Collaborative Concurrent Mapping and Localization

by

John William Fenwick

B.S. Physics, United States Air Force Academy, 1999

Submitted to the Department of Electrical Engineering and Computer Science

in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2001

Author .....

Department of Electrical Engineering and Computer Science

May 21, 2001

Certified by .....

Dr. Michael E. Cleary

The Charles Stark Draper Laboratory, Inc.

Technical Supervisor

Certified by .....

Professor John J. Leonard

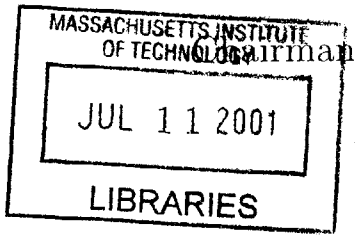
Associate Professor, Department of Ocean Engineering

Thesis Advisor

Accepted by

Professor A. C. Smith

Chairman, Department Committee on Graduate Students



ARCHIVES

[This page intentionally left blank]

Collaborative Concurrent Mapping and Localization

by

John William Fenwick

Submitted to the Department of Electrical Engineering and Computer Science
on May 21, 2001, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

Autonomous vehicles require the ability to build maps of an unknown environment while concurrently using these maps for navigation. Current algorithms for this concurrent mapping and localization (CML) problem have been implemented for single vehicles, but do not account for extra positional information available when multiple vehicles operate simultaneously. Multiple vehicles have the potential to map an environment more quickly and robustly than a single vehicle. This thesis presents a collaborative CML algorithm that merges sensor and navigation information from multiple autonomous vehicles. The algorithm presented is based on stochastic estimation and uses a feature-based approach to extract landmarks from the environment. The theoretical framework for the collaborative CML algorithm is presented, and a convergence theorem central to the cooperative CML problem is proved for the first time. This theorem quantifies the performance gains of collaboration, allowing for determination of the number of cooperating vehicles required to accomplish a task. A simulated implementation of the collaborative CML algorithm demonstrates substantial performance improvement over non-cooperative CML.

Technical Supervisor: Dr. Michael E. Cleary
Title: Member of the Technical Staff, The Charles Stark Draper Laboratory, Inc.

Thesis Advisor: Professor John J. Leonard
Title: Professor of Ocean Engineering

[This page intentionally left blank]

Acknowledgments

In spite of all my wishes, this thesis did not spontaneously write itself. I owe much thanks to those who helped and supported me through my time at MIT.

My parents, as always, provided steady support. Their gentle encouragement and pristine example provide all the direction I need for success in life.

Draper Laboratory provided the window of opportunity for higher education, and for this I will always be grateful. Dr. Michael Cleary provided much needed guidance to a novice engineer, giving me all the right advice at the right time. His patience with me was remarkable, especially considering that I always seemed to drop by at the most inopportune times. Mike's trust and confidence in my ability was the single most important element in making my MIT experience so rewarding.

Professor John Leonard is to be commended for his willingness to squeeze me into his already overloaded workload. He generously carved out a research niche for me and patiently let me follow it to completion. Many thanks also are extended to Dr. Paul Newman, who gave me innumerable tidbits of advice and guidance. Technical knowhow and lunch conversation dispensed by Rick Rikoski also helped keep me on the right track.

Thanks also to my roommates and all the other Air Force guys for helping me preserve my sanity. Red Sox games and beer were always a welcome study break. Finally, thanks to Nik, for putting everything in perspective and making my life so wonderful.

ACKNOWLEDGMENT

(21 May 2001)

This thesis was prepared at the Charles Stark Draper Laboratory, Inc., under Internal Company Sponsored Research Project 13024, Embedded Collaborative Intelligent Systems.

John W. Fenwick 21 *May* 2001

Contents

[This page intentionally left blank]

List of Figures

[This page intentionally left blank]

List of Tables

[This page intentionally left blank]

Chapter 1

Introduction

Successful operation of an autonomous vehicle requires the ability to navigate. Navigation information consists of positional estimates and an understanding of the surrounding environment. Without this information, even the simplest of autonomous tasks are impossible. An important subfield within mobile robotics that requires accurate navigation is the performance of collaborative tasks by multiple vehicles. Multiple vehicles can frequently perform tasks more quickly and robustly than a single vehicle. However, cooperation between vehicles demands the vehicle be aware of relative locations of collaborators in addition to the baseline environmental knowledge.

Current solutions to autonomous vehicle localization rely on both internal and external navigational aides. Internal navigation instruments such as gyros and odometers (on land vehicles) provide positional estimates, but are susceptible to drift and thus result in a navigation error that grows linearly with time. This unbounded error growth makes long-term autonomous operation using only internal devices impossible. Beacon systems (such as the Global Positioning System (GPS)) provide extremely accurate navigation updates, but require pre-placement of accessible beacons (satellites). In the case of GPS, this limits use to outdoor open-air environments.

Autonomous Underwater Vehicles (AUVs), as well as autonomous land vehicles operating indoors, are unable to utilize GPS as a result. Acoustic beacon arrays with known locations have been used successfully by AUVs for navigation, but deployment of such beacons is only feasible in a limited number of mission scenarios. AUVs have demonstrated the ability to localize their position and navigate using accurate *a priori* bathymetric maps, but *a priori* knowledge of the underwater environment is not always available, especially in littoral zones where the environment frequently changes in ways that would affect a shallow water AUV. A recent advance in autonomous vehicle navigation techniques, Concurrent Mapping and Localization (CML), incorporates environmental data to provide vehicle position information [46]. CML allows an autonomous vehicle to build a map of an unknown environment while simultaneously using this map to improve its own navigation estimate. This technique has been demonstrated both in simulation and on actual vehicles.

This thesis reports the execution of the logical next step in the development of CML: a CML algorithm for use by multiple collaborating autonomous vehicles. Sharing and combining observations of environmental features as well as of the collaborating vehicles greatly enhances the potential performance of CML. This thesis is demonstrates the feasibility and benefits of collaborative CML. Multiple vehicles performing CML together perform faster and more thorough mapping, and produce improved relative (and global) position estimates. This thesis quantifies the improvement in CML performance achieved by collaboration, and compares collaborative versus single-vehicle CML results in simulation to demonstrate how collaborative CML greatly increases the navigation capabilities of autonomous vehicles.

This chapter reviews the fields within mobile robotics that are most relevant to this thesis. Section 1.1 discusses the importance of collaboration, then briefly surveys current collaborative techniques in mapping and navigation. In Section 1.2, the field of single-vehicle CML is described. The intersection of these two fields is collaborative

CML, discussed in Section 1.3, which provides motivation for the work of this thesis and also reviews current collaborative CML implementations. The contribution made by this thesis is described in Section 1.5. The chapter closes with a presentation of the organization of the thesis in Section 1.6.

1.1 Collaboration in mapping and localization

A team of collaborating autonomous vehicles can perform certain tasks more efficiently and robustly than a single vehicle [1, 38], and thus have been the focus of significant study by the robotics community in the past decade. Section 1.1.1 motivates the need for teams of autonomous vehicles to localize themselves with respect to their surroundings and each other, as well as collaborative construct maps of their environments. Current work in collaborative localization is reviewed in Section 1.1.2.1, and a brief survey of collaborative mapping is presented in Section 1.1.2.2. More detailed reviews of collaboration in mobile robots are presented in Cao *et al.* [10], Parker [38], and Dudek *et al.* [18].

1.1.1 Motivating scenarios

The following discussion of possible applications for teams of autonomous vehicles that are able to perform localization and mapping demonstrates why collaborative CML is of interest.

1.1.1.1 Indoor search and rescue

Autonomous vehicles perform tasks without endangering the life of their operator, making them attractive for firefighting or search and rescue. Often firefighters or rescue personnel put themselves at great risks to search for victims inside buildings. A team of autonomous vehicles that could perform these tasks quickly and effectively

would be helped by being able to collaboratively map the building while searching for victims. Furthermore, a heterogenous combination of vehicles could include very small, speedy search vehicles, as well as bigger firefighting or rescue vehicles to be summoned when needed. Accurate navigation and mapping is essential to to perform this task, and a team of robots able to collaboratively localize and map the search area would provide the robustness and search efficiency needed for successful search and rescue.

1.1.1.2 Submarine surveillance

The military seeks the capability to covertly observe the departure of an adversary's submarines from their home ports. Detection of submarines in the open ocean is much harder than detection in shallow water at a known departure point. One current option available to the Navy is to position static acoustic listening devices on the the sea floor or on the surface. However, these are difficult to deploy, requiring divers or aircraft for delivery. Submarines themselves can also perform surveillance, but they are limited to deep water operations. The inability of such submarines to operate in shallow water and the limited range of underwater sensors gives an adversary a window of opportunity to escape detection. Surveillance would be performed much more effectively by an array of shallow water AUVs. By staying within communications range and cooperatively localizing with respect to each other, a web of properly positioned AUVs could create a barrier through which an adversary could not slip through undetected. This AUV array could be deployed quickly, easily, and covertly. This mission emphasizes the need for AUVs to share map and navigation information in order to maintain proper relative positioning as well as to detect the enemy submarine.

1.1.1.3 Close proximity airborne surveillance

A major advantage of autonomous air vehicles over conventional aircraft is the increased maneuverability envelope gained by eliminating the pilot, as their smaller size is coupled with a much higher tolerance for tight turns (which induce high 'G' forces). Capitalizing on this maneuverability while in close proximity to obstacles (such as the ground, foliage, or other aircraft) requires excellent navigation and mapping capabilities. One military mission that requires extremely low altitude, high-precision navigation and mapping is close-proximity surveillance of an unknown enemy position. Military operations benefit greatly from real-time observations, especially in the minutes leading up to an attack. Multiple autonomous air vehicles could gather this coveted information by entering into and observing an enemy position when the potential gain is worth the loss of surprise. Collision avoidance is especially crucial in this situation, since survivability demands that these vehicles be able to hide behind buildings, trees, and other terrain features. Further, relative position information is essential to avoid collisions with collaborators. Using multiple vehicles for this task increases the likelihood of mission success and decreases the amount of time required to complete the surveillance.

1.1.1.4 Summary

Enabling autonomous vehicles to share navigation and map information would greatly extend the existing performance capabilities of autonomous vehicles in a variety of vehicle domains. These existing capabilities are demonstrated by the current work in the fields of collaborative navigation and collaborative mapping, presented in the next section.

1.1.2 Elements of CML

CML is the combination of mapping and localization. Existing work on these two tasks as separate tasks is presented next.

1.1.2.1 Collaboration in navigation

Collaborative navigation is performed when multiple vehicles share navigation and sensor information in order to improve their own position estimate beyond what is possible with a single vehicle. This section surveys existing work in improving navigation through collaboration.

Ant-inspired trail-laying behaviors have been used by a team of mobile robots tasked to navigate towards a common goal [55]. In this implementation a robot communicates its path to collaborators upon successfully arriving at the goal via a random walk. Sharing this information improves the navigational ability of all the robots.

Simple collective navigation has been demonstrated in simulation using multiple 'cartographer' robots that randomly explore the environment [13]. These vehicles possess no pose estimate, and are only capable of line-of-sight communication. When one of these robots detects the goal, it transmits this fact to all other robots it currently observes, and then passes the data along in the same manner. A 'navigator' robot then uses this propagated communication signal to travel to the goal by using the cartographer robots as waypoints.

Simple relative localization between collaborators has been performed using directional beacons [50]. Vision-based cooperative localization has been performed by a team of vehicles tasked with cooperatively trapping and moving objects [49]. Tracking via vision is also used for relative localization of collaborators in an autonomous mobile cleaning system [26].

In work by Roumeliotis *et al.* [41, 42], collaborative localization is performed us-

ing a distributed stochastic estimation algorithm. Each vehicle maintains its own navigation estimate, and communicates localization information only when directly observing a collaborator.

Cooperative navigation of AUVs has been performed in work by Singh *et al.* [45]. Also, two AUVs have demonstrated collaborative operation using the same acoustic beacon array [2]. An unmanned helicopter has used a vision sensor to detect collaborating ground vehicles at globally known positions, and thus was able to localize itself [56].

A related field to collaborative localization is use of multiple vehicles to construct maps of the environment, and is explored in the next section.

1.1.2.2 Collaboration in mapping

Collaborative mapping is performed by combining sensor information from multiple vehicles to construct a larger, more accurate map. Cooperative exploration and mapping with multiple robots is reported by Mataric [32] using behavior-based control [9]. Map matching is used to combine topological maps constructed by multiple vehicles in work performed by Dedeoglu and Sukhatme [15] (Topological maps are described in Section 1.2.2.3).

Heterogenous collaborative mapping has also been investigated, as such systems can capitalize on specialization. One example is a mapping implementation comprised of 'worker' robots which constantly search the environment, and a static 'database' robot that communicates with and is visited by the worker robots [7]. The database robot maintains a centralized, global representation of the environment.

1.2 Single vehicle CML

Most collaborative navigation and mapping techniques have a single-vehicle navigation and mapping technique as the basis, since collaborating vehicles must be able to operate independently in the event of loss of communication. Robust, collision-free operation of an autonomous vehicle requires a description of the vehicular pose as well as information about the location of objects in the vehicle's environment. Only with knowledge of self and surroundings can a safe path of travel can be calculated.

Section 1.2.1 describes common navigation techniques underlying CML. Similarly, Section 1.2.2 reviews the main techniques used for autonomous vehicle mapping. The intersection of these two related fields is Concurrent Mapping and Localization (CML), in which an autonomous vehicle builds a map of an unknown environment while simultaneously using this map to improve its own navigation estimate. Existing CML work can be partitioned by the mapping approaches presented in Section 1.2.2. Feature-based CML, the subset of CML most applicable to work in this thesis, is reviewed in greater detail in Section 1.2.3.

1.2.1 Navigation techniques used in CML

Understanding the need to improve improve techniques for autonomous vehicle navigation requires an understanding of the shortcoming of current approaches to the navigation problem. A detailed survey of localization techniques can be found in Borenstein, Everett, and Feng [8]. This subsection reviews the primary methods used; dead-reckoning and inertial navigation, beacon-based navigation, and map-based navigation.

1.2.1.1 Dead reckoning and inertial navigation systems

Dead reckoning is accomplished by integrating velocity or acceleration measurements taken by the vehicle in order to determine the new vehicle position. This task is most often performed by inertial navigation systems (INS), which operate by integrating the acceleration of the vehicle twice in time to compute a new position estimate. These systems use accelerometers and gyroscopes to sense linear and angular rate. INS suffers from accuracy problems resulting from integration errors. Another internal data source for state determination on land robots is odometry, which measures wheel rotation. This estimate is affected by wheel slippage, which can be significant in a number of situations [8].

Dead reckoning is the most commonly used AUV navigation technique. Unfortunately, the vehicle is only able to measure its velocity with respect to the water column, not accounting for drift caused by ocean currents. This can be an especially significant safety hazard for AUVs that operate at low speeds and in shallow water, due to the proximity of the ocean floor. Historically, INS use in AUVs has also been made difficult by power consumption and cost. The basic problem with reliance on either dead reckoning or INS devices is the same – position error grows without bound as the vehicle travels through the environment.

1.2.1.2 Beacon-based navigation

The placement of artificial beacons at known locations allow autonomous vehicles to determine their position via triangulation. The most prevalent beacon-based navigation system is the satellite-based Global Positioning System (GPS), which provides worldwide localization with an accuracy of meters. GPS is an excellent navigation solution for a great number of mobile robot implementations, but is not applicable in all cases. Specifically, GPS signals are not strong enough to be used indoors and underwater, and in military applications GPS use can be denied by signal jamming.

Beacon-based navigation by AUVs uses an array of acoustic transponders placed in the environment. Sound pulses emanating from these beacons and prior knowledge about the transponder locations are combined to calculate the AUV position. The two primary beacon systems currently used are ultra-short baseline (USBL) and long baseline (LBL). Both systems rely on accurate beacon deployment and positioning. While beacon-based navigation is the system of choice for AUV applications, beacon deployment during covert military operations in other difficult areas (as under the polar ice cap) can be significant handicaps. Currently GPS is crucial for truly autonomous operation of unmanned air vehicles, as even the slightest of positional errors can have disasterous consequences, especially during takeoff and landing.

1.2.1.3 Map-based navigation

Often it is infeasible to provide artificial beacons for navigation. Instead, map-based navigation techniques use the natural environment and an *a priori* map for localization. By comparing sensor measurements with the ground truth map, current vehicle pose can be deduced. Thompson *et al.* [52] performed localization by matching visible hills and other naturally occuring terrain features on the horizon to an *a priori* topological map. Cozman and Krotkov [13] also visually detect mountain peaks on the horizon, then localize probabilistically using a known map. Cruise missiles have successfully used terrain contour matching (TERCOM) via a radar altimeter and an *a priori* map to localize their position [23].

1.2.2 Mapping techniques used in CML

Generally, autonomous vehicle mapping approaches can be grouped by how the map is constructed and environmental information is stored. The basic techniques of map construction elicit entirely different solution spaces, and by this criteria mapping approaches can be grouped into grid-based, feature-based, and topological-based

approaches.

1.2.2.1 Grid-based map representation

Grid-based approaches, such as those described by Moravec [34], represent the environment via an evenly-spaced grid. Each grid cell contains information about possible obstacles at that location. In most cases a probability between 0 and 1 is stored in each cell. A probability of 1 is assigned if the cell is certain to be occupied, and a probability of 0 if it is certain to be free. A map constructed in this fashion is called an occupancy or certainty grid. Mapping is performed by incorporating new measurements of the environment into the occupancy grid, and these measurements are incorporated by increasing or decreasing the probability values in the corresponding grid cells. Localization is performed by a technique called map matching. A local map consisting of recent measurements is generated and then compared to a previously constructed global map. The best map match is found typically by correlating the local to the global map, and from this match the new position estimate is generated. Based on this position, the local map is then merged into the global map.

Work by Thrun *et al.* [53] represents the current state of the art of implementations of the grid-based map representation. This method has also been implemented by Yamauchi *et al.* [57] and Salido-Tercero *et al.* [44]. Grid based map representations are simple to construct and maintain, and directly incorporate all measurement data into the map. However, grid-based approaches suffer from large space and time complexity. This is because the resolution of a grid must be great enough to capture every important detail of the world. Performance is also highly dependent on the quality and model of sensors used for the map update. Also, information is lost when measurements are assigned to grid cells.

1.2.2.2 Feature-based map representation

Feature-based approaches to mapping represent environments using a set of geometric attributes such as points, planes, and corners, and encode these landmarks in a metrically accurate map [31]. This representation has its roots in surveillance and target tracking [4].

1.2.2.3 Topological-based map representation

Topological-based approaches to mapping produce graph-like descriptions of environments [16]. Nodes in the graph represent 'significant places' or landmarks [28]. Work by Chatila and Laumond [12] exemplifies this approach. Once created, the topological model can be used for route planning or similar problem solving purposes. Arcs connecting the nodes depict the set of actions required to move between these significant places. For instance, in a simple indoor environment consisting entirely of interconnected rooms, the topological map can represent each room as a node and the actions needed to travel between rooms as arcs. Computer vision has been used to characterize places by appearance, making localization a problem of matching known places with the current sensor image [51]. The topological approach can produce very compact maps. This compactness enables quick and efficient symbolic path planning, which is easily performed by traversing the graph representation of the environment. The major drawback to the topological approach is difficulty in robustly recognizing significant places. Regardless of the sensor used, identification of significant places, especially in a complex environment (e.g. an outdoor environment), is very sensitive to point of view [27, 28]. Further, distinguishing between similar-looking significant places is difficult, in part because no metric map is maintained.

1.2.2.4 Summary

The three mapping approaches have many orthogonal strengths and weaknesses. All the approaches, however, exhibit increased computational complexity when the size of the environment to be mapped is increased. Another common difficulty with these approaches is the need to robustly handle ambiguities in the sensor measurements. The problem of data association is compounded in the feature and topological-based approach, which use frequently unreliable sensor measurements for accurate identification of features within the environment.

1.2.3 Feature-based CML

A significant challenge for mobile robotics is navigation in unknown environments when neither the map nor vehicle position are initially known. This challenge is addressed by the techniques for Concurrent Mapping and Localization (CML). CML approaches can be categorized by the map representations described in Section 1.2.2. This section focuses on the subset of the field, feature-based CML, which is the representation of choice for work presented in this thesis. Work in grid-based CML [3, 20, 21, 54, 57] is less relevant in this context.

Feature-based approaches to CML identify stationary landmarks in the environment, then use subsequent observations of these landmarks to improve the vehicle navigation estimate. An example of this approach is work by Deans and Hebert [14], which uses an omnidirectional camera and odometry to perform landmark-based CML.

Stochastic Mapping (SM) [47] (discussed in further detail in Chapter 2) provides the theoretical foundation for the majority of feature-based CML implementations. In Stochastic Mapping, a single state vector represents estimates of the vehicle location and of all the features in the map. An associated covariance matrix incorporates the uncertainties of these estimates, as well as the correlations between the estimates.

The heart of Stochastic Mapping is an Extended Kalman Filter (EKF) [5, 22], which uses sensor measurements and a vehicle dead reckoning model to update vehicle and feature estimates. Stochastic Mapping capitalizes on reobservation of static features to concurrently localize the vehicle and improve feature estimates. Analysis of theoretical Stochastic Mapping performance is presented by Dissanayake *et al.* [17] and in further detail by Newman [37].

Adding new features to the state vector produces a corresponding quadratic expansion to the system covariance matrix, and computational complexity thus becomes problematic in large environments. Feder [19] addresses the complexity issue by maintains multiple local submaps in lieu of a single, more complex global map.

Another challenge inherent in Stochastic Mapping is feature association, the process of correctly matching sensor measurements to features. Feature association techniques have been successfully demonstrated in uncluttered indoor environments [6], but remain a challenge for more geometrically complex outdoor environments.

The Stochastic Mapping approach to feature-based CML serves as the algorithmic foundation for this thesis. Current implementations of the collaborative extensions to CML are discussed in the next section.

1.3 Collaborative CML

This section reviews related work most similar to that presented in this thesis, the intersection of the fields of autonomous vehicle collaboration and CML. One navigation method for collaborative navigation and mapping uses robots as mobile landmarks for their collaborators. Kuipers and Byun [29, 30] introduce this concept, whereby some robots move while their collaborators temporarily remain stationary. This method is extended through the use of synchronized ultrasound pulses to measure the distances between team members and determination the relative position of the vehicles via

triangulation [36]. This system has been implemented on very small (5 cm) robots [24]. Another such implementation uses an exploration strategy that capitalizes on line of sight visual observance between collaborators to determine free space in the environment and reduce odometry error [40]. Drawbacks to this approach are that only half of the robots can be in motion at any given time and the robots must stay close to each other in order to remain within visual range.

An important challenge in collaborative robotics is the task of combining maps that were independently gathered by cooperating vehicles. The first step in accomplishing this task is vehicle rendezvous, the process of determining the relative location of each vehicle with respect to its collaborators. This is not trivial when vehicles have previously been out of sensor range, out of communication, or have a poor sense of global position. Rendezvous has proved workable in an unknown environment given unknown starting positions using landmark-based map matching [43]. Rendezvous has been detected visually, following which the shared maps are combined probabilistically [20]. Map merging has also been demonstrated once rendezvous is complete [25, 39].

1.4 Summary

This chapter introduced and motivated the underlying techniques for collaborative concurrent mapping and localization. Possible applications requiring improved navigation and mapping performance for multiple vehicles were presented. Current work in autonomous vehicle collaboration, navigation, and mapping was reviewed to place the work performed in this thesis into the correct context, with focus given to the the subset of CML that this thesis extends – the Stochastic Mapping approach to feature-based CML.

1.5 Contributions

This thesis makes the following contributions:

- A method for performing collaborative concurrent mapping and localization.
- A quantitative theoretical analysis of the performance gains of that collaboration method.
- An analysis of collaborative concurrent mapping and localization performance in 1-D and 2-D simulation.

1.6 Thesis organization

This thesis presents an algorithm for performing CML with multiple vehicles working cooperatively. The remainder of it is structured as follows.

Chapter 2 reviews stochastic mapping as a theoretical foundation for performing single-vehicle CML.

Chapter 3 extends the stochastic mapping algorithm to multiple vehicles. An algorithm for collaborative dead-reckoning in the absence of static environmental features is discussed. A collaborative CML algorithm is then introduced and developed. Theoretical analysis of this algorithm generates a convergence theorem that quantifies the performance gain from collaboration.

Chapter 4 applies the collaborative CML algorithm in a 1-D simulation, to explain the algorithm structure and demonstrate its performance.

Chapter 5 presents 2-D simulations of both collaborative localization and collaborative CML with varying parameters.

Chapter 6 summarizes the main contributions of this thesis and provides suggestions for future research.

Chapter 2

Single Vehicle Stochastic Mapping

Most successful implementations of feature-based CML use an Extended Kalman Filter (EKF) [5, 22] for state estimation. The class of EKF-based methods for feature-based CML is termed stochastic mapping (SM) [35, 47].

This chapter reviews the single vehicle stochastic mapping algorithm which will be extended to incorporate collaboration in Chapter 3. Section 2.1 presents the representations used in the stochastic mapping process, followed by a brief overview in Section 2.2 of the stochastic mapping algorithm itself. For a more detailed explanation refer to one of Smith, Self, and Cheeseman's seminal papers on SM [47, 48].

2.1 Models

This section presents the form of the vehicle, observation, and feature models to be used in this thesis. To preserve simplicity for presentation purposes, these models are restricted to two dimensions.

2.1.1 Vehicle model

The state estimate for an autonomous vehicle in this implementation is represented by $\mathbf{x}_v = [x_v \; y_v \; \phi \; v]^T$, storing north and east coordinates in the global reference frame as well as heading and speed. Vehicle movement $\mathbf{u}_v[k]$ due to control input is generated at time k with time T between successive updates, and consists of a change in heading and a change in speed, such that

$$\mathbf{u}_v[k] = \begin{bmatrix} 0 \\ 0 \\ T\delta\phi[k] \\ T\delta v[k] \end{bmatrix} , \tag{2.1}$$

and is assumed to be known exactly. The general form of a vehicle dynamic model can be defined as

$$\mathbf{x}_v[k+1] = \mathbf{f}(\mathbf{x}_v[k], \mathbf{u}_v[k]) + \omega_v[k] . \tag{2.2}$$

This discrete time vehicle model describes the transition of the vehicle state vector $\mathbf{x}_v$ from time k to time $k+1$ and mathematically takes into account the kinematics and dynamics of the vehicle. The function $\mathbf{f}$ is a nonlinear model that receives the current vehicle state $\mathbf{x}_v[k]$ and control input $\mathbf{u}_v[k]$ as inputs. The model is updated at times $t = kT$ for a constant period T, and can be expanded and written as

$$\mathbf{f}(\mathbf{x}_v[k], \mathbf{u}_v[k]) = \begin{bmatrix} x[k] + Tcos(\phi[k])v[k] \\ y[k] + Tsin(\phi[k])v[k] \\ \phi[k] + T\delta\phi[k] \\ v[k] + T\delta v[k] \end{bmatrix} . \tag{2.3}$$

Although this particular vehicle dynamic model is non-linear, it can be linearized using its Jacobian evaluated at time k [33]. The Jacobian is linearized based on the

vehicle state, such that

$$\mathbf{x}_v[k+1] = \mathbf{F}_v[k]\mathbf{x}_v[k] + \mathbf{u}_v[k] + \omega_v \ . \tag{2.4}$$

The dynamic model matrix $\mathbf{F}_v[k]$ is the Jacobian of $\mathbf{f}$ with respect to the vehicle state, and is defined as

$$\mathbf{F}_v[k] = -\left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}_v}\right|_{\mathbf{x}_v[k]} = \begin{bmatrix} 1 & 0 & -Tsin(\phi[k])v[k] & Tcos(\phi[k]) \\ 0 & 1 & Tcos(\phi[k])v[k] & Tsin(\phi[k]) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} . \tag{2.5}$$

Noise and the unmodeled components of the vehicle behavior are consolidated into the random vector ω_v. This vehicle model process noise is assumed to be a stationary, temporally uncorrelated zero mean Gaussian white noise process with covariance

$$\mathbf{E}[\omega_v\ \omega_v^T] = \begin{bmatrix} x_\omega & 0 & 0 & 0 \\ 0 & y_\omega & 0 & 0 \\ 0 & 0 & \phi_\omega & 0 \\ 0 & 0 & 0 & v_\omega \end{bmatrix} . \tag{2.6}$$

2.1.2 Feature model

Features are fixed, discrete, and identifiable landmarks in the environment. Repeatable observation of features is a core requirement for CML. These features can take many forms, including passive features (points, planes, and corners), or active features (artificial beacons). What constitutes a feature is entirely dependent on the physics of the sensor used to identify it. Vision systems, for instance, may be able to identify features based on color, whereas sonar and laser rangefinders use distance and reflectivity to categorize features.

