Correlated-Data Fusion and Cooperative Aiding in GNSS-Stressed or Denied Environments

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Hamid Mokhtarzadeh

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

Demoz Gebre-Egziabher

September, 2014

© Hamid Mokhtarzadeh 2014
ALL RIGHTS RESERVED

Acknowledgements

I start by thanking the Creator, to whom all thanks is due. It is difficult to acknowledge all the direct and indirect support that went into these formative years. I'm most grateful to my adviser, Professor Demoz Gebre-Egziabher. I admire and learned a great deal from your teaching philosophy, the standards to which you hold your work, and the value you place in the role and responsibility of an educator. Although the content of our discussions have evolved over the last four years, you've always projected the same level of patience and respect in these meetings. This was true in my first year as a complete novice as well as four years onwards.

I'm also grateful to Professor Gary Balas, both for my two years in his research lab during my masters degree research, and for his continued direct and indirect support. Your passion and energy translated into a lively lab environment. Among others, Arnar Hjartarson, Paw Yew Chai, Rohit Pandita, and Abhijit Chakraborty were excellent mentors during those years, for which I'm grateful. During our student-led systems seminars, the feedback from Professor Balas were invaluable as they brought together not only technical rigor, but also lessons in management and business aptitude. The ability to both formulate such rich feedback on a live presentation and simultaneously edit papers is a skill of Professor Balas I will not forget!

I'm grateful to Professor Yiyuan Zhao for serving on my preliminary oral examination committee and to both Professors Peter Seiler and Mihailo Jovanovic for serving on my final examination committee. Despite the work load that goes with the high caliber research and teaching that they do, both have always been friendly and very approachable. As I am finishing the compilation of this thesis in LaTeX, I recall my first attempt to use LaTeX more than five years ago. Professor Seiler guided me on the process, supplied a sample set of source files from his own paper, and let me borrow his LaTeX reference book. This exemplifies the type of openness and support for which I'm thankful.

I'm grateful to the researchers of the UAV Lab as the Chapter 6 results in this work are based on flight data collected and archived by their team. Past and present labmates made for a lively work environment. Saber Taghvaeeyan was a neighbor, friend, and someone I could always count on. Among others I'd like to mention David Escobar, Zhefeng Li, Chen-Chi

Chu, and more recently Trevor Layh and Jordan Larson. Our discussions about research and otherwise were refreshing. I'm particularly grateful for my overlapping time with Adhika Lie. What started as a collaboration on a project in 2010 has evolved into a valuable friendship. The discussions, hardware and software hacks, and the numerous ideas we've bounced around always added freshness to the graduate school experience. I hope you've found our time together equally beneficial.

I'd also like to recognize the support from other faculty and staff from our department and University as a whole. Outside our University, the mentorship and experience from having spent time with Suneel Sheikh at ASTER Labs and Jonathan Nichols at Google both added great practical value to my training.

I close with an acknowledgment of the support of my wife Atefeh, parents Alireza and Nahid, aunt Zohreh, my brother Ali, and many other family members. Your collective support has been the air I've breathed during these years and beyond. To put into words an acknowledgment for your role in this work would be an attempt in vain. Hence, I close by expressing my heartfelt appreciation to each of you.

Abstract

A growing number of applications require continuous and reliable estimates of position, velocity, and orientation. Price requirements alone disqualify most traditional navigation or tactical-grade sensors and thus navigation systems based on automotive or consumer-grade sensors aided by Global Navigation Satellite Systems (GNSS), like the Global Positioning System (GPS), have gained popularity. The heavy reliance on GPS in these navigation systems is a point of concern and has created interest in alternative or back-up navigation systems to enable robust navigation through GPS-denied or stressed environments.

This work takes advantage of current trends for increased sensing capabilities coupled with multilayer connectivity to propose a cooperative navigation-based aiding system as a means to limit dead reckoning error growth in the absence of absolute measurements like GPS. Each vehicle carries a dead reckoning navigation system which is aided by relative measurements, like range, to neighboring vehicles together with information sharing. Detailed architectures and concepts of operation are described for three specific applications: commercial aviation, Unmanned Aerial Vehicles (UAVs), and automotive applications.

Both centralized and decentralized implementations of cooperative navigation-based aiding systems are described. The centralized system is based on a single Extended Kalman Filter (EKF). A decentralized implementation suited for applications with very limited communication bandwidth is discussed in detail. The presence of unknown correlation between the *a priori* state and *measurement* errors makes the standard Kalman filter unsuitable. Two existing estimators for handling this unknown correlation are Covariance Intersection (CI) and Bounded Covariance Inflation (BCInf) filters. A CI-based decentralized estimator suitable for decentralized cooperative navigation implementation is proposed. A unified derivation is presented for the Kalman filter, CI filter, and BCInf filter measurement update equations. Furthermore, characteristics important to the proper implementation of CI and BCInf in practice are discussed. A new covariance normalization step is proposed as necessary to properly apply CI or BCInf.

Lastly, both centralized and decentralized implementations of cooperative aiding are analyzed and evaluated using experimental data in the three applications.

In the commercial aviation study aircraft are simulated to use their Automatic Dependent Surveillance - Broadcast (ADS-B) and Traffic Collision Avoidance System (TCAS) systems to cooperatively aid their on board INS during a 60 min GPS outage in the national airspace. An availability study of cooperative navigation as proposed in this work around representative United States airports is performed. Availabilities between 70-100% were common at major airports like LGA and MSP in a 30 nmi radius around the airport during morning to evening hours.

A GPS-denied navigation system for small UAVs based on cooperative information sharing is described. Experimentally collected flight data from 7 small UAV flights are *played-back* to evaluate the performance of the navigation system. The results show that the most effective of the architectures can lead to 5+ minutes of navigation without GPS maintaining position errors less than 200 m $(1-\sigma)$

The automotive case study considers 15 minutes of automotive traffic ($2,000+$ vehicles) driving through a half-mile stretch of highway without access to GPS. Automotive radar coupled with Dedicated Short Range Communication (DSRC) protocol are used to implement cooperative aiding to a low-cost 2-D INS on board each vehicle. The centralized system achieves an order of magnitude reduction in uncertainty by aggressively aiding the INS on board each vehicle. The proposed CI-based decentralized estimator is demonstrated to be conservative and maintain consistency. A quantitative analysis of bandwidth requirements shows that the proposed decentralized estimator falls comfortably within modern connectivity capabilities. A naive implementation of the high-performance centralized estimator is also achievable, but it was demonstrated to be burdensome, nearing the bandwidth limits.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

A growing number of applications require continuous and reliable estimates of position, velocity, and orientation. Collectively, these quantities are formally referred to as the navigation state vector, however, the successful operation of other common subsystems like guidance and control also depend heavily on this quantity [1]. The choice of a suitable navigation system involves a trade-off between accuracy and reliability, on one hand, and complexity, size, weight, and power on the other. In navigation systems based on commercially available sensors the price requirements alone disqualify most traditional navigation-grade and tactical-grade sensors. It is no surprise, therefore, that navigation systems based on automotive-grade and consumer-electronics sensors aided by Global Navigation Satellite System (GNSS) position and velocity measurements [5], generally referred to as GNSS-aided Inertial Navigation Systems (INS), are gaining popularity. The most popular of GNSSs being Global Positioning System (GPS), which is owned and operated by the United State government.

The GNSS-aided INS enables a drift free estimate of position, velocity, and attitude whereby the high-rate dead reckoning of the INS is continuously aided by low-rate GNSS position and velocity measurements. This works well so long as any loss of GNSS-signal does not exceed a few tens of seconds. The challenge, however, is that automotive/consumer-grade sensors are not suitable for extended operation in free-inertial coasting or stand-alone inertial mode. Without external aiding, the position drift of a dead reckoning system based on automotive/consumer-grade sensors is greater than 50 meters (150 ft) per minute [1], quickly rendering the navigation system unreliable for many applications. And due to the tight inter-dependence between navigation, guidance, and control functions on modern dynamical systems, like Unmanned Aerial Vehicles (UAVs) or automated cars, the safety of the entire vehicle could be compromised. Recorded instances of UAV catastrophic failure simply due to compromised navigation systems are illustrative of this inter-dependence [6]. The heavy reliance on GNSS in these navigation systems is a point of concern.

As a result there is growing interest in alternative or back-up navigation systems to enable robust navigation through GNSS-denied or stressed environments. Most attention is in finding alternative aiding measurements to again arrest the error-growth of the on-board dead reckoning system. Conceptually, any source with a known location, and from which relative measurement can be obtained, can serve as an aiding measurement to arrest the error-growth of the dead reckoning system. Examples of current candidates include camera/vision using georeferenced maps, cell-phone signals, or measurements based on other signals-of-opportunity. This work proposes aiding measurements based on networked vehicles or cooperative navigation.

1.1 Cooperative Navigation

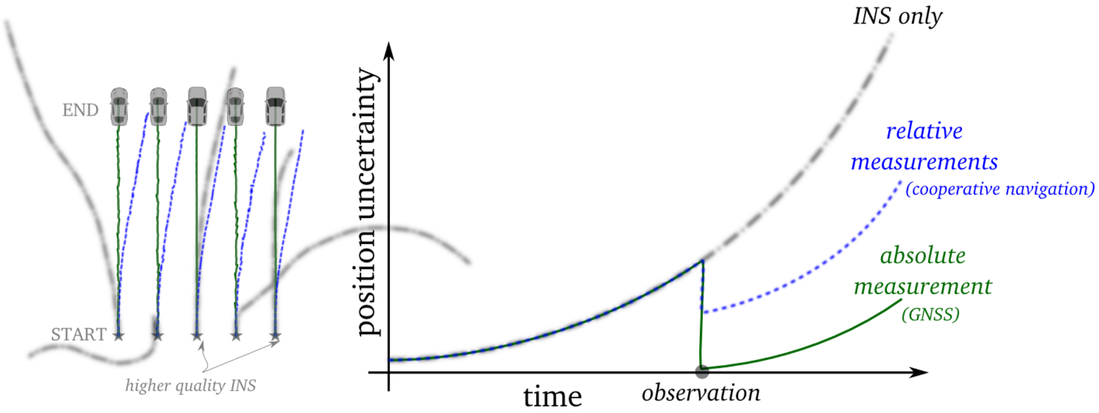


Figure 1.1: Operational benefits of using cooperative navigation to aid INS in GNSS-stressed environments. Trajectories are simulation-based; plot is qualitative.

Cooperative navigation describes a system whereby two or more vehicles, objects, or persons, collaborate to calculate a navigation solution with higher quality than would be attainable had each of them generated a solution independently. The community of users is made up of all members equipped with the sensing and communication capability to enable cooperative navigation. No spatial or temporal limit is implied by the term *community*, hence we are not restricting attention to formation-type applications. Three potential communities of users are commercial airliners, automotive traffic, or urban smart-phone users. It also may be that the community is composed of multiple user classes. For example, a cooperative navigation system developed for military applications may be composed of ground, air, and sea vehicles, all part of the same community.

In general, cooperative navigation requires users be equipped with relative-measurement sensors, like range and bearing sensors, as well as the capability to share information between the users. The information exchanged can include (all or some of) the navigation state vector estimate of the vehicle, an associated quality metric, and available sensor measurements. Together with the received information from neighboring collaborators, the relative measurements are used as aiding measurements to the on-board dead reckoning system. For example, the range between a user and a neighboring collaborator together with the estimated position of the collaborator can be used to compute corrections to a user's own position estimate. In this manner radio transmitters on participating vehicles are used opportunistically as navigation beacons, arresting the errors of the dead reckoning system and hereby mitigating the effect of the extended GNSS-outage.

Figure 1.1 schematically depicts the benefit that can be realized if the on-board dead reckoning system is aided by cooperative navigation based on inter-vehicle or relative range measurements. The left schematic shows the ground track for 5 vehicles traveling directly North under three operating conditions - INS aided by absolute position measurements from GNSS; INS aided by relative range measurements or cooperative navigation; and unaided or *free* INS. There are both high and low quality INSs in the community and the ground tracks show qualitatively the resulting position errors. A snapshot of position uncertainty as a function of time is plotted in the right schematic. Although the accuracy afforded when GNSS is available is unparalleled, it is clear to see that the cooperative navigation implementation of inter-vehicular range measurements and exchanging of position estimates among users dramatically improves the situation by arresting the rate of error growth on the dead reckoning system.

One of the key challenges in cooperative navigation is designing suitable estimators for fusing the information exchanged by members of the community. A chosen estimation architecture for cooperative navigation applications depend largely on the availability of both computational resources and communication bandwidth. When there are no communication and computational constraints, a single centralized estimator, possibly based on the Extended Kalman Filter (EKF), can be used. This centralized estimator would model all the vehicles in the community and each vehicle would broadcast on board sensor measurements to it. The centralized estimator would then process the information for the entire community to generate the best current state estimate of each vehicle and broadcast that solution to individual vehicles.

Where computation and communication constraints are present, an alternative design based on decentralized estimators on board each vehicle would be suitable. This implies that each member in the community is interested only in correcting its own state errors. This flexibility, however, comes at a cost. The flexibility and reduced computational load of decentralized estimators are largely achieved by dropping states necessary to account for inter-vehicle error correlations. Both the relative sensor measurement errors and the uncertain position estimates received from a collaborator are treated as measurement error. The standard Kalman Filter assumption of independence between the *a priori* state and the measurement error is now violated, which can lead to filter divergence [7]. If left unrestricted, cooperative navigation can introduce error loops which can amplify positioning errors [8]. Therefore, the selection of a suitable estimator for the decentralized cooperative navigation application is non-trivial. Estimation and fusion algorithms which are effective for single users or centralized filters, like the standard EKF, are problematic for decentralized cooperative applications. Alternative fusion algorithms, specifically in a decentralized implementation, are necessary to ensure proper fusion of the aiding measurements and manage the *double counting* or data fusion with unknown correlations problem.

1.2 Correlated Data Fusion

Handling correlated quantities is at the heart of many estimation problems in aerospace guidance, navigation, and control. It is the proper knowledge of correlation between error-states that enables indirect observations to provide stochastic observability to a large state vector. A well studied example of this is the aided INS. Low-rate aiding from GNSS-based position measurements are commonly sufficient to provide the stochastic observability required to estimate other error-states like gyro and accelerometer biases. The assumed dynamic model establishes the relationship between states, and therefore, inter-state correlations are computed and maintained. Hence, when a measurement for a single state is received, all correlated states stand to benefit.

There are, however, applications where information about correlation is incomplete. For example:

1. If, contrary to Kalman Filter assumptions, the measurement and or process noise are not *white*.

2. Dynamical model is uncertain, introducing unmodeled correlations.

3. Linearization errors (e.g. for covariance propagation) introduce unknown correlations.

Traditional solutions to this problem included state-augmentation and inflating process and or measurement noise [9]. State augmentation involves adding a new state to the system state-vector to effectively model the correlated quantity. For example, if the measurement is corrupted by both additive-noise and a time-varying bias, it is common to model the time-varying bias as a Markov process and augment it to the state vector. While this works very well for some applications, it can become prohibitive when the number of potentially correlated quantities is large, leading to a state vector too large and unwieldy for practical implementation.

The second solution which involves inflating the process noise is what is commonly known as *tuning the filter*. It involves changing the assumed error statistics to account for neglected/unknown correlations. When the neglected quantities are relatively small, such inflation may work quite well in practice. It is, however, partially ad hoc. Furthermore, in applications where the neglected correlations are potentially large, this approach may not work. Cooperative navigation, where significant inter-vehicle correlation is the natural result of the information sharing and inter-vehicle aiding measurements [7], is one such example.

Thus the challenge of fusion quantities with unknown correlation is a general problem on its own. Decentralized cooperative navigation, however, serves as an excellent application on which to test the utility of such algorithms.

1.3 Research Questions

The practicality of cooperative navigation hinges on addressing three key challenges: availability, estimator architecture, and solution integrity. Availability refers to the fraction of time when cooperative navigation can be utilized. The nature of cooperative navigation requires surrounding traffic to be within cooperation range, otherwise there is no *cooperation* to speak of. The estimator architecture challenge deals with the type of estimator used in the fusion of information and states of interest. The choice of estimator architecture will be driven largely by the application concept of operations. The estimator architecture must handle collaborator position uncertainty and will dictate whether each vehicle only models and estimates its own states, or if it will include collaborator states. When shared information is being relied upon, integrity becomes very important. Fault-free integrity is the problem of ensuring, in part, that under normal operating conditions the actual position errors remain smaller than estimated position errors. Faulted integrity deals with the problem of ensuring that the proposed system has the property that a vehicle with a faulty navigation system does not corrupt the otherwise proper state estimates of surrounding vehicle.

With the exception of faulted integrity, this research explores the remaining challenges and attempt to define the problem and describe possible solutions. This is done first by a general derivation of the cooperative aiding system and secondly by three detailed application case-studies: commercial aviation, UAVs, and automotive vehicles, all of which have an experimental data dimension. More specifically, for each application the following three problems are explored:

1. Under various concepts of operation, what system architecture is suited for implementing cooperative aiding to mitigate the effect of a GNSS-outage?

2. Under very limited communication bandwidth, how well can a scheme based on dead reckoning aided by decentralized cooperative navigation perform? Can a unknown correlation data fusion algorithm based on democratic information exchanges (i.e. least restrictive architecture) be devised to make this happen?

3. Given the trends for increased sensing capabilities coupled with multilayer connectivity, does it make sense to consider centralized cooperative inertial navigation schemes? If so, what is the performance attainable when using a system mechanized around a centralized system?

In all three cases, the case-study is anchored to existing or soon-to-be existing sensor/communication systems.

1.4 Prior Work

The Relative Navigation (RELNAV) function of the Joint Tactical Information Distribution System (JTIDS) is one of the early examples of a cooperative navigation system. Designed for military applications, it is a decentralized system mechanized in the Extended Kalman Filter (EKF) framework which aids dead reckoning data with time-of-arrival (i.e. range) measurements between networked vehicles [10]. The system was designed to enabled both accurate relative and absolute navigation information for the networked community. As part of the system design, instabilities caused by the neglect of correlated information in the decentralized EKF were observed [11]. This introduced the topic of source-selection, where community hierarchies were defined to prevent the formation of error loops and retard the instability caused by neglecting inter-vehicle correlations [7]. In this manner, the problem of inter-vehicle correlations was mitigated through restrictions placed on the community information exchange, and not by the estimator architecture.

