete-time noise represents errors accumulated during one sample period.

The process noise covariance, $\mathbf{Q}^v$, for the simple unicycle model defined in (2.8) has covariance entries

$$\begin{aligned} \sigma_{x_k^v} &= 0.01\,\text{m}, \\ \sigma_{y_k^v} &= 0.01\,\text{m}, \\ \sigma_{\theta_k^v} &= 0.04\,\text{rad}. \end{aligned} \tag{2.112}$$

For stationary landmarks, the process covariance $\mathbf{Q}^{s(i)}$ defined in (2.24) has covariance entires

$$\begin{aligned} \sigma_{x^{s(i)}} &= 0.03\,\text{m}, \\ \sigma_{y^{s(i)}} &= 0.03\,\text{m}. \end{aligned} \tag{2.113}$$

For moving landmarks, the parameters of Singer's process noise model defined in

(2.33) are

$$\begin{aligned} \alpha_m &= 0.5\,\text{sec}^{-1}, \\ \sigma_m &= 0.01\,\text{m/sec}^2. \end{aligned} \tag{2.114}$$

The measurement noise covariance matrices, $\mathbf{R}^{s(i)}$ in (2.44) for stationary landmarks and $\mathbf{R}^{m(i)}$ in (2.47) for moving landmarks, have the same covariance entries,

$$\begin{aligned} \sigma_r &= 0.02\,\text{m}, \\ \sigma_b &= 0.05\,\text{rad}. \end{aligned} \tag{2.115}$$

For simulation results shown in Chapters 3 and 4, the initial covariance matrix is defined as diagonal with reasonably large diagonal entries. For the SLAM filter that includes states of the vehicle and stationary landmarks, the initial covariance is set to

$$\mathbf{P}_0^{vs} = 5\mathbf{I}_{N^v+2N_0^s}, \tag{2.116}$$

where $N^v = 3$ for the simple unicycle model and N_0^s is the initial number of stationary landmarks. For the moving landmark filter, the initial covariance for each moving landmark is set to

$$\mathbf{P}_0^m = \begin{bmatrix} 1^2 & 0 & 0 \\ 0 & 0.5^2 & 0 \\ 0 & 0 & 0.005^2 \end{bmatrix}. \tag{2.117}$$

2.7 Summary

This chapter introduces notation for the SLAM problem in the context of state estimation. Two different types of unicycle models and a range-and-bearing sensor are developed to represent the mobile sensing system. Stationary landmarks are modeled with some uncertainty in motion in order to detect possible transitions from stationary to moving. Landmark motion is modeled using Singer's approach that is well-known in the target tracking literature. Background theoretical information is

provided on the important tools used in this thesis. The Kalman Filter is an optimal predictor-corrector algorithm that forms the basis for the Extended Kalman Filter that permits a SLAM state estimation algorithm in a recursive on-line implementation. State estimation requires that the system, as posed, satisfies observability conditions in order for information to be accumulated in the filter. The concept of indistinguishability is used in Chapter 3 to illustrate that the discrete-time nonlinear 2D SLAM problem as posed is not observable. Fisher's Information Matrix and the Cramer-Rao Lower Bounds are important tools in state estimation to assess both observability and accuracy. One limitation of the bound calculation is that covariance bounds are derived for deterministic trajectories in state space and do not include uncertainty in the vehicle motion over each timestep. Hypothesis testing based on the Mahalanobis distance is reviewed for detection of landmark transitions from stationary to moving, whereas a threshold test on velocity estimates is used to determine when a landmark transitions from moving to stationary. The set of model parameters used in simulation studies of Chapters 3 and 4 is selected to be representative of SLAM systems in noisy environments.

Chapter 3

Observability and Accuracy in SLAM

Chapter 2 defined the autonomous SLAM problem and reviewed theoretical background needed to investigate practical solutions for this problem. In this Chapter, the issue of observability is first addressed in Section 3.1 to show that the autonomous SLAM problem as formulated is not observable and has three degrees of freedom. Four common approaches to the unobservable problem are examined through simulations with a simple static three-landmark environment. Section 3.2 gives the proposed solution for observability of the autonomous problem and shows initial simulation results and Cramer-Rao Lower Bounds for the simple environment.

The proposed solution is shown to be locally observable and can be initialized autonomously with the methods developed in Section 3.3. The issue of accuracy in large environments is treated in Section 3.4 and suggests the possible requirement to re-select special landmarks that form the observability constraints on the system.

Addition and deletion of landmarks in the EKF filters is treated in Section 3.5. Re-selection of the special landmarks, however, has implications on observability, bias in the filter, and filter accuracy, and these are discussed in Section 3.6. All simulations in this Chapter use a simple static three-landmark environment. Further analysis of the proposed methods in large dynamic environments is conducted with simulation studies in Chapter 4.

3.1 Observability in Autonomous SLAM

The autonomous SLAM problem is developed in Section 2.1. Observability is a key condition that must be met for a filter to retain information and maintain or improve the state estimate over time. Observability for the two-dimensional continuous-time SLAM problem has been analysed in [6] and [29]. In [6] it was proposed, incorrectly, that if the coordinates of one landmark relative to the world reference frame were known, then the problem becomes observable. In [29] it was shown that knowledge of coordinates for one landmark is not sufficient, but that knowledge of the world-centric coordinates of two landmarks results in observability. However, knowledge of the coordinates of two landmarks requires external sensing of the environment and hence this does not yield a solution to the autonomous SLAM problem.

3.1.1 Observability Problem

The reason that the autonomous SLAM problem is not observable is that the problem definition does not fix the world centric frame in the environment. It is then impossible to derive a consistent estimate of world-centric vehicle and landmark positions. The following example, summarized from [3], examines why the system is unobservable. This example uses a simplification of the vehicle dynamics with $\mathbf{p}_k^v = [x_k^v, y_k^v, \theta_k^v]$ representing the vehicle pose, and the state vector

$$\mathbf{x}_k^v = \begin{bmatrix} \mathbf{p}_k^v \\ \dot{\mathbf{p}}_k^v \end{bmatrix}. \tag{3.1}$$

With the sample period T, the discrete-time vehicle motion model is written as

$$\mathbf{p}_{k+1}^v = \mathbf{p}_k^v + T\dot{\mathbf{p}}_k^v. \tag{3.2}$$

To simplify the notation, the input for the vehicle is not explicitly identified in the model, however, the derivative $\dot{\mathbf{p}}_k^v$ may, in this example, take on different values at each sample period so that the vehicle may roam through the environment with a representative control law.

The environment contains N^s stationary landmarks whose world-centric positions, with $i = 1, 2, \ldots, N^s$, are

$$\mathbf{x}^{s(i)} = \begin{bmatrix} x^{s(i)} \\ y^{s(i)} \end{bmatrix}. \tag{3.3}$$

The constant landmark state vector is

$$\mathbf{x}^s = \left[\mathbf{x}^{s(1)^T}, \ldots, \mathbf{x}^{s(N^s)^T}\right]^T. \tag{3.4}$$

The measurements at each sample period are the range $r_k^{s(i)}$ and bearing $b_k^{s(i)}$ of each landmark relative to the vehicle reference frame. These polar measurements are transformed, for the purposes of this example, to Cartesian coordinates relative to the vehicle using the transformation

$$\begin{aligned} \Delta x_k^{s(i)} &= r_k^{s(i)} \cos b_k^{s(i)} \\ \Delta y_k^{s(i)} &= r_k^{s(i)} \sin b_k^{s(i)}. \end{aligned} \tag{3.5}$$

The transformed measurements in (3.5) are related to the vehicle and landmark state vectors according to

$$\begin{bmatrix} \Delta x_k^{s(i)} \\ \Delta y_k^{s(i)} \end{bmatrix} = \mathbf{C}(\theta_k^v) \left\{ \begin{bmatrix} x^{s(i)} \\ y^{s(i)} \end{bmatrix} - \begin{bmatrix} x_k^v \\ y_k^v \end{bmatrix} \right\}. \tag{3.6}$$

where

$$\mathbf{C}(\gamma) = \begin{bmatrix} \cos\gamma & \sin\gamma \\ -\sin\gamma & \cos\gamma \end{bmatrix} \tag{3.7}$$

is an orthogonal rotation matrix.

Define an arbitrary transformation vector

$$\mathbf{T} = [T_x, T_y]^T, \tag{3.8}$$

and an arbitrary angle β and rotation matrix $\mathbf{C}(\beta)$. Now let

$$\tilde{\mathbf{x}}_k^v = \left[\tilde{x}_k^v, \tilde{y}_k^v, \tilde{\theta}_k^v, \dot{\tilde{x}}_k^v, \dot{\tilde{y}}_k^v, \dot{\tilde{\theta}}_k^v\right]^T, \tag{3.9}$$

and

$$\begin{aligned} \bar{\mathbf{x}}_k^v &= \left[\bar{x}_k^v, \bar{y}_k^v, \bar{\theta}_k^v, \dot{\bar{x}}_k^v, \dot{\bar{y}}_k^v, \dot{\bar{\theta}}_k^v\right]^T \\ &= \begin{bmatrix} \mathbf{C}(\beta)\begin{bmatrix} \tilde{x}_k^v \\ \tilde{y}_k^v \end{bmatrix} + \mathbf{T} \\ \tilde{\theta}_k^v - \beta \\ \mathbf{C}(\beta)\begin{bmatrix} \dot{\tilde{x}}_k^v \\ \dot{\tilde{y}}_k^v \end{bmatrix} \\ \dot{\tilde{\theta}}_k^v \end{bmatrix}. \end{aligned} \tag{3.10}$$

Also let

$$\tilde{\mathbf{x}}^{s(i)} = \left[\tilde{x}^{s(i)}, \tilde{y}^{s(i)}\right]^T \tag{3.11}$$

and

$$\bar{\mathbf{x}}^{s(i)} = \mathbf{C}(\beta)\tilde{\mathbf{x}}^{s(i)} + \mathbf{T}. \tag{3.12}$$

Under the terminology reviewed in Section 2.3, the following development [3] shows that, with the available measurements, the initial state $(\bar{\mathbf{x}}_0^v, \bar{\mathbf{x}^s})$ is indistinguishable from the initial state $(\tilde{\mathbf{x}}_0^v, \tilde{\mathbf{x}^s})$.

For each landmark measurement $i = 1, 2, \ldots, N^s$,

$$
\begin{aligned}
\begin{bmatrix} \overline{\Delta x}_k^{s(\imath)} \\ \overline{\Delta y}_k^{s(\imath)} \end{bmatrix} &= \mathbf{C}(\bar{\theta}_k^v) \left\{ \begin{bmatrix} \bar{x}^{s(\imath)} \\ \bar{y}^{s(\imath)} \end{bmatrix} - \begin{bmatrix} \bar{x}_k^v \\ \bar{y}_k^v \end{bmatrix} \right\} \\
&= \mathbf{C}(\bar{\theta}_0^v + T\sum_{\jmath=1}^{k} \dot{\bar{\theta}}_\jmath^v) \cdot \\
&\quad \left\{ \begin{bmatrix} \bar{x}^{s(\imath)} \\ \bar{y}^{s(\imath)} \end{bmatrix} - \left[\begin{bmatrix} \bar{x}_0^v \\ \bar{y}_0^v \end{bmatrix} + T\sum_{\jmath=1}^{k} \begin{bmatrix} \dot{\bar{x}}_\jmath^v \\ \dot{\bar{y}}_\jmath^v \end{bmatrix} \right] \right\} \\
&= \mathbf{C}(\tilde{\theta}_0^v - \beta + T\sum_{\jmath=1}^{k} \dot{\tilde{\theta}}_\jmath^v) \cdot \\
&\quad \left\{ \mathbf{C}(\beta) \begin{bmatrix} \tilde{x}^{s(\imath)} \\ \tilde{y}^{s(\imath)} \end{bmatrix} + \mathbf{T} - \left[\mathbf{C}(\beta) \begin{bmatrix} \tilde{x}_0^v \\ \tilde{y}_0^v \end{bmatrix} + \mathbf{T} + T\mathbf{C}(\beta) \sum_{\jmath=1}^{k} \begin{bmatrix} \dot{\tilde{x}}_\jmath^v \\ \dot{\tilde{y}}_\jmath^v \end{bmatrix} \right] \right\} \\
&= \mathbf{C}(\tilde{\theta}_k^v - \beta)\mathbf{C}(\beta) \left\{ \begin{bmatrix} \tilde{x}^{s(\imath)} \\ \tilde{y}^{s(\imath)} \end{bmatrix} - \begin{bmatrix} \tilde{x}_k^v \\ \tilde{y}_k^v \end{bmatrix} \right\} \\
&= \begin{bmatrix} \widetilde{\Delta x}_k^{s(\imath)} \\ \widetilde{\Delta y}_k^{s(\imath)} \end{bmatrix}. \qquad (3.13)
\end{aligned}
$$

Note that the last equality in (3.13) follows from the fact that $\mathbf{C}(\alpha-\beta) = \mathbf{C}(\alpha)\mathbf{C}(-\beta)$ and $\mathbf{C}(-\beta) = [\mathbf{C}(\beta)]^{-1}$. The result in (3.13) shows that for all landmarks $i = 1, 2, \ldots, N^s$ and for all times k, the sequence of measurements cannot distinguish between $(\bar{\mathbf{x}}_0^v, \bar{\mathbf{x}}^s)$ and $(\tilde{\mathbf{x}}_0^v, \tilde{\mathbf{x}}^s)$ under the arbitrary transformation $\mathbf{T}$ and rotation $\mathbf{C}(\beta)$, and hence the system is not observable. This example also demonstrates that there are three degrees of freedom (two in $\mathbf{T}$ and one in β) that cause the system to be unobservable.

3.1.2 Common Approaches for the Unobservable Problem

Several strategies have been proposed in the literature to implement autonomous SLAM. In this Section, four such strategies are tested in a simulation environment

to investigate performance. The environment is created as illustrated in Figure 3.1 with a stationary vehicle and three stationary landmarks, $L1$, $L2$ and $L3$. The simulated vehicle uses the simple unicycle model of Section 2.1.1, but with zero speed and turning rate inputs. A stationary vehicle removes vehicle motion in order to focus on estimation performance based primarily on filter initialization and the assumed measurement model.

The four test cases are as follows:

- **Case 3.1**: A common approach is to do nothing, or to include no information of how the world-centric frame is fixed in the environment.

- **Case 3.2**: Very often, it is assumed that the vehicle pose at time $t = 0$ is known *exactly* with respect to the world-centric frame. This often results in the initialization $\mathbf{p}_0^v = [0, 0, 0]^T$. The question is whether absolute knowledge of the pose at time $t = 0$ does anything to help with the unobservability problem.

- **Case 3.3**: In [6] it was proposed that if the world-centric coordinates of one landmark are known, then the problem becomes observable. The results for this Case demonstrate that this conclusion is incorrect, but they are included here to investigate possible improvements over Cases 3.1 and 3.2.

- **Case 3.4**: In [29] it was shown that knowledge of the world-centric coordinates of two landmarks results in observability. Having two known landmark positions results in four constraints; two coordinates for each landmark. These known coordinates are removed from the landmark state vector and are treated as known parameters in the measurement model for these two special landmarks.

Case 3.1: Ignore unobservability.

Simulation studies were conducted using the model parameters listed in Section 2.6 with the environment illustrated in Figure 3.1 for a stationary vehicle. Fisher's Information Matrix in (2.102) and results from execution of the SLAM EKF were recorded. All coordinates of the three landmarks, $L1$, $L2$, and $L3$, are contained

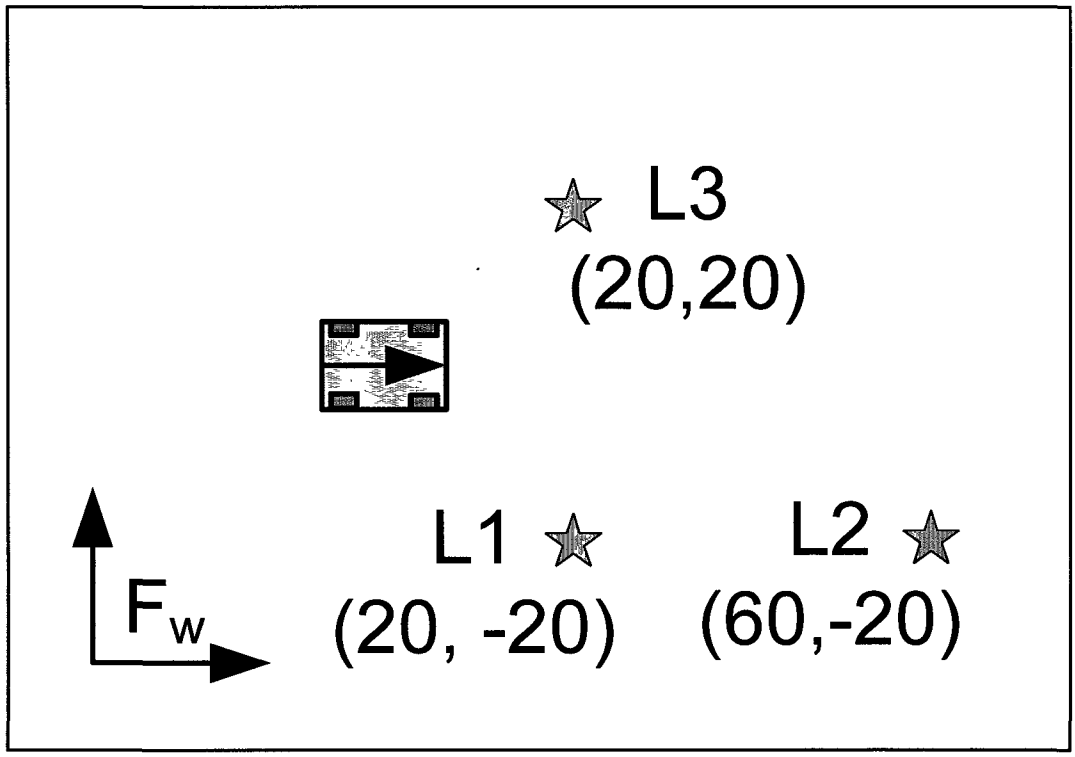

Figure 3.1: SLAM environment for testing observability.

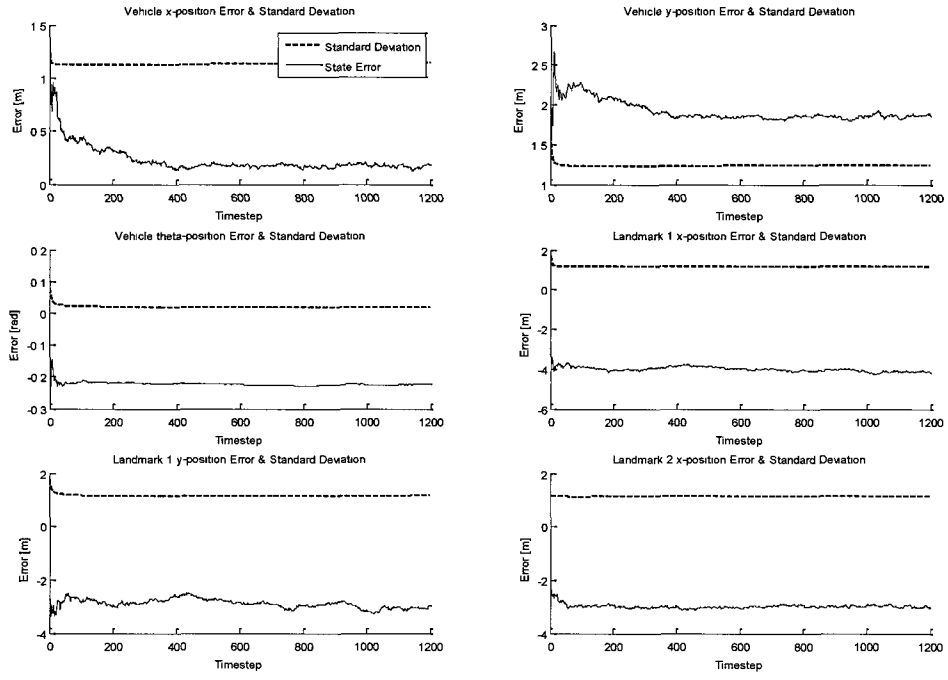

Figure 3.2: Case 3.1: Unobservable SLAM.

within the landmark state vector $\mathbf{x}_k^s$. As reviewed in Section 2.4, if Fisher's Information Matrix remains singular in the absence of prior information (i.e. $\mathbf{P}_0^{-1} = \mathbf{0}$ in (2.102)), then the system is unobservable. Fisher's Information Matrix for the system in this Case has dimensions 9×9. Using a sample period of $T = 0.05$sec., the recursive calculation of Fisher's Information Matrix was executed for 1200 timesteps. In this Case, Fisher's Information Matrix remains singular throughout the run with **three** zero singular values. This result supports the findings in Section 3.1.1 that the unobservable subspace for the system has dimension three.

Simulation results of the SLAM EKF for this Case are shown in Figure 3.2. These results demonstrate that the filter cannot correct initialization errors. In this case of no vehicle motion, the Kalman covariance reaches steady state, but all state estimates show convergence to biased results towards the end of the simulation.

Case 3.2: Absolute knowledge of initial vehicle position.

If the initial vehicle pose $\mathbf{p}_0^v$ is known with great certainty, the initial information in vehicle pose is infinite, while the initial covariance of vehicle pose is zero. This Case was simulated to compute Fisher's Information Matrix based on an initialization

$$\mathbf{P}_0^{-1} = \left[\begin{array}{c|c} \mathbf{P}_0^{v-1} & \mathbf{0}_{3\times 6} \\ \hline \mathbf{0}_{6\times 3} & \mathbf{P}_0^{s-1} \end{array}\right], \tag{3.14}$$

where $\mathbf{P}_0^{v-1} = 10^6\mathbf{I}_3$ and $\mathbf{P}_0^{s-1} = \mathbf{0}_{6\times 6}$. Fisher's Information Matrix in this Case has dimensions 9×9 and, after 1200 timesteps with $T = 0.05$sec., Fisher's Information Matrix remains singular throughout the run again with **three** zero singular values.

The SLAM EKF was executed for this Case with Kalman covariance initialization

$$\mathbf{P}_0 = \left[\begin{array}{c|c} \mathbf{P}_0^{v} & \mathbf{0}_{3\times 6} \\ \hline \mathbf{0}_{6\times 3} & \mathbf{P}_0^{s} \end{array}\right], \tag{3.15}$$

where $\mathbf{P}_0^v = 10^{-6}\mathbf{I}_3$ and $\mathbf{P}_0^s = 5\mathbf{I}_6$. Simulation results are shown in Figure 3.3 for the vehicle pose, landmark $L1$ x- and y-position, and $L2$ x-position ($L3$ y-position and $L3$ coordinates not shown for brevity). As expected, even with highly accurate knowledge of the initial vehicle pose, estimation errors show convergence to biased results due to unobservability.

Case 3.3: Absolute knowledge of one landmark position.

In [6], it was proposed that if the world-centric position of a single landmark is known, then the system becomes observable. In this Case, the position of landmark $L1$ in Figure 3.1 is assumed to be known. Vehicle-relative measurements are still available for this landmark, however, the coordinates of this landmark are removed from the state vector and treated as known parameters in the system. Over 1200 timesteps using $T = 0.05$sec, Fisher's Information Matrix is now 7×7 and, when initialized with $\mathbf{P}_0^{-1} = \mathbf{0}$, remains singular but with only **one** zero singular value. Knowledge of one landmark position results in two constraints on the system and the unobservable subspace is reduced to dimension one.

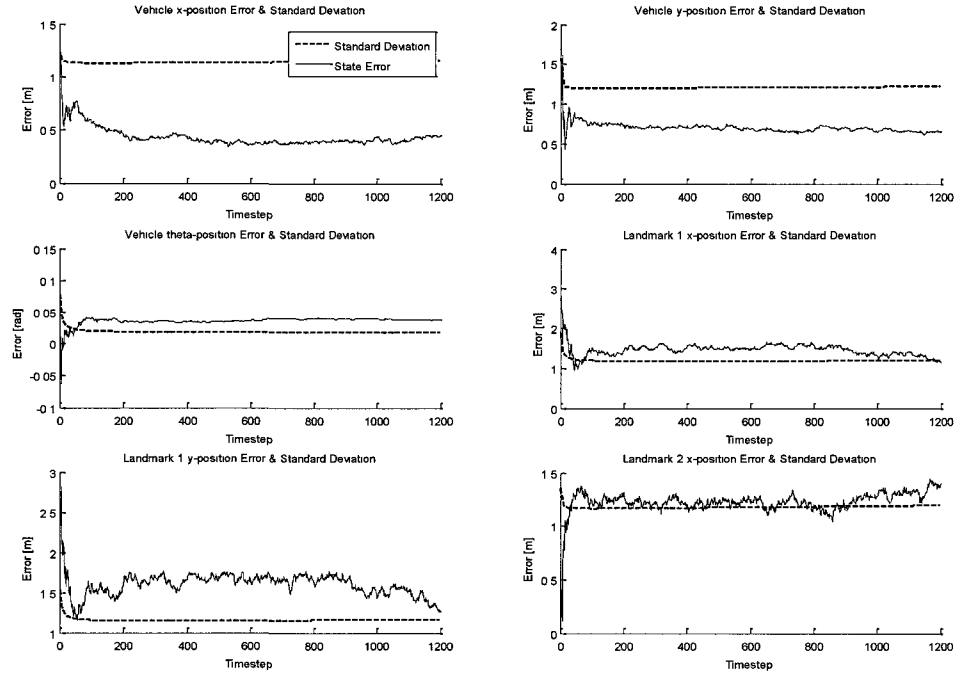

Figure 3.3: Case 3.2: Unobservable SLAM with absolutely known vehicle starting pose.