This thesis uses, without loss of generality, the least complicated of features, stationary point landmarks. This simplification reduces challenges with feature identification and interpretation, increasing the focus on the CML algorithm itself. A point feature is defined by two parameters specifying its position with respect to a global reference frame, and is observable from any angle and any distance. The feature state estimate parameters of the i^{th} point landmark in the environment are represented by

$$\mathbf{x}_{f_i} = \begin{bmatrix} x_{f_i} \\ y_{f_i} \end{bmatrix} . \tag{2.7}$$

The point feature is assumed to be stationary, so unlike the vehicle model, there is no additive uncertainty term due to movement in the feature model. Therefore the model for a point feature can be represented by

$$\mathbf{x}_{f_i}[k+1|k] = \mathbf{x}_{f_i}[k] . \tag{2.8}$$

2.1.3 Measurement model

A measurement model is used to describe relative measurements of the environment taken with on-board sensors. In this thesis range measurements are provided by sonar, which operates by generating a directed sound pulse and timing the reflected return off features in the environment. With knowledge of the speed of sound and the pulse duration, the distance to the reflecting surface is deduced. Inexpensive sonars are easily installed on mobile robots but present some significant challenges due to often ambiguous and noisy measurements. Sonar pulses have a finite beam width, producing an angular uncertainty as to the direction of the reflecting surface. Drop-outs, another problem, occur when the physical structure of the reflecting surface is too poor to generate a reflection capable of being detected. A multipath return occurs when a sonar pulse reflects off multiple surfaces before being detected, thus producing

an overly long time of flight. Lastly, when using an array of multiple sonars, crosstalk is possible. This occurs when a sonar pulse emanating from one sonar transducer is detected by another, producing an erroneous time of flight. These attributes of sonar are taken into account by incorporating noise into the sonar model.

The measurement model is used to process sonar data readings in order to determine where features are in the environment. An *actual* sonar return

$$\mathbf{z}_i[k] = \begin{bmatrix} v_{z_i}[k] \\ \varphi_{z_i}[k] \end{bmatrix} \tag{2.9}$$

consisting of a relative range $v_{z_i}[k]$ and bearing $\varphi_{z_i}[k]$ measurement is taken at time k from the vehicle with state $\mathbf{x}_v[k]$ to the i^{th} feature with state $\mathbf{x}_{f_i}[k]$. The model for the production of this reading is given by

$$\mathbf{z}_i[k] = \mathbf{h}_i(\mathbf{x}_v[k], \mathbf{x}_{f_i}[k]) + \mathbf{w}_i[k] \,, \tag{2.10}$$

where $\mathbf{h}_i$ is the observation model which describes the nonlinear coordinate transformation from the global to robot-relative reference frame. Noise and unmodeled sensor characteristics are consolidated into an observation error vector $\mathbf{w}_i[k]$. This vector is a temporally uncorrelated, zero mean random process such that

$$\mathbf{R}_i = \mathbf{E}[\mathbf{w_i}[k]\ \mathbf{w_i}[k]^T] = \begin{bmatrix} r_w & 0 \\ 0 & \varphi_w \end{bmatrix} \,, \tag{2.11}$$

where $\mathbf{R}_i$ is the observation error covariance matrix. The measurement function can be expanded and written as

$$\mathbf{h}_i(\mathbf{x}_v[k], \mathbf{x}_{f_i}[k]) = \begin{bmatrix} \sqrt{(x_{f_i}[k] - x_v[k])^2 + (y_{f_i}[k] - y_v[k])^2} \\ arctan\left(\frac{y_{f_i}[k] - y_v[k]}{x_{f_i}[k] - x_v[k]}\right) - \phi_v[k] \end{bmatrix} \,. \tag{2.12}$$

Improving navigation performance in stochastic mapping relies on comparing predicted and actual measurements of features. Predicted measurements are calculated based on the current location of the vehicle and environmental features as well as the observation model. Therefore a *predicted* sonar return taken at time k from the vehicle with state $\mathbf{x}_v[k]$ to the i^{th} feature with state $\mathbf{x}_{f_i}[k]$ has the form

$$\hat{\mathbf{z}}_i[k] = \mathbf{h}_i(\mathbf{x}_v[k], \mathbf{x}_{f_i}[k]) \ . \tag{2.13}$$

2.2 Stochastic mapping

Stochastic mapping (SM), first introduced by Smith, Self and Cheeseman [48], provides a theoretical foundation of feature-based CML. The SM approach assumes that distinctive features in the environment can be reliably extracted from sensor data. Stochastic mapping considers CML as a variable-dimension state estimation problem, where the state size increases or decreases as features are added to or removed from the map. A single state vector is used to represent estimates of the vehicle location as well as all environmental features. An associated covariance matrix contains the uncertainties of these estimates, as well as all correlations between the vehicle and feature estimates. SM capitalizes on reobservation of stationary features to concurrently localize the vehicle and improve feature estimates. The implementation of stochastic mapping applied by this thesis uses the vehicle, feature, and sonar measurement models detailed in Sections 2.2.1, 2.2.2, and 2.2.3, respectively. Assuming two stationary features, this section presents the EKF-based algorithms that constitute SM.

2.2.1 SM prediction step

Stochastic mapping algorithms used for CML use a single state vector that contains both the vehicle and feature estimates, denoted by

$$\mathbf{x}[k] = \begin{bmatrix} \mathbf{x}_v[k] \\ \mathbf{x}_f[k] \end{bmatrix} = \begin{bmatrix} \mathbf{x}_v[k] \\ \mathbf{x}_{f_1}[k] \\ \mathbf{x}_{f_2}[k] \\ \vdots \\ \mathbf{x}_{f_n}[k] \end{bmatrix}. \tag{2.14}$$

A predicted estimate given the motion commands provided to the vehicle is generated using the vehicle model described in Equations 2.4 and 2.5, producing a predicted $\mathbf{x}[k+1|k]$ with the form

$$\begin{aligned} \mathbf{x}[k+1|k] &= \begin{bmatrix} \mathbf{x}_v[k+1|k] \\ \mathbf{x}_{f_1}[k+1|k] \\ \mathbf{x}_{f_2}[k+1|k] \end{bmatrix} = \mathbf{F}[k]\mathbf{x}[k] + \mathbf{u}[k] + \omega[k] \\ &= \begin{bmatrix} \mathbf{F}_v[k] & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_v[k] \\ \mathbf{x}_{f_1}[k] \\ \mathbf{x}_{f_2}[k] \end{bmatrix} + \begin{bmatrix} \mathbf{u}_v[k] \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \omega_v[k] \\ 0 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{F}_v[k]\mathbf{x}_v[k] + \mathbf{u}_v[k] + \omega_v[k] \\ \mathbf{x}_{f_1}[k] \\ \mathbf{x}_{f_2}[k] \end{bmatrix}. \end{aligned} \tag{2.15}$$

The feature state estimates in the prediction stage are unchanged, as the features themselves are assumed to be stationary. Unlike the features, the vehicle is in motion,

and because of the uncertainty in this motion the system noise covariance model

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_v & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.16}$$

adds noise to the vehicle estimate. Associated with the state estimate $\mathbf{x}[k]$ is an estimated covariance matrix $\mathbf{P}[k]$, which has the general form

$$\mathbf{P}[k+1|k] = \mathbf{F}[k]\mathbf{P}[k|k]\mathbf{F}^T[k] + \mathbf{Q}\,. \tag{2.17}$$

In its expanded form, the estimated covariance matrix

$$\mathbf{P}[k] = \begin{bmatrix} \mathbf{P}_{vv}[k] & \mathbf{P}_{v1}[k] & \mathbf{P}_{v2}[k] \\ \mathbf{P}_{1v}[k] & \mathbf{P}_{v1}[k] & \mathbf{P}_{1N}[k] \\ \mathbf{P}_{2v}[k] & \mathbf{P}_{21}[k] & \mathbf{P}_{22}[k] \end{bmatrix} \tag{2.18}$$

contains the vehicle ($\mathbf{P}_{vv}[k]$) and feature ($\mathbf{P}_{ii}[k]$) covariances located on the main diagonal. Also contained are the vehicle-feature ($\mathbf{P}_{ri}[k]$) and feature-feature ($\mathbf{P}_{ij}[k]$) cross correlations, located on the off-diagonals. Maintaining estimates of cross correlations is essential for two reasons. First, information gained about one feature can be used to improve the estimate of other correlated features. Second, the cross correlation terms prevent the stochastic mapping algorithm from becoming overconfident, the result incorrectly assuming features are independent when they are actually correlated [11].

At each time step a prediction is made by Equation 2.4 even if no sensor measurements are taken. The prediction step, used on its own, enables the vehicle to perform dead-reckoning.

2.2.2 SM update step

The update step in stochastic mapping integrates measurements made of features in the environment in order to create a map of the environment as well as improve the vehicle's own state estimate. Sensor ranging observations measure the relative distance and orientation between the vehicle and features in the environment. Applying Equation 2.10, a predicted measurement from the vehicle to feature 1 is

$$\hat{\mathbf{z}}_1[k+1] = \mathbf{h}(\mathbf{x}_v[k+1|k], \mathbf{x}_{f_1}[k]) = \begin{bmatrix} v_{\hat{\mathbf{z}}_1[k+1]} \\ \varphi_{\hat{\mathbf{z}}_1[k+1]} \end{bmatrix}. \tag{2.19}$$

Thus a full predicted measurement set of all features in the environment is defined as

$$\begin{aligned} \hat{\mathbf{z}}[k+1|k] &= \begin{bmatrix} \hat{\mathbf{z}}_1[k+1|k] \\ \hat{\mathbf{z}}_2[k+1|k] \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{h}(\mathbf{x}_v[k+1|k], \mathbf{x}_{f_1}[k]) \\ \mathbf{h}(\mathbf{x}_v[k+1|k], \mathbf{x}_{f_2}[k]) \end{bmatrix} \end{aligned} \tag{2.20}$$

An algorithm is then used to associate the predicted measurements with the actual measurement set generated by the sonar. There are various techniques for performing this association, which is discussed further in Section 5.1. Assuming the correct association is performed, the actual sonar return structure

$$\mathbf{z}[k+1] = \begin{bmatrix} \mathbf{z}_{f_1}[k+1] \\ \mathbf{z}_{f_2}[k+1] \end{bmatrix} \tag{2.21}$$

is organized so that each actual measurement corresponds to the matched predicted measurement.

The update process starts by computing the measurement residual

$$\mathbf{r}[k+1] \;=\; \mathbf{z}[k+1] - \hat{\mathbf{z}}[k+1|k] \tag{2.22}$$

which is the difference between the predicted and actual measurements.

The residual covariance is then found

$$\mathbf{S}[k+1] = \mathbf{H}[k+1|k]\mathbf{P}[k+1|k]\mathbf{H}[k+1|k]^T + \mathbf{R}[k+1]\,, \tag{2.23}$$

where $\mathbf{R}[k+1]$ is the measurement noise covariance calculated via Equation 2.11 and $\mathbf{H}[k+1|k]$ is the measurement Jacobian. $\mathbf{H}[k+1|k]$ is calculated by linearizing the non-linear measurement function $\mathbf{h}(\mathbf{x}[k+1|k])$. Because separate vehicle and feature state models are maintained, the measurement function is expressed in block form by linearizing separately based on the vehicle and feature states. The observation model Jacobian $\mathbf{H}_v[k+1|k]$ with respect to the vehicle state can be written as

$$\begin{aligned}\mathbf{H}_v[k+1|k] &= -\left.\frac{\partial h}{\partial \mathbf{x}_v}\right|_{\mathbf{x}_v[k+1|k]} \\ &= \begin{bmatrix} \frac{x_{f_i}[k+1|k]-x_v[k+1|k]}{\sqrt{(x_{f_i}[k+1|k]-x_v[k+1|k])^2+(y_{f_i}[k+1|k]-y_v[k+1|k])^2}} & 0 & 0 \\ -\frac{y_{f_i}[k+1|k]-y_v[k+1|k]}{(x_{f_i}[k+1|k]-x_v[k+1|k])^2+(y_{f_i}[k+1|k]-y_v[k+1|k])^2} & 0 & 0 \\ \frac{y_{f_i}[k+1|k]-y_v[k+1|k]}{\sqrt{(x_{f_i}[k+1|k]-x_v[k+1|k])^2+(y_{f_i}[k+1|k]-y_v[k+1|k])^2}} & 0 & 0 \\ \frac{x_{f_i}[k+1|k]-x_v[k+1|k]}{(x_{f_i}[k+1|k]-x_v[k+1|k])^2+(y_{f_i}[k+1|k]-y_v[k+1|k])^2} & 1 & 0 \end{bmatrix}\end{aligned} \tag{2.24}$$

The negative in this equation emphasizes that the observation is a relative measurement from the vehicle to the feature. Similarly, the Jacobian of the measurement function with respect to the feature state is

$$\mathbf{H}_{f_i}[k+1|k] = \left.\frac{\partial h}{\partial \mathbf{x}_{f_i}}\right|_{\mathbf{x}_{f_i}[k+1|k]}$$

$$= \begin{bmatrix} \frac{x_{f_i}[k+1|k]-x_v[k+1|k]}{\sqrt{(x_{f_i}[k+1|k]-x_v[k+1|k])^2+(y_{f_i}[k+1|k]-y_v[k+1|k])^2}} \\ \frac{x_{f_i}[k+1|k]-x_v[k+1|k]}{(x_{f_i}[k+1|k]-x_v[k+1|k])^2+(y_{f_i}[k+1|k]-y_v[k+1|k])^2} \\ \frac{y_{f_i}[k+1|k]-y_v[k+1|k]}{\sqrt{(x_{f_i}[k+1|k]-x_v[k+1|k])^2+(y_{f_i}[k+1|k]-y_v[k+1|k])^2}} \\ \frac{x_{f_i}[k+1|k]-x_v[k+1|k]}{(x_{f_i}[k+1|k]-x_v[k+1|k])^2+(y_{f_i}[k+1|k]-y_v[k+1|k])^2} \end{bmatrix} \tag{2.25}$$

The full measurement Jacobian contains both the vehicle and feature Jacobians, and has the following form

$$\mathbf{H}[k+1|k] = \begin{bmatrix} -\mathbf{H}_v[k+1|k] & \mathbf{H}_{f_1}[k+1|k] & 0 \\ -\mathbf{H}_v[k+1|k] & 0 & \mathbf{H}_{f_2}[k+1|k] \end{bmatrix} . \tag{2.26}$$

The residual covariance presented in Equation 2.23 is then used to calculate the Kalman filter gain and update the covariance estimate of the vehicle poses. The Kalman gain for the update is defined by

$$\mathbf{K}[k+1] = \mathbf{P}[k+1|k]\mathbf{H}[k+1]^T\mathbf{S}^{-1}[k+1] . \tag{2.27}$$

The pose estimate is updated by adding the Kalman correction, which consists of the measurement residual multiplied by the Kalman gain:

$$\mathbf{x}_v[k+1|k+1] = \mathbf{x}[k+1|k] + \mathbf{K}[k+1]\mathbf{v}[k+1] . \tag{2.28}$$

The state covariance matrix $\mathbf{P}[k+1|k+1]$ is most safely updated using the Joseph form covariance update [5] because the symmetric nature of $\mathbf{P}$ is preserved. This update has the form

$$\mathbf{P}[k+1|k+1] = (\mathbf{I} - \mathbf{K}[k+1]\mathbf{H}[k+1])\mathbf{P}[k+1|k](\mathbf{I} - \mathbf{K}[k+1]\mathbf{H}[k+1])^T$$

$$+\mathbf{K}[k+1]\mathbf{R}[k+1]\mathbf{K}[k+1]^T . \tag{2.29}$$

2.3 Single vehicle CML performance characteristics

This section reviews theorems from work by Newman [37] that characterize the performance of the single vehicle CML algorithm.

Theorem 2.1 (Newman, 1999) *The determinant of any submatrix of the map covariance matrix* $\mathbf{P}$ *decreases monotonically as successive observations are made.*

The determinant of a state covariance submatrix is an important measure of the overall uncertainty of the state estimate, as it is directly proportional to the volume of the error ellipse for the vehicle or feature. Theorem 2.1 states that the error for any vehicle or feature estimate will never increase during the update step of SM. This make sense in the context of the structure of SM, as error is added during the prediction step and subtracted via sensor observations during the update step. The second single vehicle SM theorem from Newman [37] that will be utilized is

Theorem 2.2 (Newman, 1999) *In the limit as the number of observations increases, the errors in estimated vehicle and feature locations become fully correlated.*

Not only do individual vehicle and feature errors decrease as more observations are made, they become fully correlated and features with the same structure (i.e. point features) acquire identical errors. Intuitively, this means that the relative positions of the vehicle and features can be known exactly. The practical consequence of this behavior is that when the exact absolute location of any one feature is provided to the fully correlated map, the exact absolute location of the vehicle or any other feature is deduced.

While single vehicle CML produces full correlations between the vehicle and the features (and thus zero relative error), the absolute error for the vehicle and each feature does not reduce to zero. Rather, Newman asserts that

Theorem 2.3 (Newman, 1999) *In the limit as the number of observations increases, the lower bound on the covariance matrix of the vehicle or any single feature is determined only by the initial vehicle covariance at the time of the observation of the first feature.*

This theorem states that in the single vehicle CML case, the absolute error for the vehicle or single feature can never be lower than the absolute vehicle error present at the time the first feature is initialized into the SM filter.

These theorems describe performance of single vehicle CML, and will be used to analyze the collaborative CML case in Section 3.3.3.

2.4 Summary

The stochastic mapping algorithm serves as the foundation for the collaborative CML algorithm presented in the next chapter. The case presented in this chapter (single vehicle and multiple features) will be extended to multiple vehicles in the next chapter.

Chapter 3

Extending CML to Multiple Vehicles

This chapter constitutes the theoretical contribution of this thesis. First, in Section 3.1 the main challenges that a successful collaborative CML algorithm must overcome are addressed. Then a collaborative localization algorithm using a stochastic mapping framework but doing no feature mapping is presented in Section 3.2 as an intermediate step to collaborative CML. Finally, in Section 3.3 the collaborative CML algorithm itself is introduced, its performance properties are analyzed, and a convergence theorem is proved.

3.1 Critical challenges

There are three main challenges that must be addressed for any successful implementation of collaborative CML on autonomous vehicles. These collaborative challenges are in addition to problems already faced by single vehicle CML algorithms, such as correct association of measurement data and scalability, which remain problematic in collaborative CML. The first critical challenge faced by collaborative CML is

addressed in this chapter – the merging of position estimates and measurements of collaborating vehicles into a single framework.

Second, collaborative CML requires vehicles to rendezvous with each other. Often vehicles performing CML will do so independently, then travel to meet with collaborators. Upon congregating, each vehicle will possess its own navigation estimate and map of the environment. Long term execution of CML often results in a highly accurate local map of the environment relative to the vehicle, but a poor global estimate of position. Thus, would-be collaborators will need to compare and match local maps in order to initialize their positions relative to one another accurately enough to start performing CML collaboratively or to use their previous individual maps together.

Lastly, dealing with limited communication is a difficult challenge for collaborative CML implementations. Collaboration requires sharing navigation and map information in order to improve CML estimates. Therefore, maintenance of an accurate position estimate of collaborators is essential. Intermittent communication between vehicles introduces large amounts of error because of unknown motion by collaborators between updates. Communication bandwidth also is an issue, especially for underwater vehicles reliant on the slow data rates of acoustic modems. Vehicles with a high observation rate must either communicate these measurements to collaborators or attempt to distill the data themselves and then communicate their findings. Even when using a feature-based CML approach, sharing estimated feature positions also requires communicating error covariances that describe confidence in the estimates as well as correlations with other features.

3.2 Collaborative localization

The Kalman filter structure used for performing collaborative localization in the absence of static environmental features (i.e. the need to simultaneously map the envi-

ronment) provides a theoretical foundation for collaborative CML. In this section a collaborative Kalman filter is presented that tracks the positions of multiple vehicles and incorporates the measurements of the others. This uses the single vehicle CML equations presented in Section 2.2.1 and 2.2.2, and is shown assuming three collaborating vehicles. This non-mapping subset of CML is presented here to make clear the structural change made to support collaborative CML.

3.2.1 Prediction step

The state for the centralized filter contains the estimated states of the three collaborating vehicles and is represented by

$$\mathbf{x}_v = \begin{bmatrix} \mathbf{x}_v^A \\ \mathbf{x}_v^B \\ \mathbf{x}_v^C \end{bmatrix}, \tag{3.1}$$

where $\mathbf{x}_v^N$ is the state estimate for vehicle N. The system model is constructed by extending the vehicle model described in Equation 2.4 and Equation 2.5, and has the form

$$\begin{aligned} \mathbf{x}_v &= \begin{bmatrix} \mathbf{x}_v^A[k+1|k] \\ \mathbf{x}_v^B[k+1|k] \\ \mathbf{x}_v^C[k+1|k] \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{F}_v^A[k] & 0 & 0 \\ 0 & \mathbf{F}_v^B[k] & 0 \\ 0 & 0 & \mathbf{F}_v^C[k] \end{bmatrix} \begin{bmatrix} \mathbf{x}_v^A[k] \\ \mathbf{x}_v^B[k] \\ \mathbf{x}_v^C[k] \end{bmatrix} + \begin{bmatrix} \mathbf{u}_v^A \\ \mathbf{u}_v^B \\ \mathbf{u}_v^C \end{bmatrix} + \begin{bmatrix} \omega_v^A \\ \omega_v^B \\ \omega_v^C \end{bmatrix}. \end{aligned} \tag{3.2}$$

Each of the $\mathbf{F}_v^N$ matrices describes the motion for the vehicle N. Note that because the motion of each vehicle is not affected by any other vehicles, the initial system

matrices are diagonal, reflecting no cross-correlations. The system noise covariance model is also diagonal:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_v^{AA} & 0 & 0 \\ 0 & \mathbf{Q}_v^{BB} & 0 \\ 0 & 0 & \mathbf{Q}_v^{CC} \end{bmatrix}. \tag{3.3}$$

The covariance matrix $\mathbf{P}$ contains the error estimates for the individual vehicle poses, denoted by $\mathbf{P}^{ii}$ for the i^{th} vehicle, as well as the correlation estimates between vehicle estimates, each of which is termed $\mathbf{P}^{ij}$ for the correlation between the i^{th} and j^{th} vehicles. Before any information sharing $\mathbf{P}$ is represented by

$$\mathbf{P}[k+1|k] = \begin{bmatrix} \mathbf{P}^{AA}[k|k] & 0 & 0 \\ 0 & \mathbf{P}^{BB}[k|k] & 0 \\ 0 & 0 & \mathbf{P}^{CC}[k|k] \end{bmatrix}. \tag{3.4}$$

The covariance prediction equation for generating $\mathbf{P}[k|k]$ has the general form of Equation 2.17, and can be written as

$$\mathbf{P}[k+1|k] = \begin{bmatrix} \mathbf{F}_v^A[k]\mathbf{P}^{AA}[k|k]\mathbf{F}_v^A[k]^T + \mathbf{Q}_v^{AA} & 0 & 0 \\ 0 & \mathbf{F}_v^B[k]\mathbf{P}^{BB}[k|k]\mathbf{F}_v^B[k]^T + \mathbf{Q}_v^{BB} & 0 \\ 0 & 0 & \mathbf{F}_v^C[k]\mathbf{P}^{CC}[k|k]\mathbf{F}_v^C[k]^T + \mathbf{Q}_v^{CC} \end{bmatrix} \tag{3.5}$$

At each time step this prediction step occurs, regardless of whether any sensor measurements are made of collaborating vehicles. Thus, in the worst case (without any sensor measurements and communication), dead reckoning for each vehicle is maintained, and state and covariance matrices will continue to be diagonal.

3.2.2 Update Step

Collaborative localization is performed by taking direct sensor measurements of collaborating vehicles to determine their relative location. The sensor model for a range measurement of one vehicle as seen by another generates a prediction based on the estimated pose of both vehicles. The general sensor model is defined in Section 2.1.3 and, in the collaborative localization case, a measurement of vehicle B as seen by vehicle A is produced by

$$\mathbf{z}_B^A[k+1] = h(\mathbf{x}_v^A[k+1|k], \mathbf{x}_v^B[k+1|k]) + \mathbf{w}_{B_\blacksquare}^A = \begin{bmatrix} r_{\mathbf{z}_B^A[k+1]} \\ \varphi_{\mathbf{z}_B^A[k+1]} \end{bmatrix}$$

where $\mathbf{x}_v^i$ is the state estimate of vehicle i and $\mathbf{w}_{B_\blacksquare}^A$ is the noise associated with the sensor measurement $\mathbf{z}_B^A[k+1]$ from vehicle A of vehicle B. The noise process associated with measurement $\mathbf{z}_B^A[k+1]$ is zero mean and temporally uncorrelated, and has an measurement error covariance matrix defined by

$$\mathbf{R}_B^A[k+1] = E\left[\ \mathbf{w}_{B_\blacksquare}^A \mathbf{w}_{B_\blacksquare}^{A\ T}\ \right]. \tag{3.6}$$

Each measurement is then used to improve the system state and covariance. As in the single vehicle CML case, the update process starts with computing the measurement residual

$$\mathbf{r}_B^A[k+1] = \mathbf{z}_B^A[k+1] - \hat{\mathbf{z}}_B^A[k+1|k]\,, \tag{3.7}$$

which is the difference between the predicted and actual measurements. The predicted measurement of the difference in pose between vehicle A and vehicle B is generated by using the observation model and the current estimated state

$$\hat{\mathbf{z}}_B^A[k+1|k] = \mathbf{h}_B^A(\mathbf{x}_v[k+1|k]) \,. \tag{3.8}$$

The observation model is linearized applying Equations 2.25 and 2.26. The resulting observation Jacobian $\mathbf{H}_B^A[k+1|k]$ is then used in computing the residual covariance

$$\mathbf{S}[k+1] = \mathbf{H}_B^A[k+1|k]\mathbf{P}[k+1|k]\mathbf{H}_B^A[k+1|k]^T + \mathbf{R}_B^A[k+1]$$

$$= \begin{bmatrix} -\mathbf{H}_v^A[k+1|k] & \mathbf{H}_v^B[k+1|k] & 0 \end{bmatrix} \times \begin{bmatrix} \mathbf{P}^{AA}[k+1|k] & 0 & 0 \\ 0 & \mathbf{P}^{BB}[k+1|k] & 0 \\ 0 & 0 & \mathbf{P}^{CC}[k+1|k] \end{bmatrix} \begin{bmatrix} -\mathbf{H}_v^A[k+1|k] \\ \mathbf{H}_v^B[k+1|k] \\ 0 \end{bmatrix} + \mathbf{R}_B^A[k+1] \tag{3.9}$$

$$\begin{aligned} &= \mathbf{H}_v^A[k+1|k]\mathbf{P}^{AA}[k+1|k]\mathbf{H}_v^A[k+1|k]^T \\ &\quad + \mathbf{H}_v^B[k+1|k]\mathbf{P}^{BB}[k+1|k]\mathbf{H}_v^B[k+1|k]^T + \mathbf{R}_B^A[k+1] \,. \end{aligned} \tag{3.10}$$

The residual covariance is then used to calculate the Kalman filter gain and update the covariance estimate of the vehicle poses. Applying the Kalman gain update from Equation 2.27,

$$\mathbf{K}[k+1] = \begin{bmatrix} \mathbf{K}^A[k+1] \\ \mathbf{K}^B[k+1] \\ \mathbf{K}^C[k+1] \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{P}^{AA}[k+1|k] & 0 & 0 \\ 0 & \mathbf{P}^{BB}[k+1|k] & 0 \\ 0 & 0 & \mathbf{P}^{CC}[k+1|k] \end{bmatrix} \begin{bmatrix} -\mathbf{H}_v^A[k+1|k] \\ \mathbf{H}_v^B[k+1|k] \\ 0 \end{bmatrix} \mathbf{S}^{-1}[k+1]$$

$$= \begin{bmatrix} -\mathbf{P}^{AA}[k+1|k]\mathbf{H}_v^A[k+1|k]\mathbf{S}^{-1}[k+1] \\ \mathbf{P}^{BB}[k+1|k]\mathbf{H}_v^B[k+1|k]\mathbf{S}^{-1}[k+1] \\ 0 \end{bmatrix} . \tag{3.11}$$

Note that no Kalman gain is acquired for vehicle C due to the observation of vehicle B by vehicle A. The pose estimate is updated by adding the Kalman correction, which consists of the measurement residual multiplied by the Kalman gain, as follows

$$\mathbf{x}_v[k+1|k+1] = \mathbf{x}_v[k+1|k] + \mathbf{K}[k+1]\mathbf{v}_B^A[k+1] . \tag{3.12}$$

The state covariance matrix $\mathbf{P}[k+1|k+1]$ is updated using Equation 2.29, producing a covariance update defined by

$$\begin{aligned} \mathbf{P}[k+1|k+1] &= (\mathbf{I} - \mathbf{K}[k+1]\mathbf{H}_B^A[k+1])\mathbf{P}[k+1|k](\mathbf{I} - \mathbf{K}[k+1]\mathbf{H}_B^A[k+1])^T \\ &\quad + \mathbf{R}_B^A[k+1]\mathbf{K}[k+1]\mathbf{R}_B^A[k+1]^T \end{aligned} \tag{3.13}$$

Expanding and substituting using Equation 2.27, the covariance update becomes

$$\begin{aligned} &\mathbf{P}[k+1|k+1] = \mathbf{P}[k+1|k] \\ &-(\mathbf{P}[k+1|k]\mathbf{H}_B^A[k+1|k]^T\mathbf{S}^{-1}[k+1])(\mathbf{S}[k+1])(\mathbf{P}[k+1|k]\mathbf{H}_B^A[k+1|k]^T\mathbf{S}^{-1}[k+1])^T \\ &= \mathbf{P}[k+1|k] - \mathbf{P}[k+1|k]\mathbf{H}_B^A[k+1|k]^T\mathbf{S}^{-1}[k+1]\mathbf{H}_B^A[k+1|k]\mathbf{P}[k+1|k] \end{aligned} \tag{3.14}$$

$$= \begin{bmatrix} \mathbf{P}^{AA} - \mathbf{P}^{AA}\mathbf{H}_v^{A^T}\mathbf{S}^{-1}\mathbf{H}_v^A\mathbf{P}^{AA} & \mathbf{P}^{AA}\mathbf{H}_v^{A^T}\mathbf{S}^{-1}\mathbf{H}_v^B\mathbf{P}^{BB} & 0 \\ \mathbf{P}^{BB}\mathbf{H}_v^{B^T}\mathbf{S}^{-1}\mathbf{H}_v^A\mathbf{P}^{AA} & \mathbf{P}^{BB} - \mathbf{P}^{BB}\mathbf{H}_v^{B^T}\mathbf{S}^{-1}\mathbf{H}_v^B\mathbf{P}^{BB} & 0 \\ 0 & 0 & \mathbf{P}^{CC} \end{bmatrix} \tag{3.15}$$

The important conclusion from these calculations is that only the estimates from vehicle A and vehicle B are updated. The state estimate and covariance for vehicle C remain unchanged, and the estimate for vehicle C remains independent of the other vehicles. Once vehicle C is observed or shares its own measurements of the collaborators, intervehicle cross-correlations with vehicle C (currently zero) will result.