There are three categories of source selection that were considered: Democratic, Fixed-rank, and Covariance-based [7]. A democratic organization is completely unrestrictive and is shown in Figure 1.2. Each member of the organization is free to collaborate with any other member. Democratic organizations do not work and lead to divergence of the navigation solution. In the fixed-rank organization, each member of the community is given a fixed rank as shown in Figure 1.2. From the perspective of ownship, collaboration occurs only with members of superior rank, thus, making the collaboration one-way. The highest rank member does not accept information. The assignment of fixed-ranks can be appropriate for some applications, but it introduces a challenge when the community is large and continuously evolving. A covariance-based hierarchy is a modification of the fixed-rank hierarchy described. The difference is that member rank assignment is dynamic and is based on the current computed covariance.

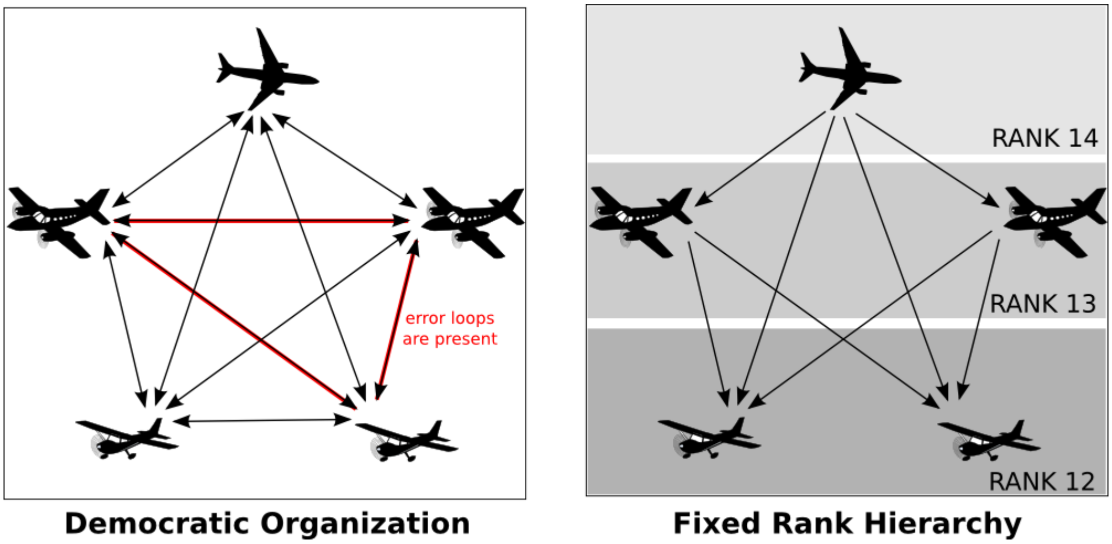


Figure 1.2: Illustration of information flow in a democratic and fixed-rank source selection scheme

Work related to handling correlated quantities in estimation has had a variety of motivations. Early work was motivated by reducing the computational load for Kalman Filters [12]. This was to be accomplished by partitioning a single dynamical system into multiple low-order sub-systems. This is similar to going from a single centralized filter to a decentralized implementation. The computational and flexibility afforded by decentralization, however, comes from dropping states that enable the filter to account for inter-state correlations. The complete loss of correlation information is detrimental to the estimator performance. The work of [13, 14] introduced the Supplemented Partitioning Approach (SPA). This overcame the complete neglect of correlation information by supplementing or inflating the process and measurement noise statistics in a systematic manner.

The application of SPA [13] together with source-selection were approaches studied for overcoming the RELNAV instabilities to prevent the formation of error loops and retard the instability caused by neglecting inter-vehicle correlations [7]. While helpful, neither SPA nor source-selection overcome the underlying challenge of data fusion in the presence of unknown correlation. Without the presence of a single centralized filter, this is the same challenge that needs to be handled for proper fusion of cooperative aiding measurements in a community of networked vehicles.

Data fusion techniques for handling unknown correlations were further developed by work in the robotics community. In the Simultaneous Localization And Mapping (SLAM) problem

[15], similar to the cooperative-navigation problem, proper accounting for the correlation states would require a single centralized filter modeling the state-vector for all vehicles and landmarks. Since the number of landmarks are often large, interest in decentralized or distributed solutions become of interest and the importance of correlations was again identified [16]. The standard Kalman Filter assumption of independent process and measurement noise were violated, and hence, the general problem of estimation in the presence of unknown correlation remained.

To deal with the inconsistency problem an algorithm known as Covariance Intersection (CI) was introduced in [17] as a solution to this problem. Covariance Intersection is a conservative fusion methodology that operates under the assumption of unknown correlation information. Applied to the SLAM application using a decentralized estimator, CI was shown to be a consistent and a computationally advantageous solution since landmark correlation terms did not need to be maintained. This has direct parallels to the decentralized cooperative navigation problem. Neighboring collaborators can be compared to moving landmarks whom ownship will use to derive corrections for its own navigation states. CI has found applications in other areas like multi-robot localization [18], decentralized spacecraft formation state estimation [19], or sensor fusion [20]. A modification to CI, known as Split Covariance Intersection (SCI) allows the CI algorithm to handle known independent quantities in the same manner as would be fused when using the standard Kalman Filter, thus improving the tightness of the attainable accuracy. SCI was applied to SLAM [21] and cooperative multi-vehicle localization [22].

The flexibility of CI to handle *any possible* correlation requires the technique to be conservative in the fusion. This can be overly-conservative in many applications. The work of [23] and later [24] provided an alternative, Bounded Covariance Inflation (BCInf)[1] , where an assumed bound on the size of the possible correlation can be used to relax some of the conservativeness of the standard CI filter.

The choice of centralized or decentralized implementations of cooperative aiding depend largely on the concept of operation under consideration. Cooperative aiding has been considered for underwater applications, where GNSS signals are either intermittent or entirely unavailable. The work of [26] looks at data fusion techniques for cooperative localization of underwater vehicles. As an alternative to the conservative data fusion of CI, this work uses careful bookkeeping and banks of filters on board each UAV to enable proper fusion of the inter-vehicle range measurements. Other techniques also have been introduced to reduce the computational requirements for maintaining such a large state vector, for example [27], through compressing the available information or delaying the covariance updates.

[1] Although first proposed by [23], the name *BCInf* given by [24] has been used in recent literature, e.g. [25], and will be used here as well.

1.5 Motivation

The practicality of decentralized cooperative aiding is demonstrated by the JTIDS RELNAV functionality. Using largely existing sensing and communication equipment on-board vehicles, and without requiring additional external infrastructure, cooperative aiding can serve to mitigate the effect of lost or degraded GNSS signals. Techniques like both source-selection and SPA were developed to mitigate the effect of unknown correlation while relying on the standard EKF framework for fusing relative measurements. In a centralized framework, these are unnecessary since the inter-vehicle correlation is known and maintained. However, in the decentralized framework, such techniques are necessary to prevent or delay filter divergence when utilizing the standard EKF. However, correlation is at the heart of cooperative navigation. Introducing techniques to retard the flow of information or the onset of correlation is in conflict with the essence of cooperative navigation, which achieves improved performance through relative measurements and information sharing. Additionally, recent advances in unknown correlation data fusion techniques introduce new prospects for handling the unknown correlations. Therefore, we propose a decentralized cooperative aiding algorithm based on correlated data fusion algorithms CI and BCInf. While this provides an opportunity to evaluate the utility of cooperative aiding in mitigating loss of absolute measurements like GNSS, it also is, as will be described next, an analysis and evaluation of the unknown correlation data fusion algorithms.

Both CI and BCInf provide answers to the general problem of estimation in the presence of unknown correlation. Although the basic fusion equations are clean and simple, there are certain properties which can be surprising, even counter-intuitive when applied to high order systems with indirectly observed states. The presentation of these techniques, and their relation to traditional Kalman filtering, has not always been clear. There is, in this authors opinion, an information gap in the literature [17, 28, 25] for the engineer, perhaps familiar with the standard Kalman filer, to understand if CI or its variants will be suitable for their application. Most of the published literature deals with low-order simulation examples. Applying these methods to more complex navigation/estimation applications is currently a significant leap. Furthermore, subtle yet important practical considerations when applying these techniques to non-simulated real world data will be uncovered by this work.

1.6 Research Contribution

The goal of this thesis is to begin answering the questions necessary to make aiding by cooperative navigation in a large and dynamic community of users an engineering reality. This required bringing together both prior art and finding modern data fusion techniques which can enable this technology. As seen by the author, the following four items are the contributions of this thesis:

1. Designed detailed architectures of cooperative aiding serving as GNSS-backup for three specific applications: commercial aviation, small UAVs, and automotive vehicles. The proposed architectures are based entirely on existing or soon to be existing hardware.

2. Analyzed and evaluated both centralized and decentralized implementations of cooperative aiding using experimental data. The commercial aviation and automotive case-studies are derived from experimental data obtained from the Federal Aviation Administration (FAA) and the US Department of Transportation (DOT), respectively. The UAV case study is arguably entirely based on experimental data collected by the University of Minnesota Unmanned Aerial Vehicle (UAV) Research Group.

3. Presented a unified derivation of the measurement update for the Kalman filter, Covariance Intersection, and Bounded Covariance Inflation filters. In doing so, their relation is clarified and the strengths and limitations of the approaches are highlighted. Important steps for successful utility of CI/BCInf are detailed. Finally, a previously overlooked covariance normalization step is presented and shown to be important for properly conducting the uncertainty trade-off inherent to both techniques.

4. Applied experimental data as part of an 11-state UAV dead reckoning navigation system to analyze decentralized CI/BCInf-based data fusion estimators.

1.7 Thesis Organization

Towards this end, this dissertation begins with a detailed description of cooperative navigation concepts of operation. General system architectures for both centralized and decentralized implementations will be described, followed by detailed description for three specific case-studies: commercial aviation, small UAVs, and automotive applications. Chapter 3 will present the mathematical details for cooperative navigation and Chapter 4 for unknown correlation data fusion algorithms. Chapters 5, 6, and 7 present case-study results, where Chapter 5 describes commercial aviation application, Chapter 6 small UAVs, and Chapter 7 automotive land-vehicle performance results. A concluding chapter closes the dissertation with final discussion and recommendations for future directions.

Chapter 2

Cooperative Navigation Concepts

2.1 Introductions

In this chapter cooperative navigation is defined and a system-level description is provided. Serving as an aiding system during GNSS-denied or stressed operation is the primary context under which cooperative navigation is discussed. Two classes of implementations, namely centralized and decentralized, are presented along with a discussion of the advantages and disadvantages of each implementation. Finally, concepts of operation and hardware systems suitable for enabling cooperative navigation in three specific applications are discussed: commercial aviation, small UAVs, and automotive applications. In all three cases existing or soon to be existing sensors and communications systems, that may serve to enable the utility of cooperative aiding for the particular application, are presented. The concept of operations and sensor system descriptions will serve to motivate the performance studies which will be shown in later chapters.

2.2 Cooperative Navigation

Cooperative navigation is a general term whose definition can vary depending on the application and context in which it is used. Other names often used synonymously are networked navigation and cooperative localization or positioning. In the context of this work, cooperative navigation is the process where by two or more vehicles collaborate to navigate with accuracy superior to the non-collaborative case. There are three components to cooperative navigation: information sharing, relative sensing, and suitable data-fusion estimators.

The information sharing component requires each vehicle be equipped with a data modem to both broadcast and receive information like estimated position or navigation status. Information sharing, with or without the other components, has significant operational advantages. For example, improved situational awareness, collision avoidance, and data sharing. This is reflected by the trend of increased communication connectivity. But without relative knowledge between vehicles, it is challenging to derive direct utility to navigational accuracy.

The second collaborative component is relative sensing. Relative sensing establishes relationships between vehicles, which when coupled with information sharing defines aiding observations used to improve navigation accuracy. Relative measurements can still be useful even if there is no information sharing. This, however, would no longer be called cooperative navigation and would instead be closer to Simultaneous Localization and Mapping (SLAM). Thus, in this work both relative sensing and information sharing are required for cooperative navigation and are required to formulate cooperative aiding measurements. Finally, a suitable data-fusion estimator is required to take advantage of the cooperative aiding measurements defined by way of the relative sensing and information exchange.

Cooperative navigation can be used as a standalone relative navigation system. In this case one vehicle in the community may be used to define the navigation-frame origin. However, if one or more users have access to absolute location and orientation information, then cooperative navigation can enable absolute positioning for the entire community of vehicles. A standalone cooperative navigation system is highly dependent on a robust relative sensing and communication capability, as well as requires the continued presence of collaborators. In practice, this is difficult to fulfill and therefore it is advantageous to equip each vehicle with an on-board navigation system and use cooperative aiding measurements to aide the on-board solution. In this manner, cooperative navigation augments the on-board navigation system and any loss of access to neighboring collaborators or disruptions in communication and sensing do not immediately jeopardize the navigation capability of the vehicle. This fits well with traditional aided dead reckoning navigation systems like aided INS. Along side traditional aiding measurements like GPS position and velocity or camera-derived measurements, cooperative navigation simply defines a new aiding measurement which can be incorporated without significant changes to existing navigation architectures. Furthermore, when traditional aiding measurements like GPS become unavailable, cooperative aiding which relies on local sensing and communication can continue to arrest the error-growth of the on board dead-reckoning systems of all properly equipped vehicles.

One significant difference between traditional aiding measurements and cooperative aiding measurements is the flow of information. In traditional aiding measurements sensors on board each vehicle provide local observations and subsequently derive aiding corrections to the respective vehicle. Multiple vehicles can use similar sensors to derive on-board corrections, and therefore may be affected by mutual error sources. The aiding however is local, and therefore whether or not there are mutual errors affecting multiple vehicles is largely inconsequential. In contrast, the information exchange inherent to cooperative aiding methods requires the proper and differential treatment of mutual vs independent error sources. Simply stated, errors from multiple independent sources can be averaged out where as only external absolute observations will remove a mutual error source. Therefore, unlike traditional aiding measurements like GPS, proper accounting of error types is critical for the successful utilization of cooperative aiding measurements.

This motivates the importance of choosing a suitable data-fusion estimator with which to mechanize the cooperative aiding measurements. The proper handling of mutual and independent error sources will be considered for two classes of implementations: centralized and decentralized.

2.2.1 Centralized Implementation

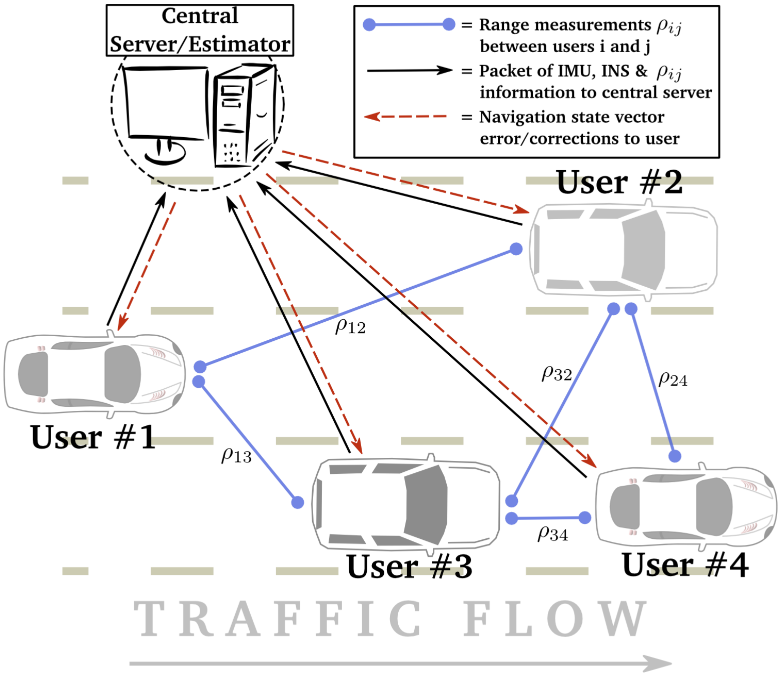

Central Server/Estimator
= Range measurements ρ_{ij} between users i and j
= Packet of IMU, INS & ρ_{ij} information to central server
= Navigation state vector error/corrections to user
User #1
User #2
User #3
User #4
ρ_{12}
ρ_{13}
ρ_{32}
ρ_{24}
ρ_{34}
TRAFFIC FLOW

Figure 2.1: Illustration of communication (and computation) requirements for centralized estimator architecture.

A centralized implementation is a natural solution to the cooperative aiding problem and implies a single estimator is used to model the navigation error-states of all vehicles in the cooperative navigation community. A centralized estimator architecture is illustrated in Figure 2.1.

The centralized estimator receives the on board sensor data for each vehicle as well as any available relative measurements. Using this information, the best available estimate of each vehicle state is computed. This information is broadcast back to each respective vehicle. By virtue of a single, possibly large, state vector, it is apparent that the centralized estimator properly handles the correlation of errors and information shared in the community. This ability to handle vehicle correlations is very significant, because, as will be described shortly, a large portion of the design challenges for cooperative navigation are in handling correlation information. Immediately obvious is the significant bandwidth and computational resources required for a centralized estimator. Also, the requirement for a single centralized estimator maybe overly restrictive for many applications especially with respect to computation and communication bandwidth requirements. This is particularly true in dynamic communities, for example commercial aircraft, where the community of users is constantly evolving.