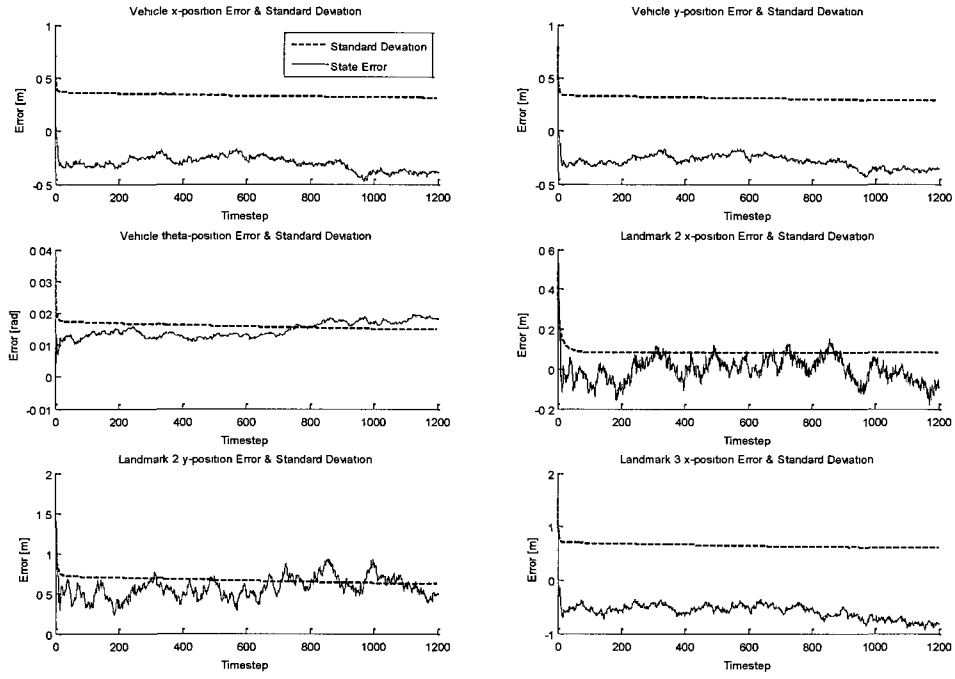

Figure 3.4: Case 3.3: Unobservable SLAM with a single fixed landmark.

The SLAM EKF was executed for this Case with all parameters set to values listed in Section 2.6. Results of this simulation are shown in Figure 3.4 for vehicle pose, $L2$ position, and $L3$ x-position ($L3$ y-position not shown). In comparison to the results of Case 3.1, the Kalman covariance is significantly reduced for the vehicle x- and y-position. However, the filter is still not able to correct for initialization errors and results in convergence to biased results.

Case 3.4: Absolute knowledge of two landmark positions.

In [29], the world-centric positions of two landmarks are assumed to be known, and this results in four constraints on the system. In this Case the positions of landmarks $L1$ and $L2$ in Figure 3.1 are assumed to be known. These landmarks are removed from the state vector and the four coordinates are treated as fixed parameters. This

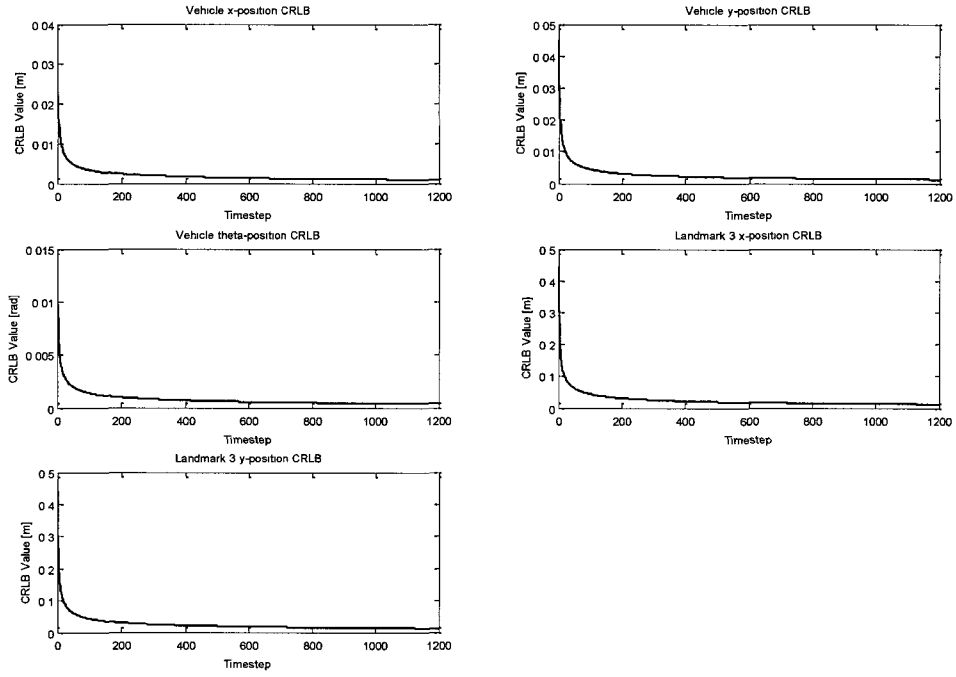

Figure 3.5: Case 3.4: Cramer Rao Lower Bounds with 2 fixed landmarks.

Case results, starting from an initialization $\mathbf{P}_0^{-1} = \mathbf{0}$, in a nonsingular Fisher's Information Matrix. Figure 3.5 shows the Cramer Rao Lower Bounds (CRLBs) over 1200 timesteps with $T = 0.05$sec. The CRLBs for this case of a stationary vehicle decrease monotonically over the run indicating that information is being accumulated in the system.

The SLAM EKF was executed for this Case with all parameters set to values listed in Section 2.6. Results of this simulation are shown in Figure 3.6 for vehicle pose and $L3$ position. These simulation results demonstrate good estimation performance in that the state errors are consistent with the Kalman standard deviations. Note that the Kalman standard deviation values lie somewhat above the CRLBs due to the uncertainty in vehicle and landmark motion modeled in the filter (represented by the covariance matrices $\mathbf{Q}^v$ and $\mathbf{Q}_k^s$). As discussed in Section 2.4, the CRLBs are

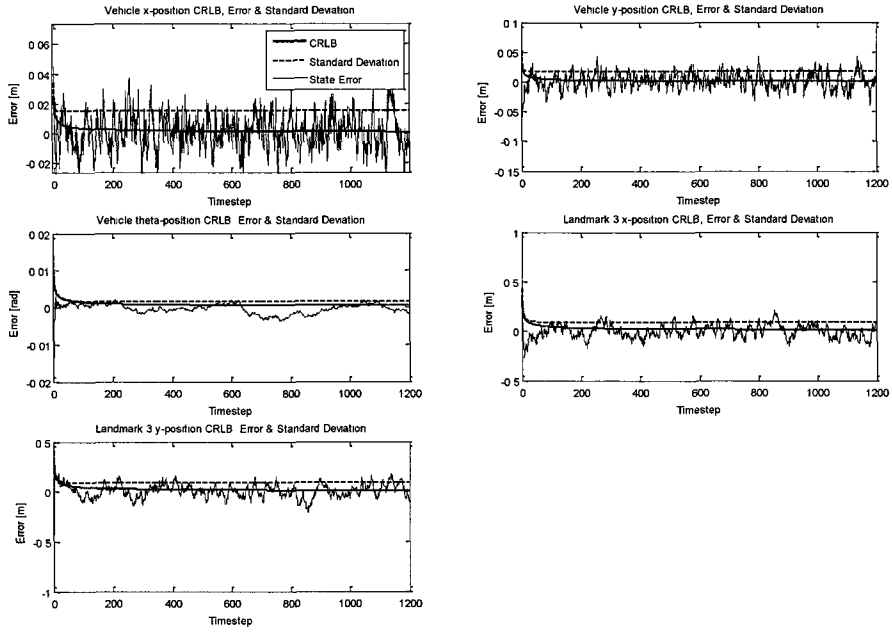

Figure 3.6: Case 3.4: SLAM EKF results with two fixed landmarks.

computed about a known deterministic trajectory, so these bounds do not include uncertainty in the dynamic models.

3.2 Observable Autonomous SLAM

Section 3.1.1 provided an example to conclude that there are three degrees of freedom in the 2D planar autonomous SLAM problem. The results shown in Cases 3.1 and 3.2 of Section 3.1.2 support this conclusion based on a null space of dimension three in Fisher's Information Matrix. If one landmark position is known, as in Case 3.3, the dimension of Fisher's null space is reduced to one. The assumption of two known landmark positions gives four constraints in Case 3.4 and represents an over-constrained system for which prior external measurements of world-centric positions of these landmarks must be available.

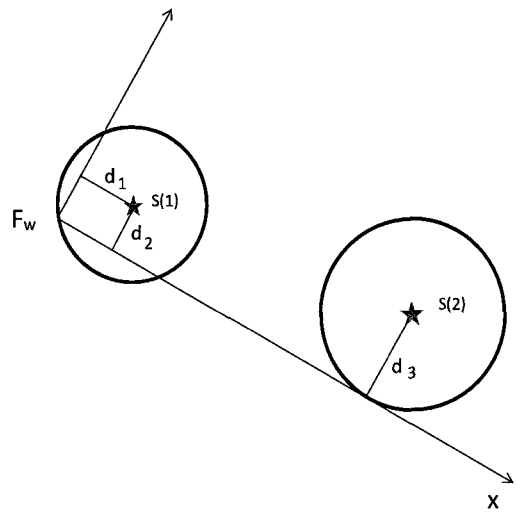

Figure 3.7: Fixing the world-centric reference frame for observable SLAM.

Following the methods in [2] [4], it is possible to specify three constraints on the SLAM system to satisfy local observability without the need for prior external measurements. This approach leads to a locally observable formulation of the autonomous SLAM problem.

Before initializing the SLAM filter, an initial survey of the environment can be conducted with a stationary vehicle. In order to fix the world-centric reference frame in the SLAM problem, as illustrated in Figure 3.7, two stationary landmarks, $s(1)$ and $s(2)$, are selected. Although not investigated within the scope of this thesis, initial identification of stationary vs moving landmarks could be implemented using the motion/stop detection of Section 2.5 and a simple linear kalman filter based solely on the range and bearing measurements with respect to the stationary vehicle frame of reference. Two special stationary landmarks are selected to form the three observability constraints without the need for prior measurements. In this thesis, $s(1)$ and $s(2)$ always identify the special landmarks that form the three observability constraints.

In order to fix a landmark-centric world reference frame, F_w, in the environment, select two stationary landmarks and three constants $\{d_1, d_2, d_3\}$. Three coordinates

of these two special landmarks, from the four coordinates available, are fixed to form the three constraints needed to give local observability. Two constraints define the position of $s(1)$,

$$\mathbf{x}^{s(1)} = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}, \tag{3.16}$$

and the third constraint fixes either the x-coordinate or the y-coordinate of $\mathbf{x}^{s(2)}$ in F_w, but leaves the other coordinate of $\mathbf{x}^{s(2)}$ to be estimated by the SLAM filter. In the following discussions, it is assumed that the y-coordinate of $\mathbf{x}^{s(2)} = [x^{s(2)}, y^{s(2)}]^T$ is fixed,

$$y^{s(2)} = d_3^y. \tag{3.17}$$

In this case the x-coordinate, $x^{s(2)}$, of $s(2)$ is included in the state vector $\mathbf{x}_k^s$ for estimation, but both coordinates of $s(1)$ and the y-coordinate of $s(2)$ are removed from $\mathbf{x}_k^s$ and treated as known parameters based on the specified values of $\{d_1, d_2, d_3^y\}$. This restriction is removed in later Sections of this thesis.

The state vector for $N^s \geq 2$ stationary landmarks now is of dimension $2N^s - 3$,

$$\mathbf{x}^s = \left[x^{s(2)}, \mathbf{x}^{s(3)^T}, \mathbf{x}^{s(4)^T}, \ldots, \mathbf{x}^{s(N^s)}\right]^T. \tag{3.18}$$

These three coordinates are removed from the process model for stationary landmarks along with the corresponding rows and columns of the process noise covariance $\mathbf{Q}_k^s$. The values of $\{d_1, d_2, d_3^y\}$ are used as known parameters in the measurement model and hence the corresponding columns are removed from the measurement model Jacobian.

It is important to note that d_1 and d_2 may be arbitrarily selected but, once d_2 has been selected, admissible values of d_3^y are bounded. The quantity $|d_2 - d_3^y|$, analysed further in Section 3.3 for autonomous initialization, represents the vertical (y-axis) separation of $s(1)$ and $s(2)$ in F_w. This separation is not admissible if it is greater than the true distance between landmarks $s(1)$ and $s(2)$; in this case the solution is complex and the EKF estimates cannot be consistent with the true environment. This problem has important implications when re-selection of special landmarks may be needed, a topic treated in Section 3.6. The true distance between $s(1)$ and $s(2)$

is not known a priori in the autonomous SLAM problem.

In practice, a natural choice to fix the world-centric frame is $\{d_1, d_2, d_3^y\} = \{0, 0, 0\}$ in which case landmark $s(1)$ lies at the origin and landmark $s(2)$ lies on the x-axis, but the position of $s(2)$ along the x-axis is estimated by the SLAM EKF. These three constraints lead to two distinct solutions for the world reference frame F_w. The two solutions are related by a rotation of F_w by 180 degrees. One solution places $s(2)$ on the positive x-axis, while the second solution places $s(2)$ on the negative x-axis. Strictly speaking, these two solutions are indistinguishable and hence the system is not globally observable. However, each of these solutions is locally observable. A measure of separation between these two indistinguishable solutions for is $2|x^{s(2)} - d_1|$, but this quantity is not known in advance for the autonomous problem. Provided the filter is initialized reasonably close to one of these solutions, and the distance between these solutions is sufficient, the filter will track that solution. If $|x^{s(2)} - d_1|$ is very small, these two solutions are very close together. In the limit, as $|x^{s(2)} - d_1| \to 0$, the two solutions become one and the system then becomes "essentially unobservable". This issue is examined further in Section 3.6.

In simulation studies of this thesis, the entire environment including the vehicle is maintained relative to a pre-defined simulation reference frame. In keeping with the recommendations for practical implementation in the preceding paragraph, simulations initially select two landmarks, $s(1)$ and $s(2)$ that are parallel to the x-axis of the simulation reference frame. In order to compare state estimates to true values in the simulation frame, the initial observability constraints are formed with $[d_1, d_2]^T$ set equal to the true simulated value of $\mathbf{x}^{s(1)}$, and d_3^y is set equal to the true simulated value of the y-coordinate of $\mathbf{x}^{s(2)}$. Under these constraints in simulation studies, and if the filter is operating correctly, the estimated states should reasonably approximate the true simulated states. The simulation environment shown in Figure 3.1 is used in Case 3.5 for initial evaluation of the proposed method.

Case 3.5: Three known landmark coordinates.

This Case simulates the same environment as used in Cases 3.1 through 3.4 with a stationary vehicle and three stationary landmarks. The special landmarks are

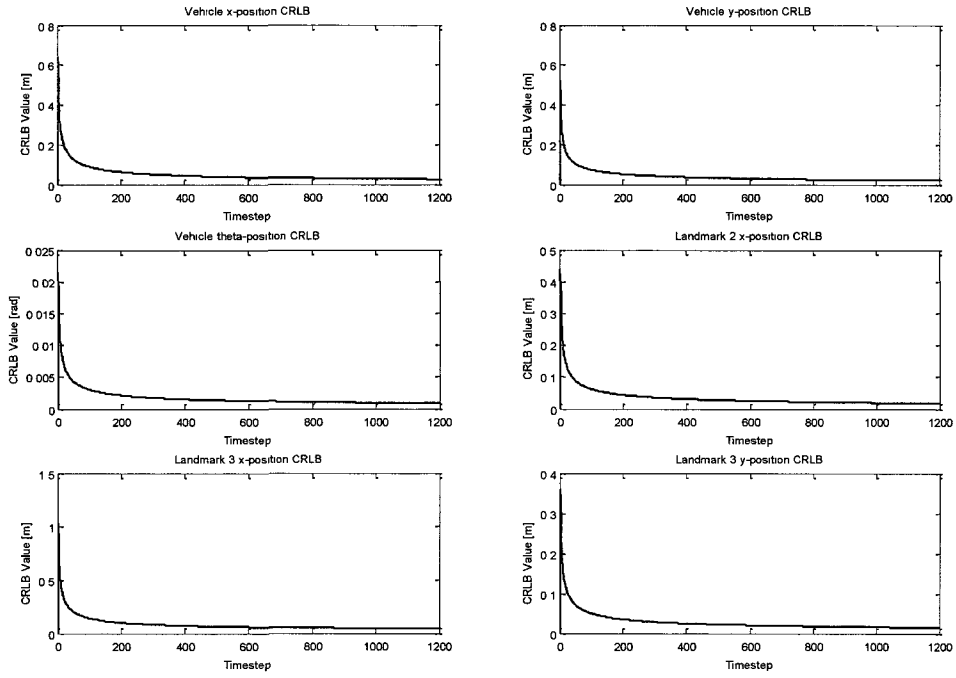

Figure 3.8: Case 3.5: CRLBs for SLAM with three observability constraints.

$s(1) = L1$ and $s(2) = L2$. The x-coordinate of $L2$ and both coordinates of $L3$ are estimated by the SLAM EKF. In the absence of prior information, $\mathbf{P}_0^{-1} = \mathbf{0}$, Fisher's Information Matrix is nonsingular and produces the CRLBs shown in Figure 3.8. These results demonstrate that it is possible to accurately estimate all states within the filter.

Figure 3.9 compares the CRLBs of vehicle pose estimation for Case 3.4 with two fixed landmarks (four constraints) to those for Case 3.5 with three constraints. As expected, the proposed three-constraint solution results in slightly higher CRLBs over the run, but the two methods produce very similar results towards the end of the 60 second simulation. Results for the SLAM EKF in this Case are shown in Figure 3.10. The Kalman standard deviation closely follows the CRLBs over the run. State estimation errors remain reasonably within the Kalman standard

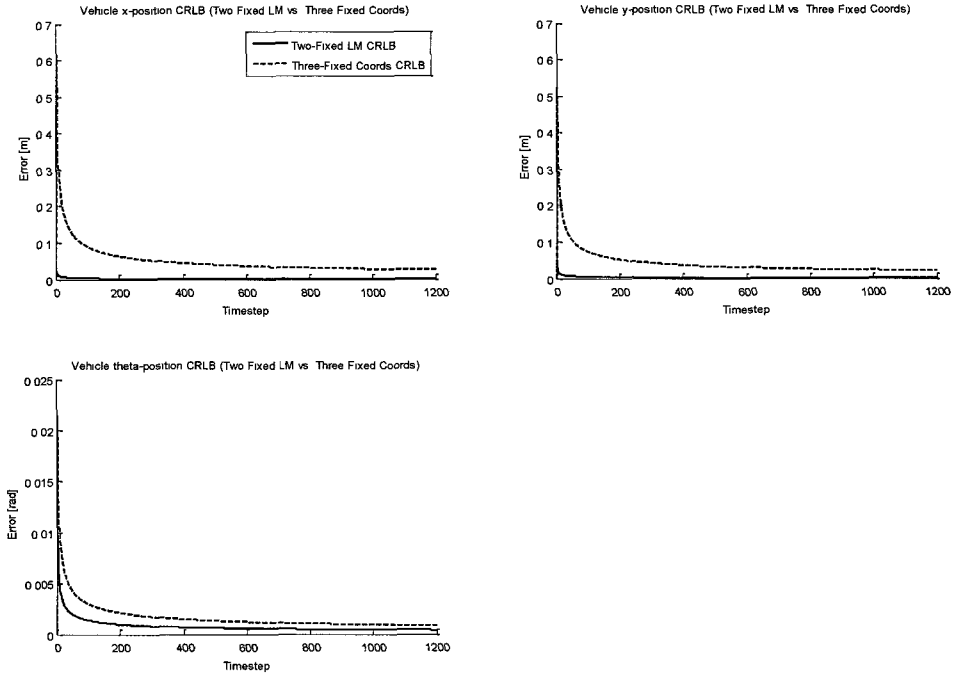

Figure 3.9: Comparison of CRLBs for Case 3.4 and Case 3.5.

deviations which indicate that the filter is performing well. The sample estimation errors for this Case also compare reasonably well with those of of Case 3.4 shown in Figure 3.6.

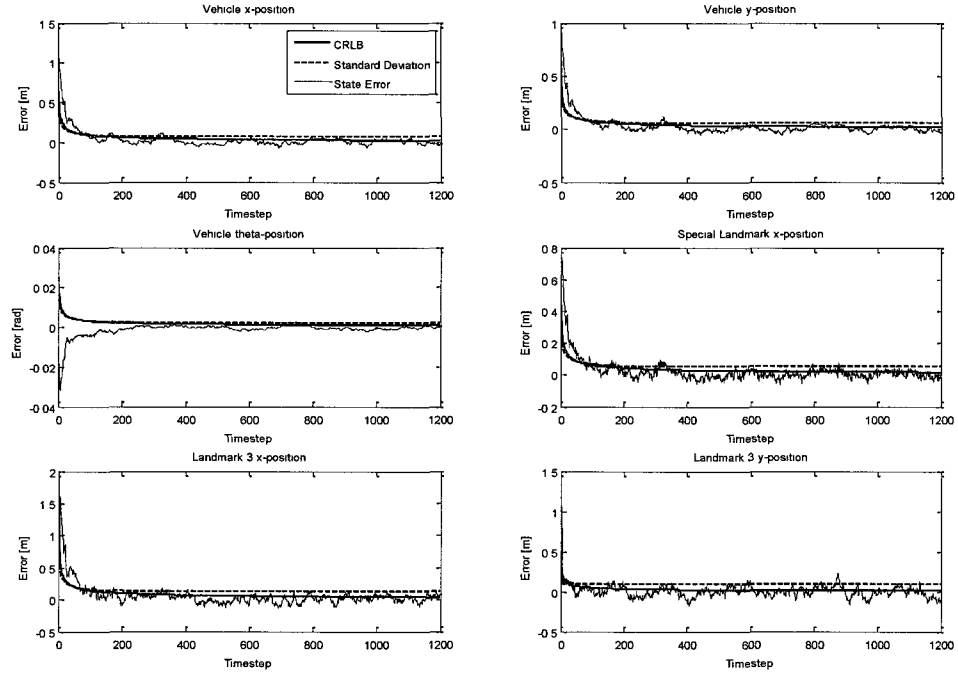

Figure 3.10: Case 3.5: EKF simulation results with three observability constraints.

3.3 Autonomous Initialization

In a practical initialization stage, the vehicle can be held stationary in the environment, and range and bearing measurements for landmarks can be obtained. Although beyond the scope of this thesis, one could employ simple state estimation (a linear Kalman Filter) to smooth initial range and bearing measurements and perform hypothesis testing as described in Section 2.5 to identify stationary landmarks. In the simulations of Chapter 4, initial range and bearing measurements are averaged over a few seconds prior to deriving initial estimates.

The three observability constraints introduced in Section 3.2 result in a world-centric reference frame that is defined relative to two special stationary landmarks $s(1)$ and $s(2)$. Two constraints are formed with $\mathbf{x}^{s(1)} = [d_1, d_2]^T$, while the third constraint, with $\mathbf{x}^{s(2)} = [x^{s(2)}, y^{s(2)}]^T$, can either be formed as $x^{s(2)} = d_3^x$, or $y^{s(2)} = d_3^y$. If one chooses to set $x^{s(2)} = d_3^x$, then the coordinate $y^{s(2)}$ is estimated in the SLAM EKF. On the other hand, if one chooses to set $y^{s(2)} = d_3^y$, then the coordinate $x^{s(2)}$ is estimated in the SLAM EKF.

The parameters $\{d_1, d_2, d_3^x\}$ or $\{d_1, d_2, d_3^y\}$ are specified in advance. Solutions for both parameter sets are derived in this Section in order to support analysis of SLAM accuracy using Fisher's Information and the CRLBs in Section 3.6 when re-selection of the special landmarks may be necessary. The derivation given here is deterministic; for example, estimation of initial values could be done using another EKF to optimize in the presence of sensor noise. The deterministic presentation gives further insight into discussions of Section 3.6. For brevity, the time subscript is dropped and details are given for the parameter set $\{d_1, d_2, d_3^y\}$ for which the y-coordinate of $s(2)$ is specified. Similar results are then stated for the parameter set $\{d_1, d_2, d_3^x\}$ for which the x-coordinate of $s(2)$ is specified. This derivation draws heavily on methods developed in [3].

Initial estimates for the vehicle pose, $\mathbf{p}^v = [x^v, y^v, \theta^v]^T$ and the estimated x-coordinate of $s(2)$, $x^s(2)$, are needed based on range and bearing measurements, $[r_1, b_1]$, from the vehicle to $s(1)$, and the range and bearing, $[r_2, b_2]$, from the vehicle

to $s(2)$. These measurements give four equations,

$$\begin{aligned} d_1 - x^v &= r_1 \cos\left(b_1 + \theta^v\right), && (3.19)\\ d_2 - y^v &= r_1 \sin\left(b_1 + \theta^v\right), && (3.20)\\ x^{s(2)} - x^v &= r_2 \cos\left(b_2 + \theta^v\right), && (3.21)\\ d_3^y - y^v &= r_2 \sin\left(b_2 + \theta^v\right). && (3.22) \end{aligned}$$

Define

$$\begin{aligned} \alpha_x &= r_1 \cos(b_1) - r_2 \cos(b_2), && (3.23)\\ \alpha_y &= r_1 \sin(b_1) - r_2 \sin(b_2), && (3.24)\\ \delta_d^y &= d_2 - d_3^y. && (3.25) \end{aligned}$$

Note that the vector from $s(2)$ to $s(1)$ expressed in the vehicle frame of reference is $[\alpha_x, \alpha_y]^T$. The quantity $|\delta_d^y|$ is the separation of $s(1)$ and $s(2)$ parallel to the y-axis of the world-centric frame F_w. Since length in the vehicle reference frame and length in the world-centric frame are identical, and given that the length of a vector cannot be less than its y-component,

$$\alpha_x^2 + \alpha_y^2 \leq |\delta_d^y|^2, \tag{3.26}$$

otherwise there is no real solution as shown below.