The general collaborative localization algorithm presented in Section 3.2 was demonstrated by Roumeliotis *et al.* [41] to perform collective localization with three vehicles. This algorithm serves as a degenerate case of the collaborative CML algorithm (presented in the next section) when no features are present.

3.3 Collaborative CML

This implementation of collaborative CML extends the single vehicle stochastic mapping algorithm to multiple vehicles employing the centralized Kalman filter structure used for collaborative localization described in Section 3.2.

3.3.1 Collaborative CML prediction step

In the collaborative CML case, as in the collaborative localization algorithm, all of the collaborating vehicle state estimates are combined into a single state vector

$$\mathbf{x}_v[k] = \begin{bmatrix} \mathbf{x}_v^A[k] \\ \mathbf{x}_v^B[k] \\ \vdots \\ \mathbf{x}_v^N[k] \end{bmatrix}, \tag{3.16}$$

where $\mathbf{x}_v^i$ is the vehicle state estimate for vehicle i.

As in single vehicle stochastic mapping, the feature state estimate of the j^{th} point landmark in the environment at time step k is represented by the position estimate

$\mathbf{x}_{f_j}[k]$, generating a combined feature estimate for this environment of

$$\mathbf{x}_f[k] = \begin{bmatrix} \mathbf{x}_{f_1}[k] \\ \mathbf{x}_{f_2}[k] \\ \mathbf{x}_{f_3}[k] \\ \vdots \\ \mathbf{x}_{f_n}[k] \end{bmatrix}. \tag{3.17}$$

In stochastic mapping, a single combined state estimate contains all of the vehicle and feature estimates, defined as

$$\mathbf{x}[k] = \begin{bmatrix} \mathbf{x}_v[k] \\ \mathbf{x}_f[k] \end{bmatrix} = \begin{bmatrix} \mathbf{x}_v^A[k] \\ \mathbf{x}_v^B[k] \\ \vdots \\ \mathbf{x}_v^N[k] \\ \mathbf{x}_{f_1}[k] \\ \mathbf{x}_{f_2}[k] \\ \vdots \\ \mathbf{x}_{f_n}[k] \end{bmatrix}, \tag{3.18}$$

The dynamic vehicle model used has the same general form as Equation 2.4 and Equation 2.5 and is functionally identical to the collaborative localization vehicle model described by Equation 3.2. Only the vehicle states are updated in the prediction step as the features are assumed to be stationary. Following Equation 2.4, the dynamic

model is

$$
\begin{aligned}
\mathbf{x}[k+1|k] &= \begin{bmatrix} \mathbf{x}_v^A[k+1|k] \\ \mathbf{x}_v^B[k+1|k] \\ \vdots \\ \mathbf{x}_v^N[k+1|k] \\ \mathbf{x}_{f_1}[k+1|k] \\ \mathbf{x}_{f_2}[k+1|k] \\ \vdots \\ \mathbf{x}_{f_n}[k+1|k] \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{F}_v^A[k] & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & \mathbf{F}_v^B[k] & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \mathbf{F}_v^N[k] & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \mathbf{I} & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{I} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x}_v^A[k] \\ \mathbf{x}_v^B[k] \\ \vdots \\ \mathbf{x}_v^N[k] \\ \mathbf{x}_{f_1}[k] \\ \mathbf{x}_{f_2}[k] \\ \vdots \\ \mathbf{x}_{f_n}[k] \end{bmatrix} \\
&\quad + \begin{bmatrix} \mathbf{u}_v^A \\ \mathbf{u}_v^B \\ \vdots \\ \mathbf{u}_v^N \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} \omega_v^A \\ \omega_v^B \\ \vdots \\ \omega_v^N \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \qquad (3.19)
\end{aligned}
$$

The system noise covariance model $\mathbf{Q}$ is diagonal and adds noise only to the vehicle states. It has the form

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}^A & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & \mathbf{Q}^B & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \mathbf{Q}^N & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} . \tag{3.20}$$

The covariance prediction equation has the same general form as that used in single vehicle CML Equation 2.17 and collaborative localization Equation 3.5,

$$\mathbf{P}[k+1|k] = \mathbf{F}[k]\mathbf{P}[k|k]\mathbf{F}^T[k] + \mathbf{Q} , \tag{3.21}$$

and updates an associated covariance matrix $\mathbf{P}[k+1|k]$ that contains all of the error information and correlations present in the $\mathbf{x}[k+1|k]$ state estimate. The error associated with each position estimate in the state estimate vector is stored as a covariance estimate $\mathbf{P}^{ij}$ and represents the covariance between the i^{th} and j^{th} elements (superscripted letters are used for vehicles and numbers for features). Thus vehicle-vehicle, feature-vehicle, and feature-feature correlation estimates are maintained. The

general collaborative CML covariance matrix is

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}^{AA} & \mathbf{P}^{AB} & \dots & \mathbf{P}^{AN} & \mathbf{P}^{A1} & \mathbf{P}^{A2} & \dots & \mathbf{P}^{An} \\ \mathbf{P}^{BA} & \mathbf{P}^{BB} & \dots & \mathbf{P}^{BN} & \mathbf{P}^{B1} & \mathbf{P}^{B2} & \dots & \mathbf{P}^{Bn} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \dots & \vdots \\ \mathbf{P}^{NA} & \mathbf{P}^{NB} & \dots & \mathbf{P}^{NN} & \mathbf{P}^{N1} & \mathbf{P}^{N2} & \dots & \mathbf{P}^{Nn} \\ \mathbf{P}^{1A} & \mathbf{P}^{1B} & \dots & \mathbf{P}^{1N} & \mathbf{P}^{11} & \mathbf{P}^{12} & \dots & \mathbf{P}^{1n} \\ \mathbf{P}^{2A} & \mathbf{P}^{2B} & \dots & \mathbf{P}^{2N} & \mathbf{P}^{21} & \mathbf{P}^{22} & \dots & \mathbf{P}^{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}^{nA} & \mathbf{P}^{nB} & \dots & \mathbf{P}^{nN} & \mathbf{P}^{n1} & \mathbf{P}^{n2} & \dots & \mathbf{P}^{nn} \end{bmatrix}. \tag{3.22}$$

Before any observations are made, no static features are present in the state estimate and there are no correlations between collaborating vehicles. Therefore $\mathbf{P}[k]$ initially is a diagonal matrix equal to Equation 3.4 with no cross-correlation terms between vehicles and no features estimated.

3.3.2 Collaborative CML update step

Each vehicle produces a set of range measurements of static features in the environment, as well as measurements of collaborating vehicles within observation range. Because of occlusion and limited sensor range, the measurement set is usually a subset of all of the features and vehicles currently contained in the state estimate. Assuming observations are made of all collaborating vehicles and static features, the complete

set of measurements produced by vehicle A at time step $k+1$ has the following form

$$\mathbf{z}^A[k+1] = \begin{bmatrix} \mathbf{z}_B^A[k+1] \\ \mathbf{z}_C^A[k+1] \\ \vdots \\ \mathbf{z}_N^A[k+1] \\ \mathbf{z}_1^A[k+1] \\ \mathbf{z}_2^A[k+1] \\ \vdots \\ \mathbf{z}_n^A[k+1] \end{bmatrix}, \tag{3.23}$$

where $\mathbf{z}_i^A[k+1]$ corresponds to the sensor measurement of the i^{th} element (vehicle or feature) as observed by vehicle A. The corresponding predicted measurement set for vehicle A is

$$\hat{\mathbf{z}}^A[k+1|k] = \begin{bmatrix} \hat{\mathbf{z}}_B^A[k+1|k] \\ \hat{\mathbf{z}}_C^A[k+1|k] \\ \vdots \\ \hat{\mathbf{z}}_N^A[k+1|k] \\ \hat{\mathbf{z}}_1^A[k+1|k] \\ \hat{\mathbf{z}}_2^A[k+1|k] \\ \vdots \\ \hat{\mathbf{z}}_n^A[k+1|k] \end{bmatrix} = \begin{bmatrix} \mathbf{h}(\mathbf{x}_v^A[k+1|k], \mathbf{x}_v^B[k+1|k]) \\ \mathbf{h}(\mathbf{x}_v^A[k+1|k], \mathbf{x}_v^C[k+1|k]) \\ \vdots \\ \mathbf{h}(\mathbf{x}_v^A[k+1|k], \mathbf{x}_v^N[k+1|k]) \\ \mathbf{h}(\mathbf{x}_v^A[k+1|k], \mathbf{x}_{f_1}[k+1|k]) \\ \mathbf{h}(\mathbf{x}_v^A[k+1|k], \mathbf{x}_{f_2}[k+1|k]) \\ \mathbf{h}(\mathbf{x}_v^A[k+1|k], \mathbf{x}_{f_3}[k+1|k]) \\ \vdots \\ \mathbf{h}(\mathbf{x}_v^A[k+1|k], \mathbf{x}_{f_n}[k+1|k]) \end{bmatrix}. \tag{3.24}$$

With collaboration, the sensor measurements taken by all collaborating vehicles are consolidated into a single measurement vector

$$\mathbf{z}[k+1] = \begin{bmatrix} \mathbf{z}^A[k+1] \\ \mathbf{z}^B[k+1] \\ \vdots \\ \mathbf{z}^N[k+1] \end{bmatrix}, \tag{3.25}$$

where $\mathbf{z}^i[k+1]$ is the set of all measurements for vehicle i. As an example, given an implementation of two vehicles and three features, the complete measurement set would take the form

$$\mathbf{z}[k+1] = \begin{bmatrix} \mathbf{z}_B^A[k+1] \\ \mathbf{z}_1^A[k+1] \\ \mathbf{z}_2^A[k+1] \\ \mathbf{z}_3^A[k+1] \\ \mathbf{z}_A^B[k+1] \\ \mathbf{z}_1^B[k+1] \\ \mathbf{z}_2^B[k+1] \\ \mathbf{z}_3^B[k+1] \end{bmatrix}. \tag{3.26}$$

As expected, the observation Jacobian $\mathbf{H}$ for a full set of sensor measurements has much the same form as the collaborative observation Jacobian presented in Equation 3.10. Sensor range measurements are treated identically for both static features and collaborating vehicles. The observation Jacobian for the two vehicle, three static

feature measurement set presented in Equation 3.26 is

$$\mathbf{H}[k+1|k] = \begin{bmatrix} -\mathbf{H}_v^A[k+1|k] & \mathbf{H}_v^B[k+1|k] & 0 & 0 & 0 \\ -\mathbf{H}_v^A[k+1|k] & 0 & \mathbf{H}_{f_1}^A[k+1|k] & 0 & 0 \\ -\mathbf{H}_v^A[k+1|k] & 0 & 0 & \mathbf{H}_{f_2}^A[k+1|k] & 0 \\ -\mathbf{H}_v^A[k+1|k] & 0 & 0 & 0 & \mathbf{H}_{f_3}^A[k+1|k] \\ \mathbf{H}_v^A[k+1|k] & -\mathbf{H}_v^B[k+1|k] & 0 & 0 & 0 \\ 0 & -\mathbf{H}_v^B[k+1|k] & \mathbf{H}_{f_1}^B[k+1|k] & 0 & 0 \\ 0 & -\mathbf{H}_v^B[k+1|k] & 0 & \mathbf{H}_{f_2}^B[k+1|k] & 0 \\ 0 & -\mathbf{H}_v^B[k+1|k] & 0 & 0 & \mathbf{H}_{f_3}^B[k+1|k] \end{bmatrix} . \tag{3.27}$$

Subsequent calculation of the residual and Kalman gain have the same form as the single-vehicle CML case presented in Section 2.2.2, and are repeated without further explanation:

$$\mathbf{r}[k+1] = \mathbf{z}[k+1] - \hat{\mathbf{z}}[k+1|k] , \tag{3.28}$$

$$\mathbf{S}[k+1] = \mathbf{H}[k+1|k]\mathbf{P}[k+1|k]\mathbf{H}[k+1|k]^T + \mathbf{R}[k+1] , \tag{3.29}$$

$$\mathbf{K}[k+1] = \mathbf{P}[k+1|k]\mathbf{H}[k+1]^T\mathbf{S}^{-1}[k+1] , \tag{3.30}$$

$$\mathbf{x}_v[k+1|k+1] = \mathbf{x}[k+1|k] + \mathbf{K}[k+1]\mathbf{v}[k+1] , \tag{3.31}$$

$$\begin{aligned} \mathbf{P}[k+1|k+1] = {} & (\mathbf{I} - \mathbf{K}[k+1]\mathbf{H}[k+1])\mathbf{P}[k+1|k](\mathbf{I} - \mathbf{K}[k+1]\mathbf{H}[k+1])^T \\ & + \mathbf{K}[k+1]\mathbf{R}[k+1]\mathbf{K}[k+1]^T . \end{aligned} \tag{3.32}$$

The key structural elements to the collaborative extension to single-vehicle CML are expansion of the state estimate vector and associated covariance matrix to incorporate more than one vehicle, combining measurements from multiple vehicles into a single measurement vector, and constructing the measurement Jacobian to reflect the

different observation sources and targets.

3.3.3 Collaborative CML performance analysis

In this section proofs are provided for the following results that characterize the performance of the collaborative CML algorithm.

1. In the limit as the number of observations increases, if there are features observed by all vehicles, or each vehicle directly observes its collaborators, all of the vehicle and feature estimates become completely correlated with each other.

2. In the limit as the number of observations increases, the covariance of each vehicle and feature estimate becomes identical and converges to a single lower covariance bound that is a function of the uncertainty of initial location estimates of the vehicles when the first feature is observed.

These results form the collaborative CML extension of single vehicle SM error convergence properties, and define the best-case performance of collaborative CML. These performance characteristics are validated via simulation in Chapter 4 and 5. The theorems derived by Newman [37] and briefly reviewed in Section 2.3 serve the theoretical basis for analyzing the performance of the collaborative CML algorithm, and will be used to prove the above results.

The full correlation property of single vehicle CML asserted in Theorem 2.2 scales to the collaborative CML case, as the second vehicle is, in essence, a moving feature in the SM structure.

However, the single vehicle CML lower performance bound does not apply to the collaborative CML case. Multiple vehicles performing CML together can attain a lower absolute error than the single vehicle initial covariance which bounds the single vehicle CML case. The collaborative lower bound is quantified in the following theorem:

Theorem 3.1 *In the collaborative CML case, in the limit as the number of observations increases, the lower bound on the covariance matrix of any vehicle or any single feature equal to the inverse of the sum the initial collaborating vehicle covariance inverses at the time of the observation of the first feature or observation of a collaborating vehicle.*

Analysis of the limiting behavior of the state covariance matrix in the collaborative case is performed by using the information form of the Kalman filter [33]. The following proof starts with the two vehicle case, then uses induction to generalize the result for any number of collaborating vehicles.

For observations of vehicle B by vehicle A during collaborative localization, the state covariance update equation can be written as

$$\mathbf{P}^{-1}[k+1|k+1] = \mathbf{P}^{-1}[k+1|k] + \mathbf{H}_B^A[k+1|k]^T(\mathbf{R}_B^A[k+1])^{-1}\mathbf{H}_B^A[k+1|k]\,. \quad (3.33)$$

Using the form for the observation model $\mathbf{H}_B^A$ presented by Equation 3.6, the state covariance update equation reduces to

$$\mathbf{P}^{-1}[k+1|k+1] = \mathbf{P}^{-1}[k+1|k] + \begin{bmatrix} -\mathbf{H}_v^A[k] \\ \mathbf{H}_v^B[k] \end{bmatrix} (\mathbf{R}_B^A[k+1])^{-1} \begin{bmatrix} -\mathbf{H}_v^A[k] & \mathbf{H}_v^B[k] \end{bmatrix}. \quad (3.34)$$

$\mathbf{P}^{-1}$ is a measure of the amount of information present in the system [33]. During the prediction step information is subtracted from the system via system noise, and then during the update step information is added back into the system via sensor measurements. Assuming there is zero system noise added during the prediction step, $\mathbf{Q}[k+1] = \mathbf{0}$, then there is zero information lost and

$$\mathbf{P}^{-1}[k+1|k] = \mathbf{P}^{-1}[k|k]\,. \quad (3.35)$$

Applying Equations 3.33 and 3.35 recursively for k observations of vehicle B by vehicle A produces

$$\mathbf{P}^{-1}[k|k] = \begin{bmatrix} \mathbf{P}^{AA^{-1}}[0] & \mathbf{0} \\ \mathbf{0} & \mathbf{P}^{BB^{-1}}[0] \end{bmatrix}$$

$$+ \begin{bmatrix} k\mathbf{H}_v^A[k]^T(\mathbf{R}_B^A[k+1])^{-1}\mathbf{H}_v^A[k] & -k\mathbf{H}_v^A[k]^T(\mathbf{R}_B^A[k+1])^{-1}\mathbf{H}_v^B[k] \\ -k\mathbf{H}_v^B[k]^T(\mathbf{R}_B^A[k+1])^{-1}\mathbf{H}_v^A[k] & k\mathbf{H}_v^B[k]^T(\mathbf{R}_B^A[k+1])^{-1}\mathbf{H}_v^B[k] \end{bmatrix} . \quad (3.36)$$

It is assumed that the size of the vehicle covariances $\mathbf{P}^{AA}$ and $\mathbf{P}^{BB}$ are identical. Note that the values of $\mathbf{P}^{AA}[0]$ and $\mathbf{P}^{BB}[0]$ need not be the same, and they are uncorrelated. Applying the common 2 x 2 matrix inversion formula

$$\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, det(\mathbf{A}) = a*d - b*c \Rightarrow \mathbf{A}^{-1} = \frac{1}{det(\mathbf{A})}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \quad (3.37)$$

to Equation 3.36 produces a determinant of

$$det(\mathbf{P}^{-1}[k|k]) = k\mathbf{P}^{AA^{-1}}[0]\mathbf{P}^{BB^{-1}}[0]$$
$$\times \left[\tfrac{1}{k}\mathbf{I} + \mathbf{H}_v^A[k]^T(\mathbf{R}_B^A)^{-1}\mathbf{H}_v^A[k]\mathbf{P}^{BB}[0] + \mathbf{P}^{AA}[0]\mathbf{H}_v^B[k]^T(\mathbf{R}_B^A)^{-1}\mathbf{H}_v^B[k] \right] . \quad (3.38)$$

The inverse of the determinant is

$$\tfrac{1}{det(\mathbf{P}^{-1}[k|k])} = \tfrac{1}{k}\mathbf{P}^{AA}[0]\mathbf{P}^{BB}[0]$$
$$\times \left[\tfrac{1}{k}\mathbf{I} + \mathbf{H}_v^A[k]^T(\mathbf{R}_B^A)^{-1}\mathbf{H}_v^A[k]\mathbf{P}^{BB}[0] + \mathbf{P}^{AA}[0]\mathbf{H}_v^B[k]^T(\mathbf{R}_B^A)^{-1}\mathbf{H}_v^B[k] \right]^{-1} . \quad (3.39)$$

Thus the resulting covariance for vehicle A is

$$\mathbf{P}^{AA}[k|k] = \tfrac{1}{k}\mathbf{P}^{AA}[0] \quad (3.40)$$

$$\times\Big[\ \tfrac{1}{k}\mathbf{I}+\mathbf{H}_v^A[k]^T(\mathbf{R}_B^A)^{-1}\mathbf{H}_v^A[k]\mathbf{P}^{BB}[0]+\mathbf{P}^{AA}[0]\mathbf{H}_v^B[k]^T(\mathbf{R}_B^A)^{-1}\mathbf{H}_v^B[k]\ \Big]^{-1} \tag{3.41}$$

$$\begin{aligned}&+\mathbf{P}^{AA}[0]\mathbf{P}^{BB}[0]\Big[\ \tfrac{1}{k}\mathbf{I}+\mathbf{H}_v^A[k]^T(\mathbf{R}_B^A)^{-1}\mathbf{H}_v^A[k]\mathbf{P}^{BB}[0]\\ &+\mathbf{P}^{AA}[0]\mathbf{H}_v^B[k]^T(\mathbf{R}_B^A)^{-1}\mathbf{H}_v^B[k]\ \Big]^{-1}\mathbf{H}_v^B[k]^T(\mathbf{R}_B^A)^{-1}\mathbf{H}_v^B[k]\end{aligned} \tag{3.42}$$

If an infinite number of observations of vehicle B are taken by vehicle A, $k\to\infty$ and the resulting vehicle covariance for vehicle A converges to

$$\begin{aligned}&\lim_{k\to\infty}\mathbf{P}^{AA}[k|k]=\mathbf{P}^{AA}[0]\mathbf{P}^{BB}[0]\\ &\times\Big[\ \mathbf{H}_v^A[k]^T(\mathbf{R}_B^A)^{-1}\mathbf{H}_v^A[k]\mathbf{P}^{BB}[0]+\mathbf{P}^{AA}[0]\mathbf{H}_v^B[k]^T(\mathbf{R}_B^A)^{-1}\mathbf{H}_v^B[k]\ \Big]^{-1}\\ &\times\mathbf{H}_v^B[k]^T(\mathbf{R}_B^A)^{-1}\mathbf{H}_v^B[k]\\ &=\mathbf{P}^{AA}[0]\mathbf{P}^{BB}[0]\\ &\times\Big[\ \mathbf{H}_v^A[k]^T(\mathbf{R}_B^A)^{-1}\mathbf{H}_v^A[k]\mathbf{P}^{BB}[0]\left(\mathbf{H}_v^B[k]^T(\mathbf{R}_B^A)^{-1}\mathbf{H}_v^B[k]\right)^{-1}+\mathbf{P}^{AA}[0]\ \Big]^{-1}.\end{aligned} \tag{3.43}$$

For simplicity assume similar vehicle models. The relative nature of making observations leads to the assumption of $\mathbf{H}_v^A[k]=-\mathbf{H}_v^B[k]$ and simplifying,

$$\lim_{k\to\infty}\mathbf{P}^{AA}[k|k]=\lim_{k\to\infty}\mathbf{P}^{BB}[k|k]=\mathbf{P}^{AA}[0]\mathbf{P}^{BB}[0]\Big[\ \mathbf{P}^{AA}[0]+\mathbf{P}^{BB}[0]\ \Big]^{-1}. \tag{3.44}$$

This result is the lower performance bound for collaborative CML with two vehicles. Note that the vehicle covariances for both vehicles becomes fully correlated and thus identical, supporting Theorem 2.2. A simpler, more intuitive conceptual result is found by taking the inverse of Equation 3.44, producing a result of

$$\lim_{k\to\infty}\mathbf{P}^{AA^{-1}}[k|k]=\lim_{k\to\infty}\mathbf{P}^{BB^{-1}}[k|k]=\mathbf{P}^{AA^{-1}}[0]+\mathbf{P}^{BB^{-1}}[0]\,. \tag{3.45}$$

Further, Equation 3.45 makes sense in the context of conservation of information. In the general case, $\mathbf{P}^{-1}$ represents the amount of information present in the system

[33]. The total amount of information in the system can never decrease, but can stay constant when no noise is added to the system. The sum of information present initially in the system is equal to the inverse of the sum of initial uncorrelated vehicle position errors, while the amount of information present after infinite observations are made is encapsulated in a single vehicle position covariance.

It is also important to note that the lower performance bound for collaborative CML is not dependent on direct observation of one vehicle by another. While direct observation improves the rate of covariance convergence, simple observation by both vehicles of a common feature is all that is required for convergence.

Equation 3.45 scales easily for collaboration with more than two vehicles. Assume that at time $t \approx \infty$ a third vehicle C, with a nonzero initial covariance uncorrelated with vehicle A and vehicle B, becomes a collaborator. Thus the lower performance bound becomes

$$\mathbf{P}^{AA^{-1}}[t] = \mathbf{P}^{BB^{-1}}[t] = \mathbf{P}^{AA^{-1}}[0] + \mathbf{P}^{BB^{-1}}[0] \; . \tag{3.46}$$

Since vehicles A and B are fully correlated at t, $\mathbf{P}^{AA^{-1}}[t]$ captures all of the information present in vehicle $\mathbf{P}^{BB^{-1}}[t]$. Thus

$$\begin{aligned} &\lim_{k\to\infty} \mathbf{P}^{AA^{-1}}[k|k] = \lim_{k\to\infty} \mathbf{P}^{BB^{-1}}[k|k] = \lim_{k\to\infty} \mathbf{P}^{CC^{-1}}[k|k] \\ &= \mathbf{P}^{AA^{-1}}[t] + \mathbf{P}^{CC^{-1}}[t] = \mathbf{P}^{AA^{-1}}[0] + \mathbf{P}^{BB^{-1}}[0] + \mathbf{P}^{CC^{-1}}[t] \; . \end{aligned} \tag{3.47}$$

In the n vehicle case, the lower performance bound becomes

$$\begin{aligned} \lim_{k\to\infty} \mathbf{P}^{AA^{-1}}[k|k] &= \lim_{k\to\infty} \mathbf{P}^{BB^{-1}}[k|k] = \ldots = \lim_{k\to\infty} \mathbf{P}^{NN^{-1}}[k|k] \\ &= \mathbf{P}^{AA^{-1}}[i_A] + \mathbf{P}^{BB^{-1}}[i_B] + \ldots + \mathbf{P}^{NN^{-1}}[i_N] \; , \end{aligned} \tag{3.48}$$

where i_n represents the initial time at which the n^{th} vehicle started collaborating, assuming that each vehicle covariance is initially uncorrelated with its collaborators. In the case of homogeneous collaborating vehicles, each with identical, initially uncorrelated error estimates, such that

$$\mathbf{P}^{AA}[0] = \mathbf{P}^{BB}[0] = \ldots = \mathbf{P}^{NN}[0] \,, \tag{3.49}$$

a relationship can be found between the final map covariance and the number of vehicles required to achieve this performance bound, defined by,

$$\begin{aligned} \lim_{k\to\infty} \mathbf{P}^{final^{-1}}[k|k] &= \mathbf{P}^{AA^{-1}}[0] + \mathbf{P}^{BB^{-1}}[0] + \ldots + \mathbf{P}^{NN^{-1}}[0] \,, \\ &= n\mathbf{P}^{NN^{-1}}[0] \,. \end{aligned} \tag{3.50}$$

Taking the determinant of both sides and solving for n produces a result of

$$n = \frac{det(\mathbf{P}^{desired^{-1}})}{det(\mathbf{P}^{NN^{-1}}[0])} \,, \tag{3.51}$$

where $\mathbf{P}^{desired}$ is the desired final map error. This result is very useful for mission planning as it allows determination of how many vehicles are required to construct a map to a desired accuracy.

3.4 Summary

This is the first formulation of CML (that is, mapping as well as localization) for multiple vehicles and quantifies the benefits collaboration provides.

This chapter presented the theoretical framework for extending stochastic mapping to multiple vehicles. First, the challenges inherent in performing collaborative CML were discussed. Second, an algorithm for performing collaborative localization

was presented as an intermediate step to collaboration in the presence of environmental features. The research program from which this thesis reports results was focused on the full CML problem, which includes feature mapping. As a result it did not derive what is herein called collaborative localization, except for elucidation purposes while writing this thesis. At that point, it was observed that the formulation for the non-mapping case had been developed separately [41]. Third, this chapter introduced a theoretical framework for a collaborative full CML algorithm. Lastly, a convergence theorem central to the cooperative CML problem was proved for the first time. This theorem quantifies the performance gain from collaboration, enabling determination of the number of cooperating vehicles required to accomplish a task.