2.2.2 Decentralized Implementation

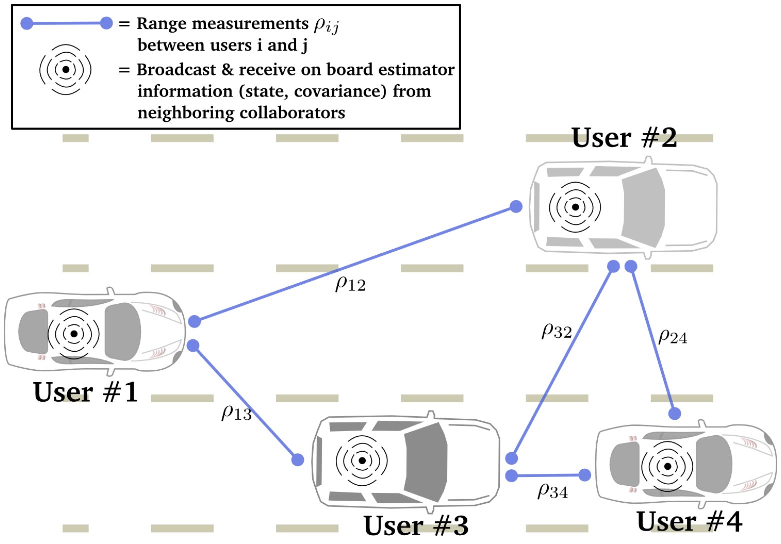

= Range measurements ρ_{ij} between users i and j
= Broadcast & receive on board estimator information (state, covariance) from neighboring collaborators
User #2
User #1
User #3
User #4
ρ_{12}
ρ_{13}
ρ_{32}
ρ_{24}
ρ_{34}

Figure 2.2: Depiction of cooperative navigation implemented based on decentralized filters.

In contrast to the centralized case, the decentralized implementation requires each vehicle to handle computation of corrections. The on-board navigation system of each vehicle exclusively models ownship error-states, and therefore, the inter-vehicle mutual errors are not readily known. A depiction of the decentralized architecture is shown in Figure 2.2. Furthermore, a block diagram of the algorithm to be implemented on each vehicle is depicted in Figure 2.3. In the decentralized framework, each vehicle is running an on board dead-reckoning system where

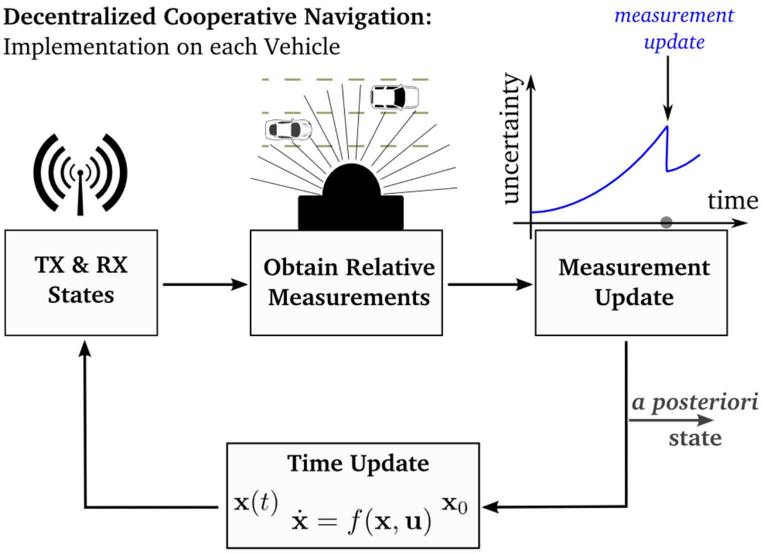

Decentralized Cooperative Navigation:
Implementation on each Vehicle
measurement update
uncertainty
time
TX & RX States
Obtain Relative Measurements
Measurement Update
a posteriori state
Time Update
$\mathbf{x}(t)$
$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$
$\mathbf{x}_0$

Figure 2.3: Diagram of cooperative navigation routine implemented on each vehicle.

only ownship error states are modeled. While the time update is unchanged, the measurement update equations are modified to both account for the fact that the collaborator states are neither part of the filter, nor is their correlation with ownship known. Hence alternate data-fusion estimators are sought to handle the possibly correlated error sources of the cooperative aiding measurements.

Note that the decentralized implementation is fundamentally different than the centralized implementation. This contrasts to some proposed distributed implementations where the centralized implementation is achieved but by distributing the computational load among the community of vehicles (e.g. [26] or [29]).

Two necessary components for cooperative navigation are relative measurements between vehicles and a data link for information exchange. The specific type of relative measurements depends on the sensors available and the application of interest. For example, many robotics applications rely on the availability of relative range and bearing measurements from sensors such as LIDAR, radar, and electro-optical cameras [30]. Similarly, the application and the associated data links will define the achievable bandwidth and therefore the suitability of centralized or decentralized implementations. In what follows we will describe the physical communication and sensing components suitable for implementing cooperative aiding measurements in two aviation and one automotive application.

2.3 Commercial Aviation

The Federal Aviation Administration (FAA) is transitioning from traditional ground-based surveillance and navigation to a more accurate airborne-based system. The new system will rely heavily on Global Positioning System (GPS) measurements for both surveillance and navigation. The requirement to mitigate the impact of a GPS outage has created a renewed interest in alternative positioning solutions, more generally known as alternative positioning navigation and timing (A-PNT). Inertial Navigation Systems (INS) are natural solutions to mitigate the effect of a GPS outage. However only navigation grade inertial sensors would be capable of operating without aiding for outages longer than several minutes. Technologies other than INS currently under consideration for A-PNT services include an improved distance measuring equipment (DME) infrastructure and passive multilateration (MLAT) [31]. One attractive characteristic of these systems is that they are largely based on existing or soon-to-be existing FAA systems and infrastructure. In alignment with this characteristic, we anchor our work to the application of an alternative backup system based on INS aided by cooperative navigation and whether it can be realized using existing or soon-to-be-existing systems. In this way we examine how cooperative navigation can be used to help mitigate the effects of GPS-denial in the future national airspace system.

Here we describe two existing systems which fulfill the communication and relative-sensing requirements for cooperative navigation in commercial aviation applications. Automatic Dependent Surveillance-Broadcast (ADS-B) can be used to share information between aircraft, independent of air traffic control, and has a defined message structure. Traffic Collision Avoidance System (TCAS) is a collision avoidance system providing relative range measurements between aircraft. The combined information from these two systems can be used in a cooperative navigation framework to aid the on board inertial navigation system. Here we detail aspects of ADS-B and TCAS that are relevant to the cooperative navigation application.

2.3.1 ADS-B

Automatic Dependent Surveillance-Broadcast (ADS-B) combines an aircraft's positioning source and avionics in cooperation with installed ground stations to create an accurate surveillance system for air traffic control. In May of 2010 the Federal Aviation Administration instituted a Final Rule which requires the majority of aircraft to have ADS-B hardware and software installed by the year of 2020 [32]. Meanwhile, ADS-B ground stations are being installed across the United States [33]. The basic operation of ADS-B is graphically depicted in Figure 2.4. Each aircraft will be required to automatically broadcast an ADS-B message at regular intervals. There are two data link options for aircraft, the first based on Universal Access Transceiver (UAT)

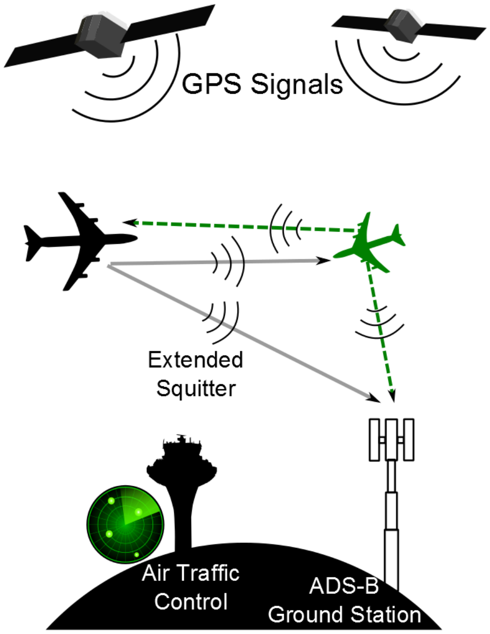
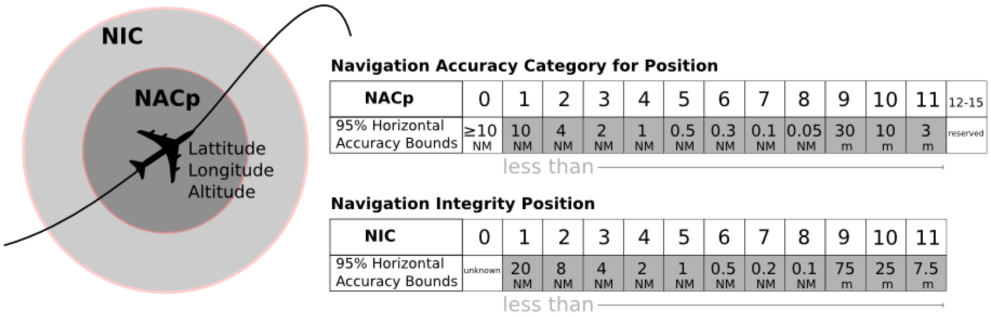


Figure 2.4: Depiction of ADS-B operation

broadcast link and the second based on the 1090 MHz Extended Squitter (ES) broadcast link [32]. The 1090 MHz ES broadcast link is the internationally accepted data link for ADS-B. However, it supports a smaller bandwidth. In contrast the UAT broadcast link can support additional information exchange and is intended to support the general aviation community. Here we focus on the 1090 MHz ES broadcast link.

The 1090 MHz ES broadcast link will be installed on the majority of commercial aircraft. Large commercial aircraft are most likely to be equipped with advanced sensor suites, enabling them to broadcast the most accurate position information receivable by surrounding aircraft. The broadcast messages have information about the aircraft position, velocity, the estimated accuracies and system integrity. The ADS-B Minimal Operational Performance Standards (MOPS) has evolved from DO-260, to DO-260A and in December of 2009 to DO-260B [34]. Two significant changes are the improvement of aircraft location broadcast as well as the improved reporting of integrity and accuracy quality information. In the cooperative navigation application these changes are significant as it is critical for each aircraft position broadcast to be accompanied with a quality metric.



Navigation Accuracy Category for Position

NACp	0	1	2	3	4	5	6	7	8	9	10	11	12-15
95% Horizontal Accuracy Bounds	≥10 NM	10 NM	4 NM	2 NM	1 NM	0.5 NM	0.3 NM	0.1 NM	0.05 NM	30 m	10 m	3 m	reserved

Navigation Integrity Position

NIC	0	1	2	3	4	5	6	7	8	9	10	11
95% Horizontal Accuracy Bounds	unknown	20 NM	8 NM	4 NM	2 NM	1 NM	0.5 NM	0.2 NM	0.1 NM	75 m	25 m	7.5 m

Figure 2.5: Mapping of broadcast position accuracy and integrity values to physical equivalents

Accuracy

Two parameters from the ADS-B message that are important to cooperative navigation applications are the position broadcast and the position accuracy metric. The mapping between the values broadcast and the physical equivalents are shown in Figure 2.5. Therefore, an ownship aircraft which receives an ADS-B broadcast from a neighboring collaborator will only have a scalar quality metric to evaluate the accuracy of the received position broadcast.

In the decentralized estimators presented as part of this work, the position covariance of the collaborator is assumed to be shared with the ownship aircraft as part of the collaborator broadcast. We acknowledge that this will not be the case in currently envisioned uses of these systems. However, this assumption is not entirely without merit as it is possible to derive an approximate covariance matrix for the collaborator position broadcast using the received scalar parameter. One possible method to achieve this was presented in a meeting for the RTCA working group developing the Airborne Surveillance and Separation Assurance Processing (ASSAP) MOPS [35].

Broadcast Interval and Latency

Airborne aircraft are required to transmit position and velocity messages at least once per second. There is a 2.0 second total latency allocation, of which 0.6 seconds can be uncompensated [36]. Latency compensation may be in the form of position and velocity extrapolation. However, any uncompensated latency must be accounted for in the broadcast NACp value, as previously defined in Figure 2.5. The possibility of both compensated and uncompensated latency must be considered in the cooperative navigation application. An aircraft flying at 400 knots and repeatedly broadcasting with 0.6 seconds of uncompensated latency may introduce a position error bias larger than 120 meters. The ADS-B message elements do not include a time stamp.

The scalar NACp parameters would be insufficient for knowing the statistical character of this broadcast error. Therefore, error models assuming uncorrelated white additive noise may lead to optimistic covariance estimates for the cooperative navigation application.

Interoperability of UAT and 1090ES

The dual link requirement set by the FAA introduces interoperability issues that should be considered. Commercial airliners or larger GA aircraft are expected to install or upgrade to the 1090 ES standard. Most other GA aircraft are expected to use UAT-compatible equipment. Aircraft will only be capable to receive information transmitted from aircraft using the same data link. Therefore, a GA aircraft equipped with UAT will not be able to directly receive the 1090 ES broadcasts of any neighboring commercial airliners. The FAA intends to use Automatic Dependent Surveillance-Rebroadcast (ADS-R) to solve this interoperability limitation. ADS-R will use ground-based transmitters to rebroadcast the received broadcasts on the alternate data link [37]. The retransmission can introduce additional delays and the effects of this step must be considered.

2.3.2 TCAS

Traffic Collision Avoidance System (TCAS) was developed to provide an added layer of collision protection, independent of air traffic control. The system uses range measurements based on round-trip timing and (if available) altitude broadcasts to alert the pilot of dangerous traffic situations. Additionally, bearing measurements are also available, but their poor quality limit them only to be used for updating visual displays and are not part of the collision avoidance logic. The Traffic Collision Avoidance System Minimum Operational Performance Standards (TCAS MOPS) [38] specifies that TCAS systems must provide reliable coverage out to a horizontal range of $R = 14$ nmi and recommends $\pm 10,000$ ft in relative altitude.

The required error specifications are that jitter be less than 50 ft RMS and bias be less than 125 ft (for Mode S). In order to utilize the TCAS measurements for navigation an appropriate error model must be developed to adequately capture the characteristics of TCAS errors. The jitter may be modeled as additive wide-band noise with a standard deviation of 50 ft. The bias however needs to be characterized before a model can be assumed.

2.4 Unmanned Aerial Vehicles

Sensor
Inertial measurement unit
3 axis accelerometer
3 axis gyroscope
3 axis magnetometer
Airspeed (*pitot-probe*)
Baro-altimeter
GPS *position* and *velocity*

Table 2.1: Sensors commonly on board small UAVs.

A growing number of applications are envisioned for small UAVs, those weighing $5 - 10$ *lbs* and small enough to fit inside the trunk of a law-enforcement squad car. The light-weight and (comparatively) low-cost characteristics of small UAVs make them attractive mobile sensing platforms which can be used for applications like border surveillance, traffic monitoring, and precision agriculture. These same characteristics, however, impose significant constraints on the choice of sensors and system architecture for the on board navigation systems. A list of sensors commonly found on board UAVs are listed in Table 2.1. At the heart of most UAV navigation systems are GPS-aided dead reckoning systems built around these sensors [39]. A comparison of Inertial Navigation System (INS) drift rates and costs for various inertial sensor qualities are shown in Table 2.2. Most small UAV navigation systems are based on automotive/consumer-grade inertial sensors, and therefore are reliant on uninterrupted access to GPS position and velocity measurements.

There is growing interest in alternative or back-up navigation systems to enable robust UAV operation through GPS-denied or stressed environments. This work proposes using networked UAVs and cooperative aiding measurements to enable UAV operation through extended durations of GPS-denial. Current Concept of Operations (CONOPS) being envisioned for UAVs make networked navigation a strong candidate backup navigation system. Figure 2.6 shows an example of law enforcement or search and rescue operations where multiple UAVs and emergency vehicles are deployed and cooperative aiding can be used. A single High-Flyer (HF) aircraft equipped with higher quality sensors operating above multiple Low-Flyer (LF) UAVs is a concept receiving attention for military applications [40, 41]. By way of cooperative aiding, the navigation information of the highest quality member would reach other neighboring vehicles.

Cooperative aiding enables UAVs to treat neighboring vehicles as *navigation beacons* and

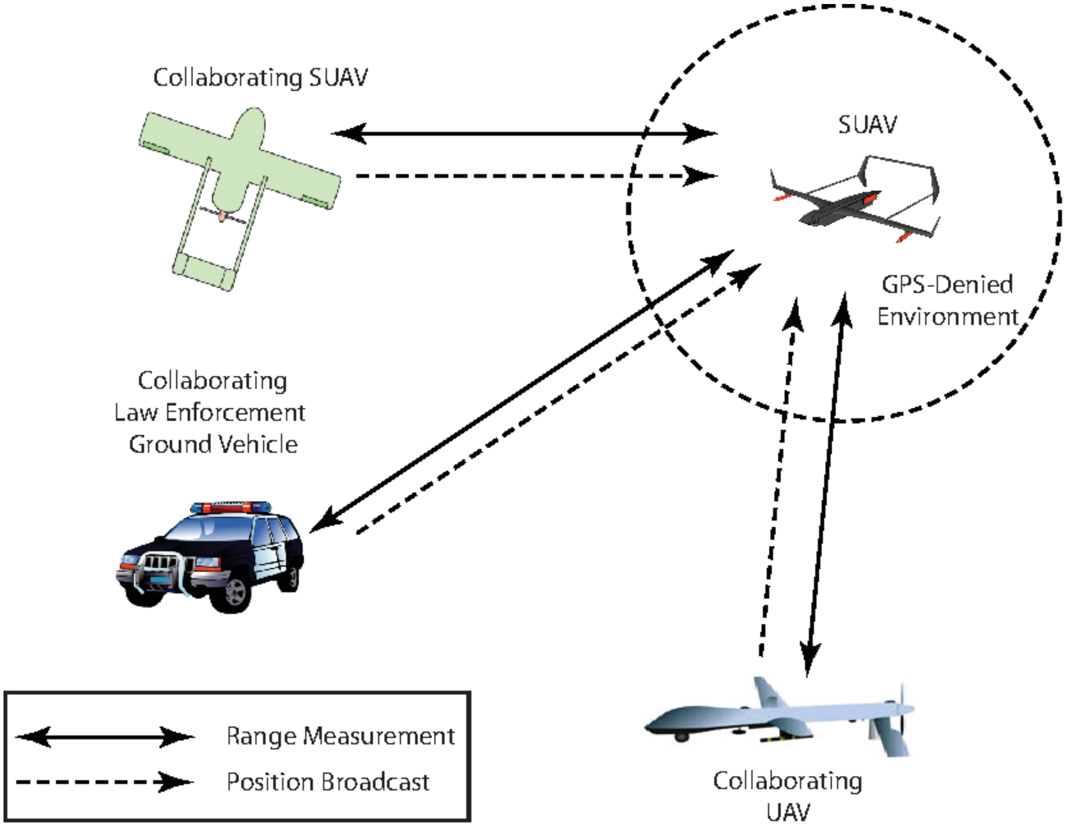


Figure 2.6: Architecture for a cooperative navigation system [1].