Subtracting (3.22) from (3.20), expanding the $\cos(\cdot)$ and $\sin(\cdot)$ and simplifying gives

$$\delta_d^y = \alpha_y \cos(\theta^v) + \alpha_x \sin(\theta^v). \tag{3.27}$$

Now set

$$\begin{aligned} \cos(\theta^v) &= \xi, && (3.28)\\ \sin(\theta^v) &= \pm\sqrt{1 - \xi^2}. && (3.29) \end{aligned}$$

Substitution into (3.27), re-arranging, and squaring both sides gives the quadratic

$$\left(\alpha_x^2 + \alpha_y^2\right)\xi^2 - 2\delta_d^y \alpha_y \xi + \left(\delta_d^{y2} - \alpha_x^2\right) = 0, \tag{3.30}$$

for which there the two simplified solutions are

$$\cos(\theta^v) = \xi = \frac{\delta_d^y \alpha_y \pm |\alpha_x| \sqrt{\alpha_x^2 + \alpha_y^2 - \delta_d^{y2}}}{\left(\alpha_x^2 + \alpha_y^2\right)}. \tag{3.31}$$

Note that admissible values of δ_d^y do not permit the discriminant $\alpha_x^2 + \alpha_y^2 - \delta_d^{y2}$ to be negative, otherwise the solutions are complex. This observation has implications in special landmark re-selection discussed in Section 3.6.

The quantity $|\alpha_x|$ is the horizontal (x-axis) separation of $s(1)$ and $s(2)$ in the vehicle reference frame. Provided $\alpha_x \neq 0$ (so that $s(1)$ and $s(2)$ are not vertical in the vehicle reference frame) one can compute, from (3.27), for each value of ξ

$$\sin(\theta^v) = \frac{\delta_d^y - \alpha_y \xi}{\alpha_x}, \tag{3.32}$$

and the two possible values of vehicle heading are then given by

$$\theta^v = \operatorname{atan2}\left(\cos(\theta^v), \sin(\theta^v)\right). \tag{3.33}$$

On the other hand, if $\alpha_x = 0$, then the two solutions are given by

$$\cos(\theta^v) = \xi = \frac{\delta_d^y}{\alpha_y}, \tag{3.34}$$

$$\sin(\theta^v) = \pm\sqrt{1 - \xi^2}. \tag{3.35}$$

and the two values of θ^v, provided $\xi \neq 1$, can be obtained from (3.33). In the case that $\alpha_x = 0$ and $\xi = 1$, this means that that $s(1)$ and $s(2)$ are vertical (y-axis) in the world reference frame and there is only one solution. This degenerate case has serious implications for estimation accuracy and is treated in Section 3.6.

For the parameter set $\{d_1, d_2, d_3^x\}$ in which the x-coordinate of $s(2)$ is specified,

similar results can be obtained using the methods above and essentially replacing α_x with α_y and δ_d^y with δ_d^x. With the same definitions as given in (3.28) and (3.29), the solutions for the quadratic are

$$\cos(\theta^v) = \xi = \frac{\delta_d^x \alpha_x \pm |\alpha_y| \sqrt{\alpha_x^2 + \alpha_y^2 - \delta_d^{x2}}}{\left(\alpha_x^2 + \alpha_y^2\right)}. \tag{3.36}$$

Again, admissible values of δ_d^x do not permit the discriminant $\alpha_x^2 + \alpha_y^2 - \delta_d^{x2}$ to be negative.

The quantity $|\alpha_y|$ is the vertical (y-axis) separation of $s(1)$ and $s(2)$ in the vehicle reference frame. Provided $\alpha_y \neq 0$ (so that $s(1)$ and $s(2)$ are not horizontal in the vehicle reference frame) one can compute for each value of ξ

$$\sin(\theta^v) = \frac{\alpha_x \xi - \delta_d^x}{\alpha_y}, \tag{3.37}$$

and the two possible values of vehicle heading are then given by

$$\theta^v = \operatorname{atan2}\left(\cos(\theta^v), \sin(\theta^v)\right). \tag{3.38}$$

On the other hand, if $\alpha_y = 0$, then the two solutions are given by

$$\cos(\theta^v) = \xi = \frac{\delta_d^x}{\alpha_x}, \tag{3.39}$$

$$\sin(\theta^v) = \pm\sqrt{1 - \xi^2}. \tag{3.40}$$

and the two values of θ^v, provided $\xi \neq 1$, can be obtained from (3.38). In the case that $\alpha_y = 0$ and $\xi = 1$, this means that $s(1)$ and $s(2)$ are horizontal (x-axis) in the world reference frame and there is only one solution. Again, this degenerate case has serious implications for estimation accuracy and is treated in Section 3.6.

Section 3.2 recommends an initialization with parameter sets $d_1 = d_2 = 0$, and then a decision of whether the special landmarks $s(1)$ and $s(2)$ will be placed on the y-axis or the x-axis of F_w (this initial decision is arbitrary for the purposes of this thesis) and hence set either $d_3^x = 0$ or $d_3^y = 0$, respectively. If the y-coordinate of

$s(2)$ is fixed, then $\delta_d^y = 0$ and

$$\cos(\theta^v) = \xi = \frac{\pm|\alpha_x|}{\sqrt{\alpha_x^2 + \alpha_y^2}}. \tag{3.41}$$

If the x-coordinate of $s(2)$ is fixed, then $\delta_d^x = 0$ and

$$\cos(\theta^v) = \xi = \frac{\pm|\alpha_y|}{\sqrt{\alpha_x^2 + \alpha_y^2}}. \tag{3.42}$$

The determination of $\sin(\theta^v)$ continues as described above in order to obtain two possible values for the vehicle heading, θ^v, relative to F_w. Either value is a valid selection for the initial orientation of F_w in the 2-dimensional environment.

For each value of θ^v, the remaining values needed for initialization of the vehicle pose can be obtained from (3.19)-(3.20),

$$x^v = d_1 - r_1 \cos(b_1 + \theta^v), \tag{3.43}$$

$$y^v = \mp d_2 - r_1 \sin(b_1 + \theta^v),, \tag{3.44}$$

and the estimated component of $\mathbf{x}^{s(2)}$ can be initialized from (3.21)-(3.22),

$$x^{s(2)} = x^v + r_2 \cos(b_2 + \theta^v), \text{ or} \tag{3.45}$$

$$y^{s(2)} = y^v + r_2 \sin(b_2 + \theta^v). \tag{3.46}$$

In the simulations of Chapter 4, the parameter set $d_1 = d_2 = d_3^y = 0$ is used, and the value of θ^v that gives a positive $x^{s(2)}$ is selected to initialize all other states. The initial positions of the remaining stationary landmarks, $\mathbf{p}^{s(j)}$, $j = 3, 4, \ldots, N_0^s$ are computed as

$$x^{s(j)} = x^v + r_j \cos(b_j + \theta^v), \text{ and} \tag{3.47}$$

$$y^{s(j)} = y^v + r_j \sin(b_j + \theta^v). \tag{3.48}$$

Figure 3.11 shows the landmark-centric world reference frames derived from the initialization algorithm with the parameter set $d_1 = 15$m, $d_2 = 0$m, and $d_3^y = 0$

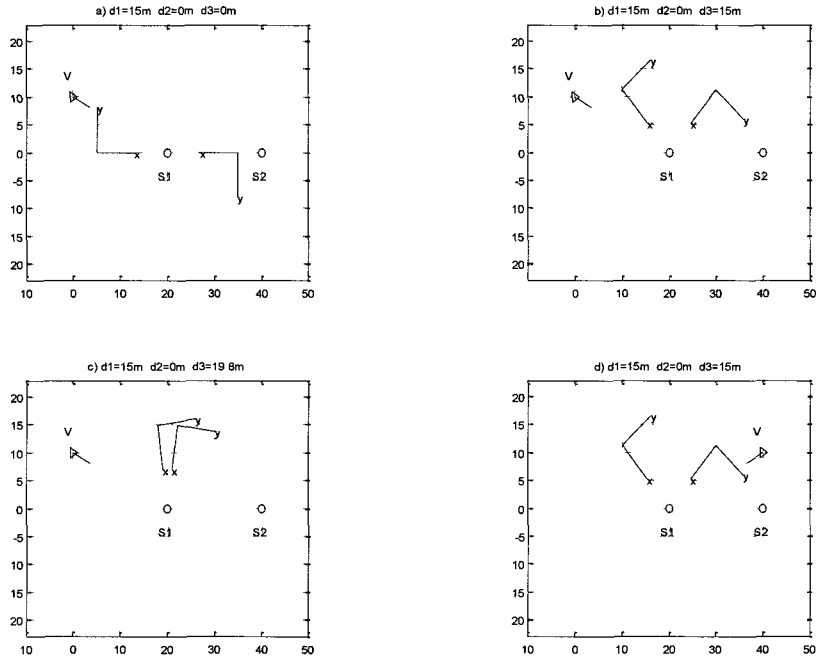

Figure 3.11: World-centric reference frames with three observability constraints.

in plot a), $d_3^y = 15$m in plot b), and $d_3^y = 19.8$m in plot c). Special landmarks $s(1)$ and $s(2)$ are separated in this environment by 20m. In plots a) through c) the vehicle is positioned at $[0, 10]$m in the simulation reference frame with heading $-30°$. Note that in all plots $\mathbf{x}^{s(1)}$ lies on the x-axis of either world frame under the constraint $d_2 = 0$. As d_3^y increases, the distance between the two solutions for the world reference frame decreases, and this has implications on estimation accuracy discussed further in Section 3.6. Plot d) illustrates that the two world reference frames are independent of the initial vehicle pose, now at position $[40, 10]$ with a heading of $-150°$; the two admissible world reference frames in this case are identical to those shown in Plot b).

3.4 Accuracy in Large Environments

In order to implement an observable SLAM formulation as described in Section 3.2, at least two stationary landmarks must be visible with vehicle sensors. In this thesis, the sensor is assumed to provide range and bearing measurements of each landmark relative to the vehicle. Although noise in these measurements is assumed to have constant covariance, the effective Cartesian error due to bearing uncertainty depends on the distance between the vehicle and the landmark as illustrated in Figure 3.12. The position of the landmark relative to the vehicle also determines how the bearing error influences estimation accuracy with respect to the world-centric reference frame. Figure 3.13 illustrates a vehicle surrounded by five landmarks. In this configuration, with the vehicle positioned as shown, landmarks L1 and L3 will have reduced estimation accuracy in the x-component of world-centric position in comparison to the y-component, while landmark L2 will have reduced estimation accuracy in the y-component in comparison to the x-component. Landmarks L4 and L5 would be expected to have roughly equal error in estimates of the x- and y-components of world-centric positions. Estimation performance in large environments is investigated in Case 3.6.

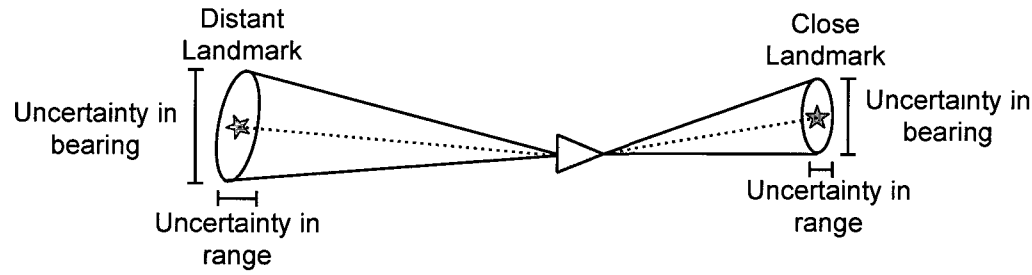

Figure 3.12: Measurement uncertainty for a range and bearing sensor.

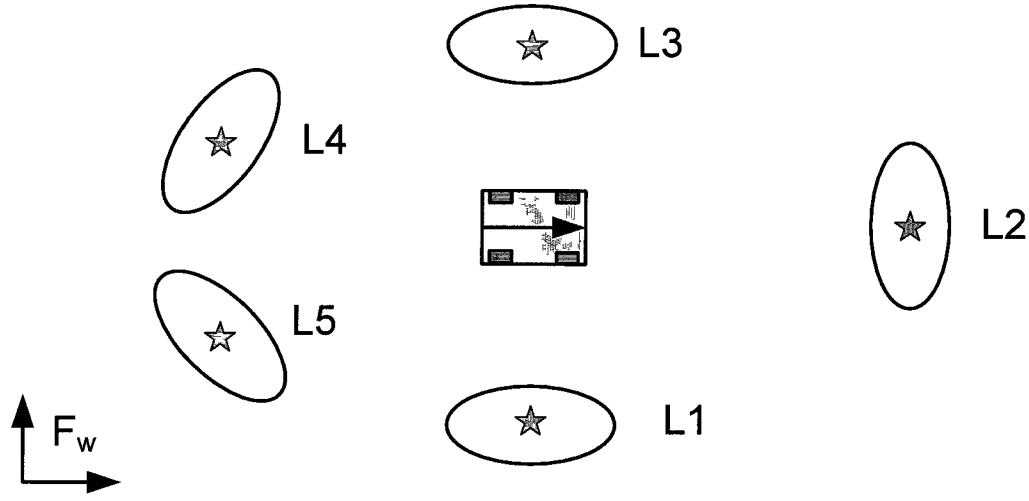

Figure 3.13: The effect of errors of a range and bearing sensor.

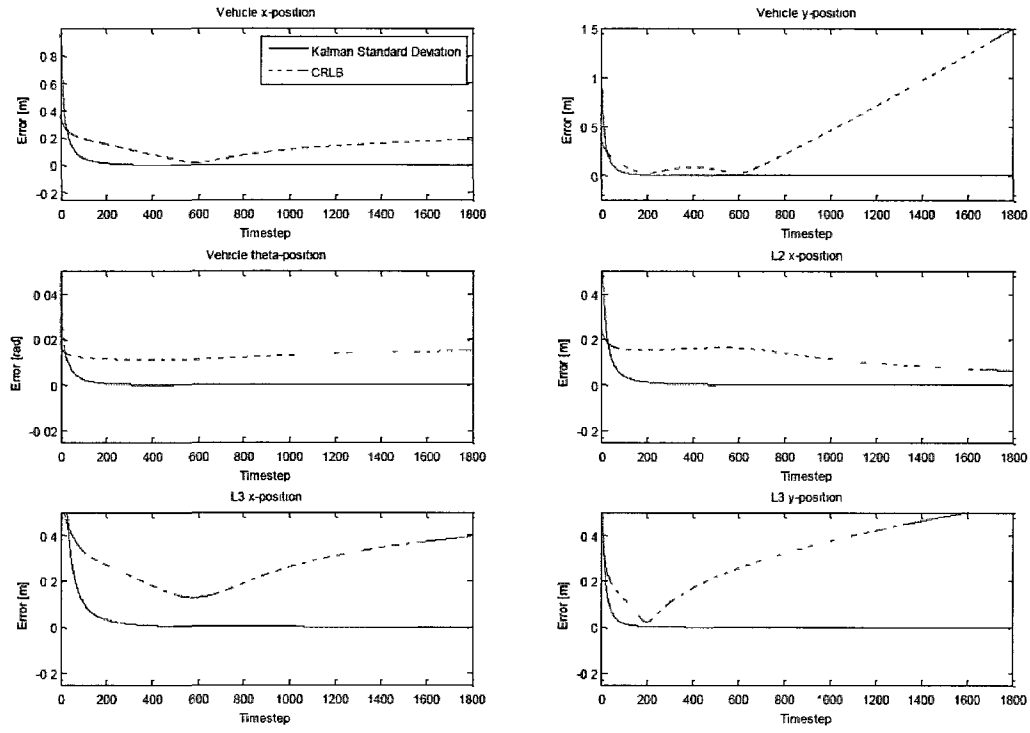

Figure 3.14: Case 3.6: Accuracy in large environments.

Case 3.6: Accuracy in Large Environments

In large environments, the vehicle may travel significant distances from the special landmark pair. The environment of Figure 3.1 is used in this case with simulation parameters as described in Section 2.6. The vehicle, using the simple unicycle model, is initially placed at the origin of the simulation reference frame and travels at a constant speed of 2m/s from left to right in Figure 3.1 along the x-axis of the simulation frame. The special landmarks are $s(1) = L1$ and $s(2) = L2$ with the y-coordinate of $s(2)$ fixed.

The special landmarks are separated along the vehicle trajectory ($L2$ is 40m to the right of $L1$ in Figure 3.1). The CRLBs and the EKF standard deviations are computed as the vehicle travels a straight path at constant velocity through the large environment. The CRLBs and the Kalman standard deviations are shown

in Figure 3.14. These results show that while the CRLBs are decreasing or constant, the standard deviations estimated by the EKF increase significantly as the vehicle moves farther away from the special landmark pair. Additionally, there are some peaks and valleys in the EKF standard deviation estimates over the run. For example, in Figure 3.14 the Kalman covariance for the y-position estimate of the vehicle shows local minima around 200 and 600 timesteps. These times coincide with the vehicle being directly above (y-axis in the Figure) $s(1)$ at 200 timesteps and $s(2)$ at 600 timesteps. These attributes are a result of the range-and-bearing sensor that is used. Although the range and bearing sensor has constant errors on each of the range and bearing channels, the noise levels in Cartesian space as the vehicle observes a landmark is dependent upon the position of the vehicle relative to the landmark being observed. The CRLB values decrease because information is being accumulated with additional measurements, however the EKF covariance diagonals increase, particularly in the vehicle y-coordinate, during the latter half of the simulation. At the end of this simulation, the vehicle has moved 180m to the right of the origin. As the vehicle moves farther and farther to the right and away from the special landmark pair, more error is introduced in the y-component of vehicle position due to the bearing error. The same trend of increasing standard deviations also occurs in the estimation of the position of landmark L3. The disparity between the CRLBs and the EKF standard deviations show, in part, how much the EKF estimation accuracy is being reduced as the distance from the vehicle to the special landmark pair increases.

3.5 Adding and Deleting Landmarks in the EKF

The EKF framework provides a mechanism for the addition or deletion of landmarks in the filter. The decision to delete a landmark may be triggered by the following events:

1. A previously stationary landmark within the SLAM filter, special or general, transitions to a moving landmark, or a previously moving landmark within the Moving Landmark Filter (MLF) transitions to stationary; or

2. One or both special landmarks used for observability moves beyond a predefined distance threshold; or

3. A landmark moves beyond the useful coverage of the sensors, or drops from sensor readings due to occlusion by other objects.

The addition of a landmark into the EKF is triggered by the following events:

1. A previously stationary landmark is added to the MLF, or a previously moving landmark is moved to the SLAM filter; or

2. A new special landmark is selected from available stationary landmarks; or

3. A new landmark comes within the useful coverage of the sensors, or now appears in sensor readings by coming into view around other objects.

In this thesis there are four classes of landmarks within the SLAM and Moving Landmark filters that require alternate steps for removal or addition:

1. Special landmark $s(1)$ of the SLAM filter for which both coordinates are treated as known parameters;

2. Special landmark $s(2)$ of the SLAM filter for which one coordinate is treated as a known parameter and the other coordinate is estimated in the filter;

3. A general stationary landmark $s(\ell)$ within the SLAM filter; and

4. A general moving landmark $m(\ell)$ within the MLF.

Landmark Removal

Removal of special landmark $s(1)$ (two known coordinates) from the SLAM filter involves setting these two parameters aside and replacing these in the filter with the new estimated coordinates of the selected replacement for $s(1)$ as fixed parameters. The measurement model re-defines $\mathbf{h}_k^{s(1)}$ and $\mathbf{R}^{s(1)}$ to be consistent with the selected replacement for $s(1)$, and the previous landmark $s(1)$ is either added to the SLAM filter as a general stationary landmark, or added to the MLF as a new landmark.

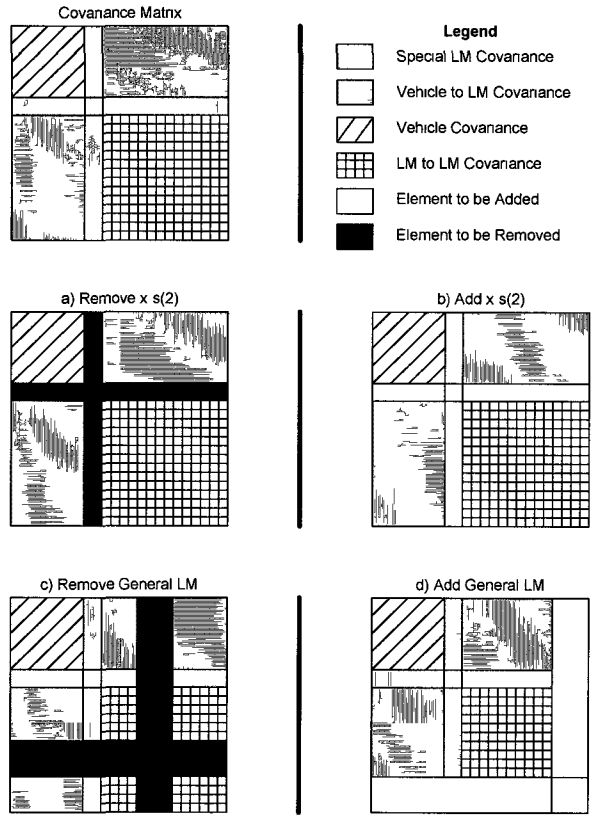

Figure 3.15: Kalman covariance matrix changes for addition or removal of landmarks.

Removal of special landmark $s(2)$ from the SLAM filter involves setting aside the fixed coordinate of $s(2)$ and replacing this with the new estimated coordinate of the selected replacement for $s(2)$ as a fixed parameter. The previously estimated coordinate of the old $s(2)$ is deleted from the state vector $\mathbf{x}_k^s$, the single row and column of the Kalman covariance matrix, as illustrated in Figure 3.15 a), are deleted as are the corresponding row and column of $\mathbf{F}_k^s$ and also of the process covariance $\mathbf{Q}_k^s$. Measurement $\mathbf{h}_k^{s(2)}$ is deleted from the measurement model along with the corresponding measurement covariance $\mathbf{R}^{s(2)}$. The previous landmark $s(2)$ is either added to the SLAM filter as a general stationary landmark, or added to the MLF as a new landmark.

Several steps must be performed to remove a non-special landmark $s(\ell)$ from the SLAM filter:

1. $\mathbf{x}_k^{s(\ell)}$ is deleted (two elements) from the state vector $\mathbf{x}_k^s$;
2. The corresponding two rows and two columns of the Kalman covariance matrix are deleted as illustrated in Figure 3.15 c);
3. The corresponding two rows and two columns of the process model $\mathbf{F}_k^s$ and $\mathbf{Q}_k^s$ are deleted; and
4. $\mathbf{h}^{s(\ell)}$ and $\mathbf{R}^{s(\ell)}$ are removed from the measurement model.

Removing landmark $m(\ell)$ from the MLF involves more elements since velocity and acceleration of each landmark are included in the state vector:

1. $\mathbf{x}_k^{m(\ell)}$ is deleted (six elements) from the state vector $\mathbf{x}_k^m$;
2. The corresponding six rows and six columns of the Kalman covariance matrix are deleted as illustrated in Figure 3.15 c);
3. The corresponding six rows and six columns of the process model $\mathbf{F}_k^m$ and $\mathbf{Q}_k^m$ are deleted; and
4. $\mathbf{h}^{m(\ell)}$ and $\mathbf{R}^{m(\ell)}$ are removed from the measurement model.

Landmark Addition

The procedure for adding landmarks into the SLAM system is referred to in some literature as *augmentation.* A new landmark can be added to the state vector based on measurements $(r_k^{M(\ell)}, b_k^{M(\ell)})$ with the initialization

$$\begin{aligned} x_k^{M(\ell)} &= x_k^v + r_k^{M(\ell)} \cos(\theta_k^v + b_k^{M(\ell)}), \\ y_k^{M(\ell)} &= y_k^v + r_k^{M(\ell)} \sin(\theta_k^v + b_k^{M(\ell)}). \end{aligned} \tag{3.49}$$

Taking the Jacobian $\mathbf{J}_k^v$ of this initialization with respect to the robot pose $\mathbf{p}_k^v = [x_k^v, y_k^v, \theta_k^v]$ gives

$$\mathbf{J}_k^v = \begin{bmatrix} 1 & 0 & -r_k^{M(\ell)} \sin(\theta_k^v + b_k^{M(\ell)}) \\ 0 & 1 & r_k^{M(\ell)} \cos(\theta_k^v + b_k^{M(\ell)}) \end{bmatrix}. \tag{3.50}$$

Taking the Jacobian $\mathbf{J}^M$ of this initialization with respect to the measurements $(r_k^{M(\ell)}, b_k^{M(\ell)})$ gives

$$\mathbf{J}_k^M = \begin{bmatrix} \cos(\theta_k^v + b_k^{M(\ell)}) & -r_k^{M(\ell)} \sin(\theta_k^v + b_k^{M(\ell)}) \\ \sin(\theta_k^v + b_k^{M(\ell)}) & r_k^{M(\ell)} \cos(\theta_k^v + b_k^{M(\ell)}) \end{bmatrix}. \tag{3.51}$$

An approximate initial position covariance for a new landmark based on the intialization in (3.49) is

$$\mathbf{P}_k^{M(\ell)} = \mathbf{J}^v \mathbf{P}_k^v \mathbf{J}_k^{vT} + \mathbf{J}_k^M \mathbf{R}^{M(\ell)} \mathbf{J}_k^{MT}, \tag{3.52}$$

where $\mathbf{P}_k^v$ is the vehicle pose covariance (upper left 3×3 block of the Kalman covariance matrix for the simple unicycle model), and $\mathbf{R}^{M(\ell)}$ is the 2×2 measurement covariance of the sensor. It should be noted that an estimated diagonal initialization for $\mathbf{P}_k^{M(\ell)}$ will also work reasonably well since the EKF algorithm will naturally compute its assumed covariance from a nonsingular initialization over a small number of iterations.

Moving a landmark to the SLAM EKF from the MLF, or to the MLF from the SLAM EKF, exploits information in the original filter by transferring current Kalman estimates and covariance submatrices for this landmark to the new filter.

Addition of special landmark $s(1)$ (two known coordinates) into the SLAM filter

involves re-assigning these two known parameters to the new estimated coordinates of the selected replacement for $s(1)$. The measurement model redefines $\mathbf{h}_k^{s(1)}$ and $\mathbf{R}^{s(1)}$ to be consistent with the selected replacement for $s(1)$. Addition of special landmark $s(2)$ into the SLAM filter involves re-assigning one parameter to the new fixed coordinate of the selected replacement for $s(2)$ (either the x or the y coordinate). The new estimated coordinate for $s(2)$ is added into the state vector $\mathbf{x}_k^s$, a single row and column of the Kalman covariance matrix, as illustrated in Figure 3.15 b), are inserted as are the corresponding row and column of $\mathbf{F}_k^s$ and also of the process covariance $\mathbf{Q}_k^s$. One element of the measurement $\mathbf{h}_k^{s(2)}$ is added to the measurement model along with the corresponding diagonal element of the covariance $\mathbf{R}^{s(2)}$.