Chapter 4

1-D Collaborative CML Simulation Results

This chapter demonstrates the fundamental principles of the Collaborative CML algorithm in the simplest environment possible – a 1-D simulation. Quantitative improvement over single vehicle CML is shown. In this implementation two vehicles move along a line, observing each other as well as stationary point features also located along the line.

There are two important reasons for the inclusion of the 1-D case into this thesis. First, the observation model $\mathbf{H}$ that encapsulates the heart of the collaborative extension to CML is linear, and becomes exceptionally easy to visualize and understand in the 1-D case. Second, simulation results in the 1-D case are not subject to EKF linearization as in the 2-D case. Thus the 1-D convergence behavior to the theoretical lower covariance bound presented in Section 3.3.3 is more obvious.

4.1 1-D algorithm structure

The state estimate for a mobile robot in the 1-D simulation is represented by the position along the world line $x_v = [x_v]$. Note that only the position is estimated; velocity is not stored in the state estimate. Vehicle control input u consists of a change in position, $u = [T\delta x]$, where T is the time between updates. The vehicle model is

$$x_v[k+1] = f(x_v[k], u[k]) + \omega_r \ . \tag{4.1}$$

The corresponding vehicle dynamic model update is defined as

$$\begin{aligned} x_v[k+1] &= F_v[k]x_v[k] + u_v[k] + \omega_v[k] \\ &= x_v[k] + T\delta x \ . \end{aligned} \tag{4.2}$$

The vehicle model process noise ω_r provides noise in the robot motion, and is assumed to be a stationary zero mean Gaussian white noise process with covariance x_ω. The model used for both vehicles in the simulation is identical. It is assumed that both vehicles observe the same features and each other at every time step.

The collaborative state vector combines the estimate from two vehicles and four features, and is defined as

$$\mathbf{x}[k] = \begin{bmatrix} \mathbf{x}_v[k] \\ \mathbf{x}_f[k] \end{bmatrix} = \begin{bmatrix} x_A[k] \\ x_B[k] \\ x_{f_1}[k] \\ x_{f_2}[k] \\ x_{f_3}[k] \\ x_{f_4}[k] \end{bmatrix} , \tag{4.3}$$

The measurement model produces a set of noisy sonar range measurements. In the collaborative case, each vehicle directly observes the other. This extra measurement is also included by the measurement model, which for vehicle A is given by

$$\mathbf{z}^A[k] = \mathbf{H}(\mathbf{x}_v^A[k], \mathbf{x}_f[k]) + \mathbf{w}[k] = \begin{bmatrix} r_{z_B^A[k]} \\ r_{z_1^A[k]} \\ r_{z_2^A[k]} \\ r_{z_3^A[k]} \\ r_{z_4^A[k]} \end{bmatrix}, \tag{4.4}$$

where $r_{z_B^A[k]}$ corresponds to the range measurement of vehicle B as observed by vehicle and B and $r_{z_i^A[k]}$ is the range to the i^{th} feature. The noise in the sonar observation is modeled by $\mathbf{w}[k]$, which is assumed to be a stationary zero mean Gaussian white noise process with covariance x_w.

The combined measurement vector for vehicle A in the collaborating case is

$$\mathbf{z}[k] = \begin{bmatrix} \mathbf{z}^A[k] \\ \mathbf{z}^B[k] \end{bmatrix} = \begin{bmatrix} r_{z_B^A[k]} \\ r_{z_1^A[k]} \\ r_{z_2^A[k]} \\ r_{z_3^A[k]} \\ r_{z_4^A[k]} \\ r_{z_A^B[k]} \\ r_{z_1^B[k]} \\ r_{z_2^B[k]} \\ r_{z_3^B[k]} \\ r_{z_4^B[k]} \end{bmatrix}. \tag{4.5}$$

Each sonar measurement is a simple estimate of the difference between the x position of the robot and the feature being observed. For instance, $z_1^A[k] = x_{f_1}[k] - x_v^A[k]$. Thus

the observation model, **H**, for a full set of sensor measurements in 1-D implementation is defined as

$$\mathbf{H} = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 \end{bmatrix} . \tag{4.6}$$

Thus the complete expanded observation model for the 1-D case is defined by

$$\mathbf{z}[k] = \begin{bmatrix} \mathbf{z}^A[k] \\ \mathbf{z}^B[k] \end{bmatrix} = \begin{bmatrix} v_{z_B^A}[k] \\ v_{z_1^A}[k] \\ v_{z_2^A}[k] \\ v_{z_3^A}[k] \\ v_{z_4^A}[k] \\ v_{z_A^B}[k] \\ v_{z_1^B}[k] \\ v_{z_2^B}[k] \\ v_{z_3^B}[k] \\ v_{z_4^B}[k] \end{bmatrix} = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_v^A[k] \\ x_v^B[k] \\ x_{f_1}[k] \\ x_{f_2}[k] \\ x_{f_3}[k] \\ x_{f_4}[k] \end{bmatrix} + \mathbf{w}[k] . \tag{4.7}$$

The error associated with each position estimate in the state estimate vector is stored as a covariance estimate $P^{ij}[k] = \sigma_x$ and represents the covariance between the i^{th}

and j^{th} feature. The combined covariance matrix has the form

$$\mathbf{P}[k] = \begin{bmatrix} P^{AA}[k] & P^{AB}[k] & P^{A1}[k] & P^{A2}[k] & P^{A3}[k] & P^{A4}[k] \\ P^{BA}[k] & P^{BB}[k] & P^{B1}[k] & P^{B2}[k] & P^{B3}[k] & P^{B4}[k] \\ P^{1A}[k] & P^{1B}[k] & P^{11}[k] & P^{12}[k] & P^{13}[k] & P^{14}[k] \\ P^{2A}[k] & P^{2B}[k] & P^{21}[k] & P^{22}[k] & P^{23}[k] & P^{24}[k] \\ P^{3A}[k] & P^{3B}[k] & P^{31}[k] & P^{32}[k] & P^{33}[k] & P^{34}[k] \\ P^{4A}[k] & P^{4B}[k] & P^{41}[k] & P^{42}[k] & P^{43}[k] & P^{44}[k] \end{bmatrix}. \quad (4.8)$$

4.1.1 Simulation parameters and assumptions

Both vehicles and all four features present in the 1-D simulation are initialized into the state estimate at the start of the simulation, and every sensor measurement is associated with the correct feature. Moreover, there is no sensor occlusion as each feature is observed at every time step. Therefore, no data association techniques attempting to match sonar returns with features are required. Vehicle motion in the simulation consists of vehicle A traveling in the positive x direction, and vehicle B traveling in the negative x direction, crossing paths at roughly the center of the world line. Table 4.1 summarizes the global parameters that are consistent for all three scenarios presented in Section 4.2.

Table 4.1: 1-D CCML simulation global parameters

number of vehicles	2
number of features	4
sampling period T	1 sec.
range measurement std. dev. x_w	0.4 m
feature probability of detection	1.0
vehicle speed u	0.2 m/s

4.2 1-D Results

This section presents results from three different scenarios. In the first scenario, the vehicles are given initial uncertainty and zero process noise, demonstrating convergence to the theoretical lower performance bound. In the second scenario, there is no initial vehicle uncertainty, but process noise is added at every time step. Finally, the third scenario demonstrates performance given both process noise and initial vehicle uncertainty.

4.2.1 1-D scenario #1

In this scenario, there is no process noise added as each vehicle moves. However, each vehicle has an initial position uncertainty, as do all of the features. As a result of the zero additive process noise, dead reckoning error stays constant as no information is lost due to movement. Tables 4.1 and 4.2 summarize the parameters used in this scenario. The initial starting location of both vehicles as well as static feature positions is shown by Figure 4-1, with the 3σ (99% highest confidence region) error bound ellipse around each vehicle and feature indicating the initial position uncertainty. Note that the circular shape of the error bound is only for visualization – all estimates are 1-D. Figures 4-4 and 4-5 demonstrate the convergence property of the theoretical lower performance bound presented in Section 3.3.3. Note that the single vehicle CML performance also demonstrates convergence toward an error less than the initial error, albeit less than the collaborative case. The single vehicle CML convergence is the result of additional position information provided by the uncorrelated initial estimates of the four static features.

Table 4.2: 1-D CCML simulation scenario #1 parameters

speed process noise std. dev. x_ω	0.0 m/s
initial vehicle position uncertainty std. dev.	0.3 m
initial feature position uncertainty std. dev.	0.4 m

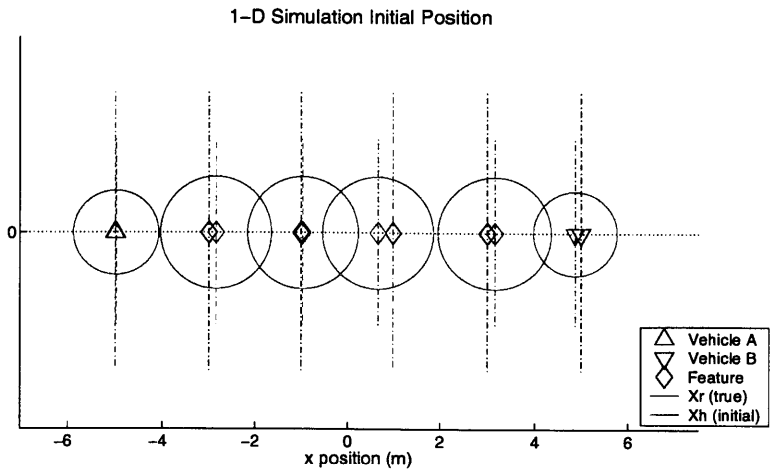


Figure 4-1: 1-D scenario #1 initial position

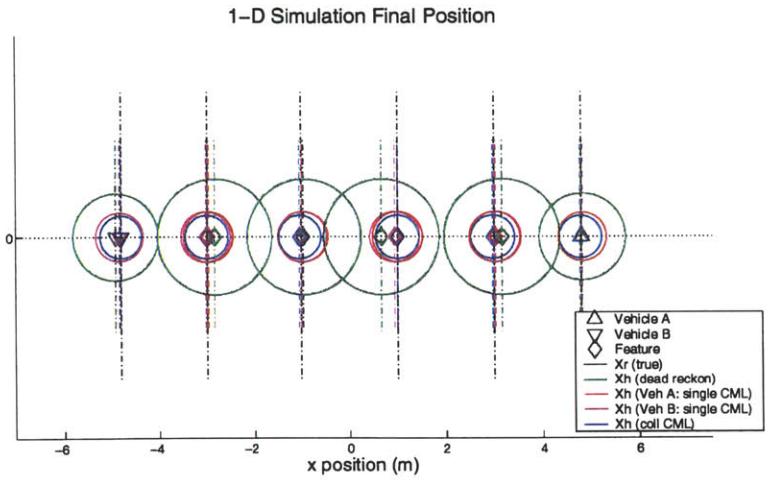


Figure 4-2: 1-D scenario #1 final position

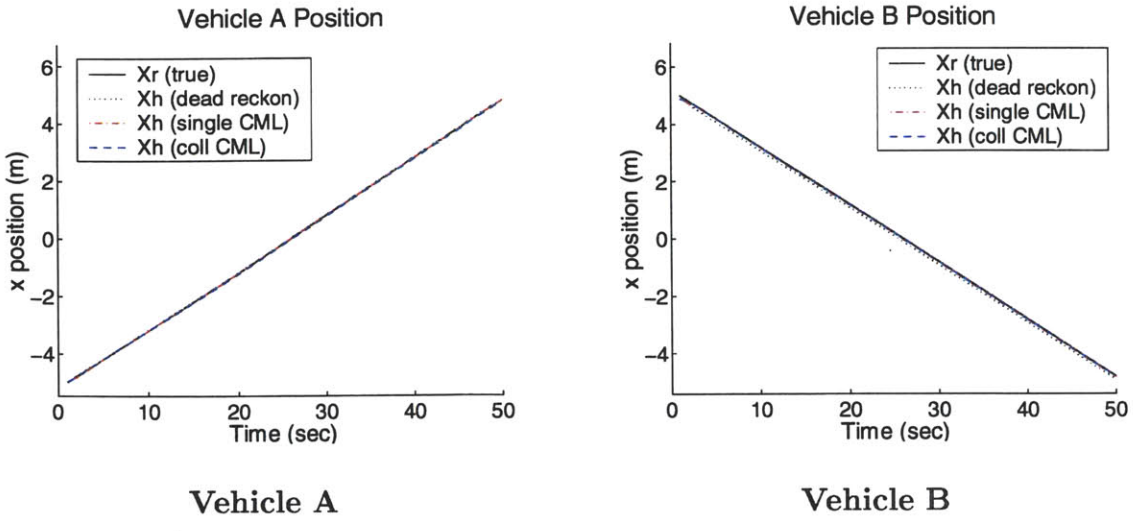
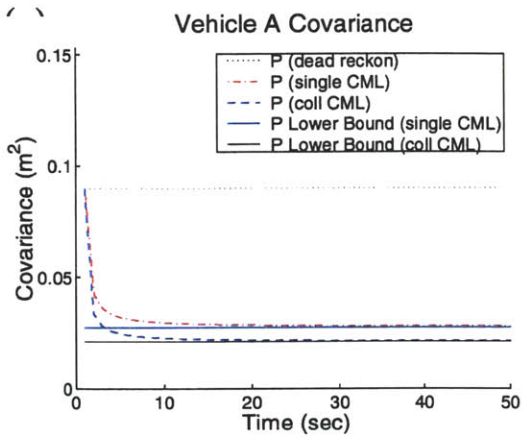
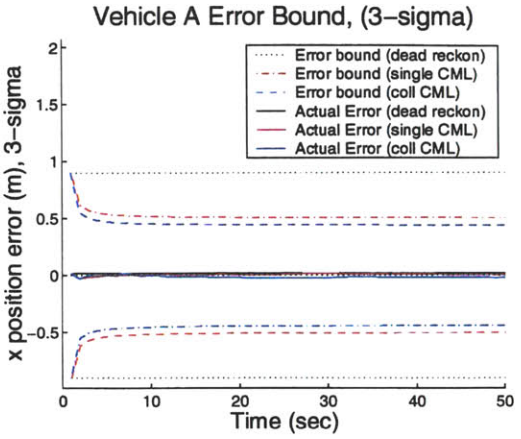


Figure 4-3: 1-D scenario #1 vehicle position versus time

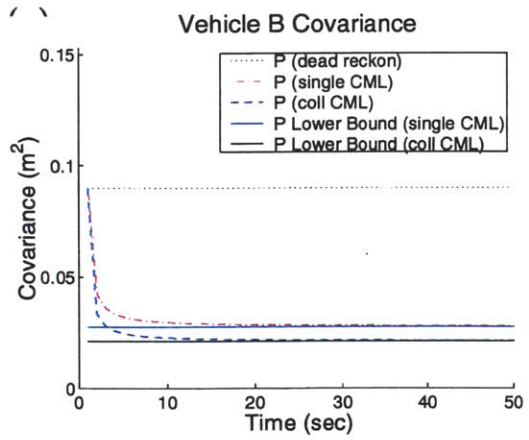


Vehicle A covariance versus time

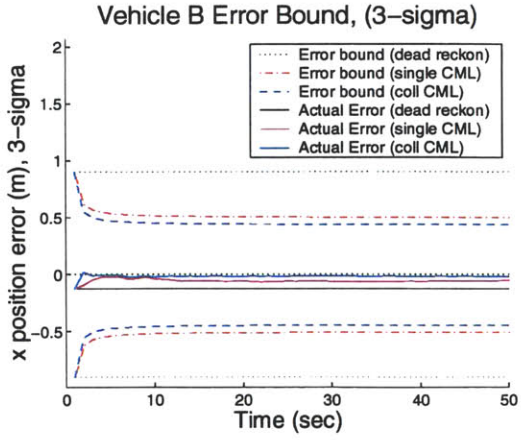


Vehicle A error versus time

Figure 4-4: 1-D scenario #1 vehicle A position error analysis, demonstrating convergence to the theoretical lower performance bound



Vehicle B covariance versus time



Vehicle B error versus time

Figure 4-5: 1-D scenario #1 vehicle B position error analysis, demonstrating convergence to the theoretical lower performance bound

4.2.2 1-D scenario #2

In this scenario, each vehicle starts with no initial position uncertainty. However, as each vehicle moves process noise is added. This scenario demonstrates how CML bounds error growth, as well as the performance improvement of collaborative CML over single vehicle CML. Tables 4.1 and 4.3 summarize the parameters used for this scenario.

Table 4.3: 1-D CCML simulation scenario #2 parameters

speed process noise std. dev. x_ω	0.15 m/s
initial vehicle position uncertainty std. dev.	0.0 m
initial feature position uncertainty std. dev.	0.4 m

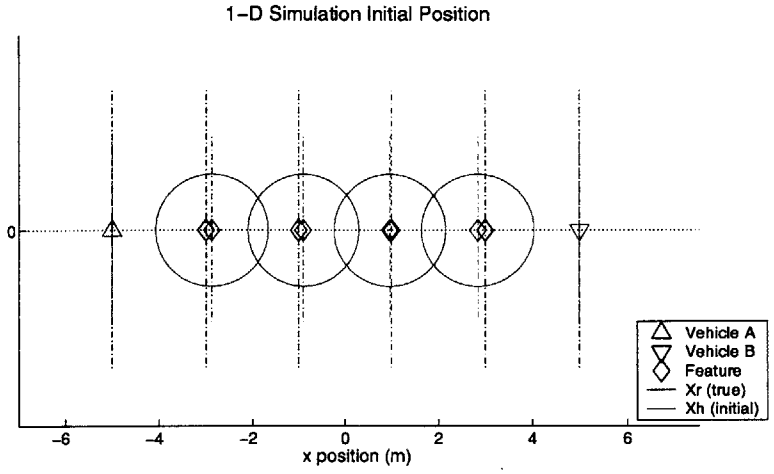


Figure 4-6: 1-D scenario #2 initial position

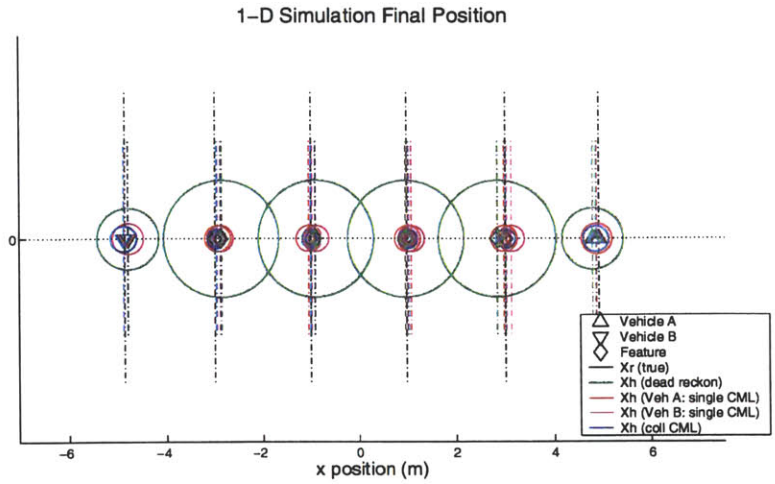
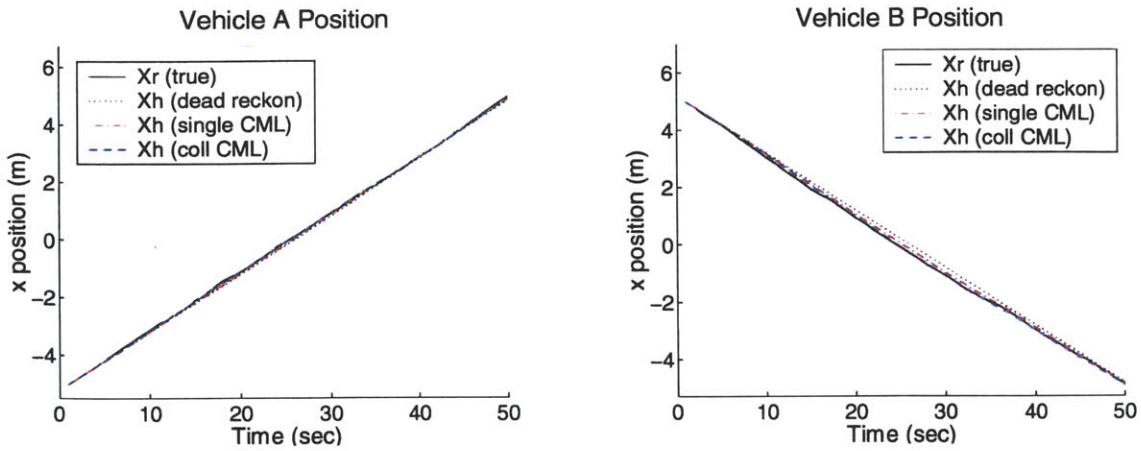
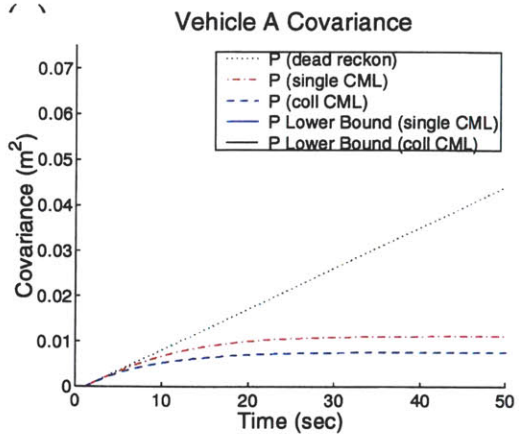


Figure 4-7: 1-D scenario #2 final position

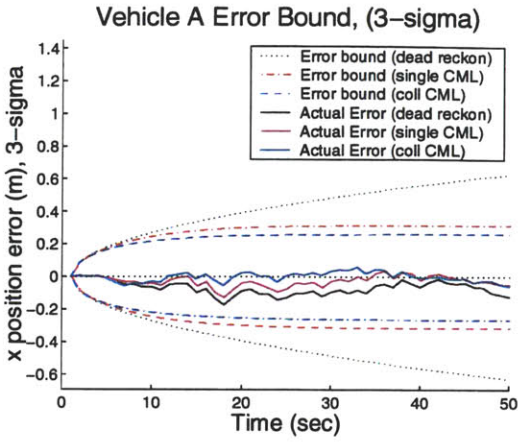


Vehicle A **Vehicle B**

Figure 4-8: 1-D scenario #2 vehicle position versus time

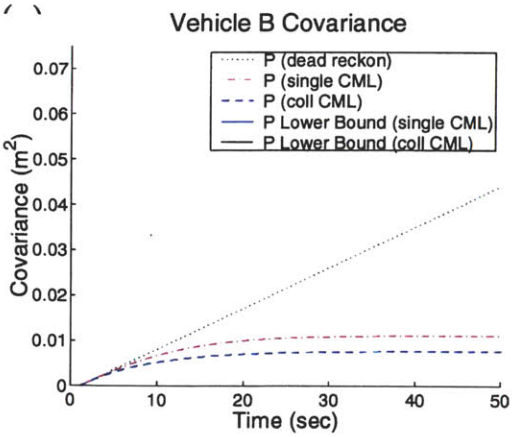


Vehicle A covariance versus time

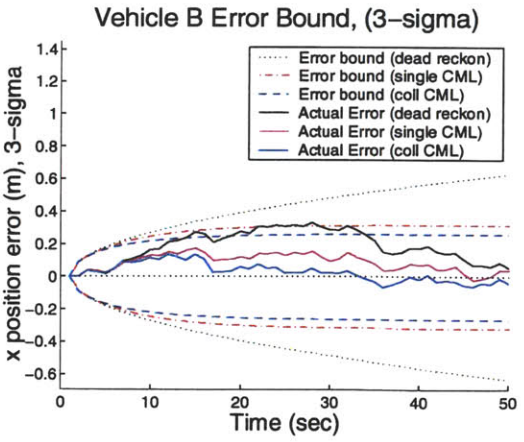


Vehicle A error versus time

Figure 4-9: 1-D scenario #2 vehicle A position error analysis



Vehicle B covariance versus time



Vehicle B error versus time

Figure 4-10: 1-D scenario #2 vehicle B position error analysis

4.2.3 1-D scenario #3

The final scenario is a combination of scenarios #1 and #2. Each vehicle starts out with an initial position uncertainty, and process noise is added in every time step. This scenario best reflects real world implementations. Tables 4.1 and 4.4 summarize the parameters used for this scenario.

Table 4.4: 1-D CCML simulation scenario #3 parameters

speed process noise std. dev. x_ω	0.15 m/s
initial vehicle position uncertainty std. dev.	0.3 m
initial feature position uncertainty std. dev.	0.4 m

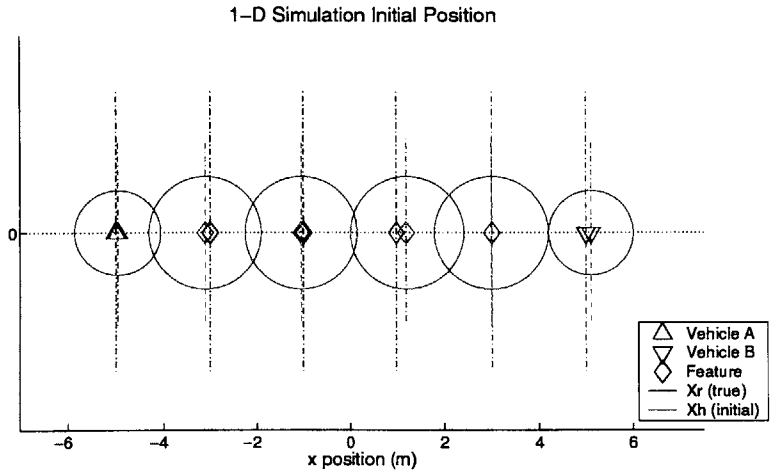


Figure 4-11: 1-D scenario #3 initial position

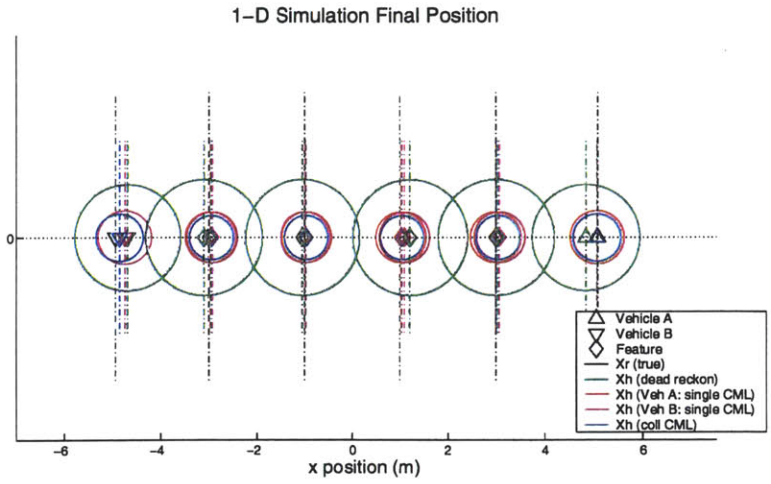


Figure 4-12: 1-D scenario #3 final position

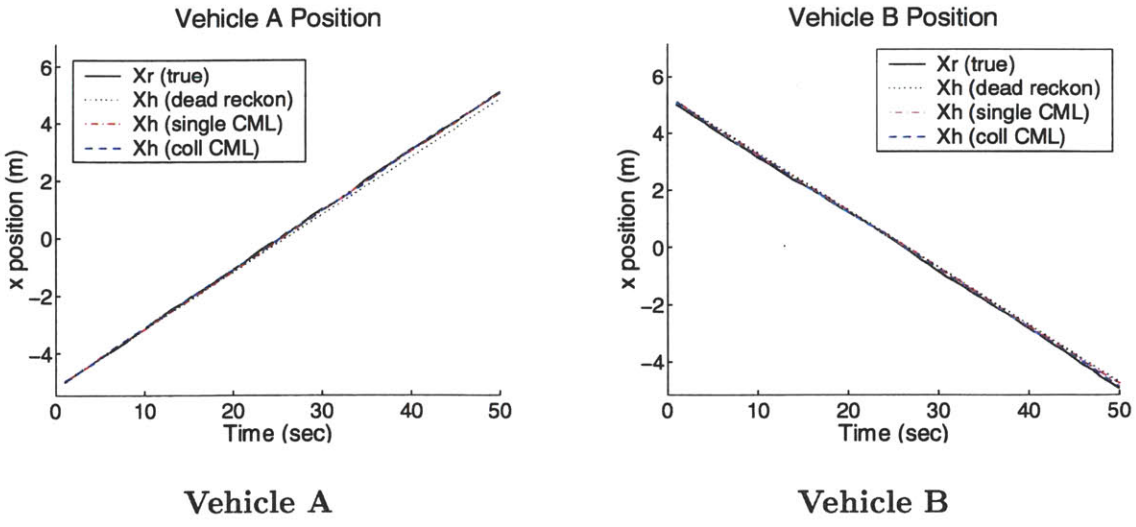
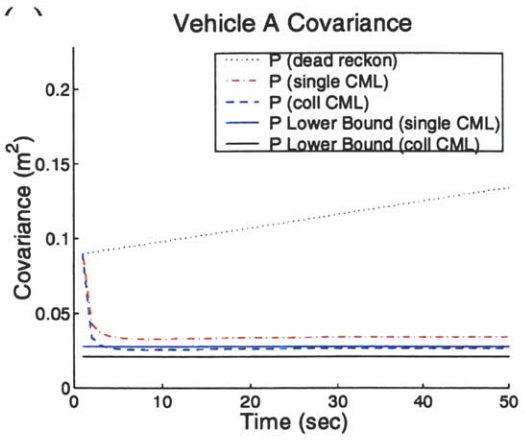
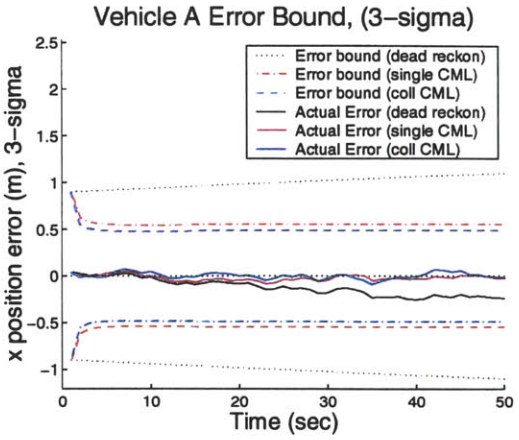


Figure 4-13: 1-D scenario #3 vehicle position versus time

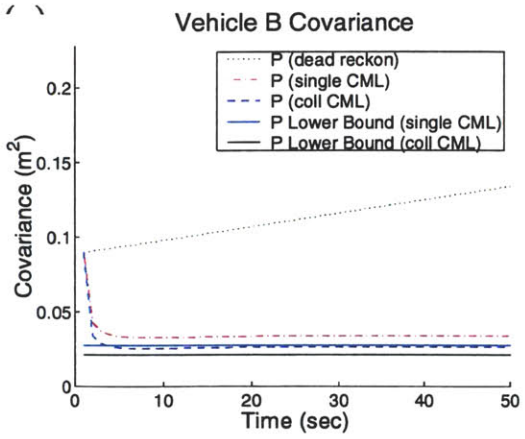


Vehicle A covariance versus time

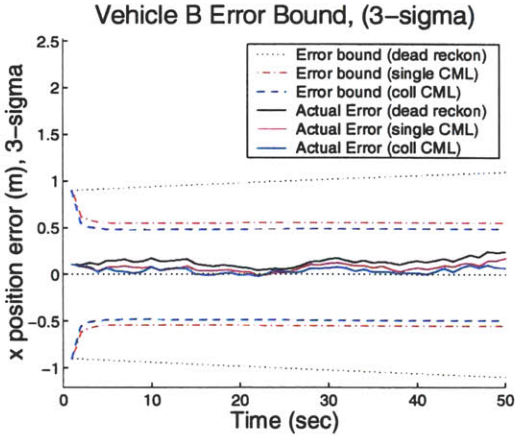


Vehicle A error versus time

Figure 4-14: 1-D scenario #3 vehicle A position error analysis



Vehicle B covariance versus time



Vehicle B error versus time

Figure 4-15: 1-D scenario #3 vehicle B position error analysis

4.3 Summary

This chapter presented a 1-D application of the collaborative CML algorithm presented in Chapter 3. This 1-D version provides a simple, easy to understand version of the collaborative extension to stochastic mapping. This extension is a simple one, adding a second vehicle to the state estimate vector and adapting the observation model to incorporate measurements from multiple vehicles. The 1-D simulation results demonstrate improved performance over single vehicle CML, and validates the theoretical lower error bound for collaboration.