Quality Moniker or Label	**Position Error Drift Rate (km/hr)**	**Approximate System Cost**	**Typical Applications Used in**
Strategic Grade	less than 0.001	\$10,000,000+	Submarines, ICBM
Navigation Grade	1.5	\$50,000 - \$100,000	Aircraft navigation
Tactical Grade	20 - 100	\$10,000 - \$20,000	Smart munitions
Automotive/Consumer Grade	100 +	\$100 - \$10,000	Cars, UAVs, Toys

Table 2.2: Quality moniker and rule of thumb for INS/IMU drift rates [1]

thereby reduce the error growth-rate of the on-board dead reckoning system as the UAV flies through an extended GPS-outage. The relative sensing and information sharing requirement for cooperative aiding will now be discussed for the UAV application.

Examples of relative-measurement sensors are time-of-arrival or relative-bearing measurements between networked vehicles. The distance or range between vehicles can be derived from the time-of-arrival measurement. Both range and bearing measurements are parts of the transponders of the Traffic Collision Avoidance System for commercial aircraft [42]. Recent studies analyzing the usage of TCAS on the Global Hawk remotely piloted vehicle confirms that TCAS-like systems are actively being considered for UAVs to fulfill *sense and avoid* requirements [43]. Therefore, it is reasonable to assume relative-sensing capabilities between UAVs and other air/ground vehicles will become standard equipment on-board UAVs in the near future. Prototype 915 MHz [44] or 2.4 GHz [45] ranging sensors, developed using commercial off-the-shelf radio-frequency hardware, claim decimeter accuracy with ranges in excess of 1 km. These may exemplify the type of performance we may expect for future UAV ranging capabilities.

The information sharing capability is a necessary component in order to define aiding measurements from the relative observations. An analogous capability again in commercial aviation is Automatic Dependent Surveillance-Broadcast (ADS-B). Mandated by the Federal Aviation Administration (FAA) to be installed on the majority of aircraft by 2020 [32], ADS-B enables aircraft to share their current position estimates with both ground receivers or other ADS-B equipped aircraft. Mode S transponders with ADS-B capability are already commercially available for UAVs [46]. Considering current rule making for safe integration of UAVs into the national airspace, it is again reasonable to believe an information sharing capability, and more precisely, regular broadcasts of current UAV estimated location and accuracy, will become standard capability on UAVs.

Therefore the relative-sensing and information sharing capability required for cooperative aiding does not impose any significant changes to standard UAV hardware.

2.5 Automotive Vehicles

The evaluation of cooperative aiding performance for ground-vehicle applications is nicely captured by an automotive case study. Current trends are towards increased automotive automation and connectivity. In 1999 Mercedes-Benz was the first car manufacturer to introduce radar based Adaptive Cruise Control (ACC) [47]. Since then many high and mid class models have incorporated ACC and new safety and comfort functions are continuing to be developed. An overview of these functions are shown in Figure 2.7. Together with the feature listed in Figure 2.7, other discussed concepts like automotive platooning or mileage-based road tax all benefit or depend on accurate vehicle sensing and location information. GPS-based positioning is severely challenged by the diverse environment of modern driving. Urban canyons, bridges and tunnels, and even neighboring vehicles all can degrade or entirely block GNSS-based signals. This dense driving

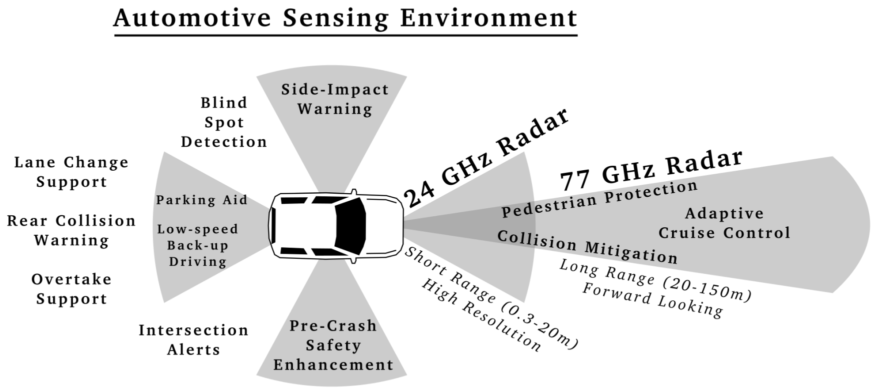


Figure 2.7: Automotive sensing environment as foreseen by manufacturers. (Based on [2] Fig. 2)

environment may, however, serve as an opportunity for cooperative-based aiding techniques. It is plausible that cooperative aiding can serve to mitigate the loss of GPS for automotive applications.

2.5.1 Radar

In order to facilitate cooperative aiding requires a relative sensing and information sharing capability. Relative sensing is soon becoming standard on new vehicles as it enables the safety and comfort features described previously. Much of these sensors are based on long range radar (LRR) and short range radar (SRR). A commercially available LRR manufactured by Bosch is shown in Figure 2.8 .Compared to alternative sensors like video, laser, or ultrasonic, the advantages of weather independence and direct acquisition of range and velocity afforded by radar make it advantageous. The LRR generally operate at 77 GHz frequency and measure range from $1 - 200$ m with a maximum field of view of ±10° [47]. The SRR are ultra-wideband and operate at 24 GHz frequency[1] , with a bandwidth of 4 GHz, and have centimeter level range accuracy up to a distance of 25 m with maximum field of view of 55° [2]. A combination of LRR and SRR serves as an excellent sensing platform to fulfill the relative sensing requirements of cooperative navigation.

2.5.2 Dedicated Short Range Communication

Current efforts on a standard communication protocol for automotive vehicles can serve to fulfill the information sharing requirement. IEEE 802.11p is an enhancement to the common

[1] Frequency allocation for SRR is not the same globally. For example, European regulations have adopted 24 GHz as an interim technology with 79 GHz as the long term solution [47].

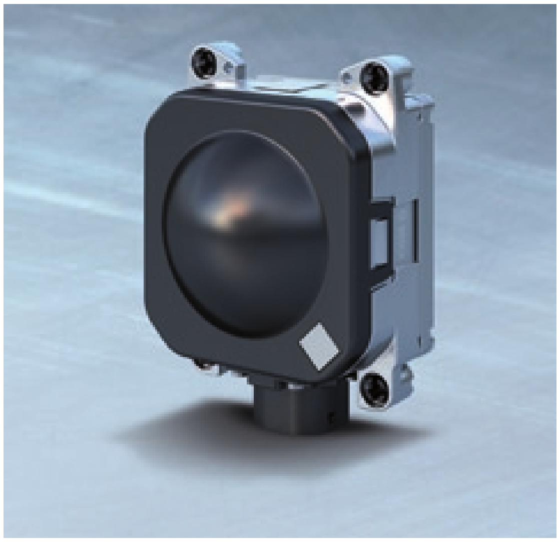


Figure 2.8: Bosch 3rd generation long range radar used in adaptive cruise control and predictive emergency braking systems [3].

Wi-Fi standard 802.11 to support data exchange between high speed vehicles and/or roadside infrastructure. Operating in a licensed 5.9 GHz frequency band, this standard was developed for rapidly changing environments where short-duration communication exchanges are required. This is the basis for the forthcoming Dedicated Short Range Communication (DSRC) technology which is supported by the U.S. Department of Transportation to enabled vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) applications [48]. DSRC is expected to handle 6 *Mbit/s* and a range greater than 1,000 feet with minimal latency [49]. The low-latency, short-range communication is well suited for the proposed decentralized estimation algorithm for cooperatively aided INS. Centralized applications can also be enabled, but would require significant infrastructure installations, relying on V2I, to get information to the central processor. However, considering the wide range of applications being considered for DSRC, it is likely such infrastructure installations will become standard along roadways.

2.6 Conclusion

In this chapter cooperative navigation was defined as process whereby two or more vehicles collaborate to navigate with accuracy superior to the non-collaborative case. A system-level description for the centralized implementation, where a single estimator is used to model the navigation error-states of all vehicles in the cooperative navigation community, as well as a a decentralized implementation, where the on-board navigation system of each vehicle exclusively models ownship error-states and must handle the computation and application of corrections locally, was given. Finally it was shown that many applications have existing sensing and communications systems which are suitable for cooperative navigation implementation. In commercial aviation the on-board collision avoidance system TCAS together with the information sharing through ADS-B would fulfill the sensing and communication requirements. Similar sensing and communication systems are detailed for UAV applications and finally automotive radar systems together with forthcoming dedicated short range communication systems would be enablers for automotive applications.

Apart from information sharing and relative sensing, a third component, namely suitable data-fusion estimators, was described as important for enabling cooperative navigation. Understanding the challenge for picking a suitable data-fusion estimator requires laying the mathematical framework for cooperative navigation. This will be the topic of the next chapter.

Chapter 3

Cooperative Navigation System Architectures

3.1 Introduction

This chapter details the mathematical model for implementing cooperative navigation. In doing so the challenges associated with choosing a suitable estimator for handling the cooperative aiding measurements, particularly in the decentralized implementation, are described. This will lay the framework for Chapter 4 where our proposed estimator is presented along with a detailed discussion of estimators handling unknown correlations.

3.2 Centralized and Decentralized

In the absence of cooperation, the navigation state dynamics of each vehicle in a community are nominally uncorrelated. The sensor errors in neighboring vehicles can be assumed independent, and hence each vehicle can run an on board dead reckoning navigation system independent of others in the community. We assume each vehicle only models ownship error states. Assuming n states modeled for each vehicle, the $n \times 1$ state vector for the i^{th} vehicle is represented as $\mathbf{x}_i$. The continuous-time state space model for the i^{th} vehicle can be written in a general form as:

$$\begin{aligned} \dot{\mathbf{x}}_i(t) &= \mathbf{f}(\mathbf{x}_i, \mathbf{w}_i) && (3.1) \\ &\approx F_i(t)\mathbf{x}_i(t) + G_i(t)\mathbf{w}_i(t) && (3.2) \end{aligned}$$

where Equation 3.2 is the linearization of the general dynamics. Matrices $F_i(t)$ and $G_i(t)$ are the system and input matrices, respectively, and $\mathbf{w}_i(t)$ is the $m \times 1$ stochastic input vector.

In this work we assume an inertial measurement unit based-dead reckoning navigation system on each vehicle is aided by cooperative navigation. With this assumption the state vector $\mathbf{x}$ is defined to be:

$$\mathbf{x} \quad = \quad \begin{bmatrix} \mathbf{p}^T & \mathbf{v}^T & \boldsymbol{\psi}^T & \mathbf{b}_a^T & \mathbf{b}_g^T \end{bmatrix}^T \tag{3.3}$$

where $\mathbf{p}$ and $\mathbf{v}$ are the position and velocity vectors, $\boldsymbol{\psi}$ is the attitude vector, and $\mathbf{b}_a$ $\mathbf{b}_g$ are the accelerometer and gyro sensor error states. The dead reckoning formulation will define the system and input matrices F_i and G_i. Details on how to form these matrices for both a 2-D INS and a 3-D airspeed based dead reckoning system for a single vehicle are provided in Appendix B and Section 6.4, respectively. A discretized version of these models will be utilized for the aviation and automotive simulations presented in later chapters. For a community of N vehicles, there would be N such models, one for each vehicle. In this manner, N estimators can be formed using Equation 3.2 where $i = 1 \ldots N$. The covariance of each vehicle's estimator, $P_{ii}(t_k) = E\left\{\tilde{\mathbf{x}}_i(t_k)\tilde{\mathbf{x}}_i^T(t_k)\right\}$, is propagated by the discrete Lyapunov equation given as:

$$\begin{aligned} \Phi_i(t_k) &\approx I + F_i(t)\Delta t \\ Q_{di}(t_k) &\approx G_i(t)Q_iG_i^T(t)\Delta t \\ P_{ii}^-(t_{k+1}) &= \Phi_i(t_k)P_{ii}^+(t_k)\Phi^T(t_k) + Q_{di}(t_k) \end{aligned} \tag{3.4}$$

where $\Phi_i(t_k)$ is the state transition matrix and $Q_i(t)$ and $Q_{di}(t_k)$ are the continuous- and discrete-time process noise, respectively. Estimators formed using this model will be decentralized estimators.

In contrast, a centralized estimator will model the states of all vehicles in the community in a single estimator. The state vector and associated continuous-time state space model for a community of N vehicles in a centralized estimator is:

$$\begin{aligned} \mathbf{X}(t) &= \begin{bmatrix} \mathbf{x}_1^T(t) & \mathbf{x}_2^T(t) & \ldots & \mathbf{x}_N^T(t) \end{bmatrix}^T \\ \dot{\mathbf{X}}(t) &= \mathbf{F}(t)\mathbf{X}(t) + \mathbf{G}(t)\mathbf{w}(t) \end{aligned} \tag{3.5}$$

where $\mathbf{X(t)}$ is shown in block vector form and is of dimension $(n \cdot N) \times 1$. Matrices $\mathbf{F}(t) \in \mathbb{R}^{nN \times nN}$ and $\mathbf{G}(t) \in \mathbb{R}^{nN \times mN}$ are the system and input matrices, respectively, and $\mathbf{w}(t) = \begin{bmatrix} \mathbf{w}_1^T(t) & \mathbf{w}_2^T(t) & \ldots & \mathbf{w}_N^T(t) \end{bmatrix}^T$ is the augmented stochastic input vector for the centralized model.

The dynamics of each individual vehicle represent a subsystem in the centralized model of Equation 3.5. Each subsystem can be considered as evolving independently. Therefore, the system and stochastic matrices are in block diagonal form:

$$\begin{aligned}\mathbf{F}(t) &= \begin{bmatrix} F_1(t) & & & \text{O} \\ & F_2(t) & & \\ & & \ddots & \\ \text{O} & & & F_N(t) \end{bmatrix} \\ \mathbf{G}(t) &= \begin{bmatrix} G_1(t) & & & \text{O} \\ & G_2(t) & & \\ & & \ddots & \\ \text{O} & & & G_N(t) \end{bmatrix} \end{aligned} \tag{3.6}$$

An estimator formed using the centralized model will be referred to as the centralized estimator. The covariance of the entire community has the following structure:

$$\mathbf{P}(t_k) = \begin{bmatrix} P_{11}(t_k) & P_{12}(t_k) & \dots & P_{1N}(t_k) \\ P_{21}(t_k) & P_{22}(t_k) & & \\ \vdots & & \ddots & \\ P_{N1}(t_k) & & \dots & P_{NN}(t_k) \end{bmatrix} \tag{3.7}$$

In this manner a single centralized estimator can be formed. The covariance of centralized estimator, $\mathbf{P}(t_k)$, is again propagated by the discrete Lyapunov equations. This is the same as the covariance-update equations for the decentralized filter given in Equation 3.4, except that there is only a single estimator and therefore any reference to i^{th} vehicle is dropped. In the absence of any collaboration, the centralized covariance matrix $\mathbf{P}(t_k)$ is block diagonal where the inter-vehicle correlation $P_{ij}(t_k) = 0$, where $i \neq j$. As will be described next, cooperation as given by the measurement model makes $P_{ij}(t_k) \neq 0$.

3.3 Relative Measurements

Relative measurements introduce the functional relationship between navigation states of two vehicles. Therefore, the handling of relative measurements is at the heart of the cooperative navigation problem. This is conceptually different than absolute measurements. Repeated absolute observations, assuming zero mean and white error statistics, implies that each measurement contains new information and, thus, the estimation error will approach zero for that state. In contrast, observing a relative quantity between two states causes a limited reduction in the estimate uncertainty because each additional measurement does not contain completely new information. The achievable benefit will depend on the initial uncertainty of the two states as well as the observation mapping between the states. Among available relative measurements, like range, position, or bearing, in this work we consider only relative range measurements.

Discrete EKF Measurement Update	
Innovation	$\tilde{\mathbf{y}} = \mathbf{y} - \hat{\mathbf{y}}$
Innovation Covariance	$S = E\{\tilde{\mathbf{y}}\tilde{\mathbf{y}}^T\}$
Kalman Gain	$K = P^- H^T S^{-1}$
Update State Estimate	$\mathbf{x}^+ = \mathbf{x}^- + K\tilde{\mathbf{y}}$
Update Covariance	$P^+ = (I - KH)P^-(I - KH)^T$ $+KRK^T$
where $H = \frac{\partial \mathbf{y}}{\partial \mathbf{x}}\vert_{\hat{\mathbf{x}}^-}$, $\mathbf{y}$ is the nonlinear measurement model.	

Table 3.1: Discrete EKF measurement update equations. For simplicity, explicit reference to (discrete) time has been dropped.

Relative range measurements can be expressed in terms of two vehicles, i and j, and will relate the respective vehicle position states $\mathbf{p}_i$ and $\mathbf{p}_j$:

$$\rho_{ij}(t_k) = ||\mathbf{p}_i(t_k) - \mathbf{p}_j(t_k)|| + v(t_k) \tag{3.8}$$

where the position states for each vehicle i and j are a subset of the complete vehicle state vectors $\mathbf{x}_i(t_k)$ and $\mathbf{x}_j(t_k)$, respectively. Linearizing Equation 3.8 leads to the following observation equation:

$$y_{ij}(t_k) = H_i(t_k)\mathbf{x}_i(t_k) + H_j(t_k)\mathbf{x}_j(t_k) + v(t_k) \tag{3.9}$$

where $H_i(t_k)$ and $H_j(t_k)$ are $1 \times n$ Jacobian vectors mapping the states of vehicles i and j to the observed quantity, respectively. The measurement noise is modeled as an additive quantity, $v(t_k)$. The standard measurement update equations for the EKF, shown in Table 3.1, require that a linearized form of the measurement equation is available. This is necessary to compute the innovation covariance and subsequently the Kalman gain. The Kalman gain is an important quantity, which ultimately maps the innovation (difference between the observed and predicted measurement) into state estimate corrections. Equation 3.9 is the linearized measurement update equation for the decentralized estimator. For a centralized estimator, multiple copies of Equation 3.9, one for each relative observation present in the community, form the measurement model.