Several steps must be performed to add a non-special landmark number $s(\ell)$ into the SLAM filter:

1. $\mathbf{x}_k^{s(\ell)}$ is added (two elements) to the state vector $\mathbf{x}_k^s$;
2. The corresponding two rows and two columns are added to the Kalman covariance matrix as illustrated in Figure 3.15 d);
3. The corresponding two rows and two columns of the process model $\mathbf{F}_k^s$ and $\mathbf{Q}_k^s$ are added; and
4. $\mathbf{h}^{s(\ell)}$ and $\mathbf{R}^{s(\ell)}$ are added to the measurement model.

Addition of landmark $m(\ell)$ into the MLF involves more elements since velocity and acceleration are included in the state vector:

1. $\mathbf{x}_k^{m(\ell)}$ is added (six elements) into the state vector $\mathbf{x}_k^m$;
2. The corresponding six rows and six columns of the Kalman covariance matrix are added as illustrated in Figure 3.15 d);
3. The corresponding six rows and six columns of the process model $\mathbf{F}_k^m$ and $\mathbf{Q}_k^m$ are added; and
4. $\mathbf{h}^{m(\ell)}$ and $\mathbf{R}^{m(\ell)}$ are added to the measurement model.

3.6 Issues With Re-selection of Special Landmarks

The results of Case 3.6 indicate that in order to maintain accuracy in a large environment a strategy may be needed to re-select the special landmarks as the vehicle is in motion. In this thesis, a method of solving this problem is proposed that involves re-assigning the special landmarks to other existing stationary landmarks within the environment. There are four major issues relevant to re-selection of special landmarks:

1. When is it necessary to re-select special landmarks?
2. What are the implications on observability?
3. What bias may be introduced into the filters through re-selection? and
4. Which coordinate of the new $s(2)$ should be kept fixed with the other coordinate being estimated in the SLAM filter, and are there any implications on filter performance?

When to re-select special landmarks.

One possible method to re-select the special landmarks is to monitor the Kalman covariance matrix to see if the diagonal elements are increasing beyond reasonable limits and, if so, then re-select one or both of the special landmarks. However, for the non-linear SLAM problem the diagonal elements of the covariance matrix may increase or decrease as the vehicle moves within relatively short distances of the special landmark pair. The proposed method is to execute a special landmark re-selection step if

- A special landmark transitions from stationary to moving; or
- A special landmark becomes occluded from view by the vehicle sensors; or
- The vehicle has travelled significantly far away from either of the special landmarks.

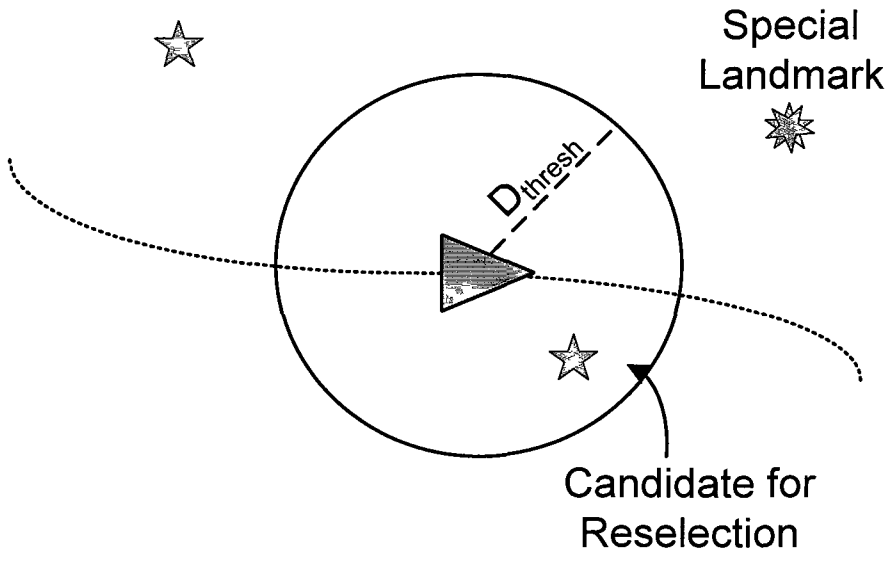

Figure 3.16: Distance threshold for re-selection of special landmarks.

As illustrated in Figure 3.16, a user-defined threshold, D_m, can be selected that represents the maximum distance that the vehicle may move away from a special landmark before re-selection of that landmark is needed. Whenever a special landmark falls outside of the D_m distance threshold, other stationary landmarks inside this distance threshold are candidates for re-selection. If no other stationary landmarks are available within this threshold distance, then the SLAM system must continue in its current form and possibly emphasize a search for a suitable stationary landmark. When multiple candidate landmarks are available within the threshold, a re-selection policy for a new special landmark should consider the following general rules:

- The landmark is selected from the existing stationary landmarks that have been maintained in the SLAM filter for at least a threshold number of timesteps; and

- The Kalman covariance of the new special landmark x and y position estimates are within pre-defined thresholds; and

- The new special landmark is within a distance of $(1-\epsilon)D_m$ from the vehicle, where ϵ forms a boundary region to prevent the new landmark from moving beyond the distance threshold within a reasonable amount of time; and
- The new special landmark lies ahead of the vehicle along its current path (smaller bearing angle $b^{s(i)}$ is prefered.)

Implications on observability.

As long as the original special landmark pair is retained, the system remains observable, but the EKF accuracy in comparison to CRLBs is degraded as the vehicle moves away from the special landmark pair. When one or both special landmarks are re-selected, the new special landmark positions that have been estimated by the EKF form the new constraints for observability. Fixing the three coordinates of these estimated new special landmarks introduces some error into the system relative to the position and orientation of F_w defined by the original special landmark pair. As the vehicle roams through a large environment, the system remains observable while the special landmarks are fixed, but this observability is relative to the current observability constraints. The error introduced in each re-selection event changes the position and orientation of F_w relative to that defined in the original initializaton, and errors may accumulate in the filter from each re-selection event. Because the system is observable following each re-selection event, and provided the reselecton is admissible, one should expect convergence to unbiased results relative to the world reference frame defined by the new observability constraints. However, each re-selection event introduces estimation errors that re-define the world reference frame relative to the environment. As a result, the proposed technique will not allow the vehicle to create accurate maps relative to the original world reference frame indefinitely.

Bias introduced through re-selection.

The original three observability constraints on the special landmark pair can be arbitrarily selected as $d_1 = d_2 = d_3 = 0$. While these original special landmarks are

retained, the system is observable and, provided the filter is working correctly, the EKF should produce estimation errors that are reasonably approximated by a zero-mean Gaussian process. At the first re-selection event, the errors in the estimated positions of the new special landmarks are therefore expected to be drawn from a zero-mean Gaussian population relative to the initial world reference frame. A sample drawn from this population, however, will have non-zero error and hence some bias is introduced into the filter, relative to the initial frame of reference, at this first re-selection event. Because of this bias, and even if the filter is operating correctly, one can no longer expect that the EKF errors will be zero mean relative to the original world reference frame. Consequently, at the second re-selection event the additional errors introduced depend on the error introduced at the first re-selection event. In this way, numerous executions of re-selection should be expected to accumulate in error relative to the initial reference frame. In very large environments with many re-selection events, one may still require loop closure techniques in order to correct for re-selection errors.

The coordinate of the new $s(2)$ to be fixed.

Re-selection of one or both special landmarks requires addition and deletion of landmarks as discussed in Section 3.5 and re-assignment of the observability constraints $[d_1, d_2, d_3^x]$ or $[d_1, d_2, d_3^y]$ defined in Section 3.3. The initial selection of these observability constraints defines the world-centric reference frame, F_w, relative to the initial choices of stationary landmarks for $s(1)$ and $s(2)$.

If $s(1)$ is to be re-selected, then $[d_1, d_2]$ are set to the current estimates of the new $s(1)$. However, as shown in Section 3.3, once d_1 and d_2 are re-selected, admissible values for either $|d_3^x|$ or $|d_3^y|$ that define the constraint on the current $s(2)$ are bounded by the true distance between $s(1)$ and $s(2)$ in the environment; if either $|d_3^x|$ or $|d_3^y|$ is greater than the true distance between the new $s(1)$ and the current $s(2)$, then the solution is complex. A similar issue arrises when the current $s(1)$ is retained, but a new $s(2)$ is selected: the new constraint on $s(2)$ (either d_3^x or d_3^y) must be admissible. The problem arises when re-selection results in the line connecting $s(1)$ and $s(2)$ lying either horizontal (x-axis) or vertical (y-axis) in F_w. Case 3.7 demonstrates the

loss of accuracy that can be expected in the filter when re-selection results in near horizontal or near vertical orientations of the special landmark pair.

Case 3.7: Special landmark re-selection.

Simulation results in this Case demonstrate that during re-selection of special landmarks it is critical to consider the orientation of the new special landmark pair relative to F_w when making the decision of which coordinate of $s(2)$ should be fixed and which coordinate should be estimated in further executions of the SLAM EKF. The simulation environments for this simulation are defined in terms of an angle γ that represents the orientation of the line connecting possible re-selections of $s(1)$ and $s(2)$:

1. The vehicle is stationary at the origin of the simulation reference frame with zero heading;

2. Special landmark $s(1) = L1 = [20, -20]$ in the simulation reference frame;

3. Special landmark $s(2) = L2$ has simulated coordinates, for each value of γ, given by
$$\mathbf{x}^{s(2)} = \begin{bmatrix} x^{s(2)} \\ y^{s(2)} \end{bmatrix} = \begin{bmatrix} 20 + 40\cos(\gamma) \\ -20 + 40\sin(\gamma) \end{bmatrix}; \text{ and} \tag{3.53}$$

4. A third stationary landmark lies at $L3 = [80, -20]$.

The simulation is conducted as follows:

1. The value of γ is initialized as $\gamma = 0$ and is incremented by $3.6°$ for 100 steps in order to cover the full $360°$;

2. For each value of γ, Fisher's Information Matrix with no prior information is computed to steady state over 1800 timesteps at a sample period of 0.05sec. for estimating either the x-coordinate of $s(2)$ or estimating the y-coordinate of $s(2)$;

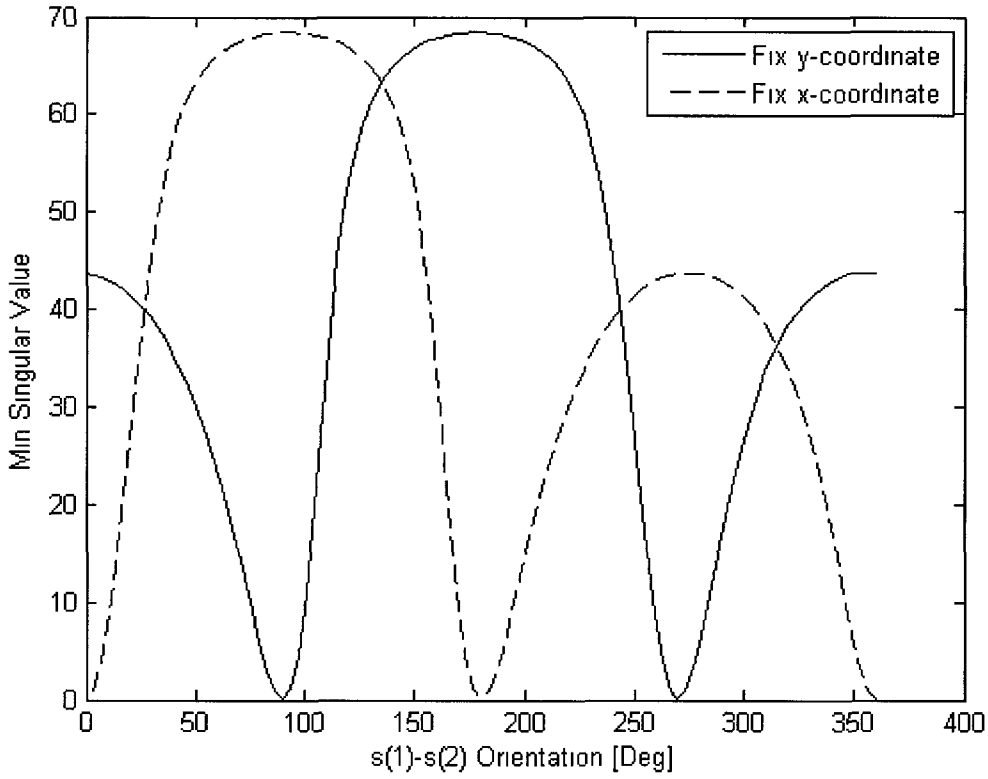

Figure 3.17: Filter information depends on special landmark orientation.

3. For each value of γ, and for either estimating the x-coordinate or the y-coordinate, a Singular Value Decomposition is computed for the final value of Fisher's Information Matrix and the minimum singular value is retained;

4. Figure 3.17 shows the minimum steady state singular values of Fisher's Information Matrix for either estimating the x-coordinate (solid line) or the y-coordinate (dashed line) of $s(2)$ for each value of γ.

As discussed in Section 2.4, if Fisher's Information Matrix remains singular in the absence of prior information, then there is an unobservable subspace in the system. As shown in Figure 3.17, if one is estimating the x-coordinate of $s(2)$ (solid line), then Fisher's Information Matrix approaches singularity as the special

landmark pair becomes vertical (y-axis) in F_w (i.e. $\gamma = 90°$ or $\gamma = 270°$). Note that the minimum singular value for these points is not exactly zero (at $\gamma = 90°$, the minimum singular value for estimating the x-coordinate is $4.0(10^{-8})$, and at $\gamma = 270°$ the minimum singular value is $6.4(10^{-9})$). However, the matrix is reported by Matlab to be singular to working precision: the ratio of the smallest singular value to the second smallest singular value is $2.5(10^{-10})$ at $\gamma = 90°$ and $4.0(10^{-11})$ at $\gamma = 270°$. If these vertical special landmarks are selected and one continues to estimate the x-coordinate of $s(2)$, then the filter behaves as if it is unobservable. The situation for estimating the y-coordinate of $s(2)$ is similar when the special landmark pair is horizontal (x-axis) in F_w (dashed line).

Fortunately, one has the option of either holding $x^{s(2)}$ fixed and estimating $y^{s(2)}$ in the filter, or holding $y^{s(2)}$ fixed and estimating $x^{s(2)}$ in the filter. In order to avoid these singular orientations, the following simple strategy is proposed:

1. For the new special landmark pair, compute

$$\begin{aligned} \delta x &= x^{s(2)} - x^{s(1)} \\ \delta y &= y^{s(2)} - y^{s(1)}. \end{aligned} \tag{3.54}$$

2. If $|\delta x| < |\delta y|$, then hold $x^{s(2)}$ fixed and estimate $y^{s(2)}$; otherwise

3. Hold $y^{s(2)}$ fixed and estimate $x^{s(2)}$.

3.7 Summary

The observability problem in SLAM has received little attention in the literature, especially for the autonomous problem for which no prior information is available concerning the environment. Section 3.1 first demonstrates that there are three degrees of freedom in the autonomous SLAM problem. Simulation results for three unobservable formulations, in Cases 3.1 through 3.3, demonstrate that the filters cannot correct for initialization errors. A current method to enforce observability by assuming that the positions of two landmarks in the environment are known

a priori gives decreasing Cramer-Rao Bounds in Case 3.4, but does not satisfy the autonomous requirement. The proposed solution given in Section 3.2 for the autonomous problem results in observability with Cramer-Rao bounds in Case 3.5 slightly above those for the two-known-landmark solution in Case 3.4. The proposed solution depends on the presence of two stationary landmarks in the environment, and can be initialized autonomously with the methods in Section 3.3. As the vehicle roams through relatively large dynamic environments as in Case 3.6, however, filter accuracy can be seriously degraded as the vehicle moves farther away from the special landmark pair. Moreover, special landmarks may have to be re-selected if they transition from stationary to moving, or become occluded from sensor view. The proposed parallel structure of the SLAM EKF and Moving Landmark EKF accommodates dynamic re-arrangement of states, Section 3.5, in order to adapt to a changing environment. Re-selection of special landmarks, Section 3.6, must exploit the option of estimating the x-coordinate of $s(2)$ or the y-coordinate of $s(2)$ in order to avoid singular orientations shown in Case 3.7 that can seriously degrade filter information and performance. Further analysis of the proposed methods in large dynamic environments is conducted with simulation studies in Chapter 4.

Chapter 4

Autonomous SLAM Simulations

Chapter 3 developed the proposed method to enforce observability in the autonomous SLAM problem. Two special stationary landmarks, $s(1)$ and $s(2)$, must be available in the environment to form the three observability constraints. Both coordinates of $s(1)$ and one coordinate of $s(2)$ are held as fixed parameters. As the vehicle moves farther away from the special landmark pair, re-selection of the special landmarks may be executed with the intent of reducing estimation errors, but this may introduce bias in the filters that cause drift in state estimates. In dynamic environments, landmarks may transition between stationary and moving. It is proposed to use two EKF structures operating in parallel: the SLAM EKF for vehicle pose and stationary landmark position estimation, and the Moving Landmark Filter (MLF) to estimate position and motion of moving landmarks. The vehicle pose estimated within the SLAM EKF is used directly in the MLF.

Simulation results are shown in this Chapter to investigate estimation performance in representative environments. Simulation software was implemented using MatLab r2008b and executed on a Dell T3400 Workstation. In a statistical sense, simulation results presented here are exploratory rather than conclusive, largely due to time constraints on the thesis work. Sample results are shown for eight test Cases, and discussions of results focus on explanations for trends in estimation errrors over the selected trajectories.

All simulations use the simple unicycle model of Section 2.1.1 for vehicle motion.

Autonomoous initialization from Section 3.3 is implemented with two stationary landmarks that are horizontal (x-axis) in the simulation reference frame and hence it is the x-coordinate of $s(2)$ that is estimated in the SLAM EKF. Estimation of the x-coordinate of $s(2)$ relative to estimation of the y-coordinate of $s(2)$ represents a rotation of the simulation reference frame by 90° and hence the results shown here are representative of both strategies.

Re-selection of special landmarks is tested at specific timesteps during the trajectory, but the complete re-selection strategy proposed in Section 3.6 has not been fully implemented. Simulations presented here use special landmarks, initial and re-selected, that are horizontal in the simulation reference frame. This restriction focuses the investigation on bias introduced through re-selection, but leaves studies related to the loss of information due to orientation of the new special landmarks to future research. Detection of landmarks that transition between stationary and moving is tested to demonstrate that it is possible to operate the two filters in parallel and achieve good estimation performance. Significant work remains for future research to completely implement and test the proposed methods. Suggestions for future research are given in Chapter 5.

Section 4.1 provides an overview of the simulations. Two environments are used to evaluate the proposed methods and investigate filter performance. Brief summaries are given to introduce the eight simulation Cases for which detailed graphical results are shown. The simulation results are presented in three sections: SLAM in static environments without re-selection of special landmarks is treated in Section 4.2; re-selection of special landmarks in static environments is investigated in Section 4.3; and Section 4.4 shows example results in dynamic environments.

4.1 Overview of Simulations

Two environments are used to investigate filter performance using the proposed methods: the first is the "straight ladder" environment shown in Figure 4.1; and the second is the circular ladder environment shown in Figure 4.2. The simulated positions of landmarks in these environments are listed in Table 4.1 and Table 4.2,

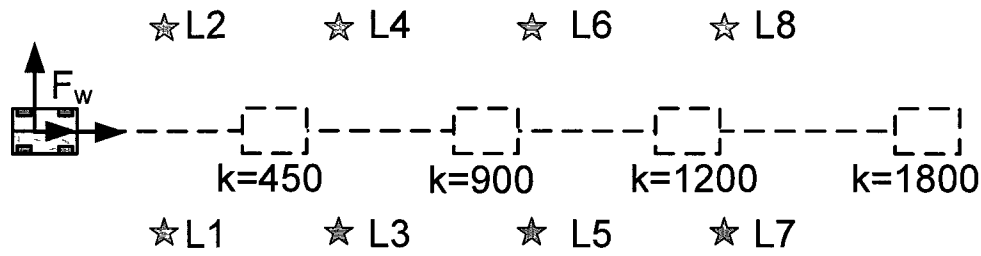

Figure 4.1: Straight ladder environment.

respectively. All simulations use the parameters defined in Section 2.6 with the initial vehicle positioned at the origin of the simulation reference frame with zero heading. These environments are relatively large due to the bearing error with standard deviation of 0.05 radians. A landmark at a distance of 100m from the vehicle will have an effective one-sigma lateral error of ±5m.

In simulations using the straight ladder environment, the total simulation time is 90sec. and the vehicle travels at a constant speed of 2m/s with zero turning rate. At the end of these simulations the vehicle x-position is 180m and is therefore approximately 160m from landmarks $L1$ and $L2$. The lateral uncertainty due to bearing error at these points is ±8m, largely in the y-coordinate of the world reference frame. For the circular ladder environment, the vehicle again travels with forward speed of 2m/s, but now with a turning rate of 0.05rad/s in order to follow a circular path of 40m radius. In this environment, the greatest distance from the vehicle to any landmark is 90m with an effective lateral error of ±4.5m.

The straight ladder environment is first used to assess any drift in filter estimates due to large distances from the special landmarks, while the circular ladder environment is used to assess repeatability as the vehicle makes two passes around the circular path. Both environments are used to explore bias introduced during re-selection or special landmarks. The straight environment is used test the parallel operation of the SLAM EKF and the MLF in dynamic environments.

Simulation results are shown for eight test Cases summarized as follows:

Table 4.1: Straight ladder landmark positions.

LM#	x-pos. (m)	y-pos. (m)
Vehicle Start Position	0	0
L1	20	-20
L2	20	20
L3	60	-20
L4	60	20
L5	100	-20
L6	100	20
L7	140	-20
L8	140	20

Table 4.2: Circular environment landmark positions

LM#	x-pos. (m)	y-pos. (m)
Vehicle Start Position	0	0
L1	0	10
L2	0	-10
L3	30	40
L4	50	40
L5	0	70
L6	0	90
L7	-30	40
L8	-50	40

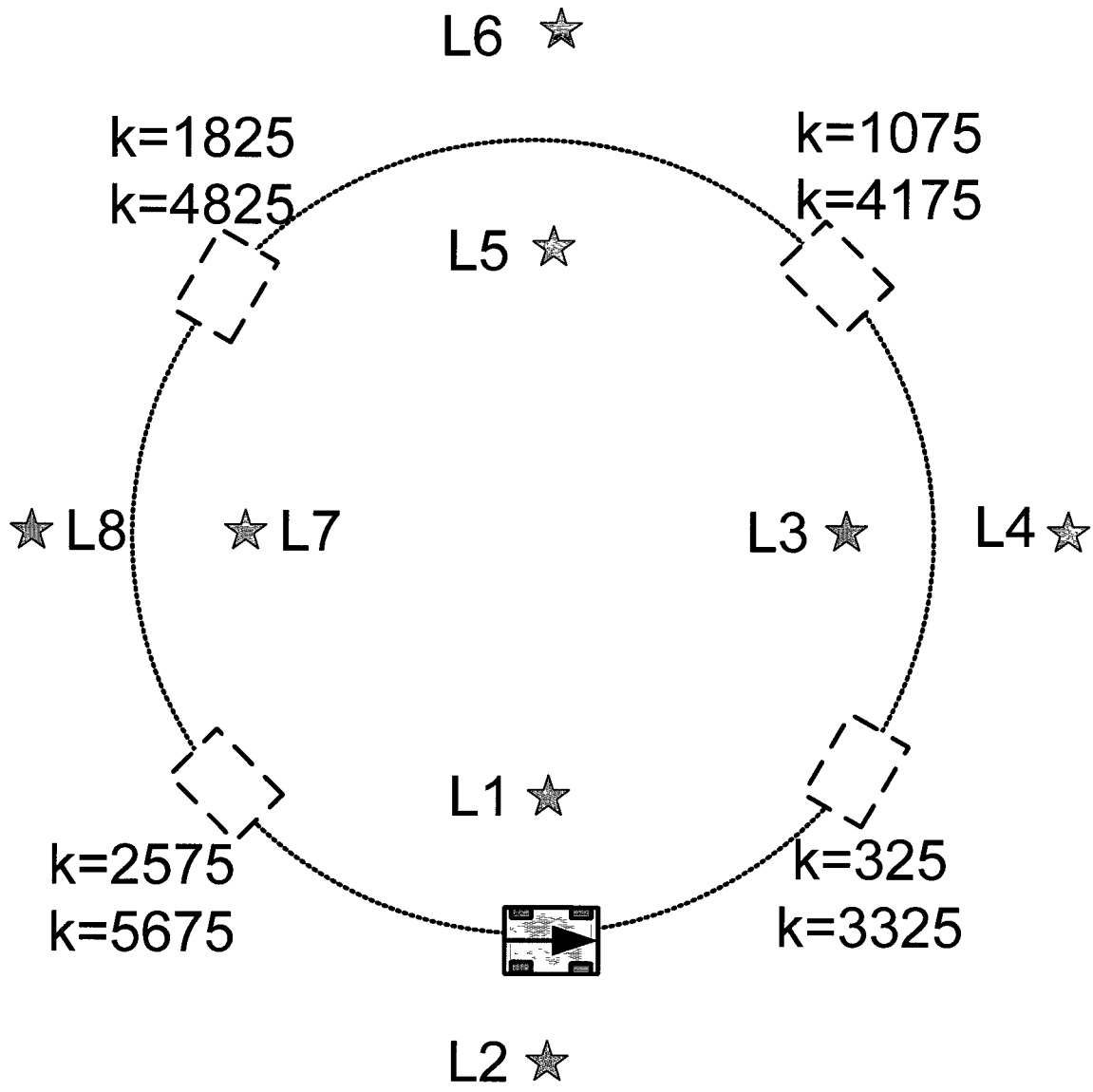

Figure 4.2: Circular ladder environment.