Chapter 5

2-D Collaborative CML Simulation Results

This chapter presents results from 2-D simulations of the collaborative localization and collaborative CML algorithms presented in Chapter 3.

5.1 Simulation assumptions

A number of assumptions are made in the simulations demonstrated in this chapter. As in the 1-D case, both vehicles and features are modeled as points. Importantly, collaborating vehicles communicate motion commands and all sensor measurements at every time step. Thus the centralized collaborative CML filter has access to the identical information as the single vehicle CML filter.

In order to apply the collaborative algorithms in real world scenarios, features must be reliably extracted from the environment. A data association strategy is utilized using sonar as a means observing collaborating vehicles and environmental features. This strategy attempts to assign measurements to the features from which they originate, generating a correct observation model $\mathbf{h}$. A delayed nearest-neighbor

approach [4] is used to identify features. Clusters of similar, sequential measurements are saved. When the cluster contains enough measurements, it is initialized as a feature into the state vector. Once a feature is represented in the state vector, all subsequent measurements are compared to the feature estimate and tested for association via a gated nearest-neighbor comparison. A sonar simulator in the 2-D simulation generates a set of noisy 'actual' measurements to be associated. No dropouts or spurious measurements are included. Note, however, that collaborating vehicles communicate their own position information and thus are initialized into the collaborative state vector *a priori.*

5.2 2-D Collaborative Localization Results

In this simulation, two vehicles simultaneously travel circuitous paths in the absence of static features. The simulation results from three different scenarios are presented. The scenario sequence mirrors the 1-D CCML simulation structure presented in Section 4.2. In the first scenario, the vehicles are given an initial uncertainty and zero process noise, demonstrating convergence to a theoretical lower performance bound. In the second scenario, there is no initial vehicle uncertainty, but process noise is added at every time step. Finally, the third scenario demonstrates performance given both process noise and initial vehicle uncertainty.

Table 5.1: 2-D Collaborative Localization simulation global parameters

number of vehicles	2
sampling period T	0.2 sec.
range measurement std. dev. x_w	0.2 m
bearing measurement std. dev. ϕ_w	10 deg
vehicle cruise speed u	0.5 m/s

5.2.1 2-D CL scenario #1

In this scenario, there is no process noise added as each vehicle moves. However, each vehicle has an initial position uncertainty. Tables 5.1 and 5.2 summarize the parameters used for this scenario. The initial starting location of both vehicles is shown by Figure 5-1, with the 3σ (99% highest confidence region) error bound ellipse around each vehicle indicating the initial position uncertainty. Figure 5-2 shows the final estimated position and path of each vehicle after 300 seconds of travel. As is to be expected with no process noise, there is no difference between the true and estimated position. In the collaborative localization portion of Figure 5-2 the final set of direct sensor measurements are also shown. A direct comparison between dead-reckoning and collaborative localization position error is made in Figure 5-3. Figures 5-4 and 5-5 show plots of the position and heading errors of the vehicles versus time, along with 3σ bounds. Position error is presented in determinant form in Figures 5-6 and 5-7. Due to the zero additive process noise, dead reckoning error will stay constant as no information is lost from movement. This plot also clearly shows the decrease in error uncertainty to the theoretical lower bound predicted by Equation 3.48. The extra information provided by the initially uncorrelated position of the collaborating vehicle produces a reduction in position uncertainty as the collaborating vehicle is directly observed.

Table 5.2: 2-D CL simulation scenario #1 parameters

x position process noise std. dev.	0.0 m/s
y position process noise std. dev.	0.0 m/s
heading process noise std. dev.	0.0 deg/s
velocity process noise std. dev.	0.0 m/s
initial vehicle x position uncertainty std. dev.	0.2 m
initial vehicle y position uncertainty std. dev.	0.2 m
initial heading position uncertainty std. dev.	0.0 deg

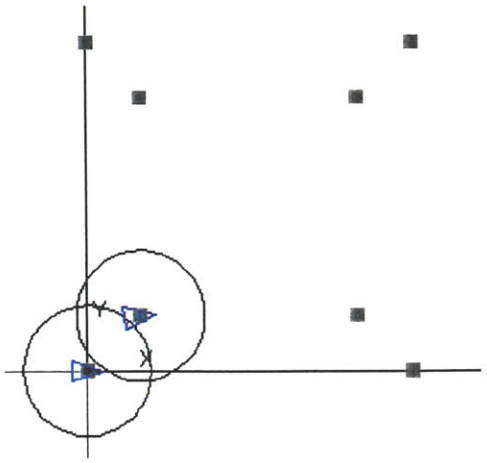


Figure 5-1: 2-D CL scenario #1 : vehicle starting position

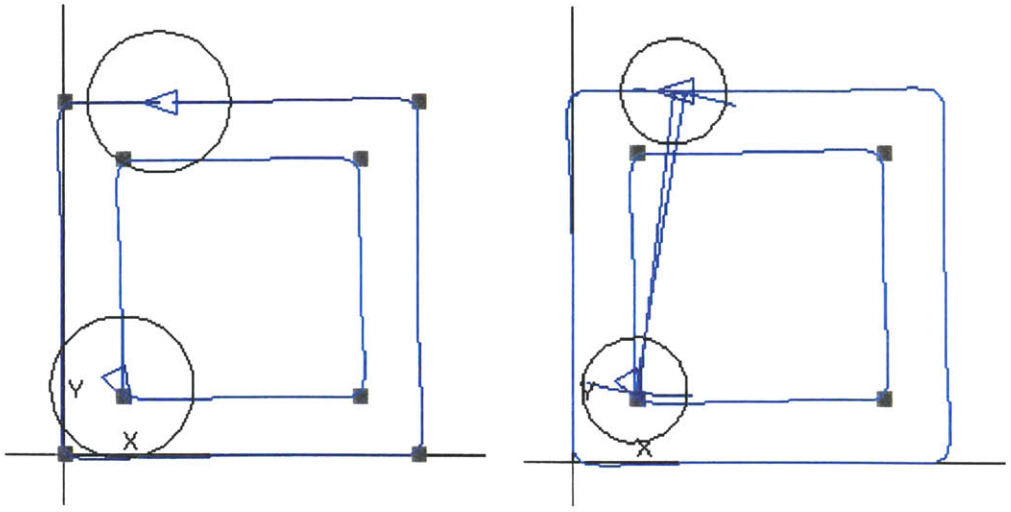


Figure 5-2: 2-D CL scenario #1 : final position estimates

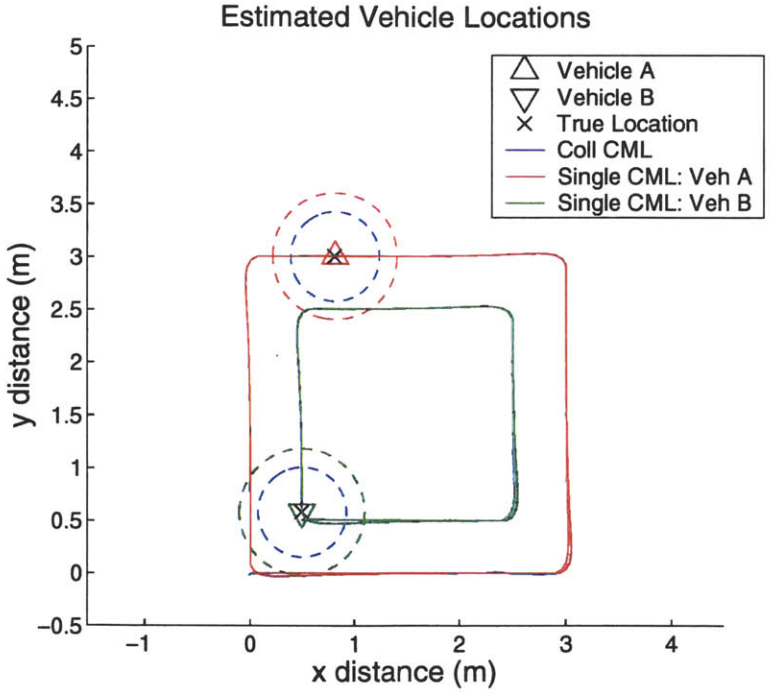


Figure 5-3: 2-D CL scenario #1 : position estimate comparison

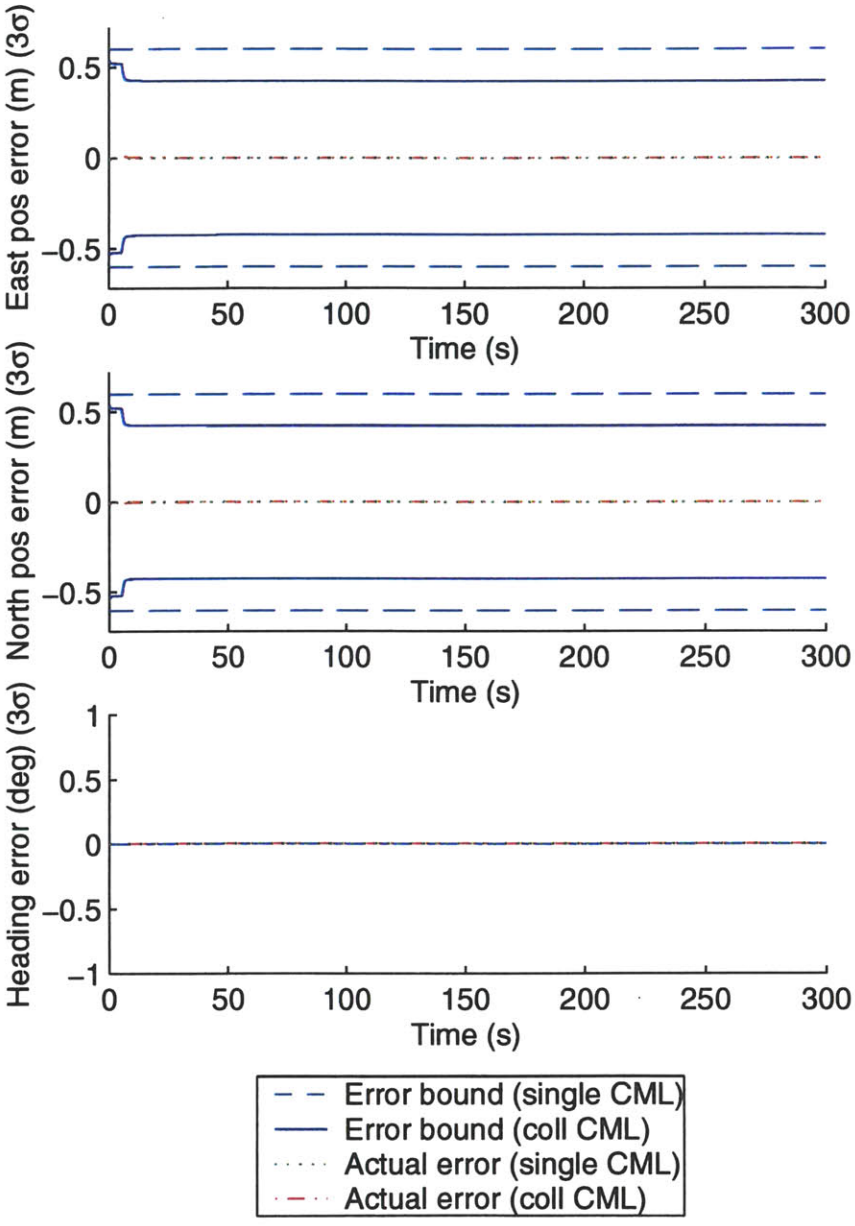


Figure 5-4: 2-D CL scenario #1 : vehicle A error comparison

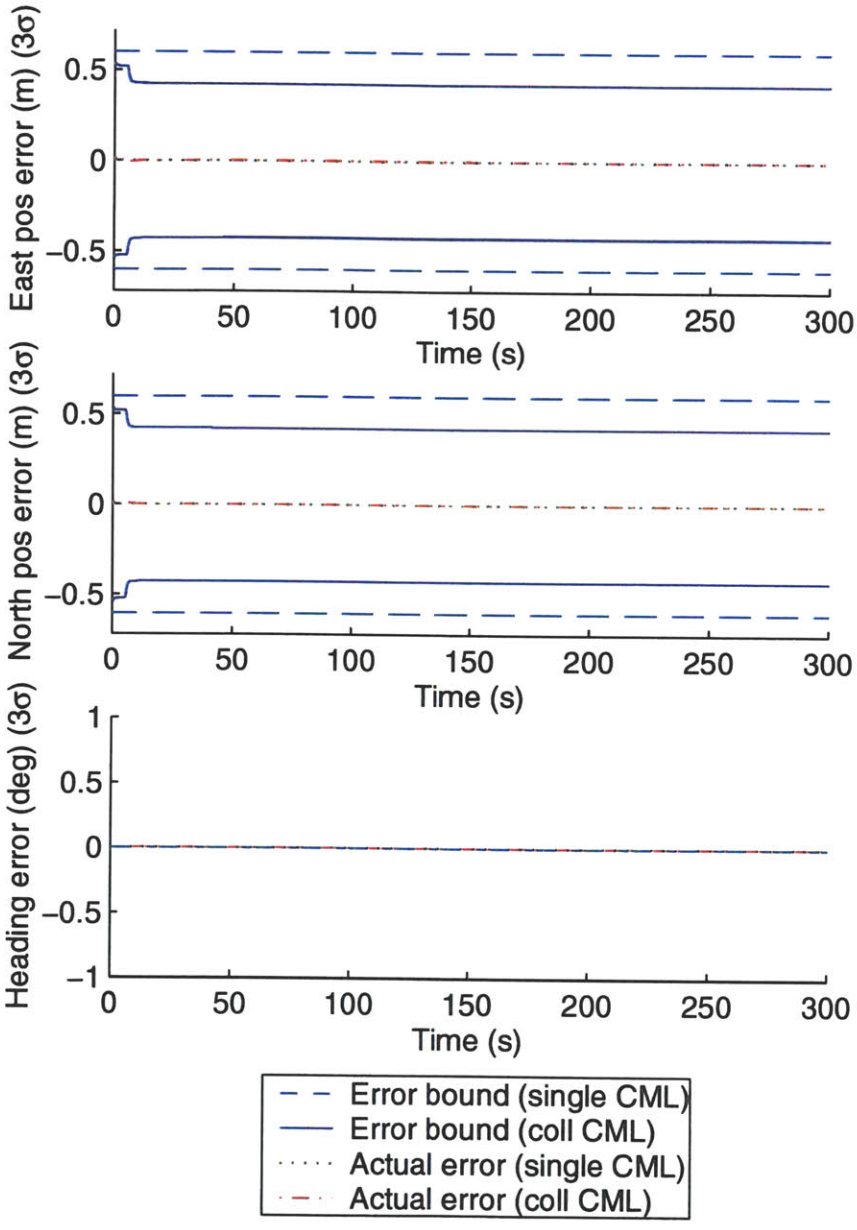


Figure 5-5: 2-D CL scenario #1 : vehicle B error comparison

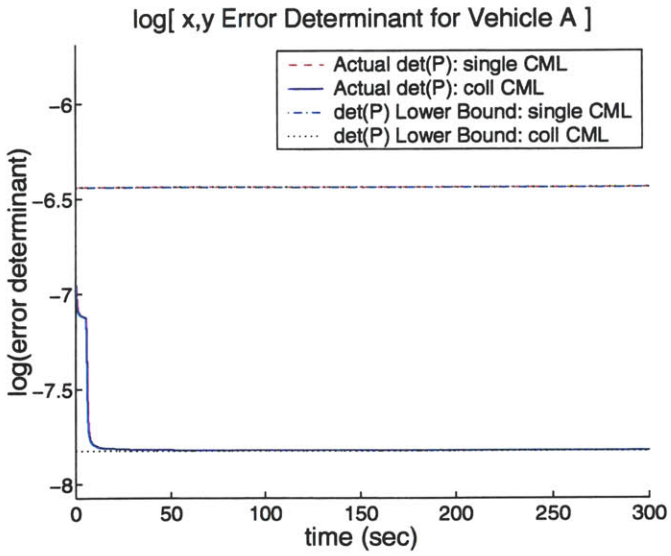


Figure 5-6: 2-D CL scenario #1 : vehicle A error determinant

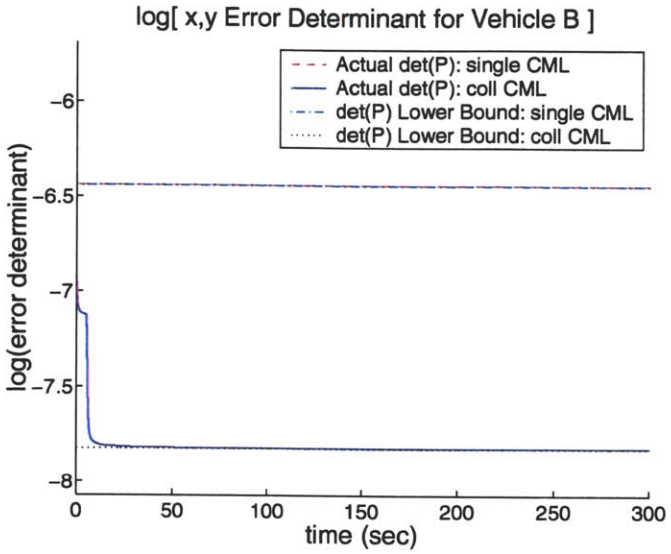


Figure 5-7: 2-D CL scenario #1 : vehicle B error determinant

5.2.2 2-D CL scenario #2

In this scenario, the starting position for each vehicle is known exactly. As each vehicle moves, process noise is added and a dead-reckoning position estimate is maintained. This scenario demonstrates that collaborative localization results in slower error growth in the presence of process noise as compared to dead-reckoning. Tables 5.1 and 5.3 summarize the parameters used for this scenario. The vehicle starting positions shown by Figure 5-8 are unchanged from Scenario #1, though the process noise produces drift in the position estimate that is evident in Figure 5-9. Figures 5-11 and 5-12 demonstrate the slower linear error growth rate produced by collaborative localization as compared to dead reckoning.

Table 5.3: 2-D CL simulation scenario #2 parameters

x position process noise std. dev.	0.075 m/s
y position process noise std. dev.	0.075 m/s
heading process noise std. dev.	0.25 deg/s
velocity process noise std. dev.	0.0 m/s
initial vehicle x position uncertainty std. dev.	0.0 m
initial vehicle y position uncertainty std. dev.	0.0 m
initial heading position uncertainty std. dev.	0.0 deg