To understand why decentralized estimation is challenging, we will start by using Equation 3.9 to form the innovation $\tilde{y}_{ij}(t)$ for the observed measurement:

$$\begin{aligned}\tilde{y}_{ij}(t_k) &= y_{ij}(t_k) - \hat{y}_{ij}(t_k) \\ &= H_i(t_k)\tilde{\mathbf{x}}_i(t_k) + H_j(t_k)\tilde{\mathbf{x}}_j(t_k) + v(t_k)\end{aligned} \tag{3.10}$$

where tilde is used to represent the difference between the true and estimated quantity. The value of the innovation and the associated statistics (captured in the innovation covariance matrix) are used to form corrections to the current state estimates of the filter. Using Equation 3.10, the covariance of the innovation is given by

$$\begin{aligned} S(t_k) &= E\{\tilde{\mathbf{y}}_{ij}(t_k)\tilde{\mathbf{y}}_{ij}^T(t_k)\} \\ &= H_i(t_k)P_{ii}^-(t_k)H_i^T(t_k) \\ &+ H_j(t_k)P_{jj}^-(t_k)H_j^T(t_k) \\ &+ R(t_k) - 2H_i(t_k)P_{ij}^-(t_k)H_j^T(t_k) \end{aligned} \tag{3.11}$$

This equation assumes unbiased estimators for vehicles i and j, and that the *a priori* errors for both vehicles are independent of the range measurement noise $v(t_k)$. Therefore, the innovation is a zero mean process. The corrections applied to the state estimates based on the relative measurements depend on the statistical properties of the innovation. As is clear from the EKF measurement update equations in Table 3.1, errors in the computation of this parameter will affect both the *a posteriori* state estimate and covariance.

Now let us examine whether each term in computing the innovation covariance, Equation 3.11, is available for general centralized and decentralized estimators. The relative measurement error statistics $R(t_k)$ can be computed and stored from prior calibration testing of the relative range measuring sensor. A centralized estimator will model the states of all vehicles in the community. Hence, the state error covariance for vehicles i and j, $P_{ii}^-(t_k)$ and $P_{jj}^-(t_k)$, would be readily available as would the cross correlation $P_{ij}^-(t_k)$.

In contrast, for the decentralized estimator the availability of each quantity depends on the vehicle under consideration. Without loss of generality we consider vehicle i as ownship and vehicle j as the collaborator. Since ownship does not model the states other than its own, it is clear that only $\mathbf{x}_i^-(t_k)$ and $P_{ii}^-(t_k)$ would be known to ownship. This introduces the necessity of information sharing as part of cooperative navigation for the decentralized estimator. Assuming, for the moment, no communication bandwidth constraints, the collaborator could transmit to ownship its state estimate and covariance, $\mathbf{x}_j^-(t_k)$ and $P_{jj}^-(t_k)$. The linearized measurement mapping matrices $H_i(t_k)$ and $H_j(t_k)$ will often depend on the current state estimates of both vehicles and hence can be formed on board ownship. Thus, the one term that remains unknown is $P_{ij}^-(t_k)$: the *a priori* cross-correlation between the states of vehicles i and j.

The inter-vehicle correlation term would be non-zero if the vehicles had cooperated previously, or if they shared a mutual collaborator in the past. Without substantial modifications, decentralized estimators on either vehicle would have no method to keep track of $P_{ij}^-(t_k)$. To simply dismiss the inter-vehicle correlation would be naive since the very nature of cooperative navigation will cause this term to grow [7]. This is the fundamental challenge in finding a suitable estimator architecture for decentralized cooperative navigation. If the inter-vehicle correlation is ignored, each additional measurement is handled as if it was new information uncorrelated with the ownship state errors. This can lead to inconsistent estimates [8], where inconsistency is defined as the difference between the estimated and true state covariance not being positive semi-definite [50]. Without proper estimates of the system uncertainty the on-board navigation system is unable to provide timely warnings in the face of excessive navigation errors. Therefore, in terms of navigation system performance metrics an inconsistent estimate represents a loss of integrity [51].

It was stated that ownship (vehicle i) can receive the collaborator (vehicle j) state estimate and covariance, $\mathbf{x}_j^-(t_k)$ and $P_{jj}^-(t_k)$. Here we expand slightly on this step and detail the implications. The received state and covariance will be estimated quantities, hence to be more precise, $\hat{\mathbf{x}}_j(t_k)$ and $\hat{P}_{jj}(t_k)$, will be received and available where the $\hat{}$ differentiates the estimated from the true quantity. Using tilde to represent estimate error, the state $\mathbf{x}_j(t_k)$ can be expressed as $\tilde{\mathbf{x}}_j(t_k)+\hat{\mathbf{x}}_j(t_k)$. The general relative measurement defined in Equation 3.9 can be rewritten as:

$$y_{ij}(t_k) = H_i(t_k)\mathbf{x}_i(t_k) + H_j(t_k)\hat{\mathbf{x}}_j(t_k) + \bar{v}(t_k) \tag{3.12}$$

where

$$\bar{v}(t_k) = H_j(t_k)\tilde{\mathbf{x}}_j(t_k) + v(t_k) \tag{3.13}$$

is the supplemented measurement noise. This formulation is necessary for the decentralized framework, where ownship needs a method to account for the uncertainty in the information received from the collaborator. In the centralized framework, this is irrelevant as the states of both vehicles are included in a single filter. This method of accounting for collaborator uncertainty is based on the general Supplemented Partitioning Approach (SPA) [13, 14].

Under two assumptions the mean of the supplemented measurement noise will be zero. Firstly, the errors in the broadcast state of vehicle j at time t_k should be independent of the errors in the range measurement received by vehicle i at time t_k. Secondly, the estimator on board vehicle j has unbiased state errors. The first assumption is reasonable and implies the operational requirement that the state broadcast occurs before the local measurement update of each vehicle. While the second requirement is satisfied in theory, modeling and linearization errors will inevitably introduce some bias. However, this is a problem with linearized estimators in general and not specific to SPA.

The variance of the supplemented measurement noise can be computed as

$$\begin{aligned} \bar{R}(t_k) &= E\{\bar{v}(t_k)\bar{v}(t_k)^T\} \\ &= H_j(t_k)E\{\tilde{\mathbf{x}}_j(t_k)\tilde{\mathbf{x}}_j^T(t_k)\}H_j^T(t_k) + E\{v(t_k)^2\} \\ &= H_j(t_k)P_{jj}(t_k)H_j^T(t_k) + R(t_k) \end{aligned} \tag{3.14}$$

Thus, the original sensor noise is inflated by mapping the uncertainty in the state of vehicle j along the direction of the relative measurement observation, $H_j(t_k)$. The entire quantity in Equation 3.14 is then treated as the new measurement noise statistic and is used in the decentralized estimation framework. The effect of the collaborator state uncertainty is no longer entirely lost, but instead is treated as additive white noise.

This is an improvement in accounting for subsystem coupling. However, it is an overly simplistic way of handling the true nature of the collaborator uncertainty. The state estimate errors in vehicle j are neither white nor is their effect additive. The first problem with this assumption is that the effect of additive white noise can be averaged out whereas, in reality, the utility of any relative measurement should be limited by the uncertainty in absolute position of the collaborator. Hence, while SPA is used to account for collaborator uncertainty, it fails to resolve the fundamental problem of unknown or neglected inter-vehicle correlation in the decentralized framework.

3.4 Conclusion

In this chapter the challenge of handling unknown inter-vehicle correlations for the decentralized cooperative navigation implementation were presented. An EKF, which is demonstrated to be suitable for centralized implementations, is unsuited for decentralized implementations. By applying SPA information about collaborator uncertainty is not entirely lost, however, there exists an unknown correlation between the *a priori* state and the measurement errors which cannot be neglected. The next chapter will show how SPA together with an alternate class of estimators can be combined to form our proposed recursive filter for decentralized cooperative navigation applications.

Chapter 4

Correlated Data Fusion Filters

4.1 Introduction

Handling correlated quantities is at the heart of many estimation problems in aerospace guidance, navigation, and control. It is the proper knowledge of correlation between error-states that enables indirect observations to provide observability to a large state vector. A well studied example of this is the aided Inertial Navigation System (INS). Low-rate aiding from GPS-based position measurements are commonly sufficient to provide the stochastic observability required to estimate other error-states like gyro and accelerometer biases. The assumed dynamic model establishes the relationship between states, and therefore, inter-state correlations are computed and maintained. Hence, when a measurement for a single state is received, all correlated states stand to benefit.

There are, however, applications where information about correlation is incomplete. For example:

1. If, contrary to Kalman Filter assumptions, the measurement and or process noise are not *white.*

2. Dynamical model is uncertain, introducing unmodeled correlations.

3. Linearization errors (e.g. for covariance propagation) introduce unknown correlations.

Traditional solutions to this problem included state-augmentation and inflating process and or measurement noise [9]. State augmentation involves adding a new state to the system state-vector to effectively model the correlated quantity. For example, if the measurement is corrupted by both additive-noise and a time-varying bias, it is common to model the time-varying bias as a Markov process and augment it to the state vector. While this works very well for some applications, it can become prohibitive when the number of potentially correlated quantities is large. This can lead to a state vector that is too large and unwieldy in practical implementation.

The second solution is what is commonly known as tuning the filter. It involves changing the assumed error statistics to account for neglected/unknown correlations. When the neglected quantities are relatively small, such inflation may work quite well in practice. It is, however, partially ad hoc. Furthermore, in applications where the neglected correlations are potentially large, this approach may not work. One such example is cooperative navigation, where significant inter-vehicle correlation is the natural result of the information sharing and inter-vehicle aiding measurements [7]. Hence, handling unknown or partially known correlated quantities for estimation in general and particularly for cooperative navigation applications, is worthy of renewed attention.

To this end this chapter has three goals. First, to describe a covariance intersection-based recursive filter suitable for decentralized cooperative navigation applications. Second, to provide a detailed derivation of general linear-unbiased estimators and show how the Kalman filter, covariance intersection filter, and bounded covariance inflation filter appear, based on assumptions about the presence of correlations. Lastly, to provide a detailed discussion describing the properties of CI/BCInf filters and practical details important for designing CI/BCInf estimators. These three goals are fulfilled by the three sections of this chapter. As part of the last section a proposed covariance normalization step, important to the proper application of CI/BCInf, is motivated and presented. Note that the discussion of CI in the first section excludes any mention of the proposed covariance normalization step.

4.2 Covariance Intersection Filter

Covariance intersection was introduced in [17] and presents a consistent method to fuse estimates of the same quantities, when the correlation between the estimates are unknown. A generalized method to handle fusion of partial information, as is the case of fusing relative measurements (e.g. range or bearing) with existing own state estimates, was presented by [19]. The general CI equations are discussed in some detail in [28] and [52].

In this section we develop a CI-based recursive filter which fuses information between an *a priori* state estimate $\hat{\mathbf{x}}^-$ and measurement $\mathbf{y}$ where the correlation between them, ρ_{xy}, is unknown. The filter generates an unbiased estimate $\hat{\mathbf{x}}^+$ and a covariance $\widehat{P}^+$ which is an overbound of the true covariance P^+ (i.e the covariance that would have been computed if ρ_{xy} was known). In mathematical terms, the CI-based filter estimates are:

$$E\left\{\hat{\mathbf{x}}^+\right\} = \mathbf{x} \tag{4.1}$$

$$\widehat{P}^+ > P^+ \tag{4.2}$$

Note that the inequality above is a matrix inequality. In practical terms it implies that the error ellipse/hyperellipsoid for $\widehat{P}^+$ overbounds that of P^+ (i.e. P^+ is contained within $\widehat{P}^+$).

When the correlation between the prior state estimate and the measured quantity is known then one may proceed to use the standard Kalman Filter measurement update equations. However, when the correlation is non-zero and unknown then CI may be used. By CI the consistent estimate $(\hat{\mathbf{x}}^+, \widehat{P}^+)$ is given as:

$$\begin{aligned} \widehat{P}^+ &= \left(\omega\left(\widehat{P}^-\right)^{-1} + (1-\omega)H^T R^{-1} H\right)^{-1} && (4.3) \\ K &= (1-\omega)\widehat{P}^+ H^T R^{-1} && (4.4) \\ \hat{\mathbf{x}}^+ &= \hat{\mathbf{x}}^- + K\left(\mathbf{y} - H\hat{\mathbf{x}}^-\right) && (4.5) \end{aligned}$$

where $\omega \in [0,1]$ is a scalar optimization parameter selected to minimize the *a posteriori* covariance $\widehat{P}^+$ in some sense (e.g. trace or determinant). In order to apply Equations 4.3-4.5, pseudocode for a system time and measurement update is presented:

1. Propagate state mean $\hat{\mathbf{x}}^+(t_{k-1})$ and covariance $\widehat{P}^+(t_{k-1})$ to t_k. The states will be propagated using the non-linear dynamic equations and covariance is propagated using Equation 3.4. The result of this step is the *a prior* state and covariance $\hat{\mathbf{x}}^-(t_k), \widehat{P}^-(t_k)$.
   This step is repeated until a measurement is available.

2. Obtain measurement $\mathbf{y}(t_k)$ with covariance $R(t_k)$ and the mapping $H(t_k)$.

   - If the measurement errors are known to be uncorrelated with the state estimate $\hat{\mathbf{x}}^-(t_k)$, then use the standard Kalman Filter measurement update equations. In this way the *a posteriori* mean and covariance $\hat{\mathbf{x}}^+(t_k), \widehat{P}^+(t_k)$ are obtained and we return to *Step* 1.
     Otherwise, proceed to *Step* 3.

3. Define a scalar cost function, denoted J and parameterized by ω, using Equation 4.3:

$$\begin{aligned} J(\omega) &= \mathrm{tr}\left\{\widehat{P}^+\right\} \\ &= \mathrm{tr}\left\{\left(\omega\left(\widehat{P}^-\right)^{-1} + (1-\omega)H^T R^{-1} H\right)^{-1}\right\} \qquad (4.6) \end{aligned}$$

   where reference to time t_k was dropped for clarity. The cost is a function of the *a priori* estimate covariance, measurement covariance, and measurement mapping.

4. Use a scalar optimization routine to find $\omega^* = \underset{\omega}{Min}\, J(\omega) : \omega \in [0,1]$.
   For example, `fminbnd` in MATLAB or `fminbound` in Scipy (a Python library), both of which use a golden section search.

5. Using ω^*, compute the *a posteriori* covariance $\widehat{P}^+(t_k)$ by Equation 4.3. Then proceed to compute the optimal gain K and subsequently the *a posteriori* estimate mean $\hat{\mathbf{x}}^+(t_k)$ by Equations 4.4 and 4.5, respectively. Return to *Step* 1.

In this work the trace was used as the cost function for the minimization as this is the same cost function minimized by the Kalman and Extended Kalman Filters. Since the cost function in Equation 4.6 is convex, it has a unique minimum which is easy to determine.

To further clarify the mechanics of Step 4 consider the simple hypothetical filtering problem depicted in Figure 4.1. In this case we are given the *a priori* covariance $\widehat{P}^-$, the measurement matrix H and the measurement noise covariance R. We wish to compute the CI gain matrix. To do this we first form the cost function of Equation 4.6 using these matrices whose numerical values are given in Figure 4.1. Then we plot the cost function versus the free parameter ω to get the curve shown in Figure 4.1. A numerical minimization (or visual inspection in this case) would return $\omega^* = 0.36$ as the choice that minimizes the trace of the of the *a posteriori* covariance $\widehat{P}^+$. Finally, using Equation 4.4 we calculate the gain matrix K associated with this value of ω.

To better understand this minimization step, it is informative to consider the cases when ω is 1 or 0. For a measurement update at time t_k, when $\omega = 1$ the CI equations collapse to:

$$\begin{aligned} \widehat{P}^+ &= \widehat{P}^- \\ \hat{\mathbf{x}}^+ &= \hat{\mathbf{x}}^- \end{aligned}$$

and if $\omega = 0$, then the *a posteriori* CI result becomes:

$$\begin{aligned} \widehat{P}^+ &= \left(H^T R^{-1} H\right)^{-1} \\ \hat{\mathbf{x}}^+ &= \left(H^T R^{-1} H\right)^{-1} H^T R^{-1} \mathbf{y} \end{aligned}$$

In the first case, the relative measurement is entirely ignored and the *a priori* state is maintained as the best estimate. In the second case, the *a priori* state is completely disregarded and instead a mean and covariance is derived exclusively using the measurement. In fact, the resulting estimate matches the weighted least-squares solution. Unless there are sufficient measurements for the entire state to be observable, we see that certain unobservable states would have infinite uncertainty.

Thus the ω parameter creates a trade-off between these two extremes, using a scalar optimization procedure to choose ω so as to minimize the *a posteriori* covariance $\widehat{P}^+$.

All other decentralized estimators discussed in prior work have tried to mitigate the effect of ignoring the cross-correlations. In contrast, CI based cooperative navigation by design takes into account the fact that the inter-vehicular correlations are unknown.