Static Environments

- **Case 4.1**: Straight ladder environment with no reselection and special landmarks $s(1) = L1 = (20, -20)$ and $s(2) = L3 = (60, -20)$. This Case is similar in configuration to the simple three-landmark environment of Chapter 3, but now with more landmarks.

- **Case 4.2**: Straight ladder environment with no reselection and special landmarks $s(1) = L1 = (20, -20)$ and $s(2) = L5 = (100, -20)$. This Case compares filter performance with the special landmarks separated by a greater distance in comparison to Case 4.1.

- **Case 4.3**: Circular ladder environment with no reselection and special landmarks $s(1) = L3 = (30, 40)$ and $s(2) = L4 = (50, 40)$. The vehicle traverses the circle twice in order to investigate repeatability of results.

Special Landmark Reselection

- **Case 4.4**: Straight ladder environment beginning with $s(1) = L1 = (20, -20)$ and $s(2) = L3 = (60, -20)$. The first re-selection occurs at timestep 600 (30 seconds) that changes $s(2)$ from $L3$ to $L5 = (100, -20)$. A second re-selection event is scheduled at 900 timesteps (45 seconds) that changes, $s(1)$ from $L1$ to $L3 = (60, -20)$.

- **Case 4.5**: Straight ladder environment beginning with $s(1) = L1 = (20, -20)$ and $s(2) = L3 = (60, -20)$. Four re-selection events are executed: at timestep 600 (30 seconds) $s(2)$ is re-selected from $L3$ to $L5 = (100, -20)$; at 900 timesteps (45 seconds) $s(1)$ is re-selected from $L1$ to $L3$; at 1200 timesteps (60 seconds) $s(2)$ is re-selected from $L5$ to $L7 = (140, -20)$; and finally at 1500 timesteps (75 seconds) $s(1)$ is reselected from $L3$ to $L5$.

- **Case 4.6**: Circular ladder environment beginning with $s(1) = L3 = (30, 40)$ and $s(2) = L4 = (50, 40)$. The re-selection schedule is most easily described by referring to the circle as a clock with 12:00 at the top in the diagram: when the vehicle is at 1:30, $s(2)$ is re-selected from $L4$ to $L7$; when the vehicle is

at 10:30, $s(1)$ is re-selected from $L3$ to $L8$; when the vehicle is at 7:30, $s(1)$ is re-selected from $L8$ back to $L3$; and when the vehicle is at 4:30, $s(2)$ is re-selected from $L7$ back to $L4$. This re-selection pattern is repeated for the second pass of the circle.

Dynamic Environments

- **Case 4.7**: Straight ladder environment as in Case 4.1 with all landmarks initially stationary. At 100 timesteps, landmark $L2$ begins to move on a sinusoidal trajectory through the environment. This motion is detected, and the landmark is moved to the MLF in order to track position, velocity and acceleration for the remainder of the trajectory.

- **Case 4.8**: Straight ladder environment as in Case 4.1 with landmarks $L1$ and $L3$ through $L8$ stationary. Landmark $L2$ is initialized within the MLF and moves in a sinusoidal trajetory. Landmark $L2$ is brought to stationary, this transition is detected in the filters, and the landmark is moved from the MLF to the SLAM EKF. Thereafter, the position of Landmark $L2$ is estiamted within the SLAM EKF.

4.2 SLAM in Static Environments

Case 4.1

This simulation uses the straight ladder environment of Figure 4.1 with no reselection and with special landmarks $s(1) = L1 = (20, -20)$ and $s(2) = L3 = (60, -20)$. This Case is similar in configuration to the simple three-landmark environment of Chapter 3, but now with more landmarks. Simulation results are shown in Figures 4.3-4.5.

Figure 4.3 shows the environment with true landmark positions denoted with a circle together with estimated positions of landmarks and the estimated vehicle pose during the simulation. In these plots, the estimated vehicle pose is drawn with a black filled wide-arrow, then this is overlaid with a white filled wide-arrow for the

true vehicle pose. One can note that towards the end of the trajectory the estimated pose is drifting slightly from the true vehicle pose. With no re-selection the system remains observable and good estimation performance is seen for landmark positions in this static environment.

Figure 4.4 shows estimation errors for the vehicle pose and the x-position of $s(2) = L3$ over this simulation. Note the different scales on the vertical axes of these plots. These plots also show the Kalman standard deviation for each state. For this simulation, the vehicle passes directly above (y-axis) $s(1)$ at timestep 200, and passes directly above $s(2)$ at timestep 600. Two local minima are seen in the standard deviation of the vehicle y-position at these timesteps because the y-position at these points relative to the special landmarks is largely determined by the range measurement channel that has higher accuracy. The local maximum in the standard deviation for the x-coordinate of $s(2)$ at timestep 600 results because it is the bearing error that influences estimation of this coordinate when the vehicle lies directly above $s(2)$. As the vehicle moves farther to the right in the Figure away from the special landmark pair, the standard deviation for the y-component of vehicle position increases significantly. As the vehicle moves farther to the right, it is the bearing error in measurements to both $s(1)$ and $s(2)$ that influences the y vehicle position. Even though drift is apparent towards the end of the trajectory, the estimation errors remain within the one-sigma Kalman standard deviations and indicate that the filter is working correctly.

It is interesting to note in Figure 4.4 that for this trajectory and environment the vehicle pose directly above $s(2)$ at timestep 600 can be estimated with greatest accuracy as indicated by the local minima in Kalman standard deviations for vehicle position. However, at this time estimation of the x-coordinate of $s(2)$ is least accurate due to the local maximum in Kalman standard deviation. These results are caused by the use of vehicle-centric polar measurements to estimate the state within a Cartesian reference frame.

Figure 4.5 shows estimation results for landmarks $L2 = (20, 20)$ that lies at the start of the trajectory and for landmark $L7 = (140, -20)$ that lies towards the end of the trajectory. The vehicle passes directly below $L2$ at 200 timesteps. As the vehicle

moves farther to the right, the Kalman standard deviation for the y-component of $L2$ increases as expected due to bearing eror. The sample errors for estimating the y-component of $L7$ are degraded relative to the standard deviation towards the end of the trajectory most likely due to the poorer performance in estimating the vehicle y-position.

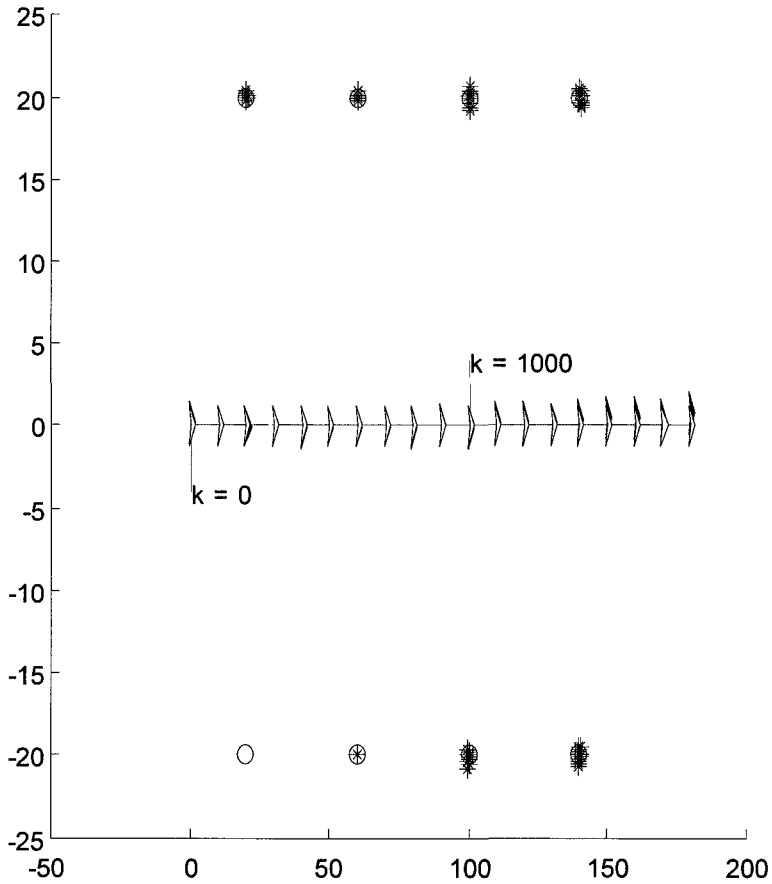

Figure 4.3: Case 4.1: Estimated vehicle trajectory and landmark positions.

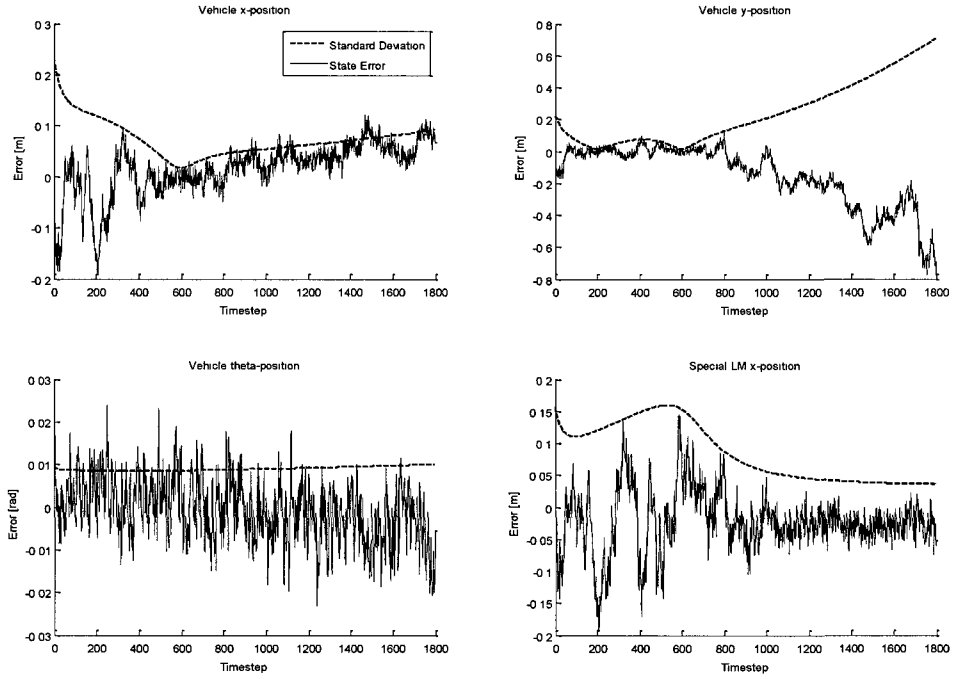

Figure 4.4: Case 4.1: Vehicle pose and special landmark coordinate.

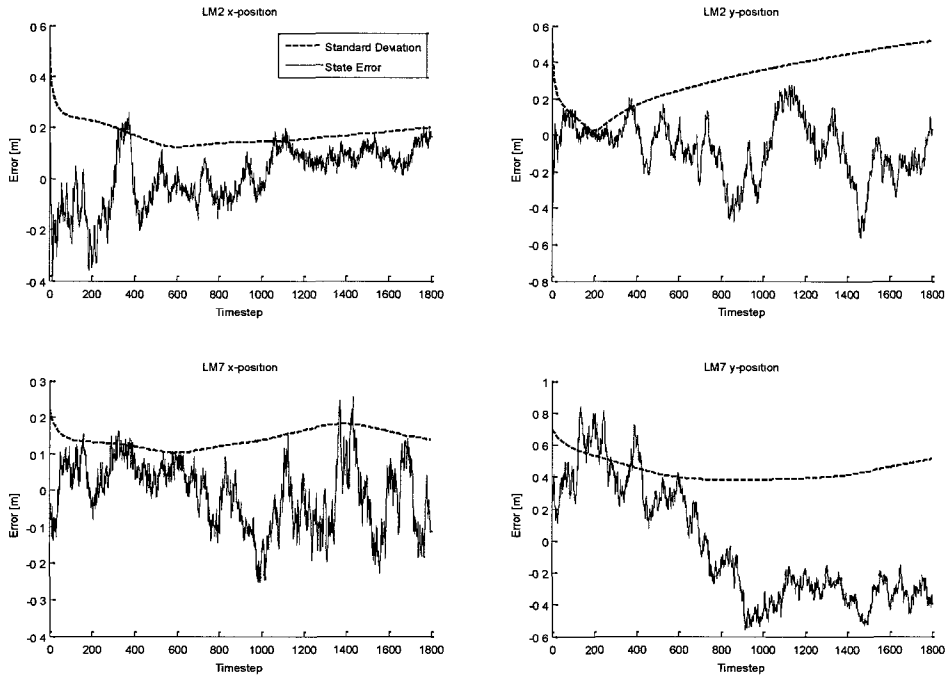

Figure 4.5: Case 4.1: Landmark position estimation.

Case 4.2

This simulation uses the straight ladder environment of Figure 4.1 with no reselection and with special landmarks $s(1) = L1 = (20, -20)$ and $s(2) = L5 = (100, -20)$. This Case is identical to Case 4.1 except that $s(2)$ is now moved 40m to the right in the environment, and a different set of random numbers are drawn from the Matlab generator. Simulation results, shown in Figures 4.6-4.8, are presented to confirm the explanations of primary observations in Case 4.1.

Figure 4.6 shows the true and estimated environment and vehicle trajectory. One can note that towards the end of the trajectory the estimated pose in this Case for the sample results shown have not drifted as much as the results in Case 4.1, but these are sample results only and this could be due to the set of random numbers drawn. Again, with no re-selection the system remains observable and good estimation performance is seen for landmark positions in this static environment.

Figure 4.7 shows estimation errors for the vehicle pose and the x-position of $s(2) = L5$ over this simulation. The vehicle passes, as in Case 4.1, directly above (y-axis) $s(1)$ at timestep 200, but now passes directly above $s(2)$ at timestep 1000. Again, two local minima are seen in the standard deviation of the vehicle y-position at these timesteps, and a local maximum in the standard deviation for the x-coordinate of $s(2)$ occurs at timestep 1000. As the vehicle moves farther to the right in the Figure away from the special landmark pair, the standard deviation for the y-component of vehicle position increases significantly similar to Case 4.1. Again the estimation errors remain within the one-sigma Kalman standard deviations and indicate that the filter is working correctly.

As noted in Case 4.1, the vehicle pose directly above $s(2)$ at timestep 1000 can be estimated with greatest accuracy, but at this time estimation of the x-coordinate of $s(2)$ is least accurate. In comparison to Case 4.1, the wider separation of $s(1)$ and $s(2)$ along the vehicle trajectory has not decreased the Kalman standard deviations for estimation of the x-coordinate and heading of vehicle pose, nor for estimation of the x-coordinate of $s(2)$. An improvement is seen in the Kalman standard deviations for estimating the vehicle y-coordinate at the end of the trajectory (roughly 0.5m for Case 4.2 and 0.8m for Case 4.1). This suggests that there may be advantages

to selecting a wide separation of the special landmarks or that drift in the y vehicle position, as seen in Case 4.1, could be reduced through re-selection of special landmarks.

Figure 4.8 shows estimation results for landmarks $L2 = (20, 20)$ and $L7 = (140, -20)$. These results are not significantly different from those for Case 4.1.

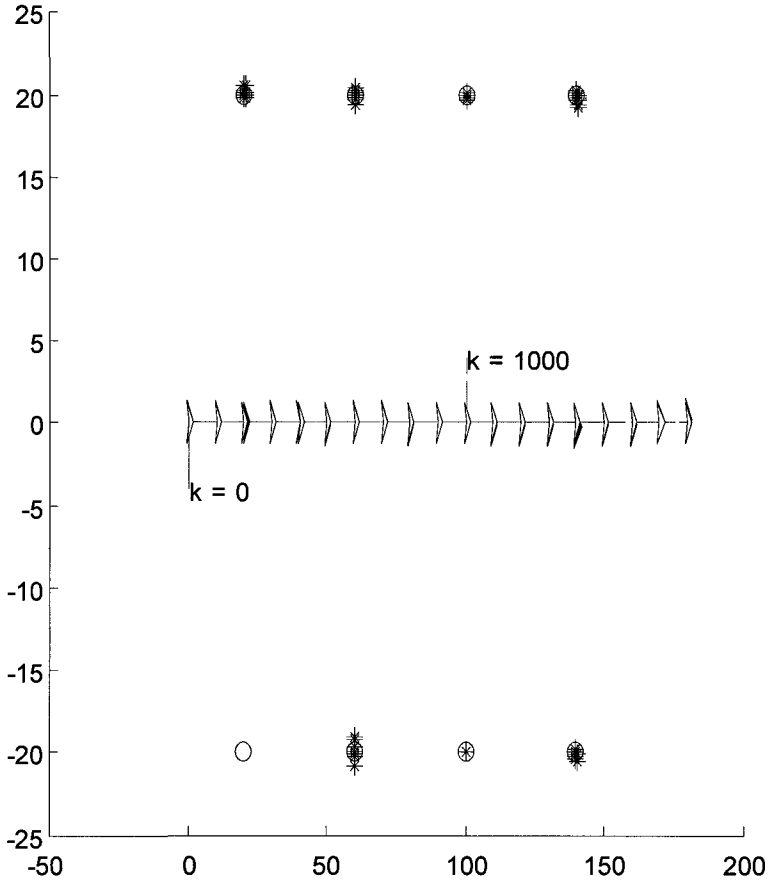

Figure 4.6: Case 4.2: Estimated vehicle trajectory and landmark positions.

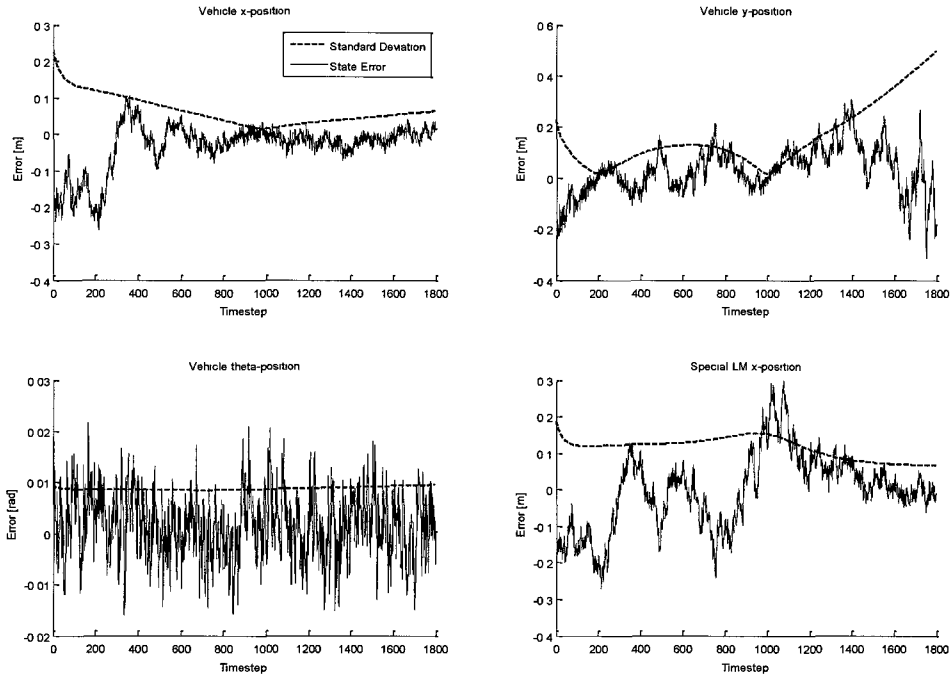

Figure 4.7: Case 4.2: Vehicle pose and special landmark coordinate.

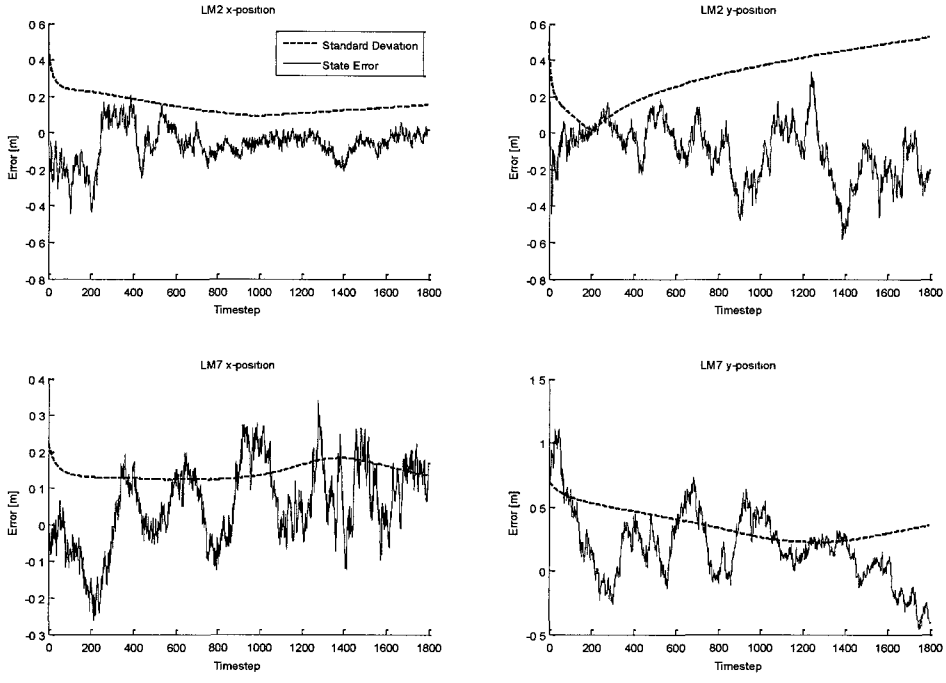

Figure 4.8: Case 4.2: Landmark position estimation.

Case 4.3

This simulation uses the circular ladder environment of Figure 4.2 with no reselection and with special landmarks $s(1) = L3 = (30, 40)$ and $s(2) = L4 = (50, 40)$. It is simplest to describe this trajectory in terms of a clock, with 12:00 at the top of the circle in Figure 4.2. The vehicle is initialized at 6:00 with zero heading and traverses the circle twice counterclockwise in order to investigate repeatability of results. (The simulation is acually run for 6000 timesteps which results in two complete passes plus additional motion so that the vehicle ends at approximately 1:00). Simulation results are shown in Figures 4.9-4.11.

Figure 4.9 shows the true and estimated environment and vehicle trajectory. After two passes of the circular trajectory, there is no apparent drift in vehicle pose estimates. With no re-selection the system remains observable and good estimation performance is seen for landmark positions in this static environment.

Figure 4.10 shows estimation errors for the vehicle pose and the x-position of $s(2) = L4$ over this simulation. The vehicle starts at 6:00, then approaches the special landmark pair $L3$ and $L4$ and passes directly between these landmarks at 3:00 or timestep 628. The vehicle continues and passes directly between landmarks $L7 = (-30, 40)$ and $L8 = (-50, 40)$ at 9:00 or timestep 1885. The local minima seen in the standard deviation for x vehicle position occur when the vehicle is at 3:00 or 9:00. At these positions the vehicle and special landmark pair lie on a line parallel to the x-axis of the simulation frame and it is the range measurement that gives the improved accuracy. The local maxima in standard deviation for x vehicle position occur when the vehicle is at 12:00 or 6:00 due to bearing errors in measurements of special landmark positions relative to the vehicle. The local minima in standard deviation for vehicle x position correspond to local maxima in the standard deviation for vehicle y position. The local minima in standard deviation for estimating the x-coordinate of $s(2)$ occur when the vehicle is at 3:00 and 9:00.

Figure 4.11 shows estimation results for landmarks $L2 = (0, -10)$ that lies at 6:00 and for landmark $L8 = (-50, 40)$ that lies at 9:00. The standard deviations for $L2$ vary only slightly during this trajectory, whereas those for $L8$ show strong local minima in the x-coordinate estimation when the vehicle is at 3:00 and 9:00.

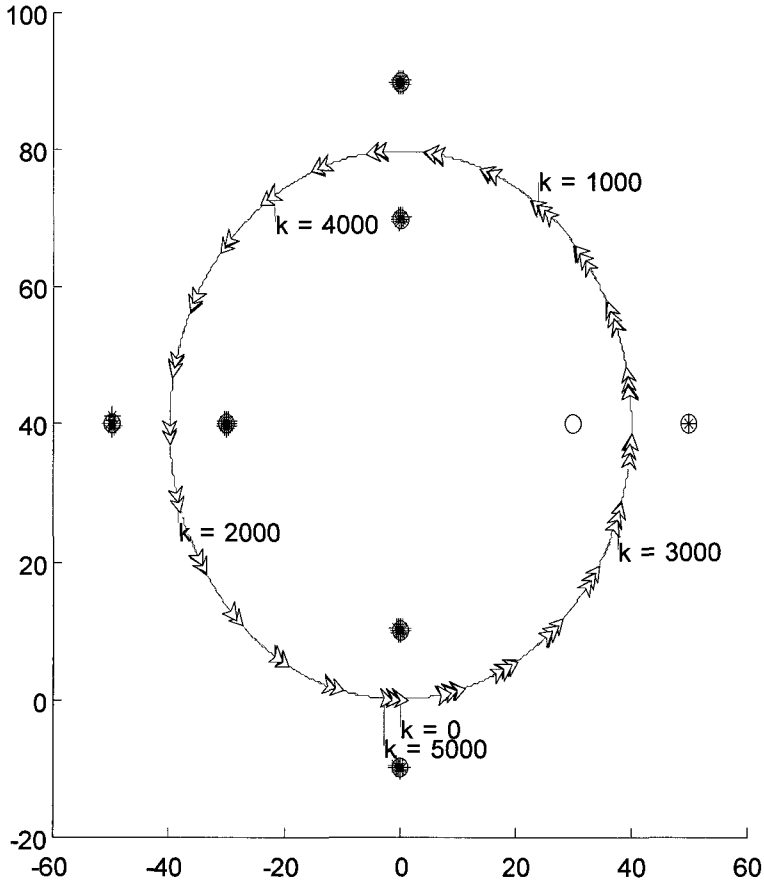

Figure 4.9: Case 4.3: Estimated vehicle trajectory and landmark positions.