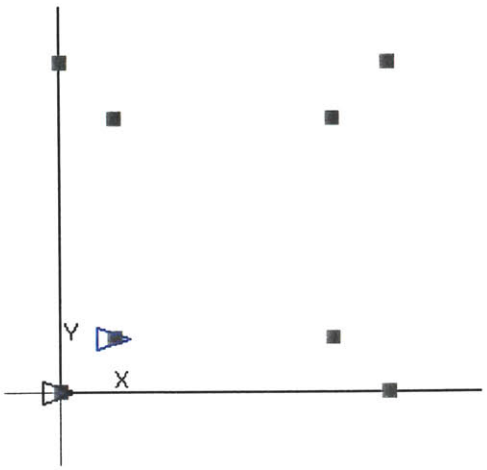


Figure 5-8: 2-D CL scenario #2 : vehicle starting position

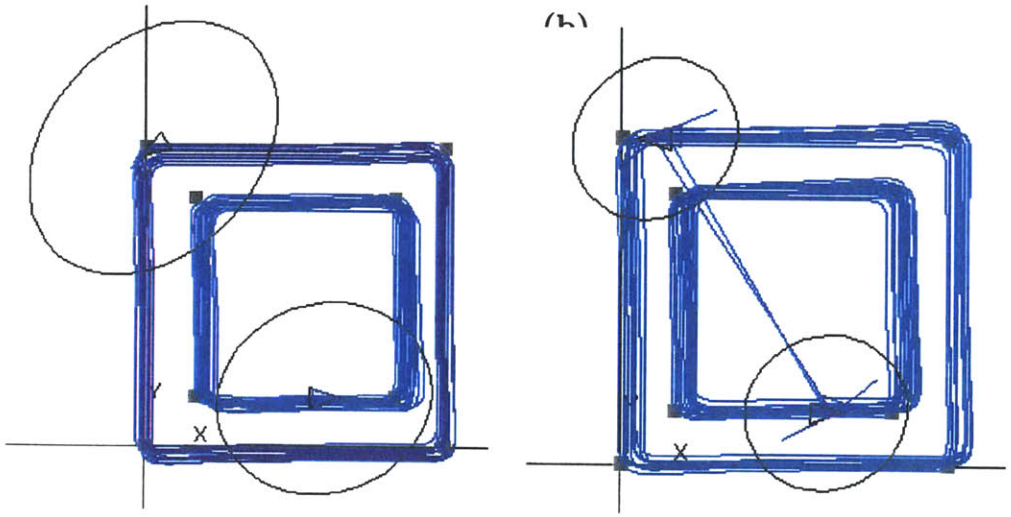


Figure 5-9: 2-D CL scenario #2 : final position estimates

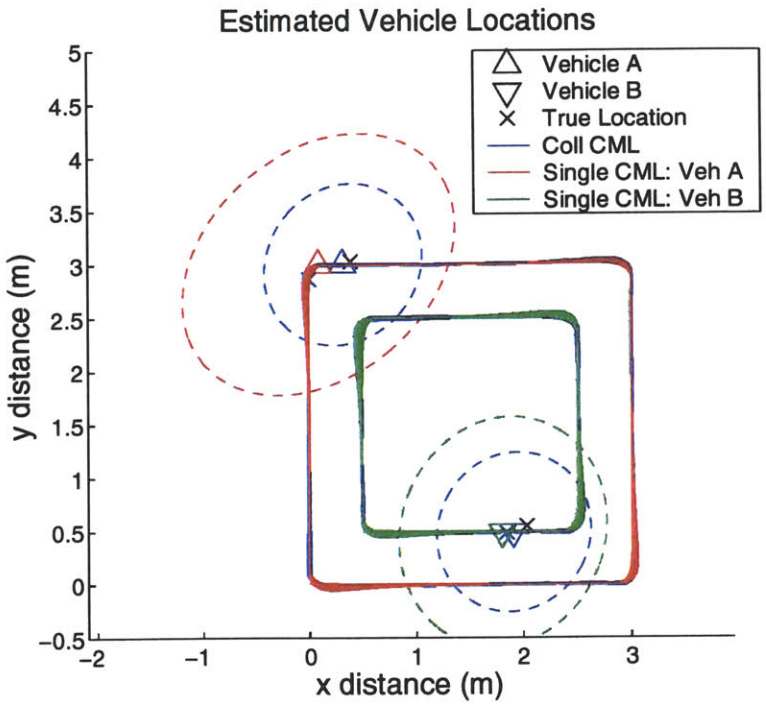


Figure 5-10: 2-D CL scenario #2 : position estimate comparison

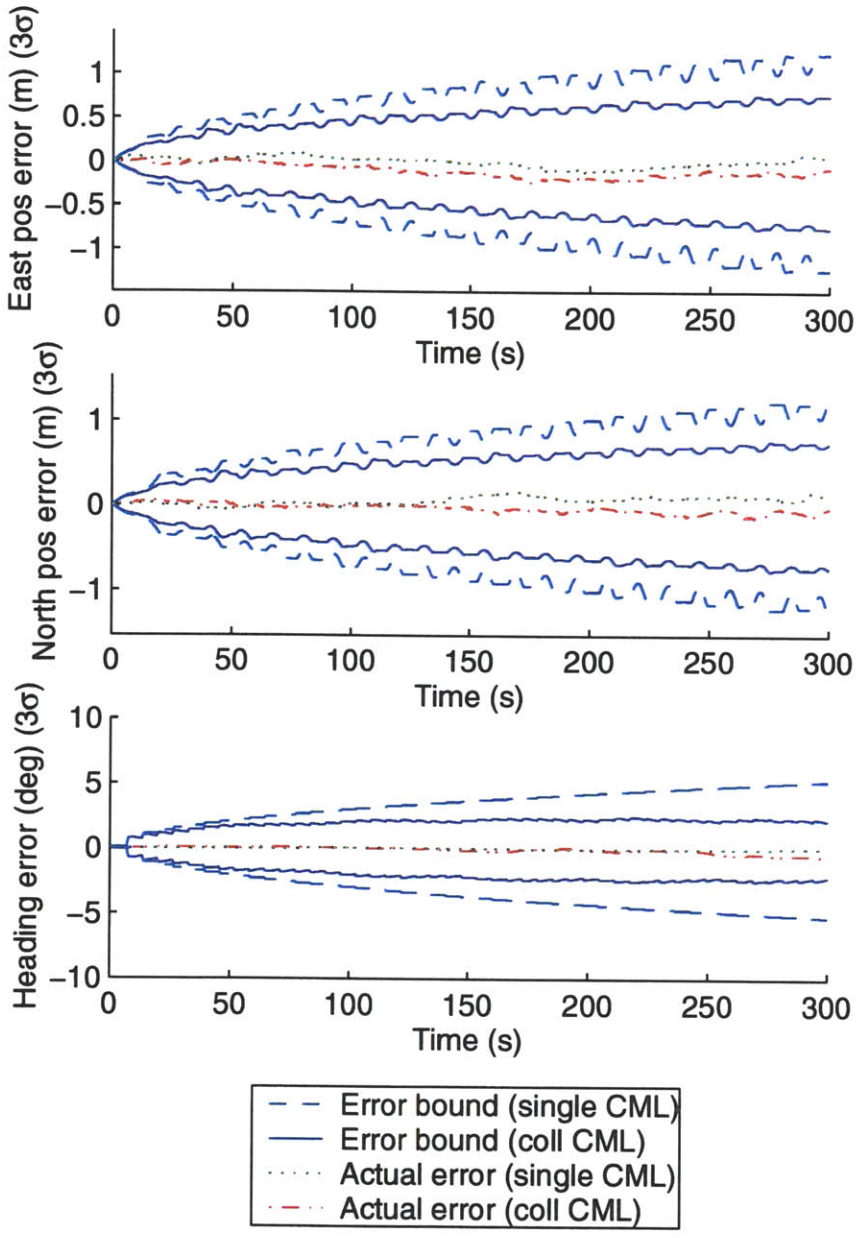


Figure 5-11: 2-D CL scenario #2 : vehicle A error comparison

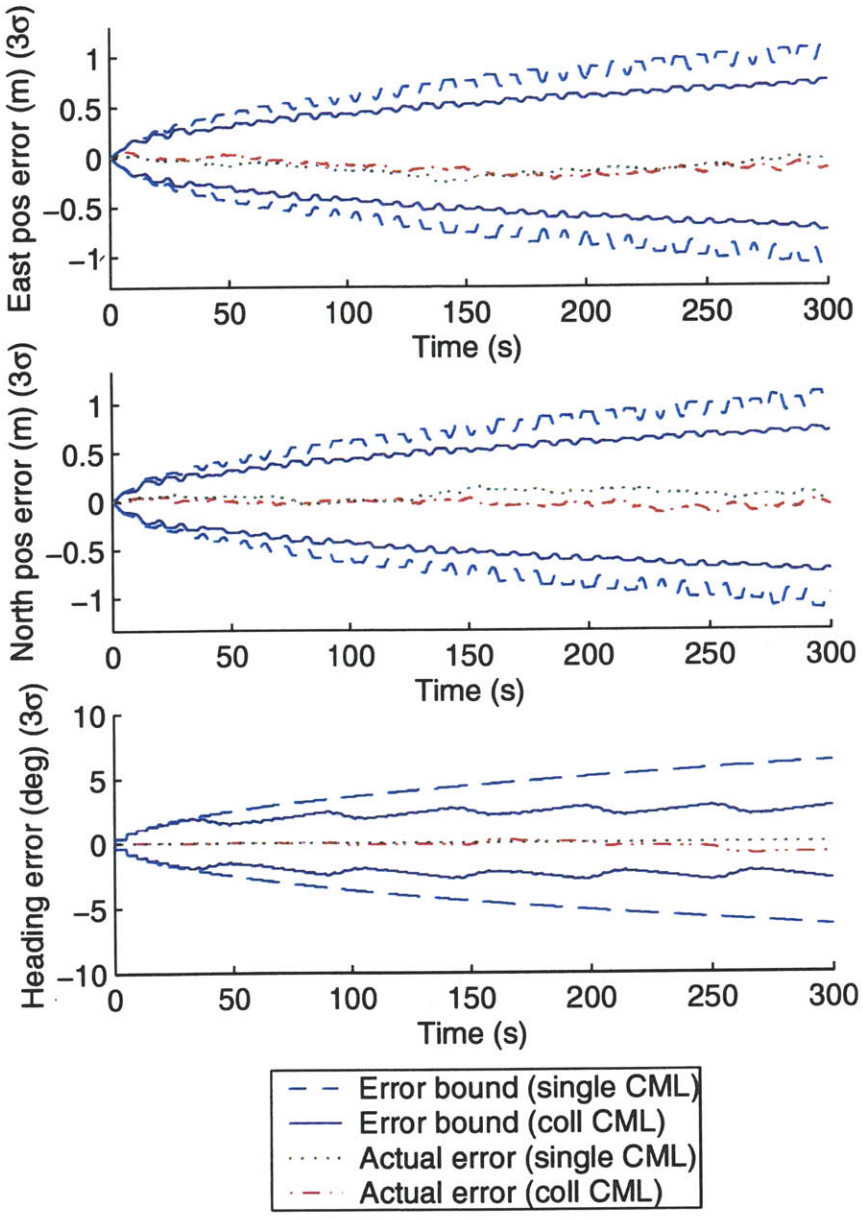


Figure 5-12: 2-D CL scenario #2 : vehicle B error comparison

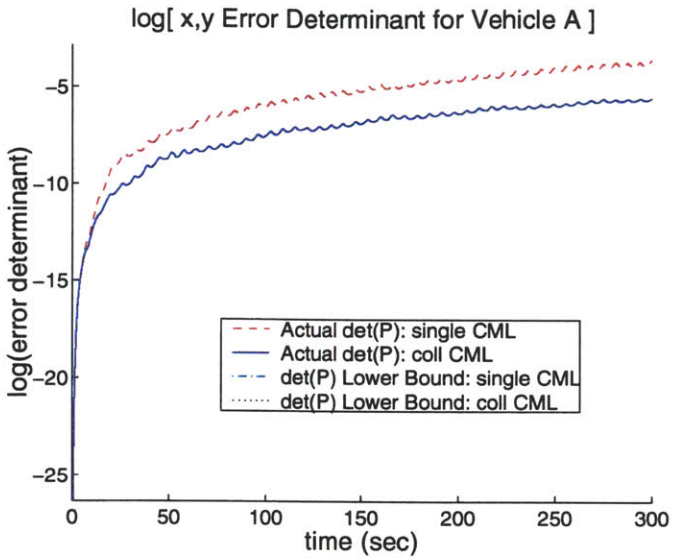


Figure 5-13: 2-D CL scenario #2 : vehicle A error determinant

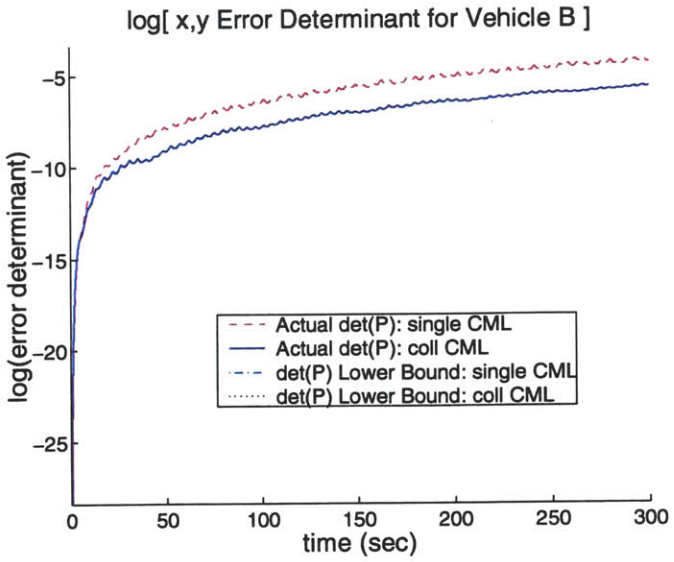


Figure 5-14: 2-D CL scenario #2 : vehicle B error determinant

5.2.3 2-D CL scenario #3

This scenario best simulates actual vehicle collaborative localization implementations, as both initial position uncertainty and process noise are present. Tables 5.1 and 5.4 summarize the parameters used for this scenario. Position error growth plotted in Figure 5-20 demonstrates the performance advantage provided by collaborative localization.

Table 5.4: 2-D CL simulation scenario #3 parameters

x position process noise std. dev.	0.075 m/s
y position process noise std. dev.	0.075 m/s
heading process noise std. dev.	0.25 deg/s
velocity process noise std. dev.	0.0 m/s
initial vehicle x position uncertainty std. dev.	0.2 m
initial vehicle y position uncertainty std. dev.	0.2 m
initial heading position uncertainty std. dev.	0.0 deg

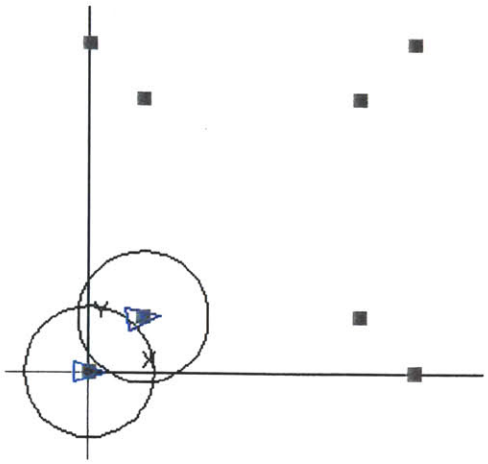
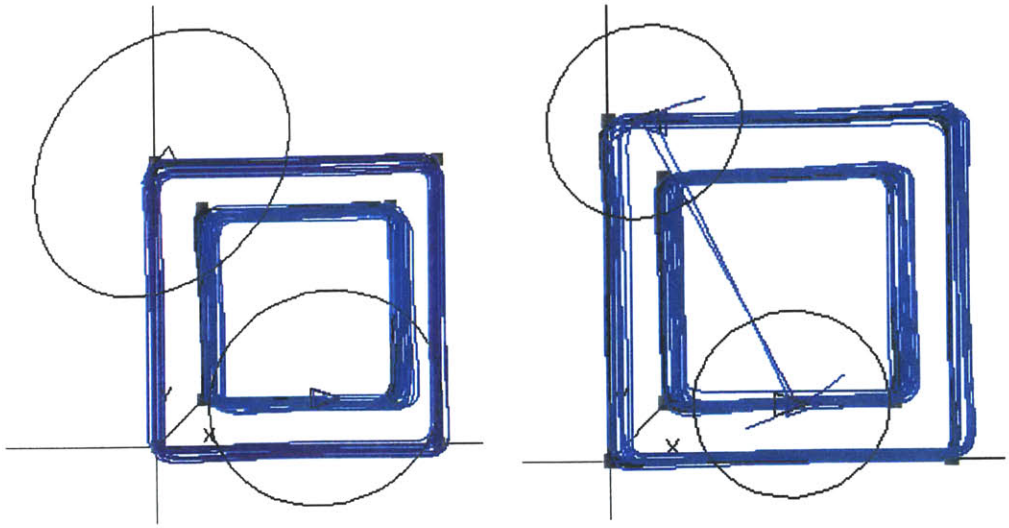


Figure 5-15: 2-D CL scenario #3 : vehicle starting position



Dead-reckoning **Collaborative localization**

Figure 5-16: 2-D CL scenario #3 : final position estimates

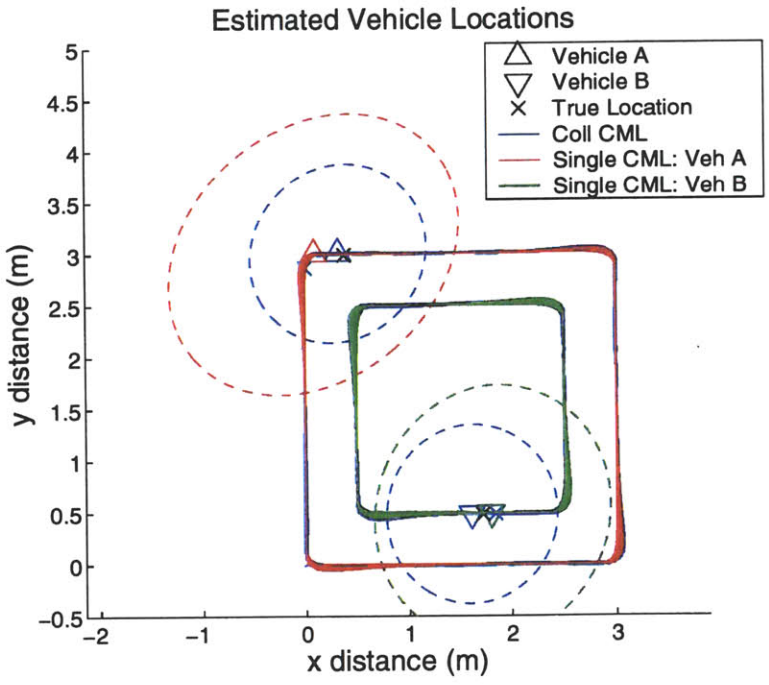


Figure 5-17: 2-D CL scenario #3 : position estimate comparison

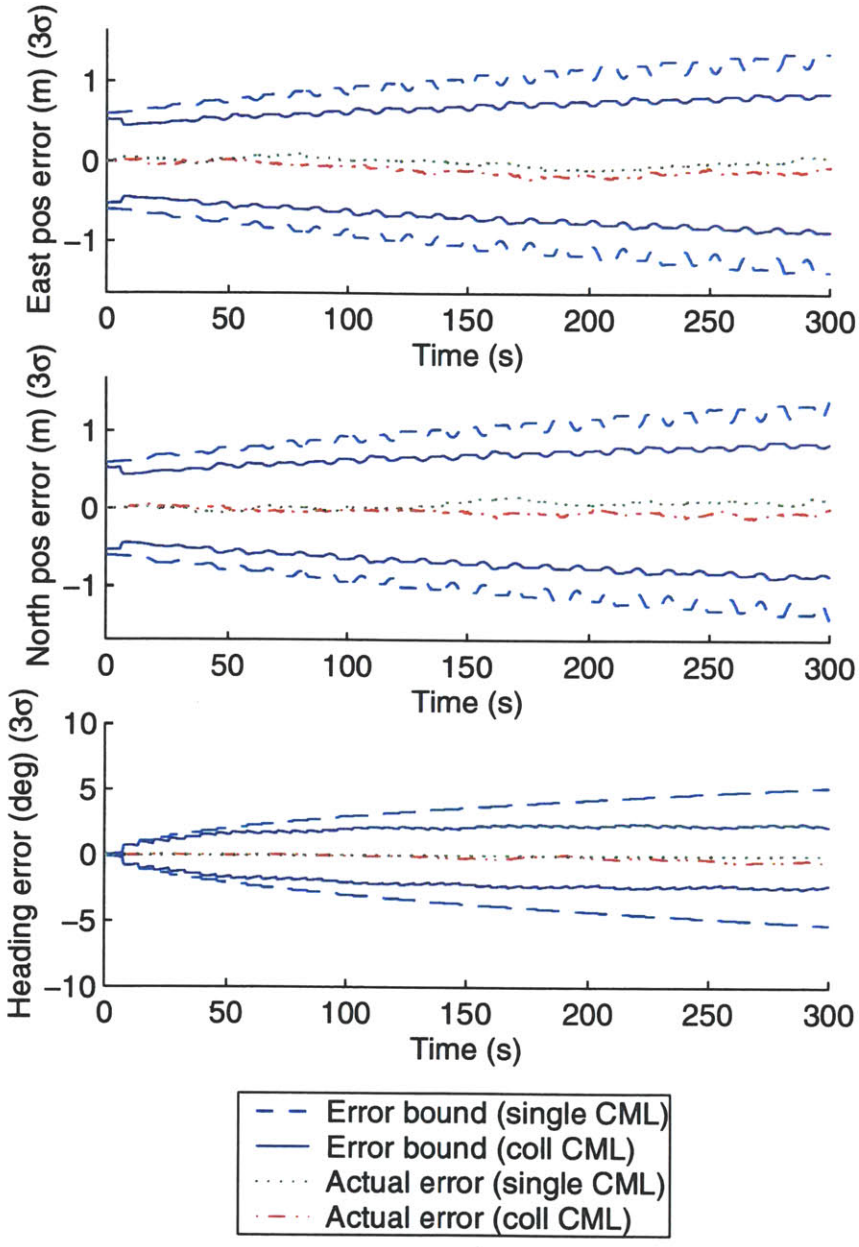


Figure 5-18: 2-D CL scenario #3 : vehicle A error comparison

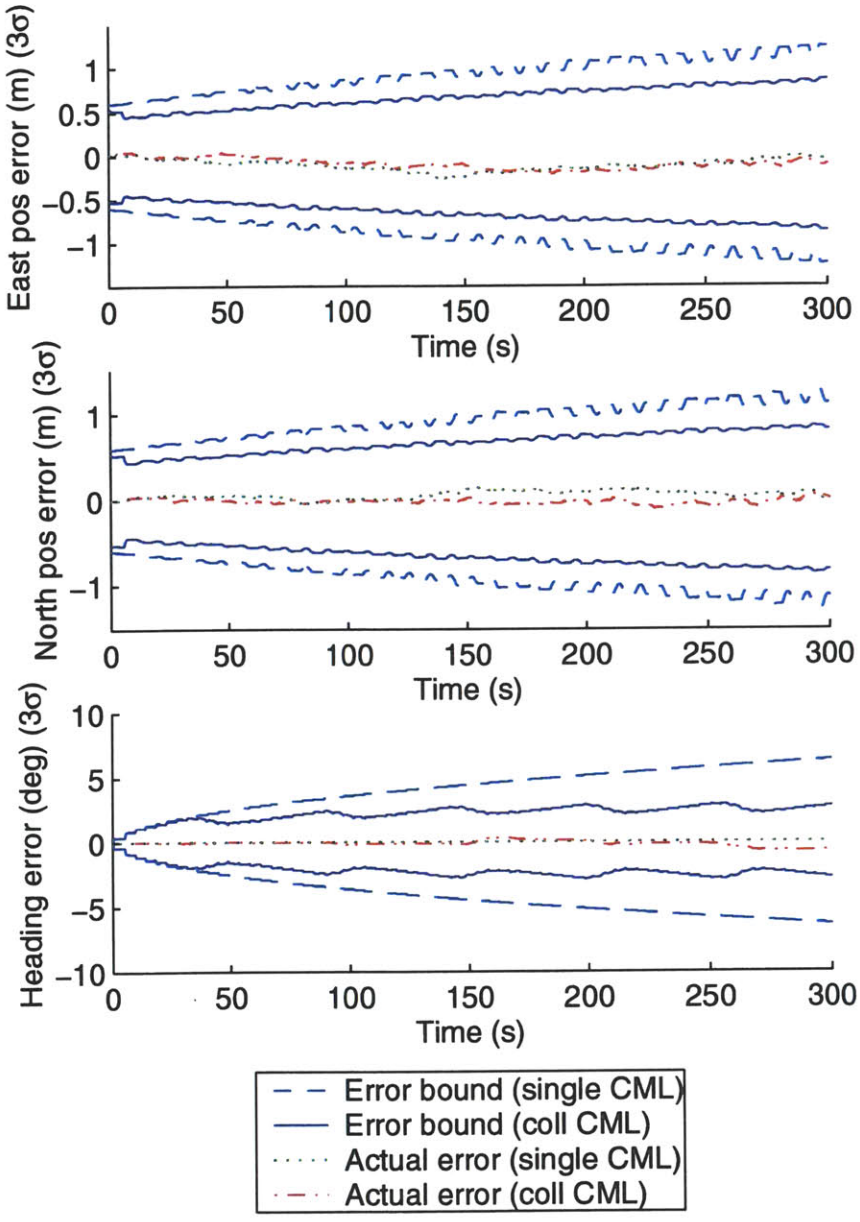


Figure 5-19: 2-D CL scenario #3 : vehicle B error comparison

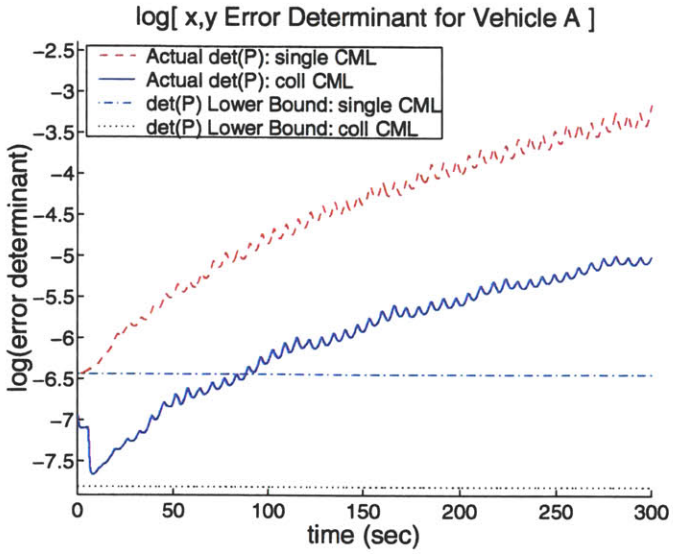


Figure 5-20: 2-D CL scenario #3 : vehicle A error determinant

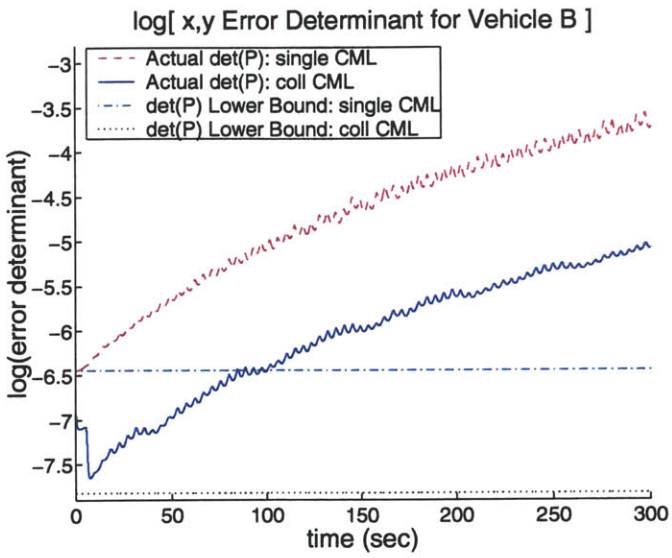


Figure 5-21: 2-D CL scenario #3 : vehicle B error determinant

5.3 2-D Collaborative CML Results

This section presents 2-D collaborative CML simulation results from three scenarios, structured similarly to the 1-D collaborative CML and 2-D collaborative localization scenarios. In the first scenario, the vehicles are given initial uncertainty and zero process noise, demonstrating convergence to a theoretical lower performance bound. In the second scenario, there is no initial vehicle uncertainty, but process noise is added at every time step. Finally, the third scenario demonstrates performance given both process noise and initial vehicle uncertainty. Table 5.5 summarizes the global parameters consistent for all three scenarios. Initial vehicle locations, feature locations, and vehicle paths are also kept consistent. Note that unlike the 1-D CML simulation, features are not initially present in the collaborative CML state estimate. Rather, the data association algorithm described in Section 5.1 is used to identify features and only then is the feature estimate added to the state vector. However, as in the 2-D collaborative localization algorithm, both collaborating vehicles are initially present

in the state estimate.

Table 5.5: 2-D Collaborative CML simulation global parameters

number of vehicles	2
number of features	4
sampling period T	0.2 sec.
range measurement std. dev. x_w	0.2 m
bearing measurement std. dev. ϕ_w	10 deg
vehicle cruise speed u	0.5 m/s

5.3.1 2-D CCML scenario #1

In this scenario, there is no process noise added as each vehicle moves. However, each vehicle has an initial position uncertainty. Because of the zero additive process noise, dead reckoning error will stay constant since no information is lost due to movement. The extra information provided by the initially uncorrelated position of the collaborating vehicle produces a reduction in position uncertainty as the collaborating vehicle is directly observed. Tables 5.5 and 5.6 summarize the parameters used for this scenario.

The initial starting location of both vehicles is shown by Figure 5-22, with the 3σ error bound ellipse around each vehicle indicating the initial position uncertainty. Figure 5-23 shows the final estimated vehicle and feature positions after 300 seconds of travel. As is to be expected with no process noise, there is no difference between the true and estimated position. These plots also display the last set of sensor range measurements taken by each vehicle. A direct comparison between vehicle position errors is made in Figure 5-24. Figures 5-25 and 5-26 show plots of the position and heading errors of the vehicles versus time, along with 3σ bounds. Vehicle position error is presented in determinant form in Figures 5-27 and . 5-28. Because of the zero additive process noise, position error never increases. However, the single vehicle CML error remains constant, supporting Theorem 2.3, which states that position uncertainty for single vehicle CML can never be lower than the initial uncertainty. This plot also clearly shows the decrease in collaborative CML error uncertainty to the theoretical lower bound predicted by Equation 3.48. The extra information provided by the initially uncorrelated position of the collaborating vehicle provides a reduction in position uncertainty as information is shared. Note that the collaborative CML error determinant generated by this simulation is slightly less that that predicted. This overconfidence is result of the linearization used to construct the measurement Jacobian. Figure 5-29 plots the feature error estimates in determinant form. Each

feature is initialized at a different time step depending on the location of the feature relative to the vehicles' outcome of the data association algorithm. These plots demonstrate the convergence of all feature estimates to the same uncertainty as the collaborating vehicles, supporting Theorem 2.2.

Table 5.6: 2-D CCML simulation scenario #1 parameters

x position process noise std. dev.	0.0 m/s
y position process noise std. dev.	0.0 m/s
heading process noise std. dev.	0.0 deg/s
velocity process noise std. dev.	0.0 m/s
initial vehicle x position uncertainty std. dev.	0.2 m
initial vehicle y position uncertainty std. dev.	0.2 m
initial heading position uncertainty std. dev.	0.0 deg

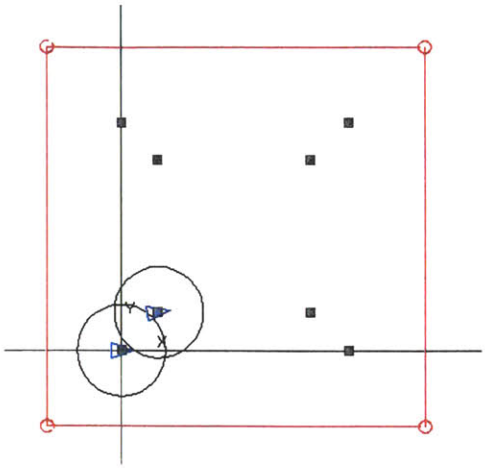
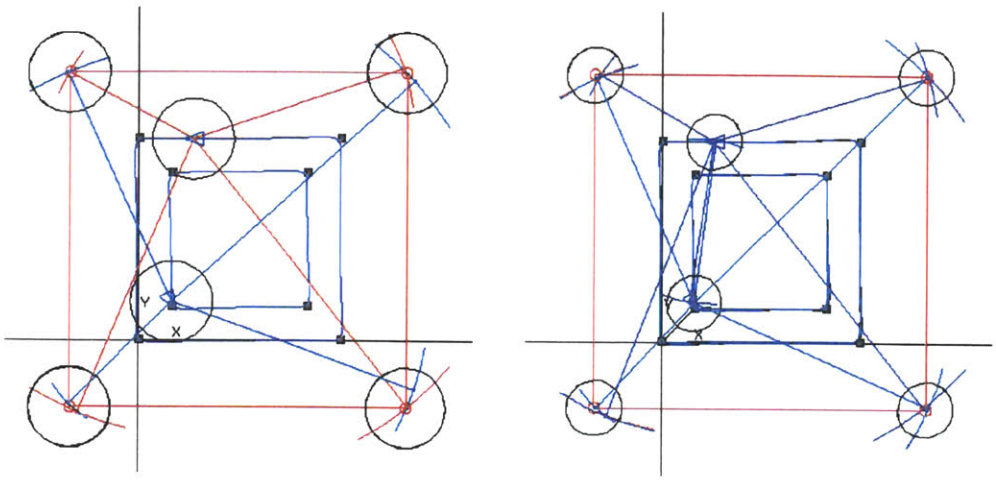