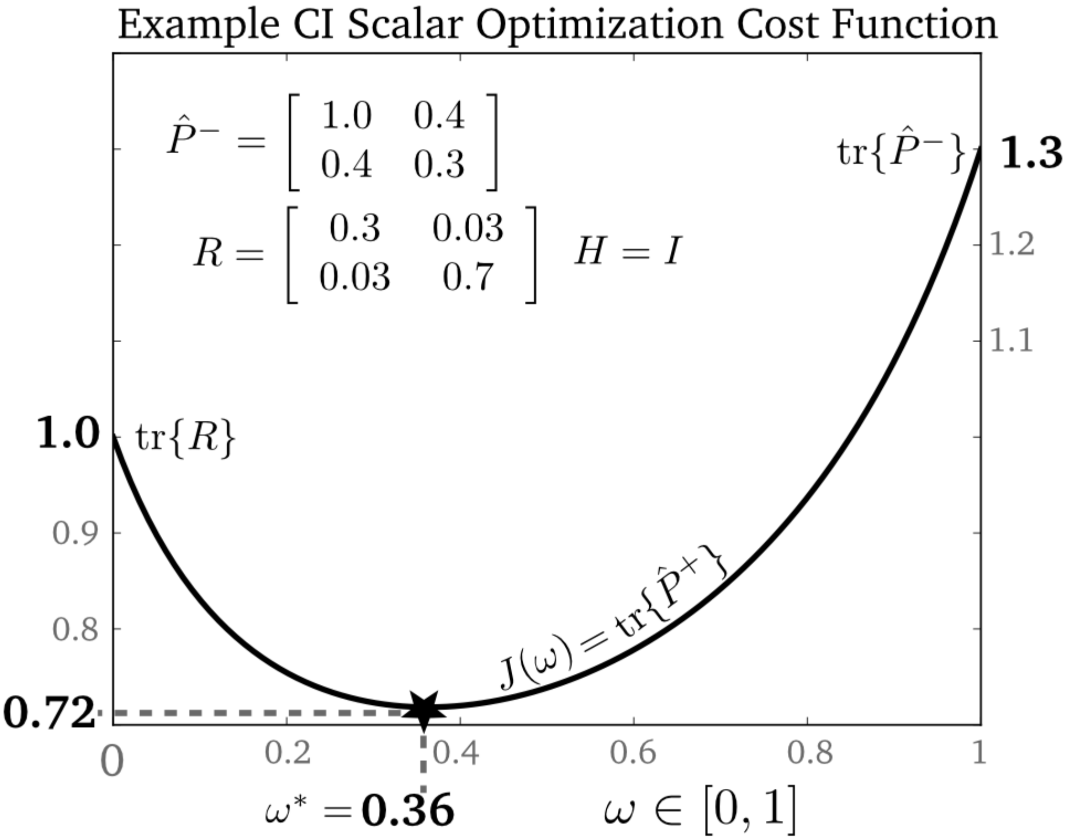


Figure 4.1: A minimization routine should select the optimal ω for each CI measurement update.

4.3 Kalman Filter and Correlated Measurements

The previous section presented a CI-based estimator suitable for decentralized cooperative navigation applications. Existing literature deriving CI often excludes a full derivation but instead present the equations and provides the intuition behind them based on geometric interpretations of the filter. It is difficult, however, to understand the source of the derivation and particularly the relation to existing data fusion techniques like Kalman filtering. Therefore, in this section we derive general correlated data fusion estimators. We begin by presenting three estimators, KF, CI, and BCInf, in notation similar to common forms of the standard Kalman Filter. This is helpful for seeing how the estimators are both similar and different without the confusion of dissimilar notation. Thereafter, we proceed with the unified derivation of these results.

4.3.1 Problem Statement

We start by generalizing the data fusion problem. To this end, the quantities being fused will be referred to as the prior state estimate vector and the measurement vector. The associated mean and covariance are $(\mathbf{x}^-, P^-)$ and $(\mathbf{y}, R)$, respectively. The linear mapping between the state-vector and the measurement is defined by H:

$$\mathbf{y} = H\mathbf{x} + \mathbf{v} \tag{4.7}$$

where $\mathbf{x}$ is the true state and $\mathbf{v}$ is the zero-mean measurement-noise vector with covariance R. Note that the measurement vector $\mathbf{y}$ is a direct or indirect measurement of the entire or a subset of the state $\mathbf{x}$. The objective is to update the *a priori* state estimate using the measurement to form the *a posteriori* state mean and covariance $(\mathbf{x}^+, P^+)$.

A linear unbiased estimator of the form:

$$\mathbf{x}^+ = K_x\mathbf{x}^- + K_y\mathbf{y} \tag{4.8}$$

is sought for this fusion. The unbiasedness requirement imposes that $K_x = I - K_yH$. Therefore, the estimator can be rewritten as:

$$\mathbf{x}^+ = \mathbf{x}^- + K(\mathbf{y} - H\mathbf{x}^-) \tag{4.9}$$

which is the common form for the standard Kalman Filter measurement-update equation. The subscript on K_y is dropped for simplicity and will hereafter be referred to as simply K. In the Kalman Filtering framework K is referred to as the Kalman gain. When the errors in the *a priori* state $\mathbf{x}^-$ and the measurement $\mathbf{y}$ are assumed uncorrelated, then the Kalman gain has a well known closed-form expression:

$$K = P^-H^T(HP^-H^T + R)^{-1} \tag{4.10}$$

and the *a posteriori* covariance can be computed as:

$$P^+ \quad = \quad (I - KH)P^-(I - KH)^T + KRK^T \tag{4.11}$$

The covariance update given by Equation 4.11 is valid for any gain K. This is commonly known as the *Joseph form* of the covariance update equation [53] and will be used in subsequent derivations. If the optimal Kalman gain defined by Equation 4.10 is used then the covariance update equation simplifies to:

$$P^+ \quad = \quad (I - KH)P^- \tag{4.12}$$

which is the form which appears commonly in the estimation literature.

4.3.2 Known Correlation P_{xy}

This work focuses on the possible presence of correlation between the *a prior* state $\mathbf{x}^-$ and the measurement vector $\mathbf{y}$. More precisely:

$$P_{xy} = E\{\Delta\mathbf{x}\Delta\mathbf{y}^T\} = P_{yx}^T \tag{4.13}$$

where the state error $\Delta\mathbf{x} = \mathbf{x}^- - \mathbf{x}$ is the difference between the *a priori* estimate and the true state, and $\Delta\mathbf{y} = \mathbf{v}$ is the measurement error. Knowledge of the correlation, P_{xy} is important both for computing the optimal fusion gain K and for subsequently computing the fused or *a posteriori* covariance P^+. If P_{xy} is known then the covariance update can be modified to account for the known correlation. This is most readily derived by rewriting the measurement update in Equation 4.9 as a single linear operation on the joint vector of the *a prior* state $\mathbf{x}^-$ and the measurement vector $\mathbf{y}$:

$$\mathbf{x}^+ = \begin{bmatrix} I - KH & K \end{bmatrix} \begin{bmatrix} \mathbf{x}^- \\ \mathbf{y} \end{bmatrix} \tag{4.14}$$

The associated covariance update and trace-minmizing gain K will be:

$$P^+ = \begin{bmatrix} (I - KH) & K \end{bmatrix} \begin{bmatrix} P^- & P_{xy} \\ P_{xy}^T & R \end{bmatrix} \begin{bmatrix} (I - KH)^T \\ K^T \end{bmatrix} \tag{4.15}$$

$$K = (PH^T - P_{xy})(HPH^T + R - HP_{xy} - P_{xy}^T H^T)^{-1} \tag{4.16}$$

where K was determined by setting $\frac{\partial \text{Trace}\{P^+\}}{\partial K} = 0$ and solving for K.

In this manner Equations 4.15 and 4.16 can be used to fuse *a prior* state $\mathbf{x}^-$ and the measurement vector $\mathbf{y}$ when their correlation P_{xy} is known. Notice that when P_{xy} is known to be zero these equations reduce to the standard Kalman gain and covariance update presented in Equations 4.10 and 4.11, respectively.

4.3.3 Unknown Correlation P_{xy}

What happens when P_{xy} is unknown or only a poor estimate of it is available? To answer this question, three cases will be considered. At the two extremes are known zero correlation and any-possible (unknown) correlation. In the middle is a case where the correlation is unknown but bounded. In all three cases the linear fusion equations presented previously will be applied. The difference will be in the assumed covariance statistics used and the reason for this will become clear in the upcoming derivation. Equations 4.9, 4.10, and 4.11 defined a general linear unbiased estimator derived for the case that the errors in the *a priori* state $\mathbf{x}^-$ and the measurement $\mathbf{y}$ are assumed uncorrelated (i.e. $P_{xy} = 0$). In order to handle the unknown correlation P_{xy} case we will rewrite these the these same equations, however, using assumed error-statistics, as indicated by the over-head *bar*:

$$\mathbf{x}^+ = \mathbf{x}^- + K(\mathbf{y} - H\mathbf{x}^-) \tag{4.17}$$

$$K = \bar{P}^- H^T (H\bar{P}^- H^T + \bar{R})^{-1} \tag{4.18}$$

$$P^+ = (I - KH)\bar{P}^-(I - KH)^T + K\bar{R}K^T \tag{4.19}$$

These equations can be applied for all the three cases. The three cases and the assumed error-statistics required for each case is presented in Table 4.1, where C_{yx} is the matrix of correlation coefficients, r_{max} defines a scalar bound on the matrix of correlation coefficients, and $\omega \in [0, 1]$ is an uncertainty trade-off parameter. All three terms will be further explained in the general derivation in Section 4.3.4.

Estimator	Assumption		$\bar{P}^-$	$\bar{R}$
(E)KF	No correlation	$P_{xy} = \mathbf{0}$	P^-	R
CI	Any allowable correlation	any valid P_{xy}	$\frac{1}{\omega}P^-$	$\frac{1}{1-\omega}R$
BCInf	Bounded correlation	$r_{max}^2 I \geq C_{yx}C_{yx}^T$	$\frac{\omega+(1-\omega)r_{max}}{\omega}P^-$	$\frac{1+\omega(r_{max}-1)}{1-\omega}R$

Table 4.1: Assumed statistics for general correlated data fusion.

What is clear from this presentation of these algorithms is their nearness to the standard Kalman Filter measurement equation which is designed to minimize the trace of the state error covariance matrix. The presence of unknown correlation between the *a priori* state and the measurement-noise simply requires carrying-out the measurement update using modified error-statistics. This is not unlike tuning, with the difference being that the inflation is done systematically to provide consistency guarantees.

Before proceeding with the derivation let us briefly define three important terms: true covariance, estimated covariance, and inflated covariance.

True, Estimated, and Inflated Covariance

A given random variable, for example the state vector $\mathbf{x}$, has a true covariance given by the following definition:

$$P = E\{(\mathbf{x} - E\{\mathbf{x}\})(\mathbf{x} - E\{\mathbf{x}\})^T\} \tag{4.20}$$

where $E\{\mathbf{x}\}$ is the mean. Calculating the true covariance requires perfect knowledge of both dynamic and measurement models and error-statistics. In practice the truth is unavailable, and hence we rely on an estimated mean and covariance $(\hat{\mathbf{x}}, \hat{P})$. Hereafter we use the *hat* to denote an estimated quantity. It is important to define two conditions that define an acceptable estimate. One requirement was stated previously and it was that the estimate should be unbiased. This implies that, on average, the estimated state mean will match the true state mean $(E\{\hat{\mathbf{x}}\} = E\{\mathbf{x}\})$.

The second condition is that of a consistent estimate. This requires that the estimated covariance always bound the true covariance. Mathematically that implies:

$$\hat{P} \geq P \tag{4.21}$$

where matrix size is measured in the matrix positive-definite sense. Later we will use an inflation on the estimated covariance $\bar{P} \geq \hat{P}$. This inflation of the *a priori* estimated covariance will be denoted using an over-head *bar*.

4.3.4 Derivation

A unified derivation for the data fusion estimators is now presented. The linear-unbiased estimator in Equation 4.9 can be written as a single linear operation on the *a priori* state and measurement vectors. This requires stacking the vectors such that the mean vector and joint covariance matrix are:

$$\left(\begin{bmatrix} \mathbf{x}^- \\ \mathbf{y} \end{bmatrix}, \begin{bmatrix} P^- & P_{xy} \\ P_{yx} & R \end{bmatrix} \right) \tag{4.22}$$

As was shown in Equations 4.14 and 4.15 where P_{xy} was assumed known, the associated *a posteriori* mean and covariance can be written as:

$$\mathbf{x}^+ = \begin{bmatrix} (I - KH) & K \end{bmatrix} \begin{bmatrix} \mathbf{x}^- \\ \mathbf{y} \end{bmatrix} \tag{4.23}$$

$$P^+ = \begin{bmatrix} (I - KH) & K \end{bmatrix} \begin{bmatrix} P^- & P_{xy} \\ P_{yx} & R \end{bmatrix} \begin{bmatrix} (I - KH)^T \\ K^T \end{bmatrix} \tag{4.24}$$

It is clear that computing the true *a posteriori* covariance depends on the true error-statistics P^-, R, and correlation $P_{xy} = P_{yx}^T$. While it is possible to replace the true error-statistics P^- and R with consistent estimates $\hat{P}^-$, $\hat{R}$, a problem, however, remains with handling the possibly unknown correlation P_{xy}. Our strategy will be to seek an uncorrelated covariance matrix which bounds the true joint covariance:

$$\begin{bmatrix} \bar{P}^- & \mathbf{0} \\ \mathbf{0} & \bar{R} \end{bmatrix} \geq \begin{bmatrix} P^- & P_{xy} \\ P_{yx} & R \end{bmatrix} \tag{4.25}$$

Once a choice of $\bar{P}^-$ and $\bar{R}$ are determined which satisfy this requirement, then the measurement update can be carried through using the standard filter Equations 4.17-4.19. This is because covariance consistency is maintained over linear operations. This can be easily shown using the definition of the positive-semidefinite condition in Equation 4.25. The matrix positive-semidefinite condition in Equation 4.25 implies:

$$\mathbf{a}^T \begin{bmatrix} \bar{P}^- & \mathbf{0} \\ \mathbf{0} & \bar{R} \end{bmatrix} \mathbf{a} \geq \mathbf{a}^T \begin{bmatrix} P^- & P_{xy} \\ P_{yx} & R \end{bmatrix} \mathbf{a} \tag{4.26}$$

for all $\mathbf{a}$. Thus, for the linear operation defined in Equation 4.23, it is easy to show that $\bar{P}^+$ computed using $\bar{P}^-$ and $\bar{R}$ will bound the true P^+ computed using P^-, R, and P_{xy}. Simply set:

$$\mathbf{a} = \begin{bmatrix} (I - KH)^T \\ K^T \end{bmatrix} \mathbf{b} \tag{4.27}$$

in Equation 4.26 and the same positive semi-definite condition is true for all $\mathbf{b}$.

As shown previously, the assumed estimator form is simply a linear operation on the joint state and measurement vectors. We assume the choice of $\bar{P}^-$ and $\bar{R}$ are a function of P^- and R. Initially we prove the above bounding condition in Equation 4.25, assuming the knowledge of true error-statistics. Once the form of $\bar{P}^-$ and $\bar{R}$ are determined, replacing the true error-statistics with consistent estimates $\hat{P}^-$ and $\hat{R}$ will continue to satisfy the bounding condition.

Bounding the Correlation

The goal is to pick $\bar{P}^-$ and $\bar{R}$ such that the condition in Equation 4.25 is satisfied for all allowable, but unknown, correlations P_{xy}. We restrict attention to a form which is a scaling of the original state and measurement error-statistics:

$$\bar{P}^- = S_x P^- S_x^T \tag{4.28}$$

$$\bar{R} = S_y R S_y^T \tag{4.29}$$

where S_x and S_y are the scaling matrices to be determined. The scalings must satisfy certain conditions, like full-rank, in order for $\bar{P}^-$ and $\bar{R}$ to be proper covariance matrices and of the same dimensions as P^- and R, respectively. Proving the bounding property for general scalings is difficult, hence we restrict attention to the class of scalings with a single degree of freedom:

$$S_x = \sqrt{\omega_x} I \tag{4.30}$$

$$S_y = \sqrt{\omega_y} I \tag{4.31}$$

where ω_x and ω_y are positive scalars. The bounding condition that needs to be satisfied can be rewritten using the assumed form of scaling.

$$\begin{aligned} \begin{bmatrix} (\omega_x - 1)P^- & -P_{xy} \\ -P_{yx} & (\omega_y - 1)R \end{bmatrix} &\geq \mathbf{0} \\ \begin{bmatrix} \bar{\omega}_x P^- & -P_{xy} \\ -P_{yx} & \bar{\omega}_y R \end{bmatrix} &\geq \end{aligned} \tag{4.32}$$

where $\bar{\omega}_x$ and $\bar{\omega}_y$, whose definition is clear, have been defined for simplicity. These scalar parameters must be chosen such that the positive-semidefinite condition is satisfied for all admissible, but unknown correlation $P_{xy} = P_{yx}^T$.

We invoke the Schur complement condition for proving positive definiteness. For the above bound to be positive definite requires both:

$$\bar{\omega}_x P^- > \mathbf{0} \tag{4.33}$$

$$\bar{\omega}_y R - P_{yx} \left(\bar{\omega}_x P^-\right)^{-1} P_{xy} > \mathbf{0} \tag{4.34}$$

The first condition simply implies that $\bar{\omega}_x > 0$ or that $\omega_x > 1$. In order to make sense of the second condition, however, it is informative to visit the Schur complement condition for positive definiteness of the original, complete, joint-covariance matrix given in Equation 4.22. Since we assume having a proper joint-covariance, this implies that the Schur complement positive definiteness condition is true:

$$R - P_{yx} \left(P^-\right)^{-1} P_{xy} > \mathbf{0} \tag{4.35}$$

Now $\bar{\omega}_x$ and $\bar{\omega}_y$ can be selected for the positive definiteness requirement in Equation 4.34 to match the known condition in Equation 4.35. For example, $\bar{\omega}_x = \bar{\omega}_y = 1$. In this manner, we have satisfied the bound we were seeking for all admissible correlations.

It turns out, however, it is advantageous to define a family of bounding covariances. This means, picking $\bar{\omega}_x$ and $\bar{\omega}_y$ in terms of a general parameter, and then optimizing on the parameter to give the optimal fusion. So rather than pick a bound and carry-out the trace-minimizing data fusion, we instead conduct the trace-minimizing data fusion in terms of a general parameter and finally pick the parameter to give the smallest *a posteriori* trace among the family of possibilities. This approach is visually depicted in Figure 4.2.

The parameter ω is used to define a family of bounding statistics. The standard Kalman filter measurement update equations (Equations 4.18-4.19) are applied. Finally, the parameter ω that minimizes the *a posteriori* covariance matrices is selected.

At this point the derivation is nearly complete for the CI filter (simply select $\bar{\omega}_x = \frac{1}{\bar{\omega}_y} = \bar{\omega}$). We will, however, go one step further before coming to CI as a special case.