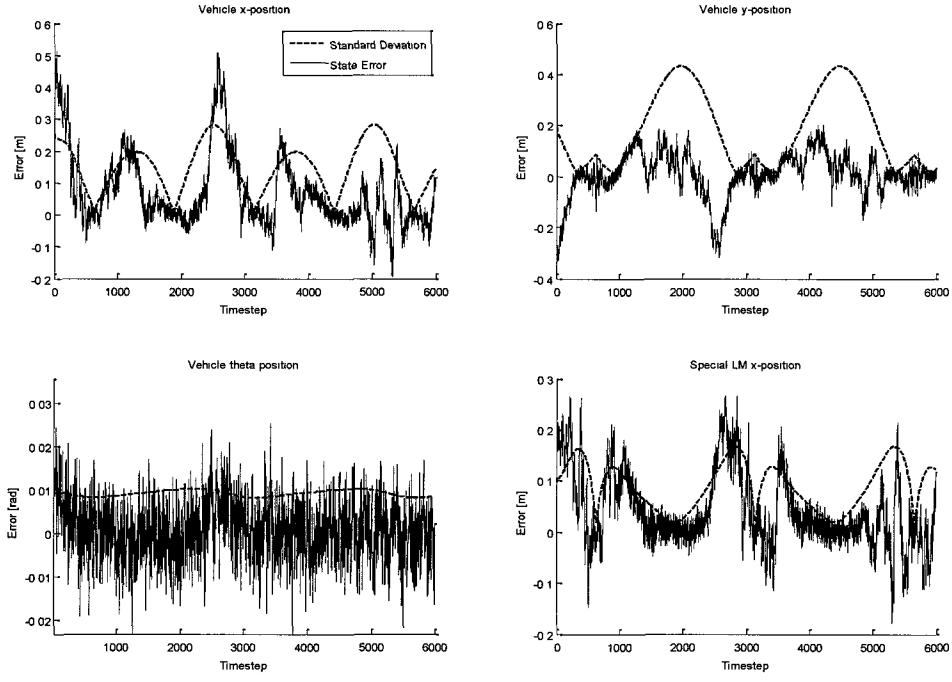

Figure 4.10: Case 4.3: Vehicle pose and special landmark coordinate.

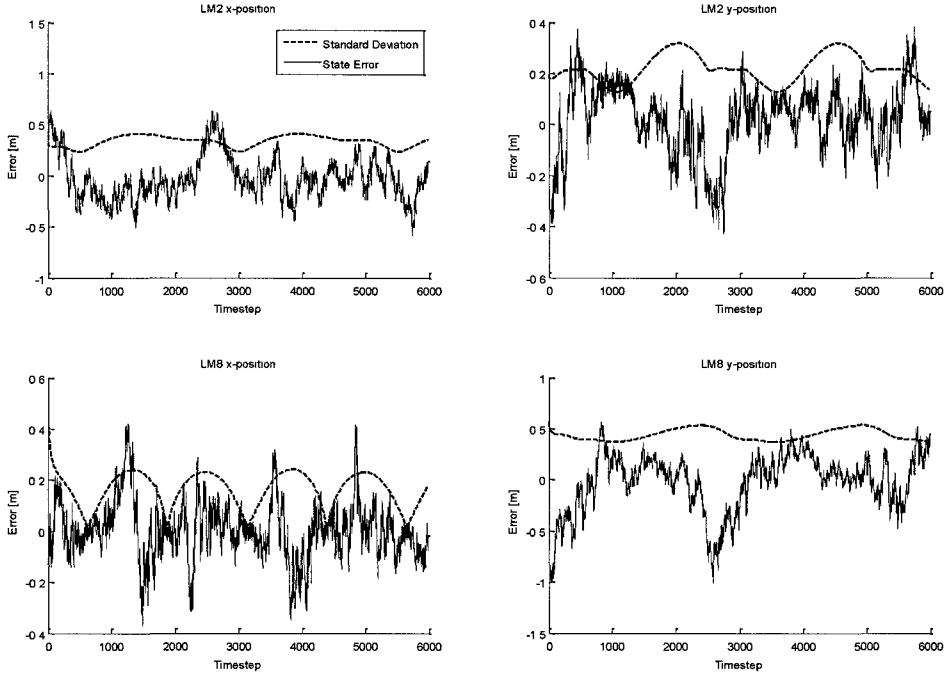

Figure 4.11: Case 4.3: Landmark position estimation.

4.3 Re-Selection of Special Landmarks

Case 4.4

This simulation uses the straight ladder environment of Figure 4.1 with special landmarks initially selected as $s(1) = L1 = (20, -20)$ and $s(2) = L3 = (60, -20)$ as in Case 4.1. The first re-selection event occurs at timestep 600 (30 seconds) that changes $s(2)$ from $L3$ to $L5 = (100, -20)$. This is the point during the trajectory, as discussed in Case 4.1, at which the vehicle position estimate should be most accurate. A second re-selection event is scheduled at 900 timesteps (45 seconds) that changes $s(1)$ from $L1$ to $L3 = (60, -20)$. Simulation results are shown in Table 4.3 and Figures 4.12-4.14. Although many simulation runs produced better results, these results are shown to emphasize possible errors introduced through re-selection.

Figure 4.12 shows the true and estimated environment and vehicle trajectory. This plot also shows the re-selection events with an "X" on the vehicle trajectory for re-selection of $s(2)$ and a "+" for re-selection of $s(1)$. The estimated vehicle pose is drawn with a black filled wide-arrow, then this is overlaid with a white filled wide-arrow for the true vehicle pose. One can note that towards the end of the trajectory the estimated pose is drifting from the true vehicle pose even more than the drift seen in the example results of Cases 4.1 and 4.2.

The system is initialized with (d_1, d_2, d_3^y) set to the true simulated coordinates, $d_1 = x^{s(1)}$, $d_2 = y^{s(1)}$ and $d_3^y = y^{s(2)}$. As a result, from timestep 0 through to the first re-selection at timestep 600, the world reference frame F_w is aligned with the simulation reference frame. The vehicle travels at 2m/s along the x-axis of F_w and passes directly above (y-axis) $s(1)$ at timestep 200, then passes directly above $s(2)$ at timestep 600. At timestep 600 the current estimate of the y-coordinate of $L5$ is captured and used as the new observability constraint d_3^y. The current estimate of the x-coordinate of $L5$ is used to initialize continued estimation of this coordinate of the new $s(2)$ in the filter. The landmark $L3$, the old $s(2)$, is added as a stationary landmark in the SLAM EKF and estimation of both coordinates continues similar to other general stationary landmarks.

Table 4.3 shows the error introduced through re-selection of $s(2)$ at timestep 600

is significant (-41cm). In these example results, the captured y-coordinate of $L5$ is -19.59m when the true value is -20m. At this time, the constraints (d_1, d_2) on $s(1)$ remain fixed at their original values. The error introduced through re-selection of $s(2)$ results in a translation and rotation of F_w relative to the simulation reference frame as illustrated in Figure 3.11.

Figure 4.13 shows estimation errors for the vehicle pose over this simulation. One can immediately see the impact of the re-selection event at timestep 600 on vehicle position estimates. The x vehicle position error rises to a roughly constant bias, and the y vehicle position shows a sharp drop due to the d_3^y error, followed by continued drift that may be due to distance from the special landmark pair. At timestep 900, Table 4.3 shows the additional error introduced through re-selection of $s(1)$ from its original value of $L1$ to $L3$, which was the original landmark used for $s(2)$. These errors are somewhat smaller (-11cm in x and -6cm in y) than those encountered in the re-selection of $s(2)$.

Figure 4.14 shows estimation results for landmarks $L2 = (20, 20)$ that lies at the start of the trajectory and for landmark $L8 = (140, 20)$ that lies towards the end of the trajectory. The Kalman standard deviations do not show sharp transitions at the re-selection events, however biased estimates are produced following re-selection. The error is most notable in the y position of $L8$ that may be due primarily to the poor estimation of the vehicle y-coordinate towards the end of the trajectory.

Table 4.3: Case 4.4: Re-selection errors for special landmarks. Errors for the constraints are true-estimated (m).

Timestep	Event	x-error	y-error
600	$s(2)$ from $L3$ to $L5$	–	-0.413
900	$s(1)$ from $L1$ to $L3$	-0.111	-0.066

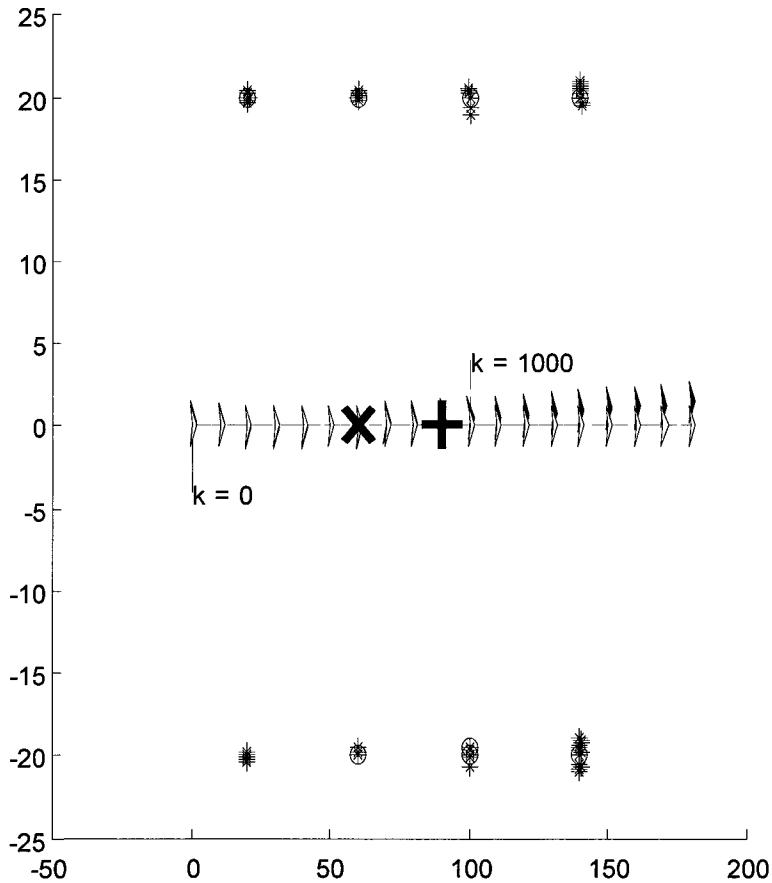

Figure 4.12: Case 4.4: Estimated vehicle trajectory and landmark positions.

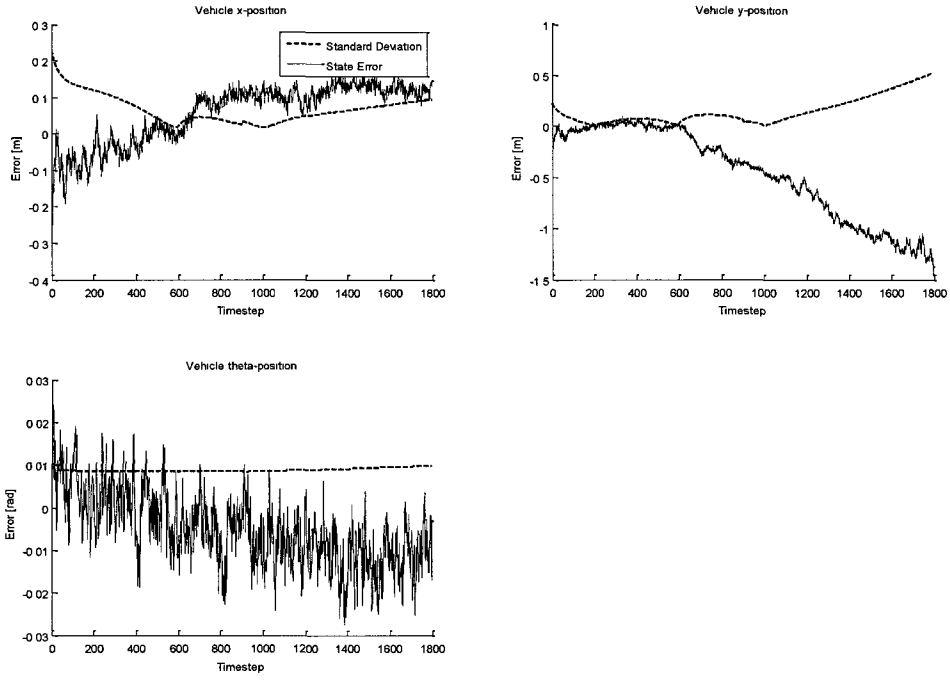

Figure 4.13: Case 4.4: Vehicle pose estimation.

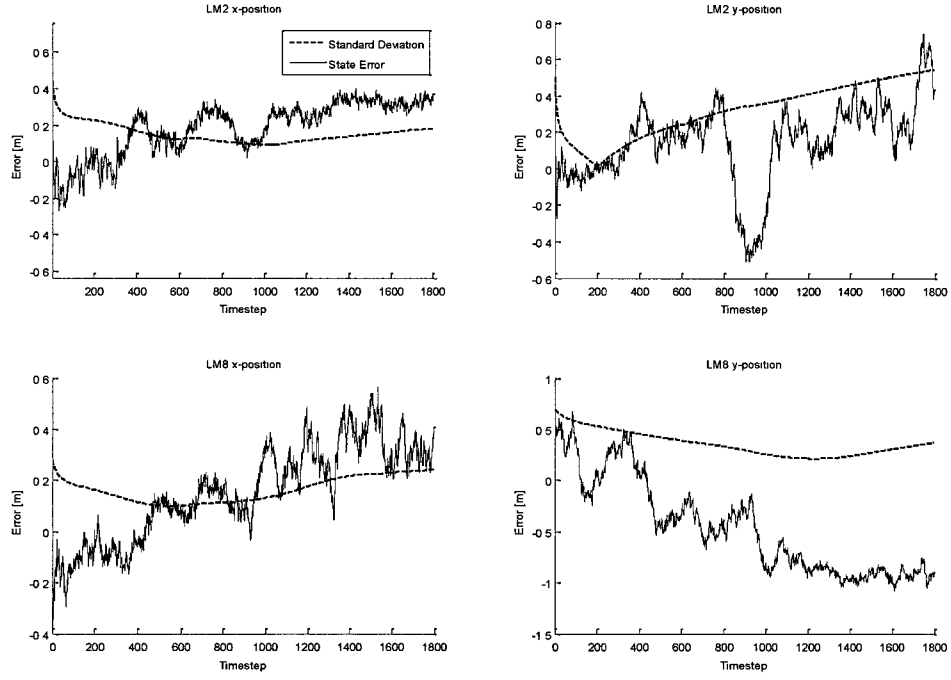

Figure 4.14: Case 4.4: Landmark position estimation.

Case 4.5

This simulation uses the straight ladder environment of Figure 4.1 with special landmarks initially selected as $s(1) = L1 = (20, -20)$ and $s(2) = L3 = (60, -20)$ as in Case 4.1 and Case 4.4. Four re-selection events are executed: at timestep 600 (30 seconds) $s(2)$ is re-selected from $L3$ to $L5 = (100, -20)$; at 900 timesteps (45 seconds) $s(1)$ is re-selected from $L1$ to $L3$; at 1200 timesteps (60 seconds) $s(2)$ is re-selected from $L5$ to $L7 = (140, -20)$; and finally at 1500 timesteps (75 seconds) $s(1)$ is reselected from $L3$ to $L5$. The first two re-selections are the same as those used in Case 4.4. Simulation results are shown in Table 4.4 and Figures 4.15-4.17. Many simulation runs were conducted in this Case and the example results shown are typical of those produced. These results suggest that there may be some advantages to the re-selection strategy even though, as shown in Case 4.4, bias introduced in re-selection can significantly degrade estimation performance.

Figure 4.15 shows the true and estimated environment and vehicle trajectory. This plot also shows the re-selection events with an "X" on the vehicle trajectory for re-selection of $s(2)$ and a "+" for re-selection of $s(1)$. One can note that towards the end of the trajectory the estimated pose is not drifting from the true vehicle pose and the system maintains reasonable accuracy in estimation of the landmark positions.

The system is initialized with (d_1, d_2, d_3^y) set to the true simulated coordinates, $d_1 = x^{s(1)}$, $d_2 = y^{s(1)}$ and $d_3^y = y^{s(2)}$ as in Case 4.4. Re-selections of $s(1)$ and $s(2)$ are conducted in the same manner as described for Case 4.4. Table 4.4 shows the error introduced at each re-selection event. The re-selection of $s(2)$ at timestep 600 introduces only 10cm of error into the system, whereas re-selection of $s(1)$ at timesteps 900 and 1500 introduce slightly more error. The error(s) introduced at each re-selection result in a translation and rotation of F_w relative to the simulation reference frame as illustrated in Figure 3.11.

Figure 4.16 shows estimation errors for the vehicle pose over this simulation. The major difference between these results and those for Case 4.4 is that the Kalman standard deviation for the y vehicle component is reduced and this indicates that more information is being retained within the filter.

Table 4.4: Case 4.5: Re-selection errors for special landmarks. Errors for the constraints are true-estimated (m).

Timestep	Event	x-error	y-error
600	$s(2)$ from $L3$ to $L5$	–	-0.099
900	$s(1)$ from $L1$ to $L3$	0.134	-0.260
1200	$s(2)$ from $L5$ to $L7$	–	0.114
1500	$s(1)$ from $L3$ to $L5$	0.131	-0.099

Figure 4.17 shows estimation results for landmarks $L2 = (20, 20)$ that lies at the start of the trajectory and for landmark $L8 = (140, 20)$ that lies towards the end of the trajectory. The state estimation errors remain within the one-sigma Kalman standard deviations which indicates that the filter is producing reasonable results.

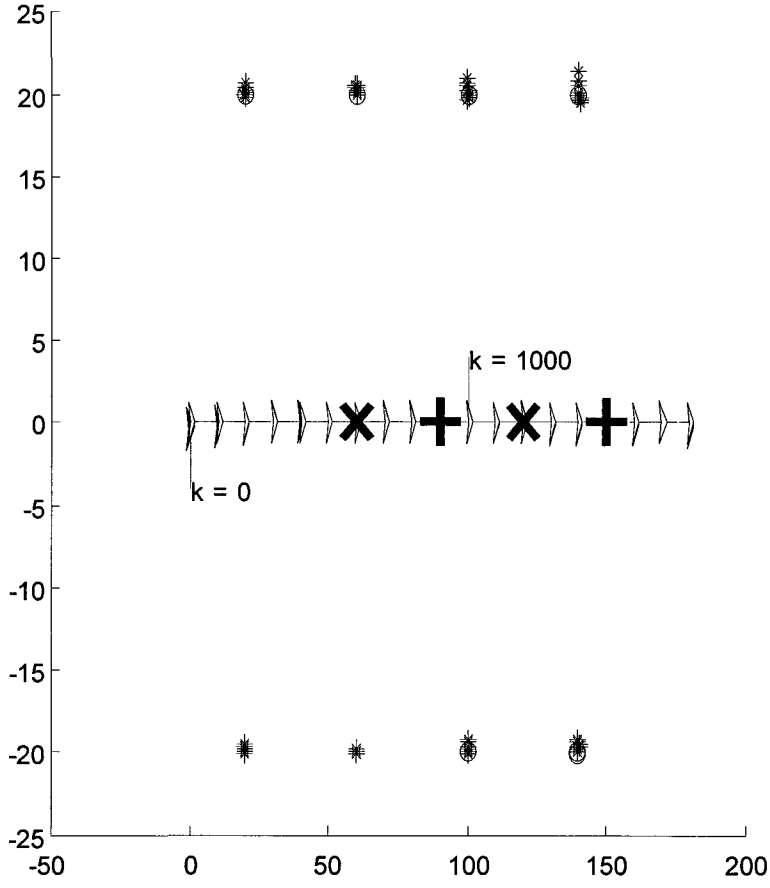

Figure 4.15: Case 4.5: Estimated vehicle trajectory and landmark positions.

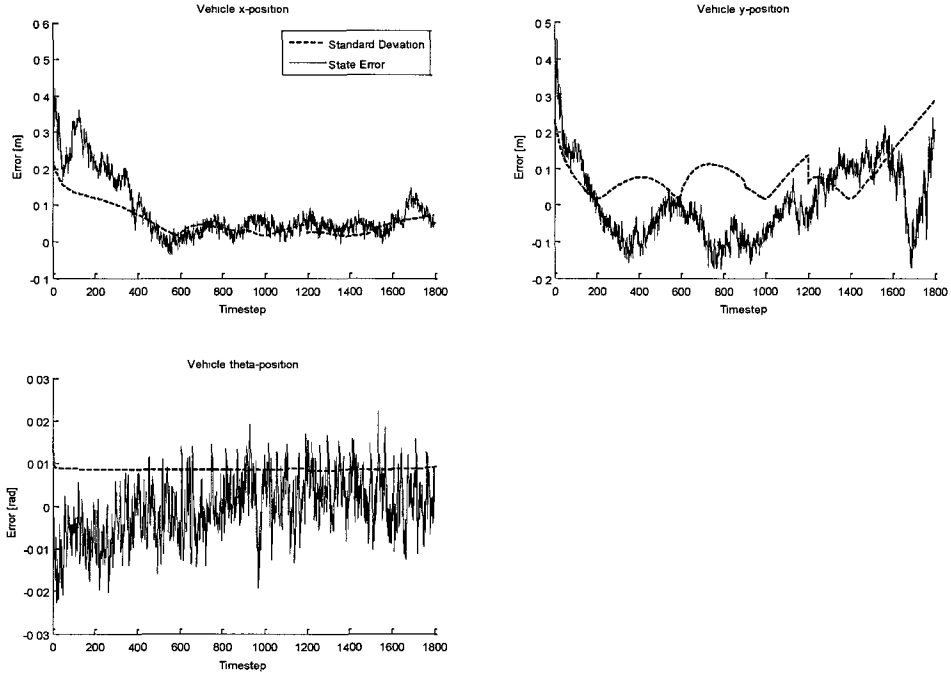

Figure 4.16: Case 4.5: Vehicle pose estimation.

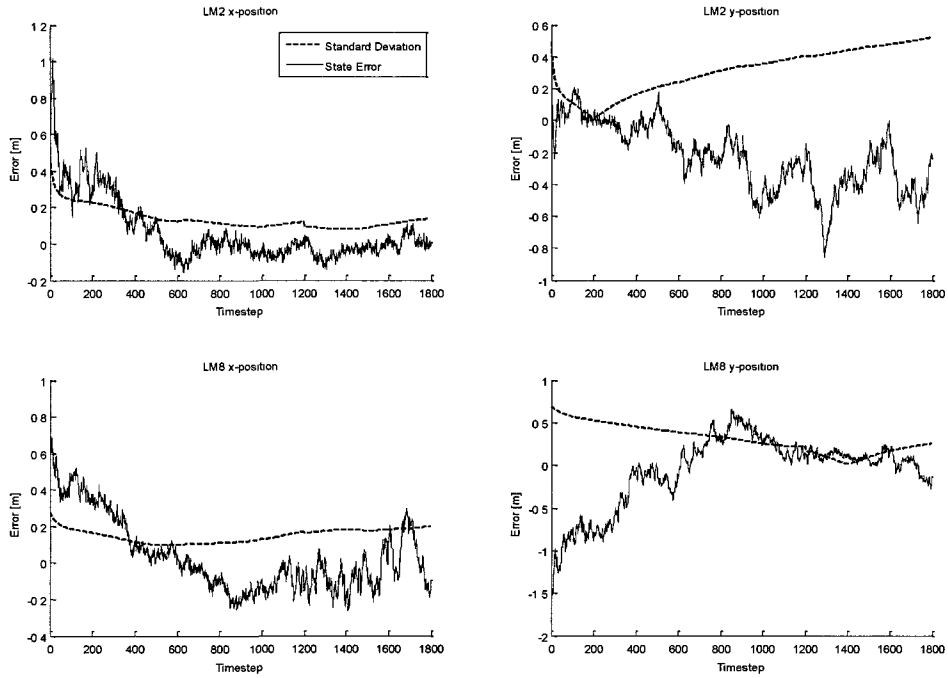

Figure 4.17: Case 4.5: Landmark position estimation.

Case 4.6

This simulation uses the circular ladder environment of Figure 4.2 beginning with $s(1) = L3 = (30, 40)$ and $s(2) = L4 = (50, 40)$. The re-selection schedule is listed in Table 4.5. Describing the circle as a clock with 12:00 at the top in the diagram, when the vehicle is at 1:30 $s(2)$ is re-selected from $L4$ to $L7$, when the vehicle is at 10:30 $s(1)$ is re-selected from $L3$ to $L8$, when the vehicle is at 7:30 $s(1)$ is re-selected from $L8$ back to $L3$, and when the vehicle is at 4:30 $s(2)$ is re-selected from $L7$ back to $L4$. This re-selection pattern is repeated for the second pass of the circle. Simulation results are shown in Table 4.5 and Figures 4.18-4.20.

Figure 4.18 shows the true and estimated environment and vehicle trajectory. This plot also shows the re-selection events with an "X" on the vehicle trajectory for re-selection of $s(2)$ and a "+" for re-selection of $s(1)$. On the scale shown in this Figure, good performance is seen in estimating the vehicle pose and landmark positions.

Table 4.5 shows the error introduced at each re-selection event. The most significant errors appear to be introduced when re-selecting $s(2)$. Re-selection errors during the second pass of the circle are larger than those of the first pass as more and more errors are accumulated in the system.

Figure 4.19 shows estimation errors for the vehicle pose over this simulation. These errors are reasonably small during the first pass of the circle, but one can see that during the second pass the errors oscillate with increasing amplitude. Vehicle heading is also biased in the second pass of the circle. Figure 4.17 shows estimation results for landmarks $L5 = (0, 70)$ and for landmark $L6 = (0, 90)$ that lie at the top of the circle. During the second pass of the circle, the estimation errors drift due to errors introduced during re-selection.

Table 4.5: Case 4.6: Re-selection errors for special landmarks. Errors for the constraints are true-estimated (m).