Figure 5-22: 2-D CCML scenario #1 : vehicle starting position



Single vehicle CML **Collaborative CML**

Figure 5-23: 2-D CCML scenario #1 : final position estimates

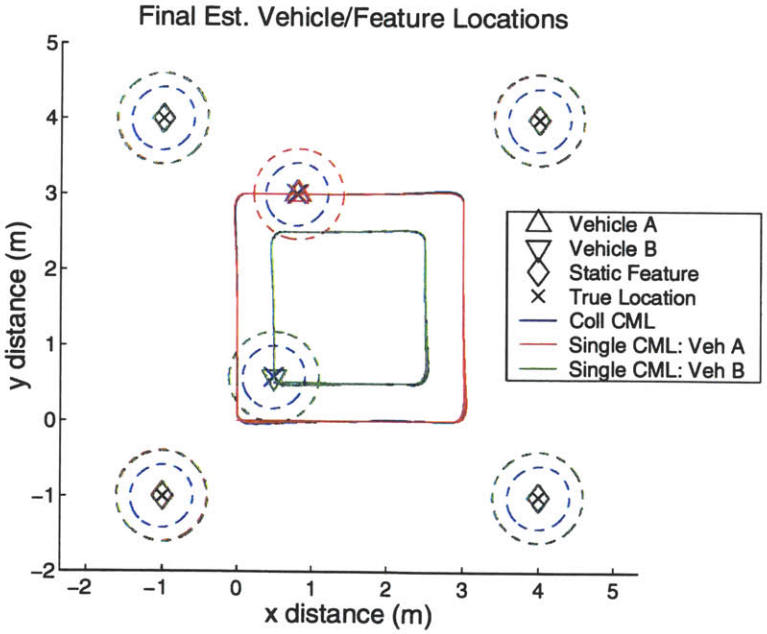


Figure 5-24: 2-D CCML scenario #1 : position estimate comparison

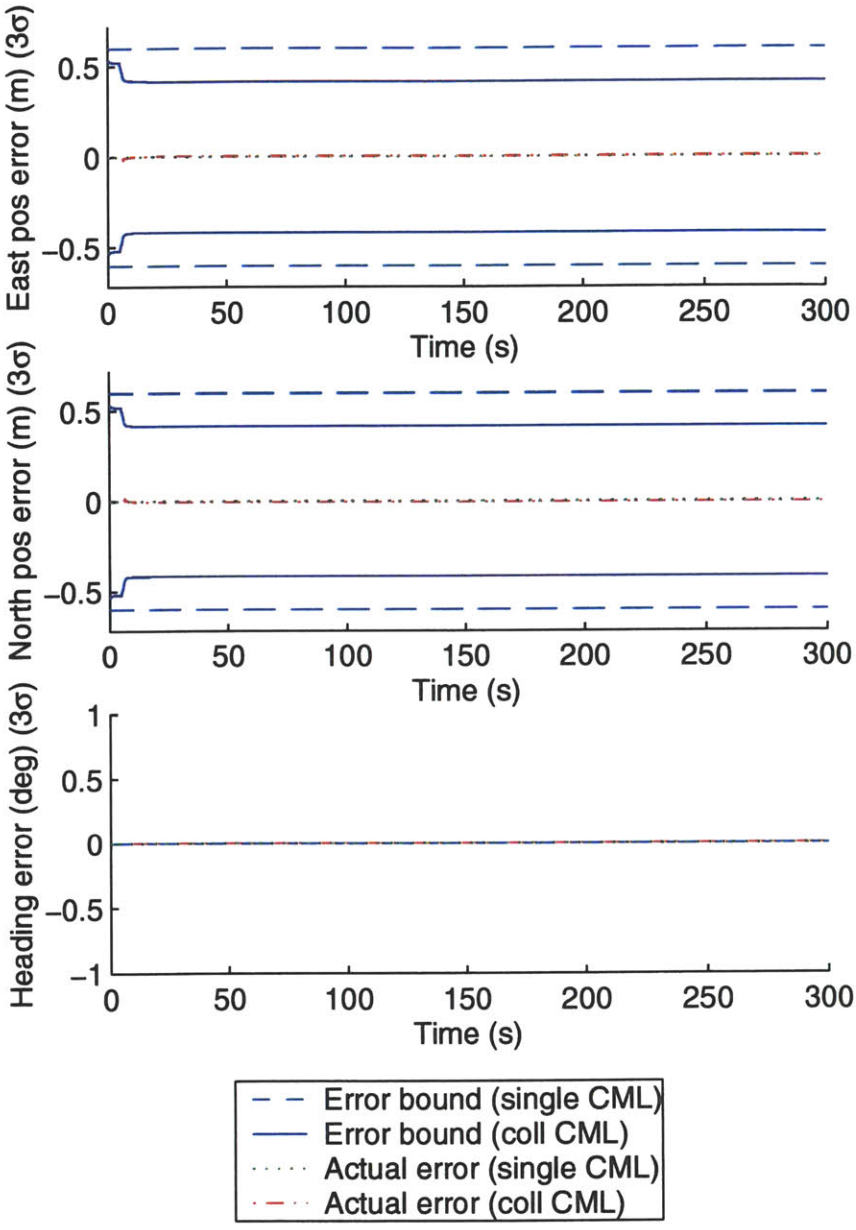


Figure 5-25: 2-D CCML scenario #1 : vehicle A error comparison

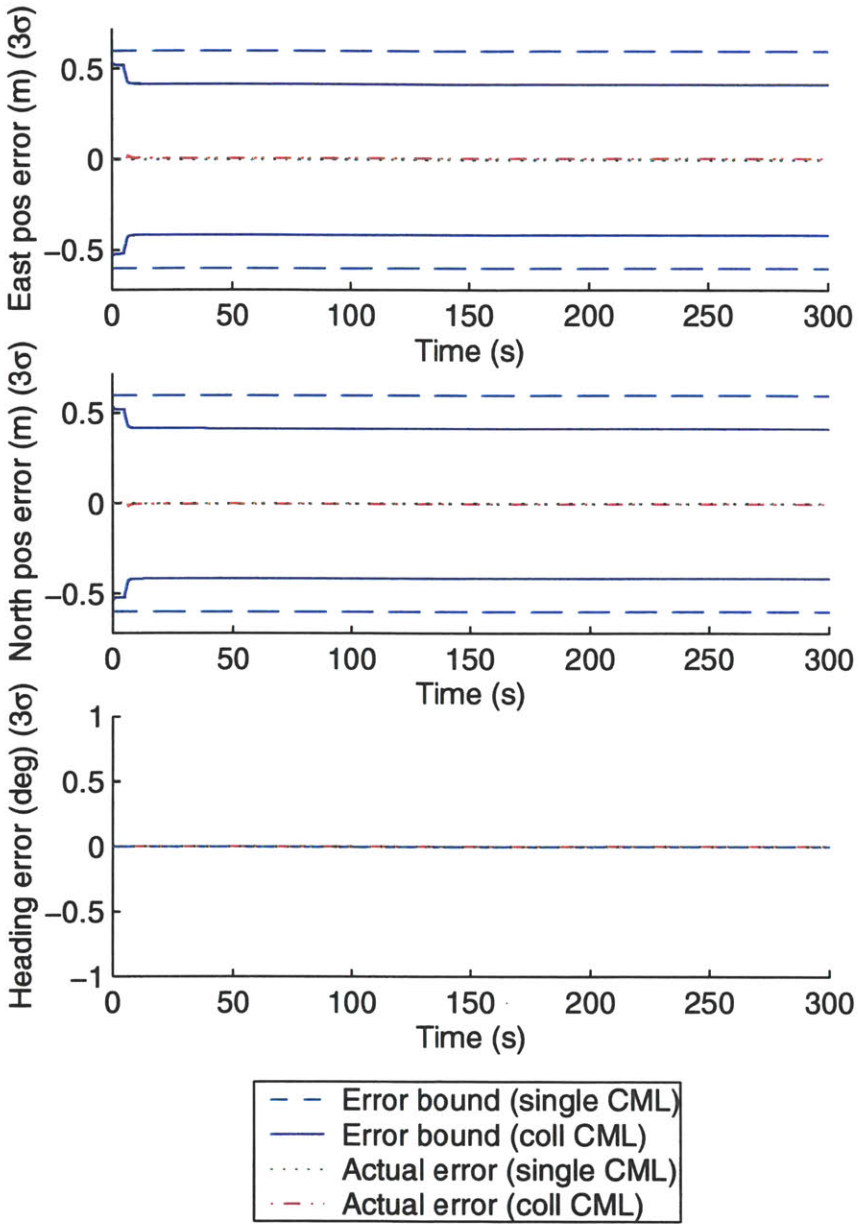


Figure 5-26: 2-D CCML scenario #1 : vehicle B error comparison

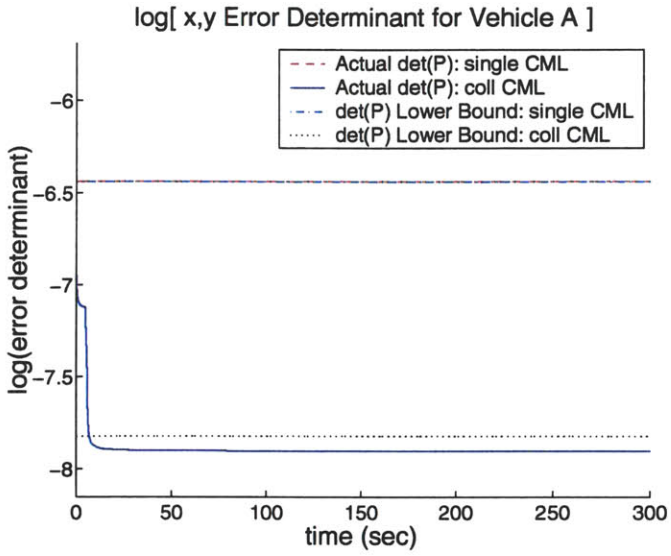


Figure 5-27: 2-D CCML scenario #1 : vehicle A error determinant

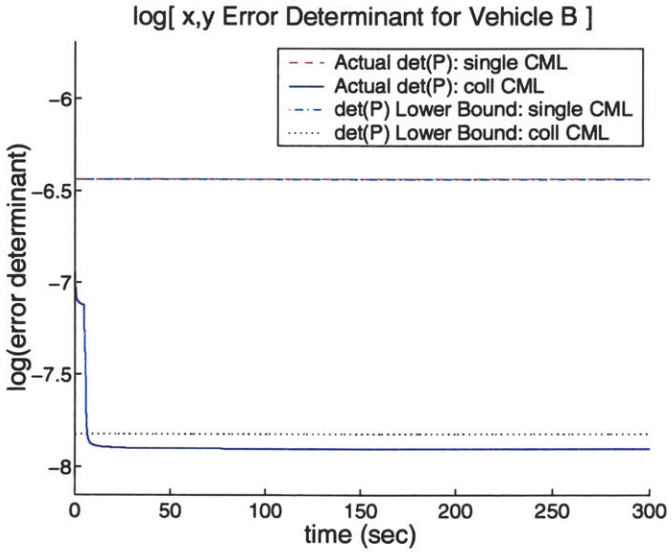
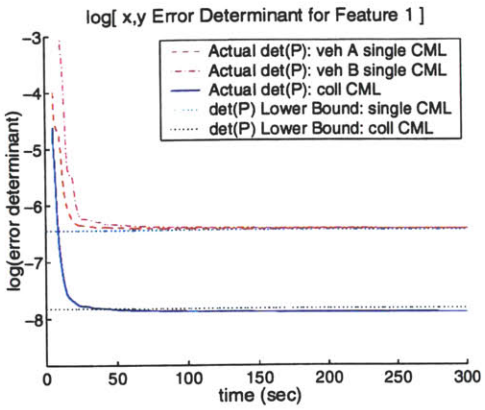
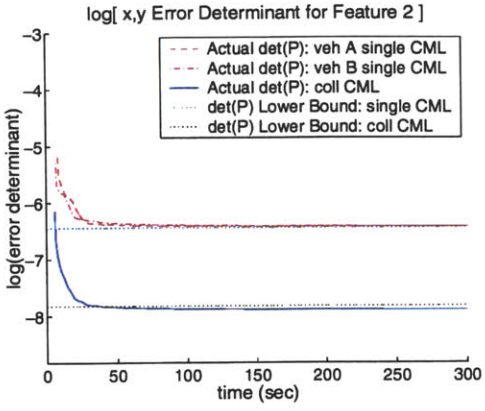


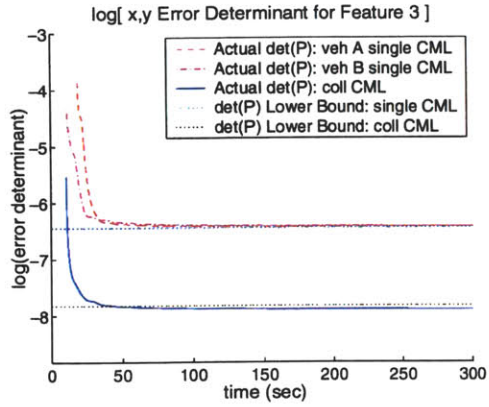
Figure 5-28: 2-D CL scenario #1 : vehicle B error determinant



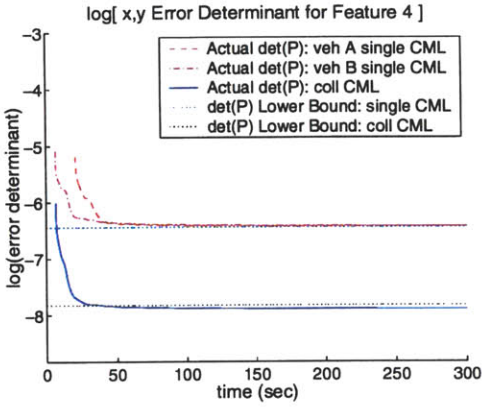
Feature 1 error versus time



Feature 2 error versus time



Feature 3 error versus time



Feature 4 error versus time

Figure 5-29: 2-D CCML scenario #1 : feature error determinant comparison

5.3.2 2-D CCML scenario #2

In this scenario, the starting position for each vehicle is known exactly. As each vehicle moves, process noise is added. Tables 5.5 and 5.7 summarize the parameters used for this scenario. The vehicle starting positions shown by Figure 5-30 are unchanged from Scenario #1, though the process noise produces drift in the position estimate that is evident in Figure 5-31. These plots also include the dead-reckoning position estimates. Note that because motion commands are based on an estimated position, the final vehicle locations for the single vehicle and collaborative CML are different, as reflected by Figure 5-32. Error bound and actual error comparison for each vehicle are presented in Figures 5-33 and 5-34. Error determinant plots for the vehicles are presented Figure 5-37. While both single and collaborative CML produces bounded error growth, the performance advantage of collaboration is evident.

Table 5.7: 2-D CCML simulation scenario #2 parameters

x position process noise std. dev.	0.1 m/s
y position process noise std. dev.	0.1 m/s
heading process noise std. dev.	0.2 deg/s
velocity process noise std. dev.	0.0 m/s
initial vehicle x position uncertainty std. dev.	0.0 m
initial vehicle y position uncertainty std. dev.	0.0 m
initial heading position uncertainty std. dev.	0.0 deg

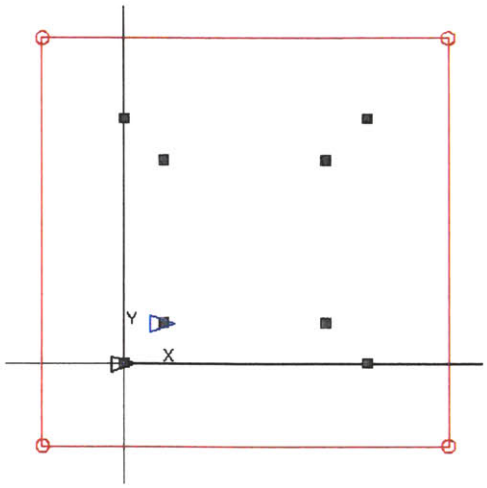
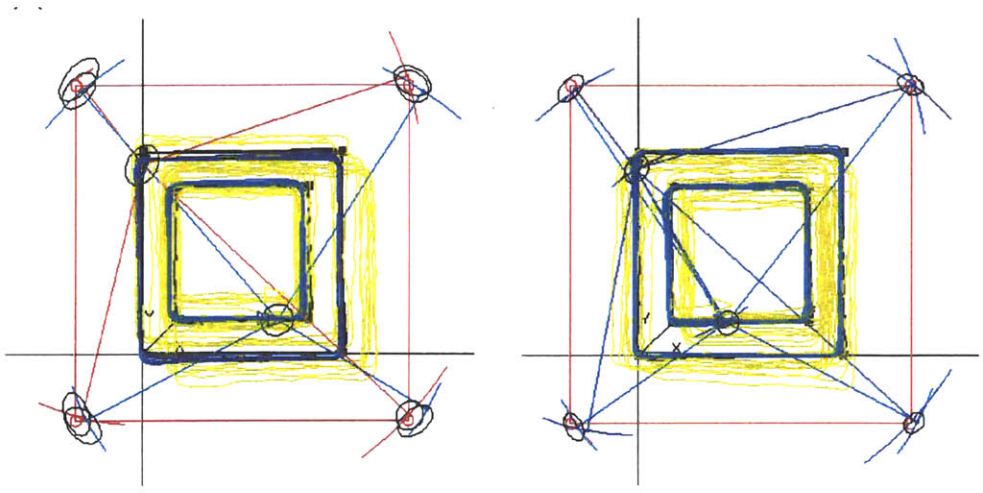


Figure 5-30: 2-D CCML scenario #2 : vehicle starting position



Single vehicle CML **Collaborative CML**

Figure 5-31: 2-D CCML scenario #2 : final position estimates

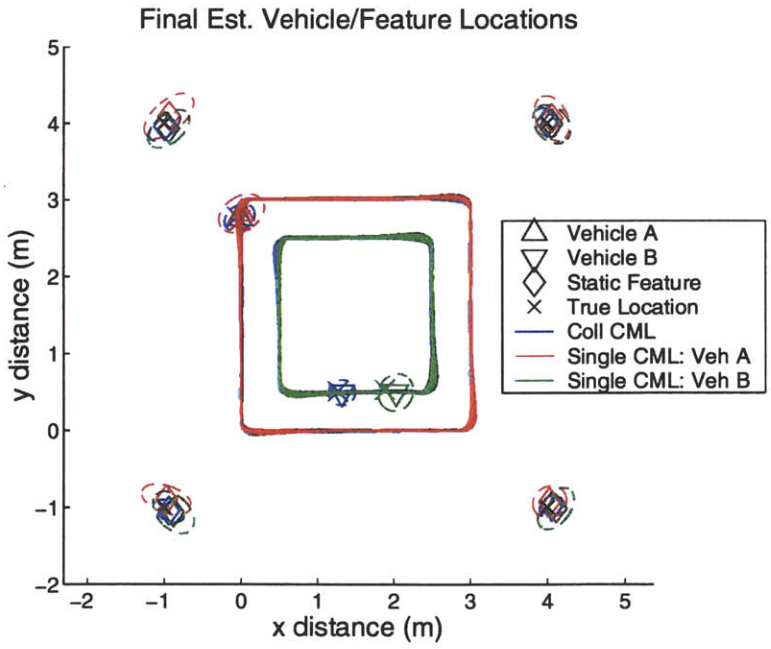


Figure 5-32: 2-D CCML scenario #2 : position estimate comparison

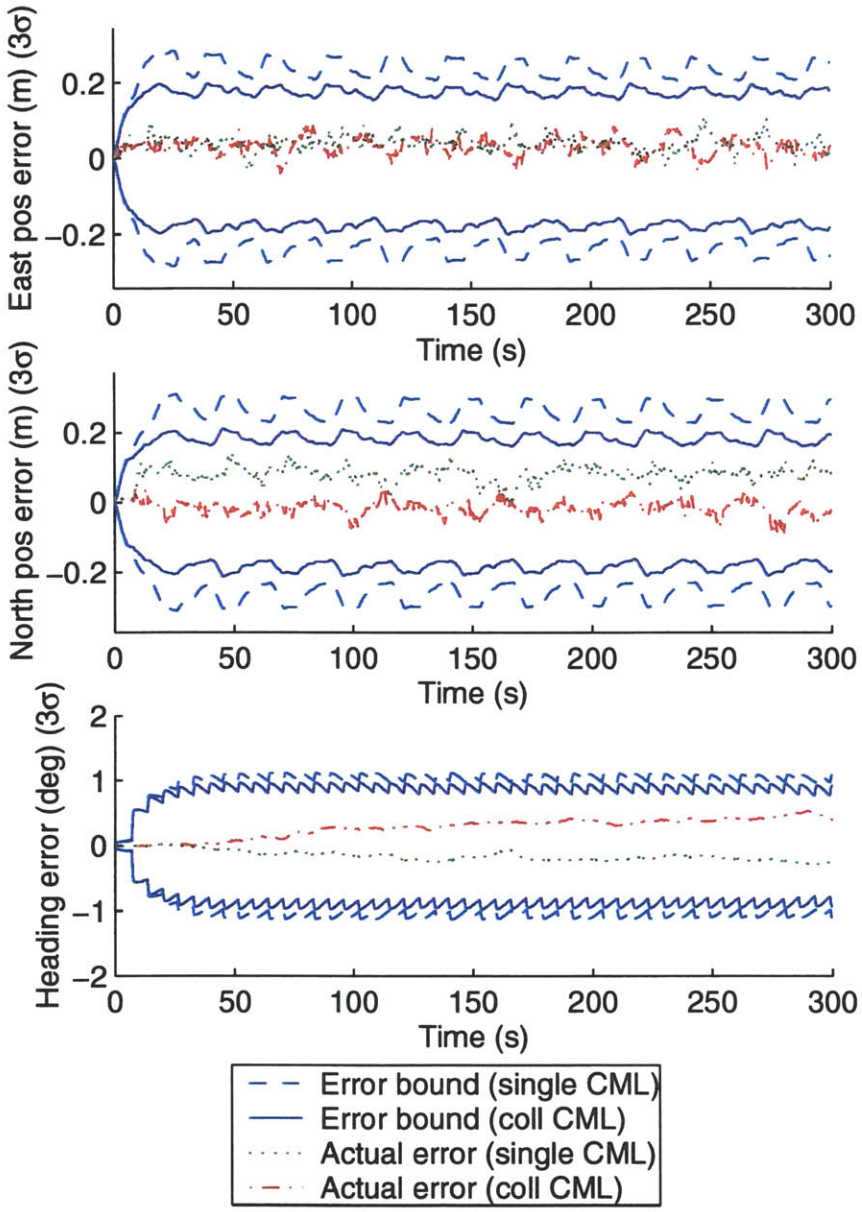


Figure 5-33: 2-D CCML scenario #2 : vehicle A error comparison

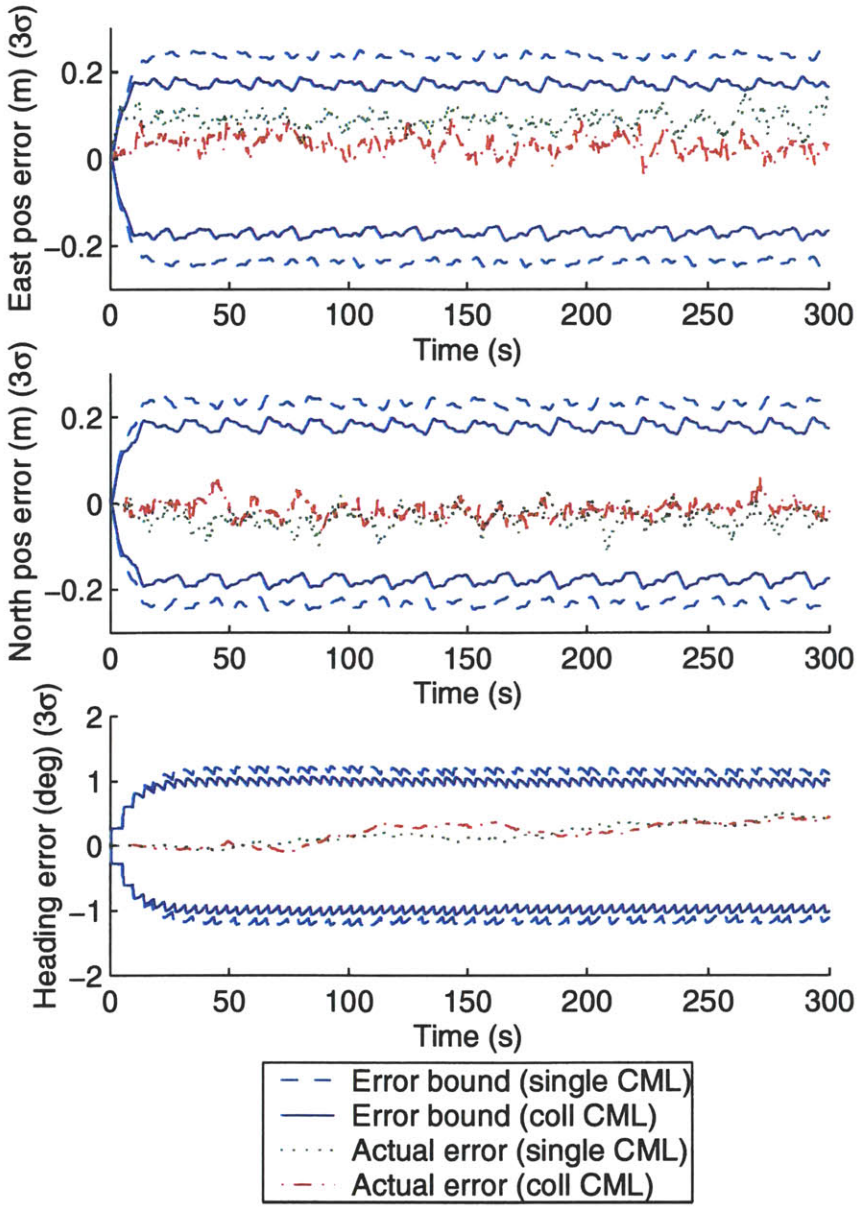


Figure 5-34: 2-D CCML scenario #2 : vehicle B error comparison

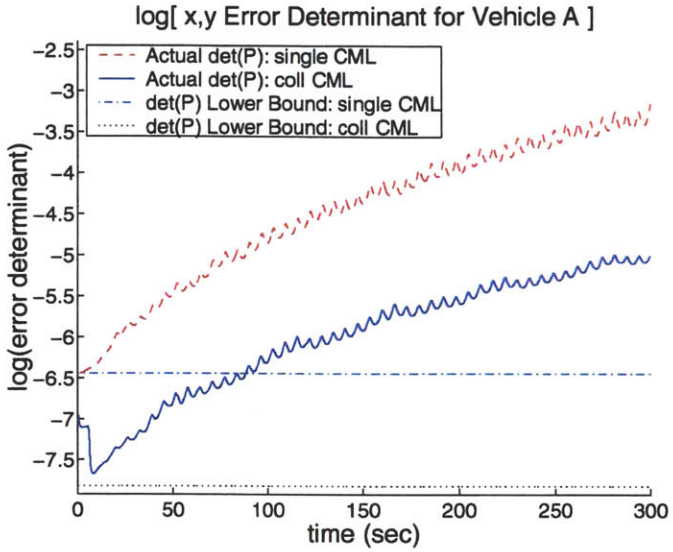


Figure 5-35: 2-D CCML scenario #2 : vehicle A error determinant

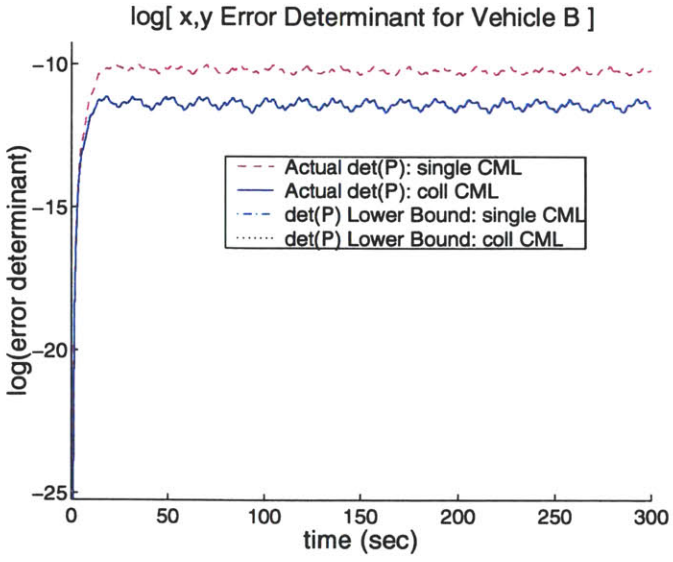
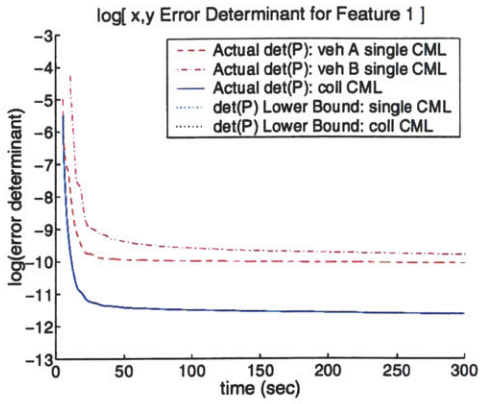
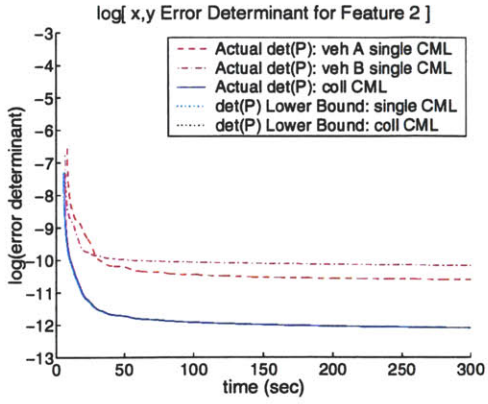


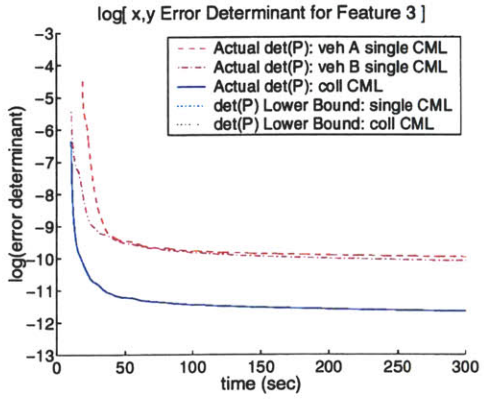
Figure 5-36: 2-D CL scenario #2 : vehicle B error determinant



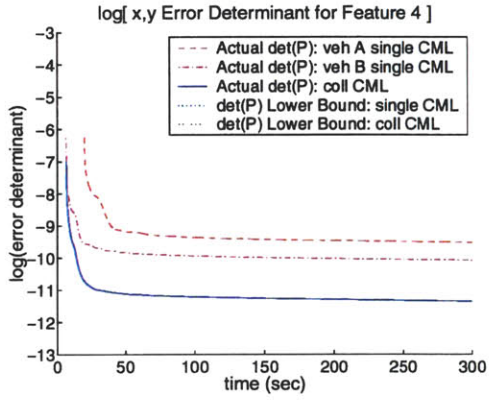
Feature 1 error versus time



Feature 2 error versus time



Feature 3 error versus time



Feature 4 error versus time

Figure 5-37: 2-D CCML scenario #2 : feature error determinant comparison

5.3.3 2-D CCML scenario #3

This scenario best simulates actual vehicle implementations, as both initial position uncertainty and process noise are present. Tables 5.5 and 5.8 summarize the parameters used for this scenario. Figures 5-41 and 5-42 demonstrate that vehicle position uncertainty stabilizes above the theoretical lower bound in the presence of process noise.