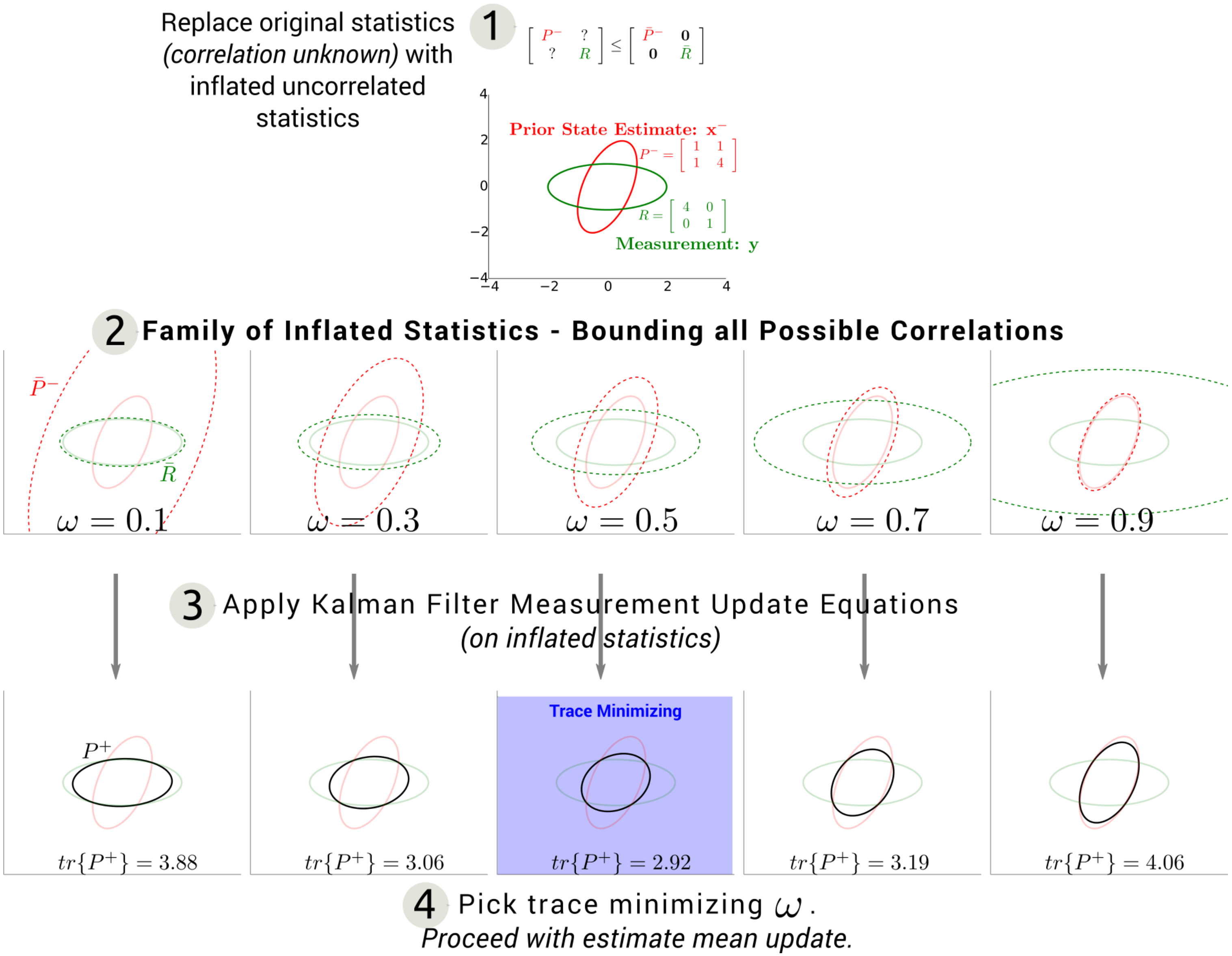

Replace original statistics (correlation unknown) with inflated uncorrelated statistics
1
$\begin{bmatrix} P^- & ? \\ ? & R \end{bmatrix} \leq \begin{bmatrix} \bar{P}^- & \mathbf{0} \\ \mathbf{0} & \bar{R} \end{bmatrix}$
Prior State Estimate: $\mathbf{x}^-$
$P^- = \begin{bmatrix} 1 & 1 \\ 1 & 4 \end{bmatrix}$
$R = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}$
Measurement: $\mathbf{y}$
4
2
0
−2
−4
−4
−2
0
2
4
2
Family of Inflated Statistics - Bounding all Possible Correlations
$\bar{P}^-$
$\bar{R}$
$\omega = 0.1$
$\omega = 0.3$
$\omega = 0.5$
$\omega = 0.7$
$\omega = 0.9$
3
Apply Kalman Filter Measurement Update Equations
(on inflated statistics)
Trace Minimizing
P^+
$tr\{P^+\} = 3.88$
$tr\{P^+\} = 3.06$
$tr\{P^+\} = 2.92$
$tr\{P^+\} = 3.19$
$tr\{P^+\} = 4.06$
4
Pick trace minimizing ω.
Proceed with estimate mean update.

Figure 4.2: Visual illustration of correlated data fusion.

Restricting Admissible Correlation

Thus far the goal of finding an uncorrelated joint-covariance matrix that bounds the original joint-covariance (i.e Equation 4.25) was conducted for "all admissible", but unknown correlation $P_{xy} = P_{yx}^T$. This is the most general case, where "all admissible" implies any amount of correlation may exist, so long as conditions for a proper joint-covariance matrix are satisfied. There are applications, however, where one may want to impose some restrictions on the "admissible" correlation. For example, in the scalar case the correlation coefficient can be used:

$$\rho = \frac{p_{xy}}{\sqrt{p_x p_y}} \tag{4.36}$$

where p_x and p_y are variance of x and y respectively, and p_{xy} is the cross covariance between x and y. Imposing that $|\rho| \leq r_{max}$, where $0 \leq r_{max} < 1$ will limit the admissible size of the correlation p_{xy}.

In the non-scalar case, a similar process can be followed, although it does not carry the same intuitive appeal as the scalar case. First, one defines the matrix of correlation coefficients:

$$C_{yx} \quad = \quad L_R^{-1} P_{yx} L_P^{-T} \tag{4.37}$$

where L_R and L_P are the Cholesky decomposition of R and P. That is, $P = L_P L_P^T$ and $R = L_R L_R^T$. Using this definition, the Schur complement condition for a proper joint-covariance given in Equation 4.35 can be rewritten as:

$$\begin{aligned} L_R L_R^T - P_{yx} L_P^{-T} L_P^{-1} P_{xy} &> \mathbf{0} \\ I &> C_{yx} C_{yx}^T \end{aligned} \tag{4.38}$$

This condition must be true for any admissible correlation. But this bound can be tightened, as was done in the case of bounding the size of the scalar correlation coefficient. More precisely, prior testing (where, for example, the true correlation P_{xy} is known) will allow one to compute C_{yx}. From this, one can determine the maximum singular value of C_{yx}, which we denote as r_{max}. Using the maximum singular value of C_{yx} we can replace Equation 4.38 with a tighter bound:

$$r_{max}^2 I \quad \geq \quad C_{yx} C_{yx}^T \tag{4.39}$$

This will define a true condition, which will inspire our choice of $\bar{\omega}_x$ and $\bar{\omega}_y$. We can rewrite the positive definiteness condition in Equation 4.34 that needs to be satisfied in terms of the matrix correlation coefficient notation:

$$\bar{\omega}_y I > \frac{1}{\bar{\omega}_x} C_{yx} C_{yx}^T \tag{4.40}$$

The choice of scaling is inspired by comparing Equation 4.39 with the Equation 4.40:

$$\begin{aligned} \bar{\omega}_x &= \frac{r_{max}}{\gamma} && (4.41) \\ \bar{\omega}_y &= \gamma r_{max} && (4.42) \end{aligned}$$

where γ is a scalar parameter. The bounding condition in Equation 4.40 now matches the true condition of Equation 4.39 so long as $\gamma > 0$ and $0 < r_{max} \leq 1$. The parameter γ is used to define a family of bounding joint-covariance matrices. Its exact value will be chosen as part of the data fusion step to minimize the size of the *a posteriori* covariance.

Note that $r_{max} = 0$ implies that $P_{xy} = 0$, i.e. the measurement noise is independent of the *a priori* state errors. In that case the standard Kalman Filter equations could be applied. If instead, $r_{max} = 1$, then any possible P_{xy} could exist, like is assumed in CI. And anything in-between will imply knowledge bounding the admissible size of the correlation-coefficient matrix.

The selection of r_{max} is done by picking the maximum singular value of the matrix of correlation coefficients C_{yx} (or C_{xy}). In general this will be different than the absolute value of the largest correlation coefficient present in the matrix. This point is important as terminology used in existing literature (e.g. [25]) may mistakenly lead one to believe that r_{max} can be selected using the absolute value of the maximum cross correlation coefficient. While this may work for certain cases, it is not generally true as will be demonstrated by a simple example.

Consider the following joint covariance matrix:

$$\left[\begin{array}{c|c} P^{-} & P_{xy} \\ \hline P_{yx} & R \end{array}\right] = \left[\begin{array}{cc|c} 1 & 1 & 8.1 \\ 1 & 16 & 18 \\ \hline 8.1 & 18 & 81 \end{array}\right]$$

The associated matrix of correlation coefficients C_{yx} can be computed using Equation 4.37:

$$C_{yx} = \left[\begin{array}{cc} 0.9 & 0.2840 \end{array}\right]$$

Hence the largest correlation coefficient is 0.9. The objective is to pick the bound r_{max} such that Equation 4.39 is known to be true. If $r_{max} = 0.9$ is selected, the bound is not satisfied:

$$\begin{aligned} r_{max}^2 I &\geq C_{yx} C_{yx}^T \\ (0.9)^2 &\ngeq 0.890 \end{aligned}$$

Instead the maximum singular value of C_{yx}, which is 0.944, can be used to correctly pick the bound r_{max}. This example demonstrates how selecting the maximum correlation coefficient is not the proper way to choose r_{max}.

Choice of Inflated Covariance

Having picked scalings which satisfy the bounding condition, we now summarize the final choice of inflated error-statistics $\bar{P}^-$ and $\bar{R}$. To do so we use the definition of $\bar{\omega}_x$ and $\bar{\omega}_y$, given in Equation 4.32, and replace $\gamma > 0$ with $\frac{\omega}{1-\omega}$ where $\omega \in [0, 1]$. This is done to match common notation in existing literature on CI.

$$\bar{P}^- = \frac{\omega + (1-\omega)r_{max}}{\omega} P^- \tag{4.43}$$

$$\bar{R} = \frac{1 + \omega(r_{max} - 1)}{1 - \omega} R \tag{4.44}$$

This matches the presented inflated error-statistics in Table 4.1 for KF, BCInf, and CI when r_{max} is 0, a general r_{max}, and r_{max} is 1, respectively. In practice one would use the consistent estimates $\hat{P}^-$, $\hat{R}$ in place of true error-statistics P^-, R and the results would not change.

This concludes the derivation for these three data fusion algorithms.

4.4 Practical Implementation Issues

Although the basic correlated data fusion equations are clean and simple, there are certain properties which can be surprising, even counter-intuitive. For an engineer perhaps familiar with the standard Kalman Filter, there are several properties that should be explained in order to help them determine how CI/BCInf might work in their application. Towards this end, this section describe three topics are important when applying CI/BCInf:

1. Inherent uncertainty trade-off
2. The number of states and effect on the trade-off
3. Necessity of units normalization
4. BCInf and choosing correlation bound r_{max}.

Topics one through three will focus on CI. The same discussion applies to BCInf so long as $r_{max} \neq 0$. The first two topics are important in explaining some of the non-obvious properties of CI/BCInf. The third topic is an important step as part of applying these algorithms which to the best of our knowledge have not been described in existing literature. The last topic will describe the challenges of picking r_{max}.

4.4.1 Uncertainty Trade-off

We begin by repeating a common geometric illustration of how CI works, but then proceed to analyze a less discussed implication. Consider a two-state estimator (x_1, x_2) with *a priori* state and covariance $\hat{\mathbf{x}}^-, P^-$. A partial observation $\mathbf{y}$ only along the x_1 direction (i.e. $H = [1, 0]$) is available with a measurement noise covariance R. There is, however, an unknown degree of correlation between the errors in the *a priori* state and the measurement. Assume the following numerical values for the covariance matrices:

$$P^- = \begin{bmatrix} 1 & 0 \\ 0 & 0.3 \end{bmatrix} \tag{4.45}$$

$$R = \begin{bmatrix} .1 \end{bmatrix} \tag{4.46}$$

$$H = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{4.47}$$

In words, the *a priori* estimate uncertainty is relatively good along the x_2 direction, however, the measurement is only available along the x_1 direction. An observation along the x_1 direction should reduce this uncertainty in x_1 but it may not be clear what the effect will be in x_2. Applying the CI fusion equations, the trace minimizing result is:

$$P^+ = \begin{bmatrix} 0.26 & 0 \\ 0 & 0.43 \end{bmatrix} \tag{4.48}$$

which occurs at $\omega = 0.69$. The associated constant-probability error-ellipses for this example are drawn in Figure 4.3.

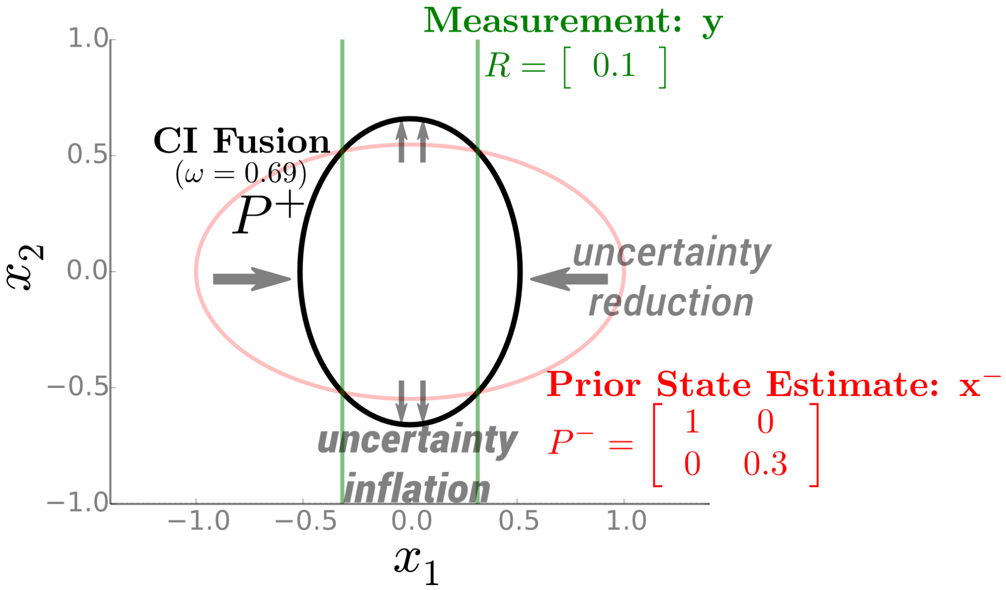


Figure 4.3: Numerical example of uncertainty trade-off inherent to CI

It is clear that the *a posteriori* uncertainty was reduced considerably along the x_1 direction. This follows our intuition as a reflection of the low-uncertainty measurement along the x_1 direction. The *a posteriori* uncertainty along the x_2 direction is more interesting, however, and in the authors opinion has received less attention. While the prior covariance was 0.3, the posterior covariance along the x_2 direction is actually larger, 0.43. This requires an explanation: Why does a measurement update add to the uncertainty?

In the standard Kalman Filter measurement update, there is neither uncertainty reduction nor inflation along a direction not directly measured if there is no correlation between x_1 and x_2. If there exists correlation between the observed and the non-observed state, there will be an uncertainty reduction. However, in this example we see the inflation in uncertainty (from 0.3 to 0.43) along the non-observed direction.

This inflation can get worse (or better) if ω is varied. The solution using the trace-minimizing $\omega = 0.69$ is shown. However, any other choice of ω is still a valid (i.e. consistent) measurement update. At the limits, as ω approaches 0, the fused covariance P^+ will converge to $(H^T R^{-1} H)^{-1}$, which minimizes the uncertainty along the x_1 direction at the cost of accepting infinite uncertainty along the x_2 direction. This is neglecting the *a priori* state estimate and instead proceeding with an estimate formed only using the measurement.

At the other extreme, if $\omega = 1$ is used, the measurement will be neglected. In this case there will be no inflation along the direction not directly measured, however, no benefit will occur to the observed direction either. The *a prior* becomes the *a posteriori* and the measurement is unused.

This sort of uncertainty trade-off can exists even for full observations. The above example can be repeated using a full observation of the state vector ($H = \mathbf{I}$). For example:

$$P^- = \begin{bmatrix} 1 & 0 \\ 0 & 0.3 \end{bmatrix} \tag{4.49}$$

$$R = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.7 \end{bmatrix} \tag{4.50}$$

Applying the CI fusion equations, the trace minimizing result is:

$$P^+ = \begin{bmatrix} 0.19 & 0 \\ 0 & 0.41 \end{bmatrix} \tag{4.51}$$

which occurs at $\omega = 0.52$. Again we observe a trade-off: significant uncertainty reduction along the x_1 direction at the cost of slight inflation in uncertainty along the x_2 direction.

This trade-off was highlighted by Arambel and Mehra in [19], however, it's our belief that the trade-off and the implications have received little attention. This is likely because many of the published examples used to illustrate the performance of CI (or its variants) have only a few states (e.g. 2-D position and heading) and have ample full-state observations available which hammer the state uncertainty covariance frequent enough to make the effect of the uncertainty trade-off negligible.

In contrast, if CI is applied to an example where measurements are less frequent, the trade-off is very noticeable and can be disconcerting. For example, in Chapter 6 we will consider an 11-state UAV dead-reckoning system operating in a GNSS-denied environment where blind application of CI-based filters will lead to unexpected results. If a (possibly correlated) position measurement from a radar is fused with the state vector using CI, one may observe the position uncertainty shrink by 100 meters and the attitude uncertainty grow by 10°!