Timestep	Event	x-error	y-error
942	$s(2)$ from $L4$ to $L7$	–	-0.315
1571	$s(1)$ from $L3$ to $L8$	-0.002	-0.394
2199	$s(1)$ from $L8$ to $L3$	-0.127	0.400
2827	$s(2)$ from $L7$ to $L4$	–	0.731
3456	$s(2)$ from $L4$ to $L7$	–	-0.819
4084	$s(1)$ from $L3$ to $L8$	-0.196	-1.203
4712	$s(1)$ from $L8$ to $L3$	0.195	0.221
5341	$s(2)$ from $L7$ to $L4$	–	0.649

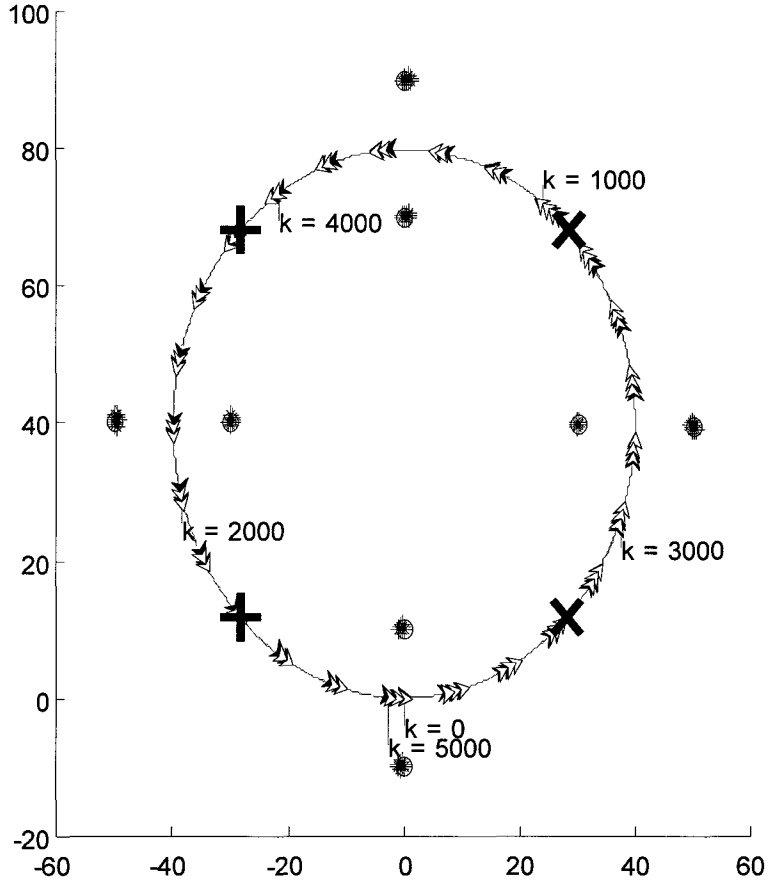

Figure 4.18: Case 4.6: Estimated vehicle trajectory and landmark positions.

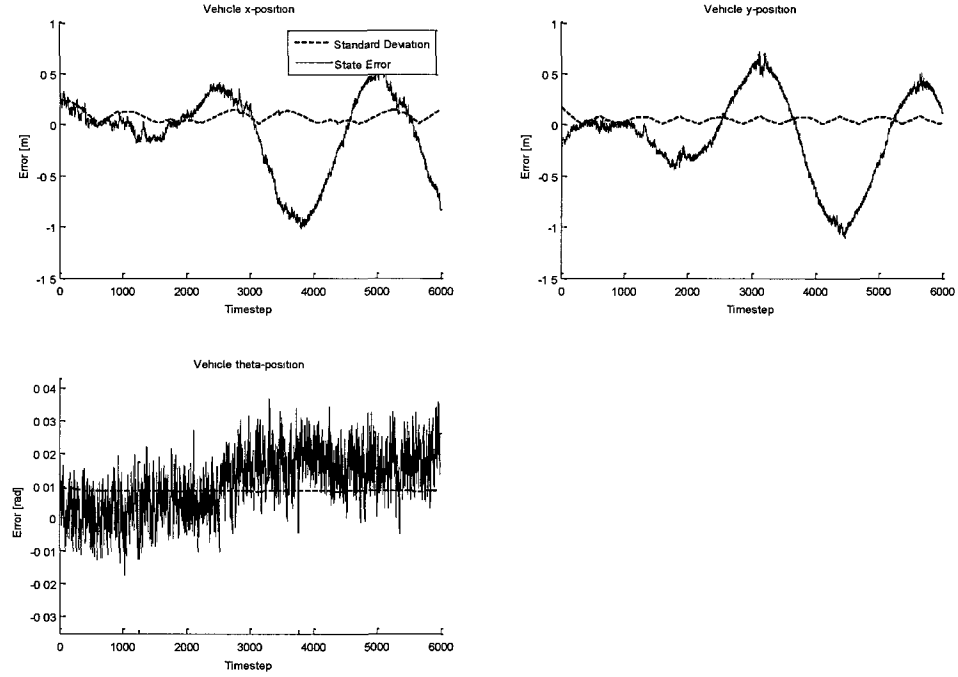

Figure 4.19: Case 4.6: Vehicle pose estimation.

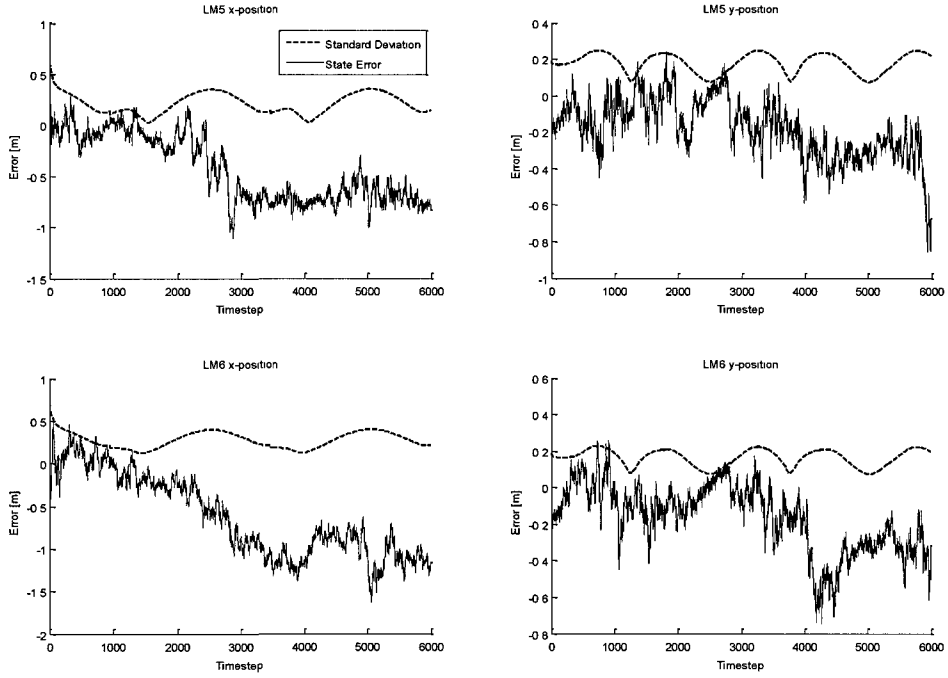

Figure 4.20: Case 4.6: Landmark position estimation.

4.4 SLAM in Dynamic Environments

Case 4.7

This simulation uses the straight ladder environment of Figure 4.1 with all landmarks initially stationary. The special landmarks are $s(1) = L1$ and $s(2) = L3$. At each timestep, following the Kalman prediction stage, all stationary landmarks are checked for possible motion using the hypothesis testing approach reviewed in Section 2.5. The hypothesis test is conducted at the 99.999 percent confidence level using the Mahalanobis distance threshold of 13.52. At 100 timesteps, landmark $L2$ begins to move on a sinusoidal trajectory through the environment. True simulated motion of the landmark is implemented with a smooth control law unknown to the filters with a rise time of approximately 1sec to a step input of desired velocity. This motion is detected, and the landmark is moved from the SLAM EKF to the MLF in order to track position, velocity and acceleration of the moving landmark using Singer's model for the remainder of the trajectory. Simulation results are shown in Figures 4.21-4.25.

Figure 4.21 shows the true and estimated environment and vehicle pose, as well as the motion of landmark $L2$. The SLAM EKF remains observable throughout the trajectory with no re-selection of special landmarks. Good estimation performance is seen for the SLAM EKF similar to the results for Case 4.1. Figure 4.22 shows the vehicle pose estimation for this trajectory. Again the results are comparible to those for Case 4.1, as are the results for estimation of $L4$ and $L7$ position shown in Figure 4.23.

At timestep 100, landmark $L2$ starts to move in a sinusoidal trajectory. The motion is detected in the SLAM EKF at timestep 147. The delay in detection of the onset of motion is due in part to the smooth simulated motion of the landmark that requires approximately 1sec. (20 timesteps) to reach the desired speed. With very slow landmark motion, the prediction errors in the EKF may not be significant relative to measurement errors. Detection of motion with a very high confidence level (99.999 percent) also results in some delay. The resulting delay in motion detection is on the order of 1sec.

Once motion is detected, landmark $L2$ is removed from the SLAM EKF and added to the MLF using the current estimate of the landmark postition and covariance. Figure 4.24 shows the true and estimated landmark position derived within the MLF using Singer's model. Only slight lag and overshoot is seen over the sinusoidal trajectory which indicates that the MLF is operating correctly. Figure 4.25 shows estimation errors for position and velocity of the moving landmark in the MLF. A small bias is visible in position estimation due to expected lag in the filter, but velocity estimates are close to zero mean. Estimation errors remain within the one-sigma Kalman standard deviations and indicate that the MLF is operating correctly. Note that errors in the SLAM EKF can propagate to the MLF because the current vehicle pose estimated in the SLAM EKF is used directly within the MLF. The converse is not true because errors within the MLF do not propagate into the SLAM EKF.

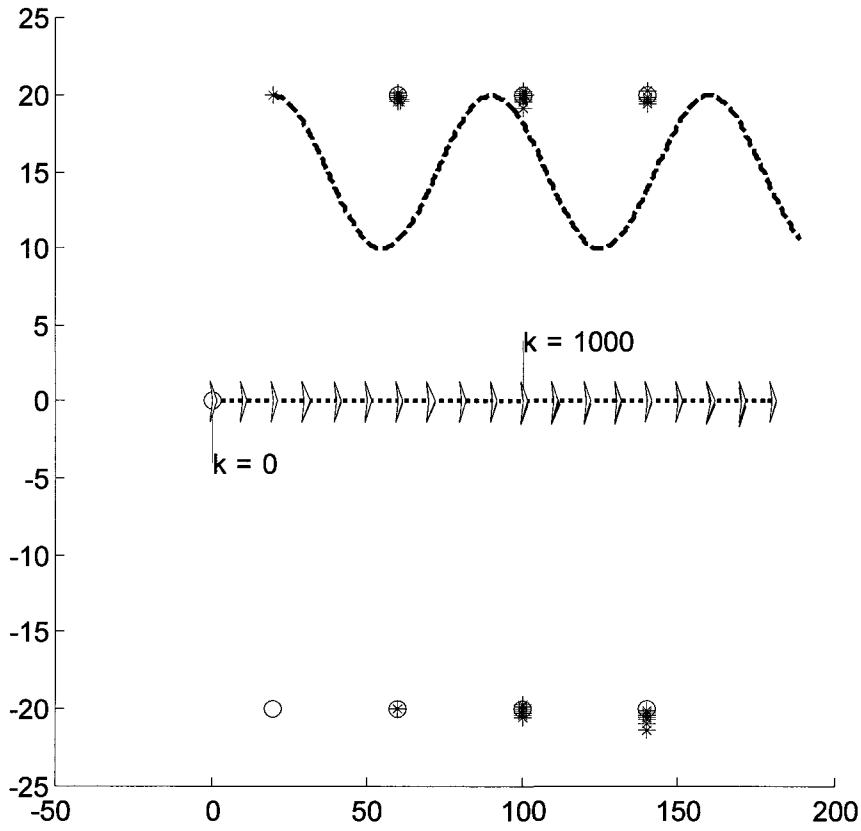

Figure 4.21: Case 4.7: Estimated vehicle trajectory and landmark positions.

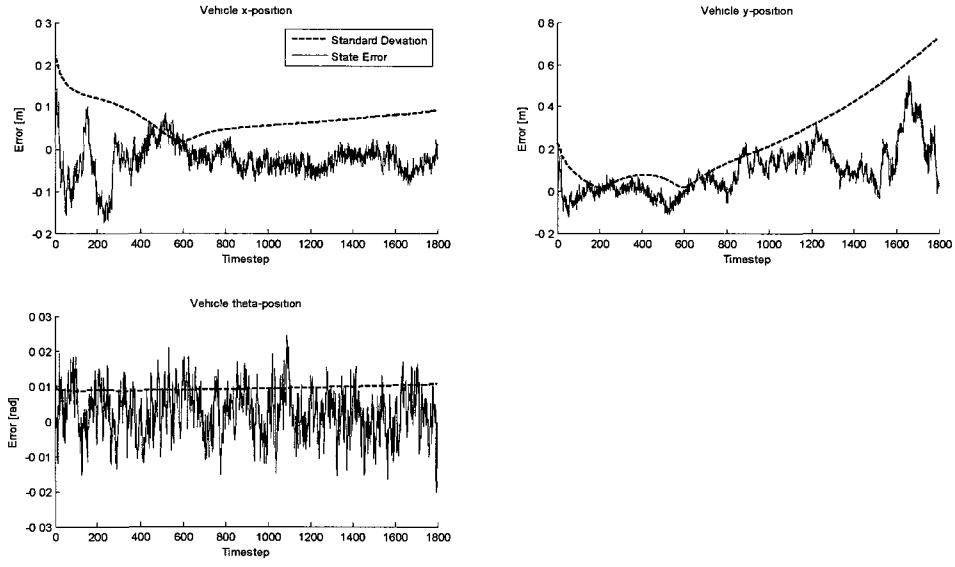

Figure 4.22: Case 4.7: Vehicle pose estimation errors.

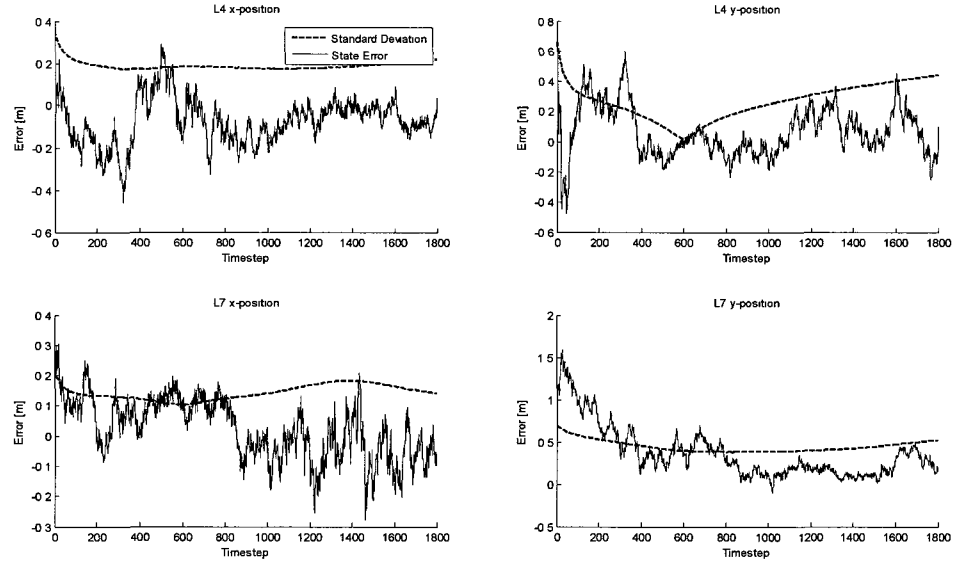

Figure 4.23: Case 4.7: Landmark estimation errors.

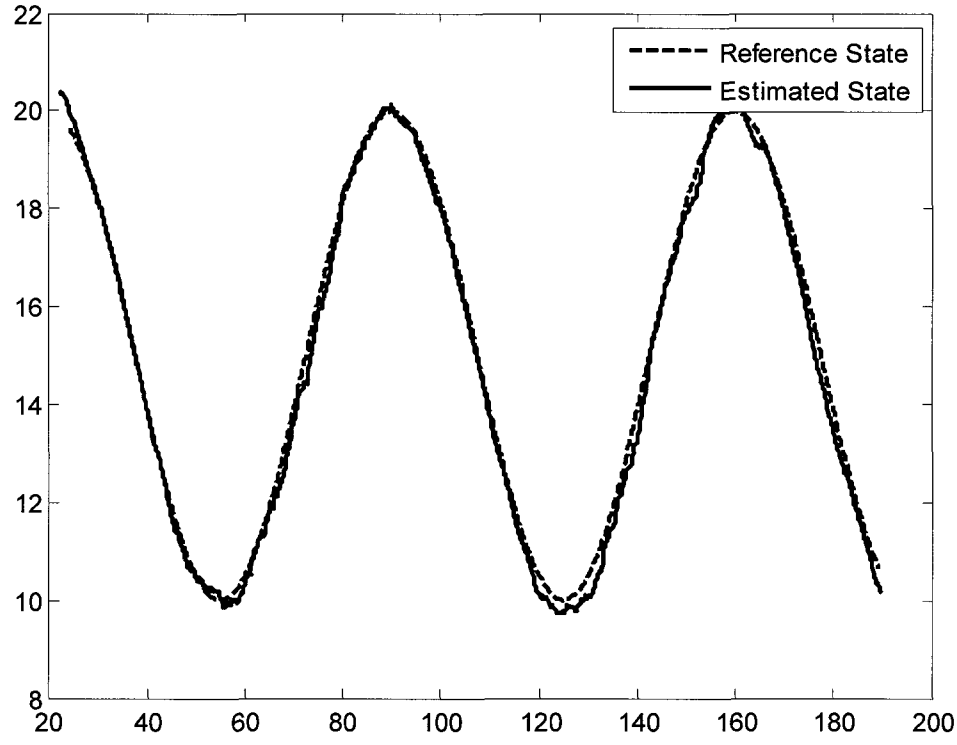

Figure 4.24: Case 4.7: Moving landmark true and estimated position.

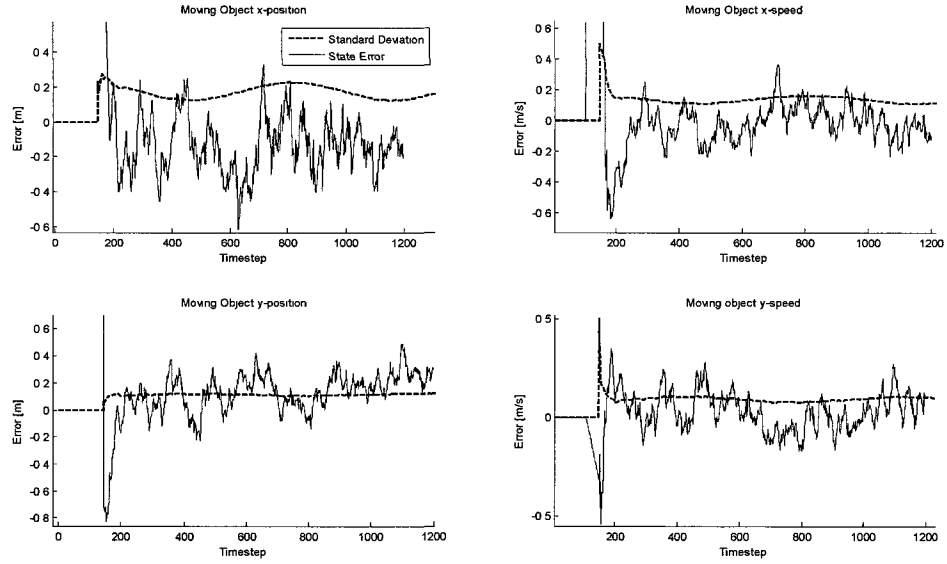

Figure 4.25: Case 4.7: Moving landmark position and velocity estimation errors.

Case 4.8

This simulation uses the straight ladder environment of Figure 4.1 with landmarks $L1$ and $L3$-$L8$ stationary throughout the run. The special landmarks are $s(1) = L1$ and $s(2) = L3$. Landmark $L2$ is initialized within the MLF with the same initial position and sinusoidal trajectory used in Case 4.7. Landmark $L2$ is brought to a stop at timestep 500 (25 sec.). At each timestep, following the Kalman correction stage (note that this was the prediction stage in Case 4.7), all moving landmarks within the MLF are checked for possible stop conditions using methods reviewed in Section 2.5. In this simulation, a landmark is determined to have transitioned from moving to stationary if the estimated landmark speed has magnitude less than 10cm/s for 40 consecutive timesteps (2sec.). These parameters were found to produce reliable results following numerous trials. Once a landmark within the MLF is determined to have stopped, that landmark is deleted from the MLF and moved to the SLAM EKF as a general stationary landmark. Current estimates of the landmark's position and covariance at the time of detection within the MLF are transferred to the SLAM EKF for initialization purposes. Simulation results are shown in Figures 4.26-4.30.

Figure 4.26 shows the true and estimated environment and vehicle pose, as well as the motion of landmark $L2$. The SLAM EKF, that is initialized with only stationary landmarks but is augmented with $L2$ following stop detection, remains observable throughout the trajectory with no re-selection of special landmarks. Good estimation performance is seen for the SLAM EKF similar to the results for Case 4.1. Figure 4.27 shows the vehicle pose estimation for this trajectory. Again the results are comparible to those for Case 4.1.

At timestep 500, landmark $L2$ is brought to zero speed. At this time, the vehicle is at position $(50, 0)$m. Figure 4.28 shows the true and estimated positions of $L2$ while motion is being estimated within the MLF. There are initial transients in the MLF, however the estimates closely follow the true trajectory. One can note the larger estimation errors in the bearing channel once the landmark is stopped due to the relative position of the vehicle and landmark at this time. Figure 4.29 shows the estimation errors for $L2$ position and velocity while this landmark is within the

MLF. The landmark is determined to have stopped at timestep 648 and is moved to the SLAM EKF. Figure 4.30 shows position estimation errors for $L2$ after it has been moved to the SLAM EKF. The position of $L2$ after it has stopped is approximately $(60, 10)$m in the environemnt. Thereafter the vehicle continues to move to the right, and hence estimation accuracy in the y component of $L2$ is expected to be reduced as indicated by the sharp increase in Kalman standard deviation.

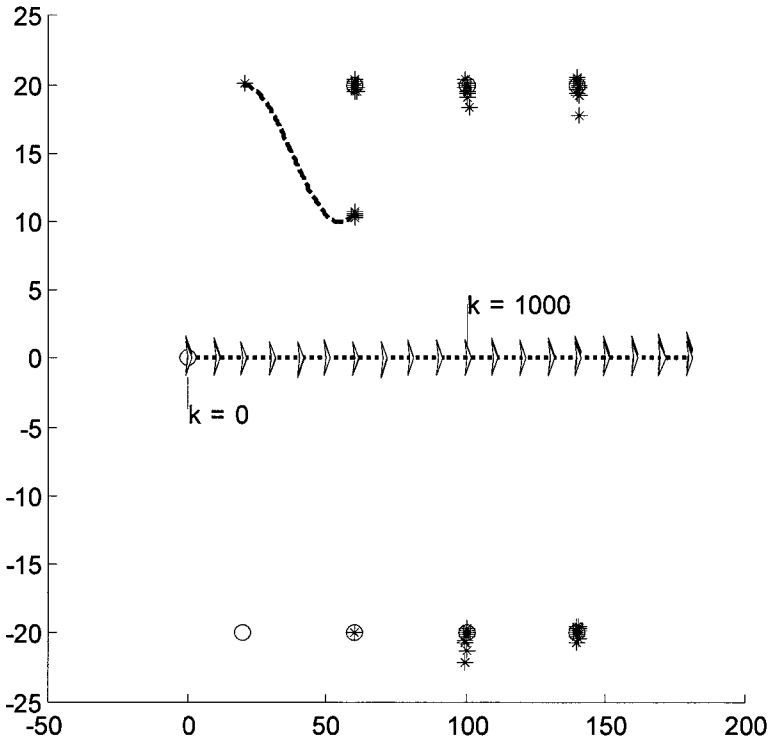

Figure 4.26: Case 4.8: Estimated vehicle trajectory and landmark positions.

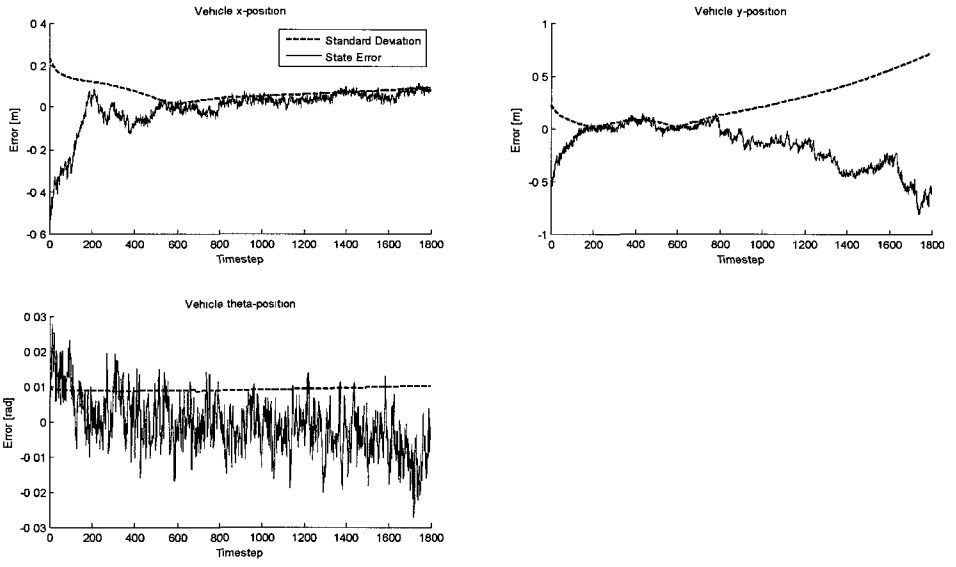

Figure 4.27: Case 4.8: Vehicle pose estimation errors.