Table 5.8: 2-D CCML simulation scenario #3 parameters

x position process noise std. dev.	0.2 m/s
y position process noise std. dev.	0.25 m/s
heading process noise std. dev.	0.2 deg/s
velocity process noise std. dev.	0.0 m/s
initial vehicle x position uncertainty std. dev.	0.075 m
initial vehicle y position uncertainty std. dev.	0.075 m
initial heading position uncertainty std. dev.	0.0 deg

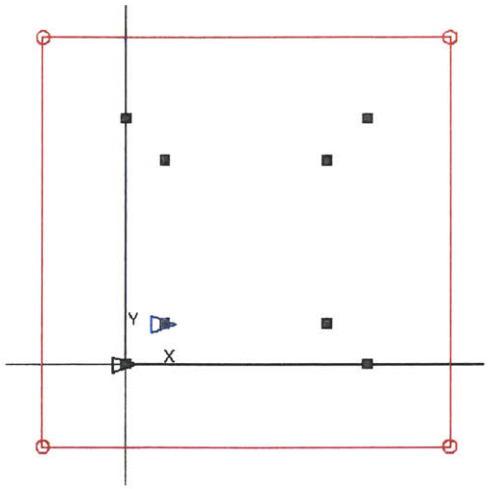
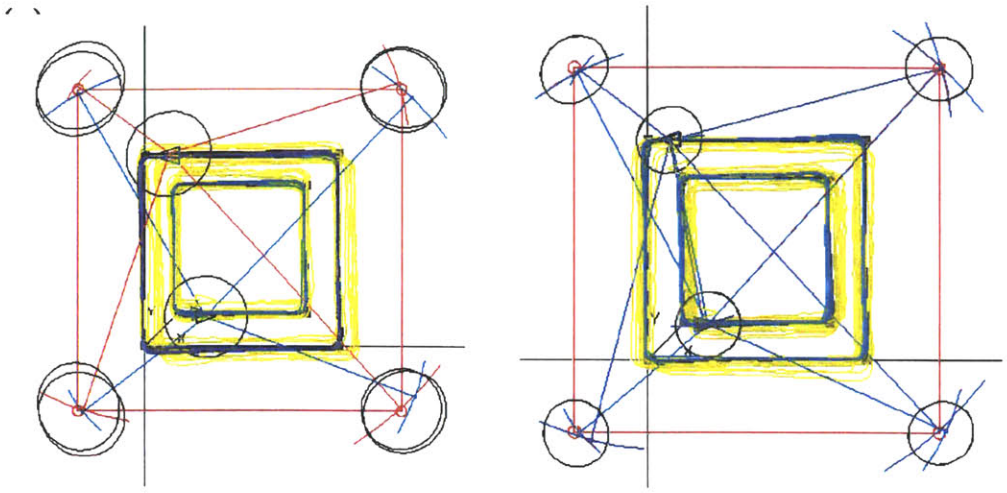


Figure 5-38: 2-D CCML scenario #3 : vehicle starting position



Single vehicle CML **Collaborative CML**

Figure 5-39: 2-D CCML scenario #3 : final position estimates

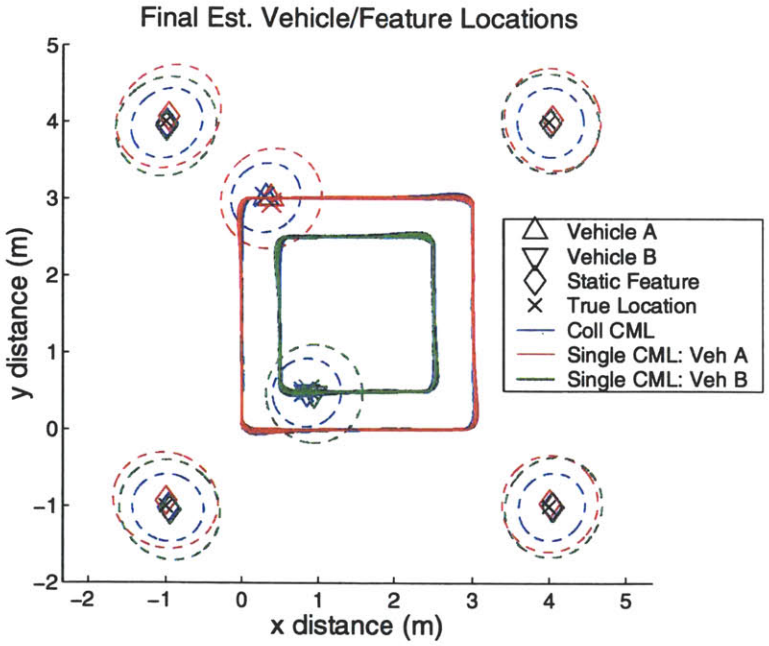


Figure 5-40: 2-D CCML scenario #3 : position estimate comparison

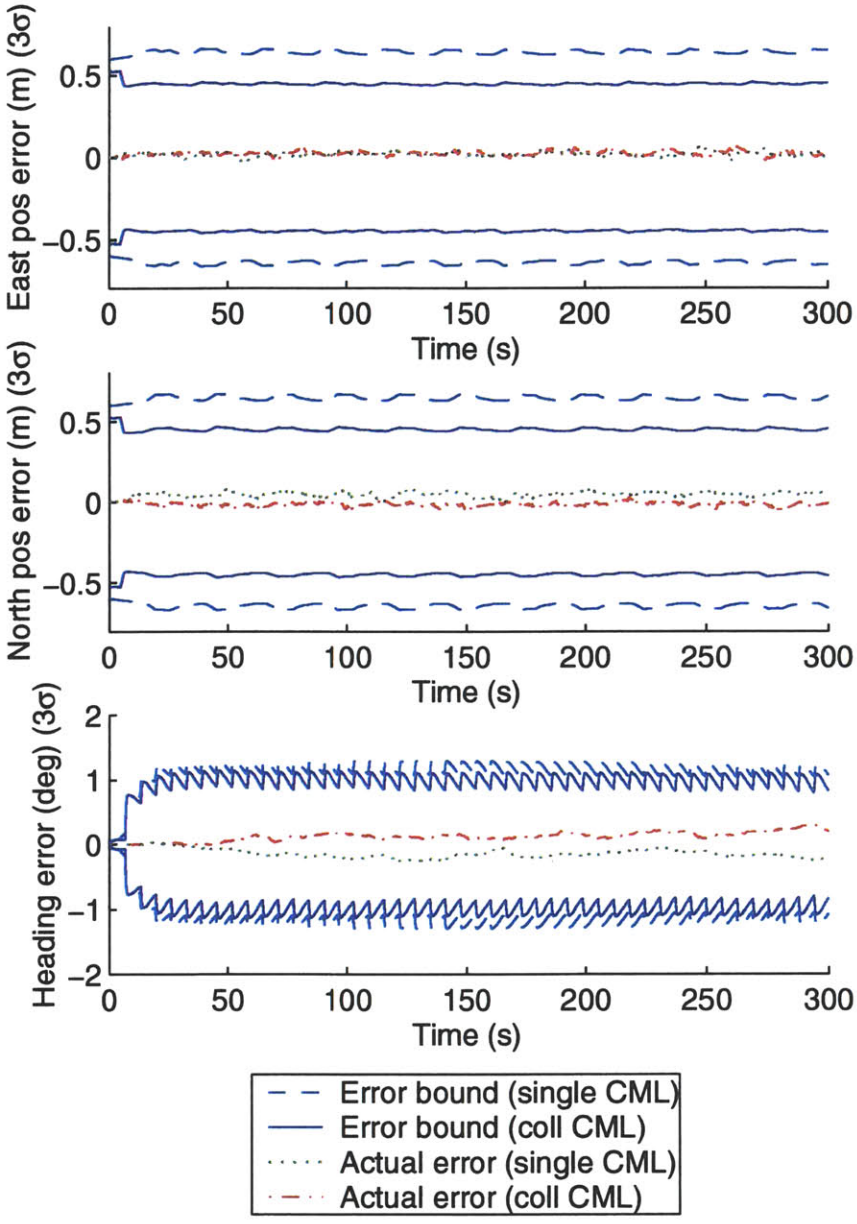


Figure 5-41: 2-D CCML scenario #3 : vehicle A error comparison

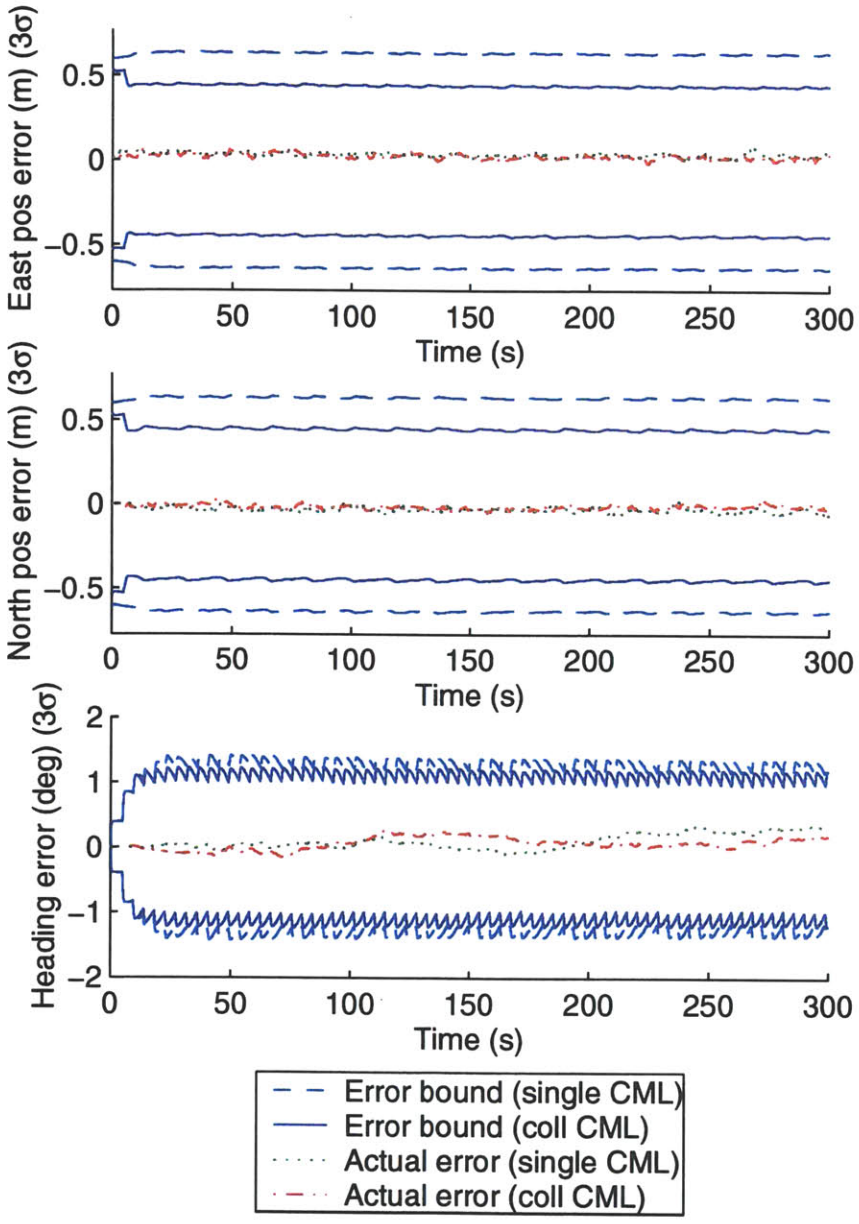


Figure 5-42: 2-D CCML scenario #3 : vehicle B error comparison

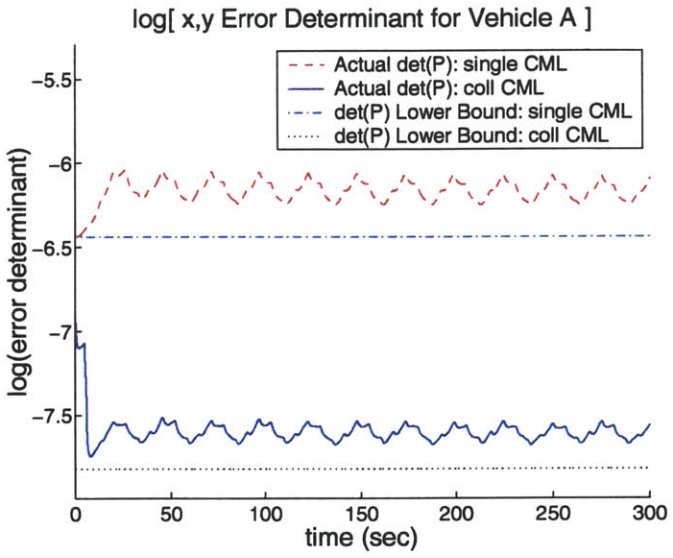


Figure 5-43: 2-D CCML scenario #3 : vehicle A error determinant

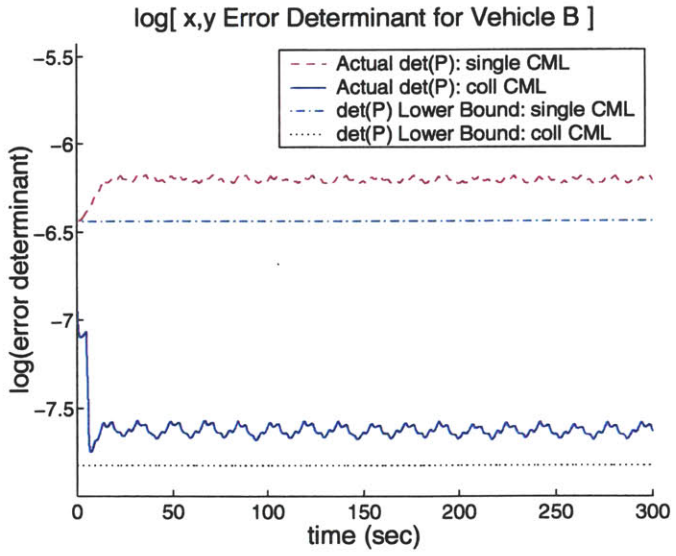
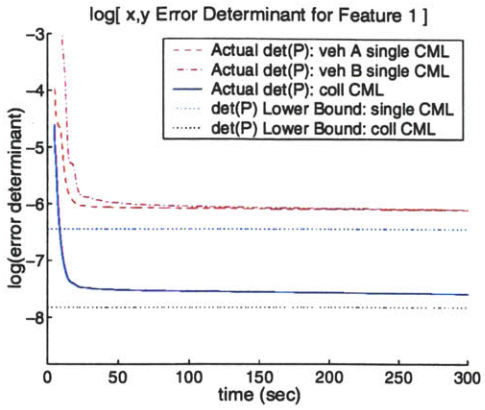
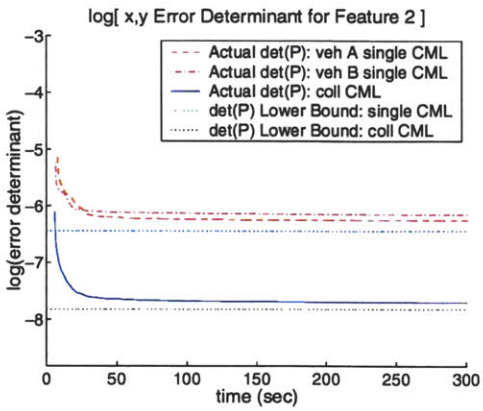


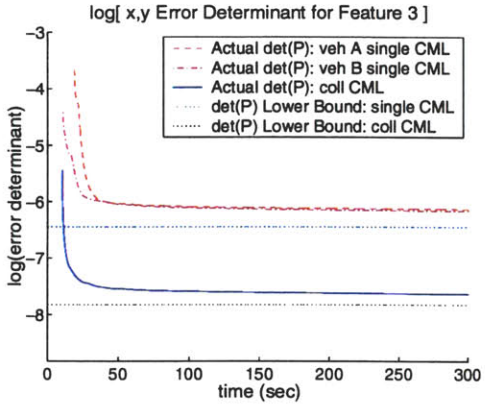
Figure 5-44: 2-D CL scenario #3 : vehicle B error determinant



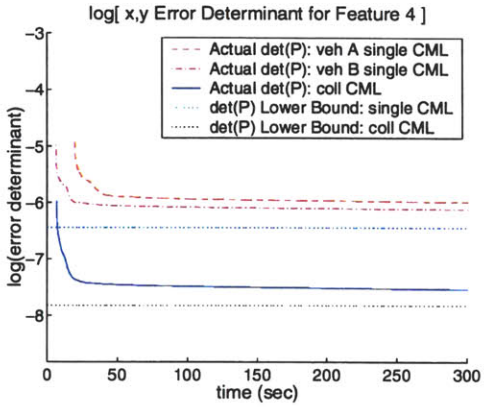
Feature 1 error versus time



Feature 2 error versus time



Feature 3 error versus time



Feature 4 error versus time

Figure 5-45: 2-D CCML scenario #3 : feature error determinant comparison

5.4 Summary

This chapter presented simulation results for vehicle navigation collaboration with and without the presence of static features in the environment. Collaborative localization provides a substantial performance improvement over dead reckoning. In the presence of process noise, collaborative localization still results in unbounded linear error growth, but at a slower growth rate than dead reckoning. The Kalman filter based structure used for collaborative localization serves as an intermediary step to collaborative CML. The collaborative CML algorithm presented in Section 3.3 in simulation proves to be superior to single vehicle CML.

Chapter 6

Conclusions and Future Research

This chapter summarizes the contributions of this thesis and presents suggestions for further collaborative CML research.

6.1 Thesis contributions

This thesis enabled multiple vehicles to collaboratively map an environment more quickly and robustly than a single vehicle. Current algorithms for this concurrent mapping and localization (CML) problem have been implemented for single vehicles, but do not account for extra positional information available when multiple vehicles operate simultaneously. This thesis presented an innovative technique for combining sensor readings for multiple autonomous vehicles, enabling them to perform collaborative CML. In addition, a lower algorithmic performance bound has been proven, enabling determination of the number of cooperating vehicles required to accomplish a given task. This quantifies intuitive performance benefits that result from using more than one vehicle for mapping and navigation, which were validated in simulation.

6.2 Future research

Any successful collaborative CML algorithm has to be based on a successful single-vehicle CML algorithm. Single-vehicle CML issues such as correct association of sensor measurements and map scalability for larger environments remain significant constraints on any CML implementations. That said, the field of cooperation in robotics is vast and largely unexplored.

This thesis quantifies intuitive performance benefits that result from using more than one vehicle for mapping and navigation. Even within this collaborative CML framework there is a substantial amount of analysis left unperformed. Suggested analysis directions include larger numbers of collaborating vehicles as well as heterogeneous vehicles. Another interesting topic is the required rate of information flow if collaborators mutually observe static features, but not one another. Addressing the challenges of rendezvous and limited communication, as discussed in Section 3.1, are logical next steps in producing a robust, effective collaborative CML implementation.

Bibliography

[1] R. Arkin and T. Balch. Cooperative multiagent robotic systems. In *Artificial Intelligence and Mobile Robots*, pages 277–296. MIT Press, 1998.

[2] D. Atwoood, J. Leonard, J. Bellingham, and B. Moran. An acoustic navigation system for multiple vehicles. In *Proceedings, International Symposium on Unmanned Untethered Submersible Technology*, pages 202–208, 1995.

[3] T. Balch and L. Parker. Guest editorial. *Autonomous Robots Special Issue on Heterogeneous Multi-Robot Systems*, 2000.

[4] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association.* Academic Press, 1988.

[5] Y. Bar-Shalom and X. Li. *Estimation and tracking: Principles, techniques and software.* Artech House, 1993.

[6] M. Betke and L. Gurvits. Mobile robot localization using landmarks. *IEEE Transactions on Robotics and Automation*, 13(2):251–263, 1997.

[7] A. Billard, A. Ijspeert, and A. Martinoli. A multi-robot system for adaptive exploration of a fast changing environment: Probabilistic modelling and experimental study. *Connection Science*, 11(3/4):357–377, 2000.

[8] J. Borenstein, H. Everett, and L. Feng. *Navigating mobile robots: Sensors and techniques.* A. K. Peters, Ltd., Wellesley, MA, 1996.

[9] R. Brooks. Intelligence without reason. *MIT AI Lab Memo 1293*, 1991.

[10] Y. Cao, A. Fukunaga, A. Kahng, and F. Meng. Cooperative mobile robotics: Antecedents and directions. In *Proceedings, IEEE/TSJ International Conference on Intelligent Robots and Systems*, 1995.

[11] J. Castellanos, J. Tardos, and G. Schmidt. Building a global map of the environment of a mobile robot: The importance of correlations. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 1053–1059, 1997.

[12] R. Chatila and J. Laumond. Position referencing and consistent world modeling for mobile robots. In *Proceedings, IEEE International Conference on Robotics and Automation.* IEEE, 1985.

[13] W. Cohen. Adaptive mapping and navigation by teams of simple robots. *Robotics and Autonomous Systems*, 18:411–434, 1996.

[14] M. Deans and M. Hebert. Experimental comparison of techniques for localization and mapping using a bearing only sensor. In *Proceedings of the Seventh International Conference on Experimental Robotics*, Hawaii, Dec 2000.

[15] G. Dedeoglu and G. Sukhatme. Landmark-based matching algorithm for cooperative mapping by autonomous robots. In *Proceedings of the Fifth International Symposium on Distributed Autonomous Robotic Systems*, 2000.

[16] X. Deng and A. Mirzaian. Competitive robot mapping with homogeneous markers. *IEEE Transactions on Robotics and Automation*, 12:532–542, Aug 1996.

[17] M. Dissanayake, P. Newman, H. Durrant-Whyte, S. Clark, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. In *In Sixth International Symposium on Experimental Robotics*, 1999.

[18] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes. A taxonomy for multi-agent robotics. *Autonomous Robots*, pages 375–397, 1996.

[19] H. Feder. *Simultaneous stochastic mapping and localization.* Ph.D. thesis, MIT Dept. of Mechanical Engineering, 1999.

[20] D. Fox, W. Burgard, H. Kruppa, and S. Thrun. A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots*, 8(3):325–344, June 2000.

[21] D. Fox, W. Burgard, S. Thrun, and A. Cremers. Position estimation for mobile robots in dynamic environments. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, Madison, WI, 1998.

[22] A. Gelb. *Applied Optimal Estimation.* MIT Press, 1973.

[23] J. Golden. Terrain contour matching (TERCOM): a cruise missile guidance aid. *Image Processing for Missile Guidance*, 238:10–18, 1980.

[24] R. Grabowski, L. Navarro-Serment, C. Paredis, and P. Khosla. Heterogeneous teams of modular robots for mapping and exploration. *Autonomous Robots*, 8(3):293–308, June 2000.

[25] K. Ishioka, K. Hiraki, and Y. Anzai. Coorperative [sic] map generation by heterogenous autonomous mobile robots. In *Proceedings, Workshop on Dynamically Interacting Robots*, pages 58–67, 1993.

[26] D. Jung, J. Heinzmann, and A. Zelinsky. Range and pose estimation for visual servoing on a mobile robotic target. In *Proceedings, IEEE International Conference on Robotics and Automation*, 1998.

[27] B. Kuipers and Y. Byun. A qualitative approach to robot exploration and map learning. In *Proceedings, IEEE Workshop on Spatial Reasoning and Mulit-Sensor Fusion*, pages 390–404, 1987.

[28] B. Kuipers and Y. Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and Autonomous Systems*, 8:47–63, 1991.

[29] R. Kurazume and S. Nagata. Cooperative positioning with multiple robots. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 1250–1257, 1994.

[30] R. Kurazume and S. Nagata. Study on cooperative positioning system - optimum moving strategies for CPSII. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 2896–2903, 1998.

[31] J. Leonard and H. Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *Proceedings, IEEE International Workshop on Intelligent Robots and Systems*, pages 1442–1447, Osaka, Japan, 1991.

[32] M. Mataric. Coordination and learning in multi-robot systems. *IEEE Intelligent Systems*, pages 6–8, 1998.

[33] P. Maybeck. *Stochastic models, estimation and control*, volume 1. Academic Press, New York, 1979.

[34] H. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, pages 61–74, 1988.

[35] P. Moutarlier and R. Chatila. Stochastic multisensory data fusion for mobile robot location and environment modeling. In *In Fifth International Symposium on Robotics Research*, Tokyo, Japan, 1989.

[36] L. Navaro-Serment, C. Paredis, and P. Khosla. A beacon system for the localization of distributed robotic teams. In *Proceedings of International Conference on Field and Service Robotics*, Aug 1999.

[37] P. Newman. *On the structure and solution of the simultaneous localization and mapping problem.* Ph.D. thesis, University of Sydney, March 1999.

[38] L. Parker. Current state of the art in distributed autonomous mobile robotics. *Distributed Autonomous Robotic Systems 4*, 4:3–12, 2000.

[39] N. Rao, V. Protopopescu, and V. Manickam. Cooperative terrain model acquisition by a team of two or three point-robots. In *Proceedings of International Conference in Robotics and Automation*, volume 2, pages 1427–1433, 1996.

[40] I. Rekleitis, G. Dudek, and E. Milios. Multi-robot collaboration for robust exploration. In *Proceedings of International Conference in Robotics and Automation*, pages 3164–3169, San Francisco, CA, Apr 2000.

[41] S. Roumeliotis and G. Bekey. Distributed multi-robot localization. In *Proceedings of the Fifth International Symposium on Distributed Autonomous Robotic Systems*, pages 241–250, Knoxville, TN, 2000.

[42] S. Roumeliotis and G. Bekey. Synergetic localization for groups of mobile robots. In *Proceedings of IEEE Conference on Decision and Control*, Sydney, Australia, Dec 2000.

[43] N. Roy and G. Dudek. Collaborative robot exploration and rendezvous: Algorithms, performance bounds and observations. *Autonomous Robots*, 8(3), June 2000.

[44] J. Salido-Tercero, C. Paredis, and P. Khosla. Continuous probabilistic mapping by autonomous robots. In *Proceedings of the 1999 International Symposium on Experimental Robotics*, March 1999.

[45] H. Singh, J. Catipovic, R. Eastwood, L. Freitag, H. Henriksen, F. Hover, D. Yoerger, J. Bellingham, and B. Moran. An integrated approach to multiple AUV communications, navigation, and docking. In *In IEEE Oceans*, pages 59–64, 1987.

[46] C. Smith. *Integrating mapping and navigation.* Ph.D. thesis, MIT Department of Ocean Engineering, 1998.

[47] R. Smith, M. Self, and P. Cheeseman. A stochastic map for uncertain spatial relationships. In *Fourth International Symposium on Robotics Research.* MIT Press, 1987.

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

[49] J. Spletzer, A. Das, R. Fierro, C. Taylor, V. Kumar, and J. Ostrowski. Cooperative localization and control for multi-robot manipulation. In *submitted to International Conference on Intelligent Robots and Systems*, Hawaii, Dec 2001. URL: http://www.cis.upenn.edu/mars/sdftko-iros01.pdf [cited 12 Apr 2001].

[50] K. Sty. Using situated communication in distributed autonomous mobile robots. In *Seventh Scandinavian Conference on Artificial Intelligence (SCAI01)*, 2001.

[51] H. Tagare, D. McDermott, and H. Xiao. Visual place recognition for autonomous robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2530–2535, 1998.

[52] W. Thompson, T. Henderson, T. Colvin, B. Dick, and C. Valiquette. Vision-based localization. In *Proceedings of the DARPA Image Understanding Workshop*, pages 491–498, 1993.

[53] S. Thrun, A. Bucken, W. Burgard, D. Fox, T. Frohlinghaus, D. Hennig, T. Hofmann, M. Krell, and T. Schimdt. Map learning and high-speed navigation in RHINO. *MIT/AAAI Press*, 1998.

[54] S. Thrun, W. Burgard, and D. Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31:29–53, 1998.

[55] R. Vaughan, K. Stoy, G. Sukhatme, and M. Mataric. Whistling in the dark: cooperative trail following in uncertain localization space. In *Proceedings of International Conference on Autonomous Agents*, Barcelona, Spain, 2000.

[56] R. Vaughan, G. Sukhatme, J. Mesa-Martinez, and J. Montgomery. Fly spy: lightweight localization and target tracking for cooperating ground and air robots. In *Proceedings of the Fifth International Symposium on Distributed Autonomous Robotic Systems*, 2000.

[57] B. Yamauchi, A. Schultz, and W. Adams. Integrating exploration and localization for mobile robots. *International Journal of Robotics Research*, 7(2):217–230, 1999.