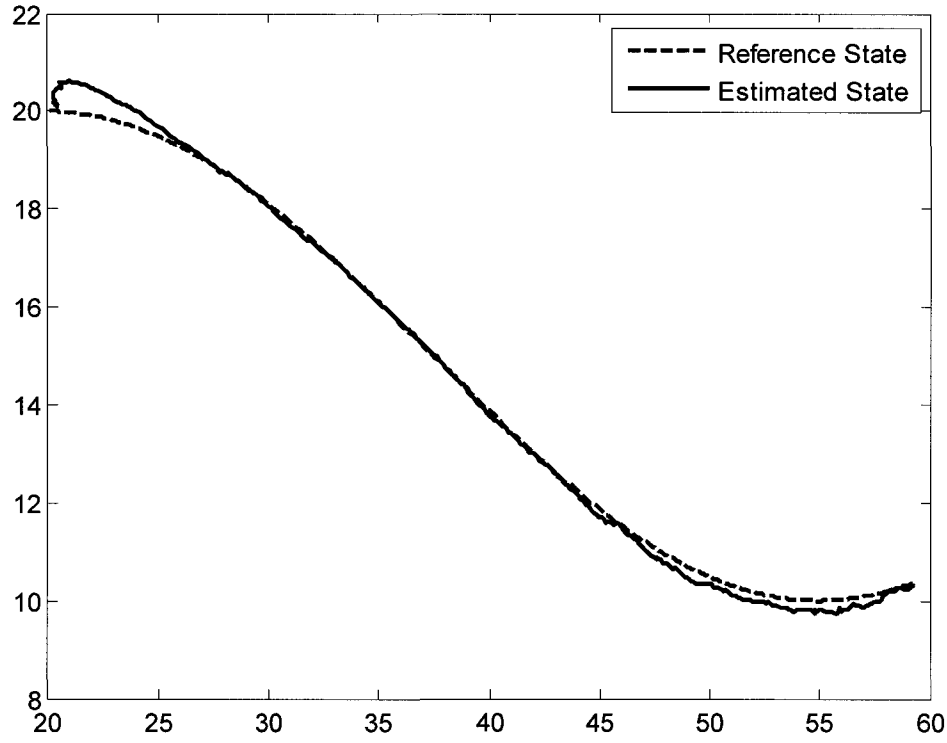

Figure 4.28: Case 4.8: Moving landmark true and estimated position.

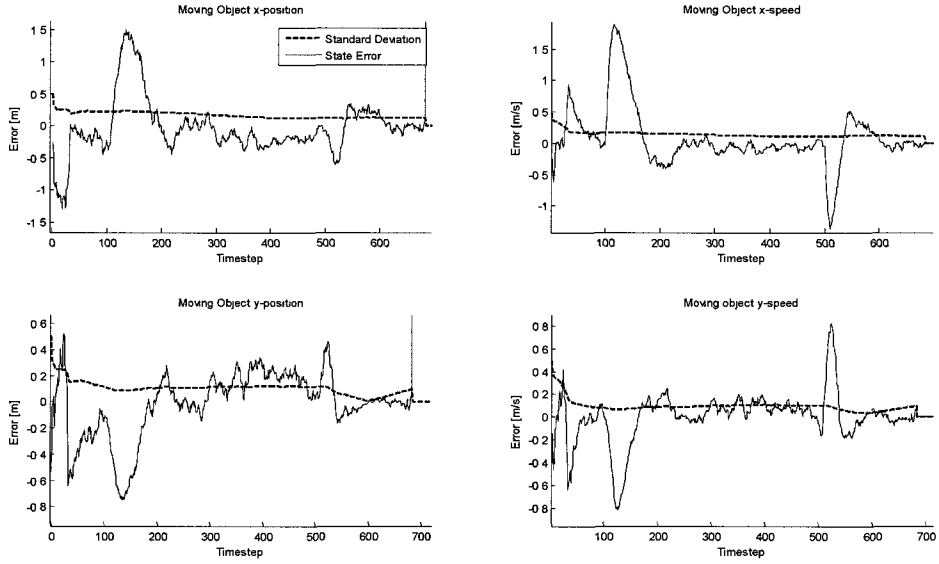

Figure 4.29: Case 4.8: MLF Moving landmark position and velocity estimation errors.

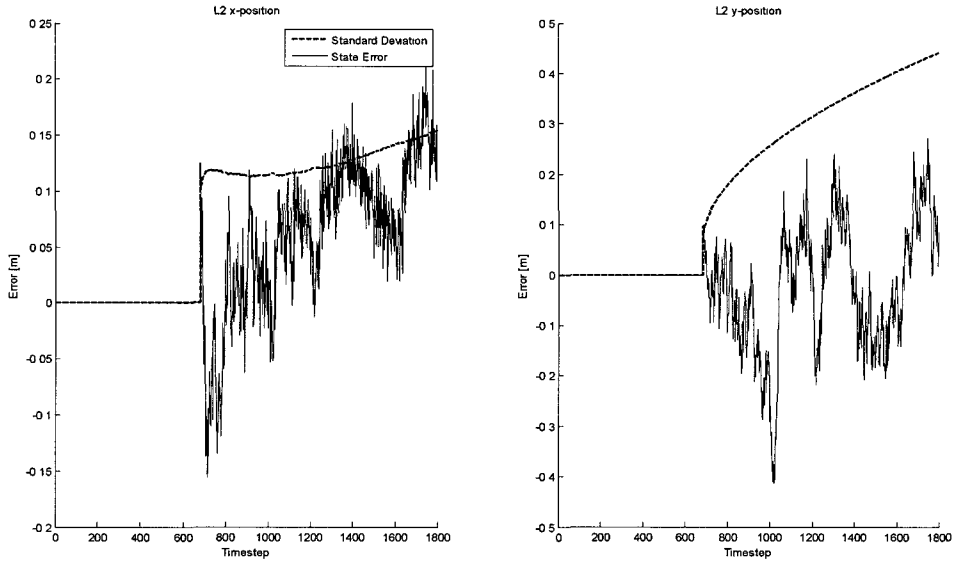

Figure 4.30: Case 4.8: SLAM EKF stationary estimation errors after stop detection.

4.5 Summary

This Chapter presents results for eight simulation Cases that investigate the behaviour of the proposed solution for observable autonomous SLAM in 2D dynamic environments. Two environments are used in the simulations: the straight ladder environment uses simple vehicle motion and is used to assess estimation performance as the vehicle moves large distances from the special landmark pair; the circular ladder environment adds complexity to vehicle motion with changing heading and is used to assess repeatability of results.

Section 4.2 shows three test Cases for static environments without re-selection of special landmarks. In these Cases the system remains observable and good estimation performance is demonstrated. However, as the vehicle moves farther from the special landmark pair estimation performance is degraded. As a result the proposed method cannot be expected to yield acceptable results in very large environments over long time periods. This issue is not specific to the proposed three-constraint solution in that similar problems would be encountered under the assumption that all four coordinates of two special landmarks are measured in advance.

Section 4.3 shows three test Cases in which the special landmarks are re-selected during the trajectory. The focus of these investigations was on bias introduced through the re-selection process. Although re-selection can reduce the Kalman covariance and estimation drift as the vehicle moves farther from the original special landmarks, bias introduced through multiple re-selections can accumulate in the filter. In practice, one may have little choice but to re-select the special landmarks if either or both start moving, or become occluded from sensor view, or simply are too far from the vehicle to give reasonable measurements. Further research is needed to fully develop and evaluate re-selection strategies, including information loss due to special landmark orientation.

Section 4.4 shows two test Cases that demonstrate the feasibility of operating two Extended Kalman Filters in parallel, the SLAM EKF for vehicle pose and static landmark positions, and the Moving Landmark Filter (MLF) for estimating position and motion of moving landmarks. The vehicle pose estimated within the SLAM EKF is used directly in the MLF. The SLAM EKF includes some uncertainty in the

dynamic model for stationary landmarks and hence this filter could track very slow motion of general landmarks without losing observability. However, special landmark $s(1)$ for which both coordinates are fixed must remain stationary and one coordinate of $s(2)$ must be constant, otherwise the solution is seriously degraded. The parallel structure with two EKFs presents a more reliable method to retain accuracy in the SLAM EKF and also to estimate more general motion of moving landmarks. The two Cases shown demonstrate that motion detection can be accomplished using hypothesis testing and the Mahalanobis distance, and that stop detection can be implemented with a threshold test and delay.

Chapter 5

Conclusions and Future Work

A fully autonomous robot is one that must be able to perform SLAM. The ability to solve SLAM online in real-time will allow robots to function for longer periods of time without human control. In the autonomous SLAM problem formulated in the literature, measurements of the environment are only available to the robot from vehicle-mounted sensors. For example, a laser or other vision system mounted on the robot can provide range and bearing measurements of environmental landmarks relative to the vehicle. Speed or intertial navigation sensors may also be available on the robot. However, absolute robot pose measurements relative to a pre-defined world reference system (such as might be provided by GPS) and absolute world-centric measurements of landmark positions are not included in the autonomous SLAM problem. All measurements are therefore vehicle-centric. The objective of autonomous SLAM is then to construct a world-centric map of the environment and to maintain an estimate of vehicle pose and motion within this world-centric map.

Chapter 1 provides a review of relevant literature. The SLAM problem is generally posed as a state estimation problem for dynamic systems. Many different SLAM techniques fail to consider the problem of observability for the autonomous problem. The autonomous SLAM problem as posed is not completely observable and, as a result, bounded state estimation errors cannot be guaranteed. In dynamic environments, landmarks may be stationary or moving, or may transition from stationary to moving or from moving to stationary. SLAM for dynamic environments must

therefore include capabilities in motion detection and tracking of moving landmarks relative to the stationary landmarks that define the environmental map. The robot pose and motion state and the environmental map state together form the complete state for the dynamic system that is updated with each measurement set received.

This thesis focusses on the two-dimensional planar autonomous SLAM problem for dynamic environments and a solution based on the Extended Kalman Filter. All measurements are vehicle-centric, and no prior environmental information is available to the robot. Landmark position measurements relative to the vehicle are provided in the form of range and bearing. The vehicle pose and motion state and the environmental state are estimated relative to a world-centric reference frame that is not specified in advance, but is defined by the estimation algorithm.

Chapter 2 introduces notation for the SLAM problem in the context of state estimation. Two different types of unicycle models and a range-and-bearing sensor are developed to represent the mobile sensing system. Stationary landmarks are modeled with some uncertainty in motion in order to detect possible transitions from stationary to moving. Landmark motion is modeled using Singer's approach that is well-known in the target tracking literature. Background theoretical information is provided on the important tools used in this thesis. The Kalman Filter is an optimal predictor-corrector algorithm that forms the basis for the Extended Kalman Filter that permits a SLAM state estimation algorithm in a recursive on-line implementation. State estimation requires that the system, as posed, satisfies observability conditions in order for information to be accumulated in the filter. The concept of observability in terms of indistinguishability is reviewed for nonlinear discrete-time systems. Fisher's Information Matrix and the Cramer-Rao Lower Bounds are important tools in state estimation to assess both observability and accuracy. Hypothesis testing based on the Mahalanobis distance is reviewed for detection of landmark transitions from stationary to moving, whereas a threshold test on velocity estimates is used to determine when a landmark transitions from moving to stationary. The set of model parameters used in simulation studies is selected to be representative of SLAM systems in noisy environments.

Chapter 3 first demonstrates that there are three degrees of freedom in the

autonomous SLAM problem. Simulation results for three unobservable formulations demonstrate that the filters cannot correct for initialization errors. A current method to enforce observability by assuming that the positions of two landmarks in the environment are known in advance gives decreasing Cramer-Rao Bounds, but does not satisfy the autonomous requirement. The proposed solution for the autonomous problem results in observability with Cramer-Rao bounds slightly above those for the two-known-landmark solution. The proposed solution depends on the presence of two stationary landmarks in the environment, and can be initialized autonomously. As the vehicle roams through relatively large environments, however, filter accuracy can be seriously degraded as the vehicle moves farther away from the special landmark pair. Moreover, special landmarks may have to be re-selected if they transition from stationary to moving, or become occluded from sensor view. The proposed parallel structure of the SLAM EKF and Moving Landmark EKF accommodates dynamic re-arrangement of states in order to adapt to a changing environment. Issues of observabilty and accuracy when re-selection of special landmarks is required are assessed. In particular it is shown that singular orientations of special landmarks can seriously degrade filter information and performance.

Chapter 4 presents results for eight simulation Cases that investigate the behaviour of the proposed solution for observable autonomous SLAM in 2D dynamic environments. Two environments are used in the simulations: the straight ladder environment involves simple vehicle motion and is used to assess estimation performance as the vehicle moves large distances from the special landmark pair; the circular ladder environment adds complexity to vehicle motion with changing heading and is used to assess repeatability of results. For static environments without re-selection of special landmarks the system remains observable and good estimation performance is demonstrated. However, as the vehicle moves farther from the special landmark pair estimation performance is degraded. As a result the proposed method cannot be expected to yield acceptable results relative to the initial world-centric frame in very large environments over long time periods. Simulations of Chapter 4 investigate possible bias that can be introduced through the re-selection process. Although re-selection can reduce the Kalman covariance and estimation drift as the

vehicle moves farther from the original special landmarks, bias introduced through multiple re-selections can accumulate in the filter. Final simulations in Chapter 4 demonstrate the feasibility of operating two Extended Kalman Filters in parallel, the SLAM EKF for vehicle pose and static landmark positions, and the Moving Landmark Filter (MLF) for estimating position and motion of moving landmarks. The vehicle pose estimated within the SLAM EKF is used directly in the MLF. The parallel structure with two EKFs presents a reliable method to retain accuracy in the SLAM EKF and also to estimate general motion of moving landmarks.

Further research is needed to fully develop and evaluate the mothods proposed in this thesis. The following items are promising extensions of the thesis work:

- Simulation studies throughout the thesis use the popular simple unicycle model for vehicle motion. Chapter 2 describes how the vehicle model can be easily extended to include vehicle speed and turning rate in the SLAM state vector. Although vehicle turning rate, especially in 2D environments, can be accurately measured with vehicle sensors, on-board speed measurements typically suffer from wheel slip or skidding. Testing of the proposed methods with the extended unicycle model in order to estimate vehicle motion from the SLAM solution for stationary landmarks, with or without on-board measurements, could assist or replace other non-contact ground speed sensors currently in use (such as 5th wheels or downward-loooking video).

- Chapter 3 analyzed several important issues related to re-selection of special landmarks. The simulation results shown in Chapter 4 investigated possible accumulation of bias relative to the initial world-centric frame due to multiple re-selection events. Information loss due to special landmark orientation was demonstrated in Chapter 3, but has not been fully investigated in simulations. Further work is needed to assess and improve the re-selection strategy suggested in Chapter 3. The capability to intelligently re-select these landmarks is important to filter accuracy in large environments.

- Methods to detect motion or stopping of landmarks were reviewed in Chapter 2 and tested in Chapter 4. The accuracy of the SLAM EKF is most sensitive

to motion in its special landmarks. Hence more sensitive motion detection for these landmarks is needed. General "stationary" landmarks could remain within the SLAM EKF with very slow/smooth landmark motion, so motion detection could be relaxed somewhat for these general landmarks as opposed to the special landmarks. Stop detection within the MLF is not critical to filter performance, but is used to reduce computational complexity of the MLF implementation. Adaptation of the parallel EKFs to the dynamic environment needs further development and testing, particularly in tuning of the filters and selection of detection parameters to achieve reliable performance.

- In a statistical sense, the simulation results of Chapter 4 are exploratory rather than conclusive. Although the example results shown are representative of those produced over a large number of trial runs, more complete simulation studies of the proposed methods are needed. Monte-Carlo simulations could be conducted with statistics collected over many trial runs to produce more conclusive findings on filter performance.

- Results of Chapter 4 demonstrate that accurate local maps can be produced with observable state estimation when re-selection is not required. Re-selection enforces new observability constraints on the system that can produce an accurate environmental map relative to the re-selected world-centric frame, but with some bias relative to the original world-centric frame. This suggests that multiple local maps could be produced using either the same vehicle or multiple vehicles. Fusion of multiple local maps to form a large map could then be investigated.

- This thesis considered a 2D environment. Although 2D environments have many indoor applications, extensions of the problem and proposed solution to 3D environments is needed to explore outdoor unstructured terrain. In 2D, three observability constraints can be defined arbitrarily (all zeros) to yield an observable solution and good estimation accuracy. In 3D, the number of observability constraints needed is increased to six [2]. Models for 3D vehicle dynamics and 3D measurements of the environment would have to be

developed to support extensions to the 3D problem.

Landmark detection in raw sensor data, the data association problem, dynamical modelling of unmanned vehicles, various formulations of the SLAM problem, and other implementations of state estimation algorithms continue to be explored in current research throughout the world. The end goal is implementaton and testing with a real vehicle for navigation through and exploration of unknown real environments.

References

[1] Raja Chatila Abelkarim Souici, Abdelaziz Ouldali. Observable formulation slam implementation. *Robot Motion and Control*, 396:339–348, 2009.

[2] V.C. Aitken. *Sliding Mode State Estimation for Nonlinear Discrete-time Systems: Applications in Image Sequence Analysis.* PhD thesis, Carleton University, Ottawa, April 1995.

[3] V.C. Aitken. On the 2d discrete-time slam observability problem, unplublished research notes, November 2010.

[4] Victor C. Aitken and Howard M. Schwartz. A comparison of rotational representations in structure and motion estimation for maneuvering objects. *IEEE Tranactions on Image Processing*, 4:516–520, April 1995.

[5] J. Andrade-Cetto, T. Vidal-Calleja, and A. Sanfeliu. Unscented transformation of vehicle states in slam. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 323 – 328, April 2005.

[6] Juan Andrade-Cetto and Alberto Sanfeliu. The effects of partial observability in slam. *International Conference on Robotics and Automation*, pages 397–402, 2004.

[7] Yaakov Bar-Shalom, Thiagalingam Kirubarajan, and X.-Rong Li. *Estimation with Applications to Tracking and Navigation.* John Wiley & Sons, Inc., New York, NY, USA, 2002.

[8] Adrian N. Bishop and Patric Jensfelt. A stochastically stable solution to the problem of robocentric mapping. In *ICRA'09: Proceedings of the 2009 IEEE international conference on Robotics and Automation*, pages 1540–1547, Piscataway, NJ, USA, 2009. IEEE Press.

[9] Howie Choset, Kevin M. Lynch, Seth Hutchinson, George A. Kantor, Wolfram Burgard, Lydia E. Kavraki, and Sebastian Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementations.* MIT Press, Cambridge, MA, June 2005.

[10] Michael Csorba. *Simultaneous Localization and Map Building.* PhD thesis, Robotics Research Group, Department of Engineering Science, University of Oxford, Oxford England, 1997.

[11] Samuel. Davey, Defence Science, and Technology Organisation (Australia). *Simultaneous localisation and map building using the probabilistic multi-hypothesis tracker.* DSTO Information Sciences Laboratory, Edinburgh, S. Aust., 2005.

[12] S. Majumder H. Durrant-Whyte S. Thrun M. de Battista. An approximate bayesian method for simultaneous localization and mapping. *IEEE Transactions on Robotics and Automation (submitted)*, 2004.

[13] M. W. M. G. Dissanayake, P. Newman, Hugh F. Durrant-Whyte, Steve Clark, and M. Csorba. An experimental and theoretical investigation into simultaneous localisation and map building. In *The Sixth International Symposium on Experimental Robotics VI*, pages 265–274, London, UK, 2000. Springer-Verlag.

[14] M.W.M.G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (slam) problem. *Robotics and Automation, IEEE Transactions on*, 17(3):229 –241, June 2001.

[15] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localisation and mapping (slam): Part i the essential algorithms. *IEEE Robotics and Automation Magazine*, 2:2006, 2006.

[16] I. Esteban, O. Booji, Zivcovik Z., and Krose B. Slam for extremely large environments. *ASCI08*, 2008.

[17] R.M. Eustice, H. Singh, J.J. Leonard, and M.R. Walter. Visually mapping the RMS Titanic: Conservative covariance estimates for SLAM information filters. *International Journal of Robotics Research*, 25(12):1223–1242, December 2006.

[18] G.F. Franklin. *Digital Control of Dynamic Systems.* Addison-Wesley, 1997.

[19] A. Gelb. *Applied optimal estimation.* MIT Press, 1974.

[20] Simon Haykin. *Kalman Filtering and Neural Networks.* Wiley-Interscience, October 2001.

[21] Matthew Walter Franz Hover and John Leonard. Slam for ship hull inspection using exactly sparse extended information filters. *2008 IEEE International Conference on Robotics and Automation*, May 2008.

[22] Guoquan P. Huang, Anastasios I. Mourikis, and Stergios I. Roumeliotis. Analysis and improvement of the consistency of extended kalman filter based slam. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 473–479, May 2008.

[23] Guoquan P. Huang, Anastasios I. Mourikis, and Stergios I. Roumeliotis. On the complexity and consistency of ukf-based slam. In *Proceedings of the 2009 IEEE international conference on Robotics and Automation*, ICRA'09, pages 3878–3885, Piscataway, NJ, USA, 2009. IEEE Press.

[24] Guoquan P. Huang, Anastasios I. Mourikis, and Stergios I. Roumeliotis. Observability-based rules for designing consistent ekf slam estimators. *Int. J. Rob. Res.*, 29(5):502–528, 2010.

[25] Simon J. Julier and Jeffrey K. Uhlmann. A new extension of the kalman filter to nonlinear systems. In *SPIE AeroSense Symposium*, pages 182–193, 1997.

[26] A.G. Kallapur, I.R. Petersen, and S.G. Anavatti. A discrete-time robust extended kalman filter for uncertain systems with sum quadratic constraints. *Automatic Control, IEEE Transactions on*, 54(4):850 –854, April 2009.

[27] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[28] Choyong Koperski. *Multirobot FastSLAM for Large Domains.* PhD thesis, Department of the Air Force, Air Force Institute of Technology, Wright-Patterson Air Force Base Ohio, 2007.

[29] Kwang W. Lee, W. S. Wijesoma, and I. G. Javier. On the observability and observability analysis of slam. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 3569–3574, Oct 2006.

[30] Joshua Marshall. *Lecture Notes: MECH 5502 - Guidance Navigation and Control.* Carleton University, Ottawa, 2008.

[31] Luis Montesano Javier Minguez and Luis Montano. Modeling the static and the dynamic parts of the environment to improve sensor-based navigation. *International Conference on Robotics and Automation*, April 2005.

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

[33] Michael Montemerlo and Sebastian Thrun. Simultaneous localization and mapping with unknown data association using fastslam. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, volume 2, pages 1985–1991, September 2003.

[34] Michael Montemerlo, Sebastian Thrun, and William Whittaker. Conditional particle filters for simultaneous mobile robot localization and people-tracking. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 695–701, 2002.

[35] Guoquan P. Huang Anastasios I. Mourikis and Stergios I. Roumeliotis. Generalized analysis and improvement of the consistency of ekf-based slam. *Multiple Autonomous Robotic Systems Laboratory*, January 2008.

[36] Arthur G. O. Mutambara. *Decentralized Estimation and Control for Multisensor Systems.* CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 1998.

[37] Jose Neira and Juan D. Tardos. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17:890–897, 2001.

[38] Paul Newman and Kin Ho. Slam loop closing with visually salient features, 2005.

[39] Paul Michael Newman. *On the Structure and Solution of the Simultaneous Localisation and Map Building Problem.* PhD thesis, Australian Centre for Field Robotics The University of Sydney, Sydney Australia, March 1999.

[40] Fredrik Orderud. Comparison of kalman filter estimation approaches for state space models with nonlinear measurements, 2005.

[41] K.V. Ramachandra. *Kalman Filtering Techniques for Radar Tracking.* Dekker Publishing, New York, 1 edition, 2000.

[42] Ioannis M. Rekleitis. A particle filter tutorial for mobile robot localization. *Center for Intelligent Machines*, 2002.

[43] Robert Sim. Active exploration planning for slam using extended information filters. In *In Proc. 20th Conf. Uncertainty in AI*, 2004.

[44] R. Singer. Estimating Optimal Tracking Filter Performance for Manned Maneuvering Targets. *IEEE Transactions on Aerospace Electronic Systems*, 6:473–483, July 1970.

[45] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. *Autonomous robot vehicles*, pages 167–193, 1990.

[46] Randall C. Smith and Peter Cheeseman. On the representation and estimation of spatial uncertainly. *Int. J. Rob. Res.*, 5(4):56–68, 1987.

[47] J.H. Taylor. The cramer-rao estimation error lower bound computation for deterministic nonlinear systems. *IEEE Trans. Automatic Control*, 24(2):343–344, 1979.

[48] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141, 2000.

[49] Sebastian Thrun. Probabilistic algorithms in robotics. *AI Magazine*, 21:93–109, 2000.

[50] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Auton. Robots*, 5:253–271, July 1998.

[51] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics.* Intelligent robotics and autonomous agents. MIT Press, September 2005.

[52] Sebastian Thrun, Daphne Koller, Zoubin Ghahramani, Hugh Durrant-Whyte, and Andrew Ng. Simultaneous mapping and localization with sparse extended information filters: Theory and initial results. *WAFR*, 2002.

[53] Sebastian Thrun, Michael Montemerlo, Daphne Koller, Ben Wegbreit, Juan Nieto, and Eduardo Nebot. Fastslam: An efficient solution to the simultaneous localization and mapping problem with unknown data association. *Journal of Machine Learning Research*, 2004.

[54] Chi Hay Tong. *Consistency Comparisons of EKF-SLAM with Investigations into Robust EKF-SLAM.* PhD thesis, Department of Engineering, University of Toronto, 2008.

[55] Matthew R. Walter, Ryan M. Eustice, and John J. Leonard. Exactly sparse extended information filters for feature-based slam. *Int. J. Rob. Res.*, 26:335–359, April 2007.

[56] Chieh-Chih Wang. *Simultaneous Localization, Mapping and Moving Object Tracking.* PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, April 2004.

[57] Chieh-Chih Wang, Charles Thorpe, and Sebastian Thrun. Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 842–849, 2003.

[58] Greg Welch and Gary Bishop. An introduction to the kalman filter. Technical report, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 1995.

[59] Jian xun Li and Zhong liang Jing. Practical system for tracking multiple maneuvering targets. *Society of Photo-Optical Instrumentation Engineers*, 42:2439–2451, August 2003.

[60] Tun Yang. Mapping and localisation for mobile robots. Master's thesis, Carleton University, Ottawa, Ontario, Canada, 2004.