Explanation

It is illustrative to rewrite the original CI covariance update equation in information form [53]:

$$\mathcal{I}^{+} = \omega \mathcal{I}^{-} + (1-\omega)\mathcal{I}^{y} \tag{4.52}$$

where $\mathcal{I}^{+}$, $\mathcal{I}^{-}$, and $\mathcal{I}^{y}$ are the *a posteriori*, *a priori*, and measurement information matrices, respectively and $H = \mathbf{I}$ is assumed for simplicity. Let us consider using the determinant as the minimization criteria. Since $det(A^{-1}) = 1/det(A)$, the objective of minimizing the determinant of P^{+} is equivalent to maximizing the determinant of $\mathcal{I}^{+}$. The choice of ω will be varied to maximize the size of the linear combination of *a priori* and measurement information.

At the edge cases ω is selected as 0 or 1 and no fusion occurs. Otherwise, there will exist a combination of information which results in a smaller size than either individual information matrices. This introduces the trade-off. The prior will contain one or several low-information states. The measurement will have one or more high-information states, where by high we mean greater than the information available in the *prior*. If this wasn't so, then the edge case ($\omega = 1$) would occur. Thus, to take advantage of the measurement high-information states, ω is decreased from 1 towards 0. This process will, necessarily, down-weight the existing *a priori* high information states. In other words, inflating the uncertainty along one state in order to decrease the uncertainty in another state. This is the trade-off being observed.

4.4.2 Number of States

Because of the uncertainty trade-off, the presence of additional error-states, sometimes referred to as nuisance parameters, also has interesting implications.

As mentioned earlier, many of the published examples involving CI are of low-order (e.g. 2 position and 1 heading) and ample observations (e.g. full-state measurements). Many classical navigation positioning applications (e.g. INS or INS/GPS) are of higher order. The 3 positions, velocities, and attitude parameters result in a 9-state system. Additionally, sensor error have to be estimated on-line and fed back to improve the open-loop operation. Assuming one error-state for each axis of the accelerometer and gyro and the total state vector is already 15 states. Available observations are usually never full-state, hence raising the concern whether CI will be suitable to handle fusing (possibly) correlated measurements. This is a legitimate concern, and as it will be shown, increasing the number of states can degrade the achievable performance when using CI.

In practice, it is common to begin covariance analysis using a high-fidelity high-order estimator. If CI is used, the design engineer would be surprised to discover the system is effectively operating in open-loop and the measurements are largely neglected ($\omega \approx 1$).

Given a single measurement condition, the utility of the measurement will depend on the number of states present in the estimator. In reality, additional states change the shape of the uncertainty trade-off which is conducted as part of the data fusion step. In general, the presence of additional states will reduce the derivable benefit from the same measurement. This is visually shown in Figure 4.4 where the cost-function is plotted for the same measurement, but for varying number of (not directly measured) states present in the estimator.

The number of states defines the the number of terms present in the cost function and the penalty associated with accepting the measurement information becomes heavier. So adding new states changes the shape of the cost function. This can be demonstrated for a general diagonal (for simplicity) *a priori* covariance.

Consider an N-state estimator with a diagonal covariance matrix:

$$\mathbf{P}^- = \begin{bmatrix} p_{11} & & & \\ & p_{22} & & \\ & & \ddots & \\ & & & p_{NN} \end{bmatrix} \tag{4.53}$$

Assume fusing a possibly correlated scalar measurement of the first state (i.e. $H = [1, 0, \dots, 0]$) with variance r using CI. As per the CI equations, the fused covariance would be:

$$\begin{aligned} \mathbf{P}^+ &= \left[\omega(\mathbf{P}^-)^{-1} + (1-\omega) r \mathbf{H}^T \mathbf{H}\right]^{-1} \\ &= \begin{bmatrix} \frac{r p_{11}}{\omega r + (1-\omega) p_{11}} & & & \\ & \frac{p_{22}}{\omega} & & \\ & & \ddots & \\ & & & \frac{p_{NN}}{\omega} \end{bmatrix} \end{aligned} \tag{4.54}$$

And thus the associated cost function will be:

$$\mathbf{J}(\omega) = \text{Trace}\{\mathbf{P}^+\} = \frac{r p_{11}}{\omega r + (1-\omega) p_{11}} + \frac{1}{\omega} \sum_{i=2}^{N} p_{ii} \tag{4.55}$$

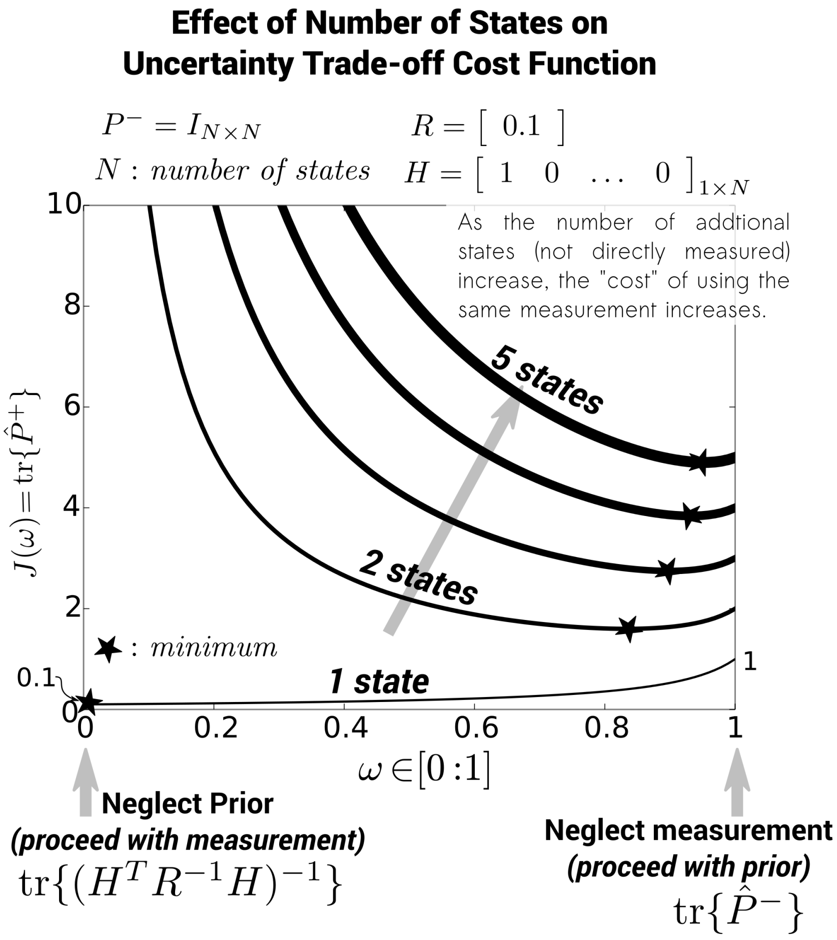


Figure 4.4: Numerical example changing cost function as number of states increase.

Hence, as N increases, the shape of the cost function changes and the minimum will shift towards $\omega = 1$ so as to avoid inflating the right-hand summation.

Note that, if justified, the fusion can be conducted at a non-optimal ω such that the observed state still benefits regardless of N. For example in Figure 4.4, using the optimal ω for $N = 2$ to do the fusion for $N = 5$. In this case the *a posteriori* variance of the observed state will be unchanged for N equal to 2 or 5. However, the penalty will be larger variances along the non-observed states for $N = 5$. This is because the fusion for $N = 5$ was done at a non-optimal ω. Therefore conducting a fusion at a non-optimal ω comes at the cost of further uncertainty inflation along the other states resulting in the non-optimal $tr\{\mathbf{P}^{+}\}$.

Hence we see built into the selection of ω is an uncertainty trade-off between state uncertainties. Whether this trade-off is *worth it* is defined by the optimization criteria. To conduct a meaningful trade-off, however, we will show that it is important to normalize the state vector covariance. Ignoring this normalization step can lead to poor selection of ω and an overall poor data fusion.

4.4.3 Units and Scaling

In [19] Mehra and Arambel introduce the concept of Weighted CI as an optional method to affect the above uncertainty trade-off in favor of states of importance. What we demonstrate here, however, is that a scaling or a normalizing step is critical to conducting a meaningful uncertainty trade-off. To skip this step, in the presence of states with mixed units, can produce variable results. To our knowledge, this normalization step has not been highlighted in existing literature.

Meters and Seconds We demonstrate the challenge by attaching physical meaning to the previous two- state example with a partial observation. Consider the problem of measuring range using the time of flight of an RF signal. In this problem we assume the states are distance and clock bias measured in meters and seconds, and that the observation is a range measurement. The *a priori* and measurement covariance matrices are:

$$P^- = \begin{bmatrix} 1m^2 & 0 \\ 0 & 0.3s^2 \end{bmatrix} \tag{4.56}$$

$$R = \begin{bmatrix} .1m^2 \end{bmatrix} \tag{4.57}$$

$$H = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{4.58}$$

For this simple example the trace of the *a posteriori* matrix P^+ can be written explicitly in terms of ω.

$$tr\{P^+\} = J(\omega) = \frac{1}{\omega + (1-\omega)10} m^2 + \frac{.3}{\omega} s^2 \tag{4.59}$$

where, as determined earlier, the trace minimization occurs at $\omega = 0.69$ and the associated *a posteriori* covariance is:

$$P^+ = \begin{bmatrix} 0.26m^2 & 0 \\ 0 & 0.43s^s \end{bmatrix} \tag{4.60}$$

So, in this example given a (possibly correlated) range measurement with $1-\sigma$ error of 0.32 *meters*, the position estimate uncertainty was halved (1 *meter* to 0.51 *meter*) and the clock-bias estimate uncertainty was inflated from 0.55 *seconds* to 0.66 *seconds*.

Notice the units of the two terms in the cost function Equation 4.59. The first is in units of m^2 while the second is s^2. Without an appropriate units normalization this uncertainty trade-off lacks meaning. This can be further illustrated if the same example is revisited, but assuming the clock bias state is maintained in units of *milliseconds*.

Meters and Milliseconds The previous example is redone, but by using a coordinate transformation to use units of position in meters and clock bias in milliseconds. The same initial uncertainties in the new units will change the *a priori* covariance to:

$$P^- = \begin{bmatrix} 1m^2 & 0 \\ 0 & 3\times10^5 ms^2 \end{bmatrix} \tag{4.61}$$

while the measurement H matrix will not change in this case. The CI cost function in this case will be:

$$tr\{P^+\} = J(\omega) = \frac{1}{\omega + (1-\omega)10} m^2 + \frac{3\times10^5}{\omega} ms^2 \tag{4.62}$$

The cost function has changed dramatically. The cost function, which has terms with mixed units, now heavily penalizes any inflation in clock-bias uncertainty. Thus, despite having a position observation which was significantly better than the prior position estimate, the optimal CI fusion is to neglect the measurement. Trace minimization occurs at $\omega = 1$ and the *prior* simply becomes the *posterior*.

This example demonstrates the variability of results that can occur when normalization is ignored. Published example usage of CI often have states like position and orientation - to neglect scaling would mean possibly (very) different results depending on the units chosen (*m*, *cm*, *radians*, *deg*, etc...). However, many of these examples are low-order systems, like 2 position and 1 heading state, and are aided by ample full-state observations. This may mask potental problems. Applications with higher-order estimators and utilizing partial-state observations will likely suffer more from the effects of neglected normalization and mixed units. The consequences of neglected normalization is demonstrated using an 11-state UAV airspeed-based dead reckoning system in Section 6.5.3.

Normalization

The above example demonstrates the importance of normalizing the *a priori* state vector before to forming the cost function. A normalized (unit-less or uniform units) covariance should be used for picking the trace-minimizing ω. Once found, the original state and covariance matrices can be used to carry out the state and covariance update. Thus we see the normalization step is needed only to conduct a meaningful minimization (i.e. uncertainty trade-off) to determine the optimal ω parameter. A linear transformation matrix T is defined to normalize the units on the *a priori* state vector $\tilde{\mathbf{x}}^-$:

$$\begin{aligned} \tilde{\mathbf{x}}^- &= T\mathbf{x}^- && (4.63) \\ \tilde{P}^- &= TP^-T^T && (4.64) \\ \tilde{H} &= HT^{-1} && (4.65) \end{aligned}$$

Then proceed with the selection of ω using $\tilde{P}^-$, R, and $\tilde{H}$. The measurement covariance does not need to be explicitly normalized since the transformed $\tilde{H}$ captures the new relationship between the original measurement and the normalized state vector.

The choice of transformation matrix T affects the optimization result. A meaningful method to normalize the covariance matrices we propose is to use desired uncertainty sizes:

$$T = \begin{bmatrix} \frac{1}{\sigma_{1_{goal}}} & & & \\ & \frac{1}{\sigma_{2_{goal}}} & & \\ & & \ddots & \\ & & & \frac{1}{\sigma_{N_{goal}}} \end{bmatrix} \qquad (4.66)$$

In this manner the uncertainty trade-off will work in favor of states that have more uncertainty than the desired outcome.

Revisiting the previous example, if a position variance of 0.2 m^2 and a clock bias of 0.5 s^2 are the desired outcome, then we define a state transformation and work with the associated normalized *a priori* covariance matrix:

$$\begin{aligned} T &= \begin{bmatrix} \frac{1}{\sqrt{.2}m} & 0 \\ 0 & \frac{1}{\sqrt{.5}s} \end{bmatrix} \\ \tilde{P}^- &= TP^-T^T = \begin{bmatrix} 5 & 0 \\ 0 & 0.6 \end{bmatrix} \\ \tilde{H} &= HT^{-1} = \begin{bmatrix} \sqrt{.2}m & 0 \end{bmatrix} \end{aligned}$$

The trace minimizing ω now is 0.57. Using this value of ω, the *a posteriori* covariance is:

$$\mathbf{P}^+ = \begin{bmatrix} 0.20m^2 & 0 \\ 0 & 0.53s^s \end{bmatrix} \qquad (4.67)$$

Notice how the uncertainty trade-off was guided towards the goal uncertainties: Give slack if the uncertainty is better than required and instead extract information in the directions where the performance is currently below the desired standards. Notice that if the above scaling method is applied, regardless of the original problem clock bias units, the CI result would be the same. Furthermore, there is nothing preventing the use of a time-varying normalization. This is used in the UAV cooperative navigation case study.

This section showed how normalization can, and should be, used when doing the minimization step in CI.

4.4.4 Implementation

Pseudocode for a measurement update based on CI or BCInf is presented:

1. Start with the *a prior* state and covariance $\hat{\mathbf{x}}^-(t_k), \hat{P}^-(t_k)$
2. Obtain measurement $\mathbf{y}(t_k)$ with covariance $R(t_k)$ and the mapping $H(t_k)$.
3. Find the trace-minimizing parameter ω^* (reference to time t_k is dropped for clarity):
    (a) Define a normalization matrix T using Equation 4.66 and form the normalized state covariance $\tilde{P}^-$ and measurement mapping $\tilde{H}$ using Equations 4.64 and 4.65.
    (b) Form inflated error-statistics $\bar{P}^-$ and $\bar{R}$ by means of Table 4.1 using the normalized $\tilde{P}^-$ and the original R. This will be in terms of a general ω.
    (c) Express P^+ in terms of ω by Equations 4.18 and 4.19, and use this to define a scalar cost function, denoted J and parameterized by ω:
    $$J(\omega) = \text{tr}\left\{P^+(\tilde{P}^-, R, \tilde{H}, \omega)\right\}$$
    (d) Find $\omega^* = \underset{\omega}{Min}\, J(\omega) : \omega \in [0, 1]$ using a scalar optimization routine.
    For example, *fminbnd* in MATLAB or *fminbound* in Scipy (a Python library), both of which use a golden section search.
4. Reform $\bar{P}^-$ and $\bar{R}$ using Table 4.1 and the original (not normalized) $\hat{P}^-(t_k)$, $R(t_k)$, and $\omega = \omega^*$.
5. Complete the fusion by computing the fusion gain K, update the state mean $\hat{\mathbf{x}}^+(t_k)$, and compute the *a posteriori* covariance $\hat{P}^+(t_k)$ using Equations 4.18-4.19

4.4.5 BCInf and Choosing Correlation Bound

Section 4.3.4 described how knowledge on the size of the possible (but unknown) cross covariance between $\mathbf{x}$ and $\mathbf{y}$ can improve the information gained through the fusion. This bound was defined in Equation 4.39 and is specified by choosing a single scalar parameter r_{max}. As described previously $r_{max} = 0$ is equivalent to fusing $\mathbf{x}$ and $\mathbf{y}$ using the standard Kalman Filter and $r_{max} = 1$ is equivalent to CI fusion. This single parameter defines the data fusion assumptions and therefore its choice changes the behavior of the BCInf estimator. The range $0 < r_{max} < 1$ is interesting as it forms a hybrid between expected results of both KF and CI. It is useful to explore a simple demonstrative example in order to build intuition on the effect of r_{max} on the data fusion result. This will be followed by a discussion on picking r_{max} in a justifiable manner and the challenges involved with this process.

Understanding r_{max}

By way of a demonstrative example we explore how the fusion result changes as r_{max} spans the range of 0 to 1. More specifically:

1. The effect on both observed and not directly measured states.
2. How the result changes with repeated measurements.
3. The presence or lack of of inter-state correlation.

Consider a single data fusion step updating a two-state *a priori* state vector $\hat{\mathbf{x}}^-$ with scalar measurement y directly observing the first state (i.e. $H = [1, 0]$). There exists a possibly bounded but unknown amount of correlation between state estimate $\hat{\mathbf{x}}^-$ and measurement y. BCInf is used to fuse $\hat{\mathbf{x}}^-$ and y in order to form the *a posteriori* state estimate $\hat{\mathbf{x}}^+$. The exact realizations or mean values are not of interest but rather we focus on the statistics as captured by the covariance. The standard deviation of two *a posteriori* state estimates, σ_1 and σ_2 are plotted as a function of the assumed r_{max} used in the BCInf fusion. The numerical values assumed for the *a priori* and measurement covariance statistics are included in the result figures. In all cases unit variance is assumed for both *a priori* states.

Observed vs Not Directly Measured States This example is shown in Figure 4.5 and shows the hybrid nature of BCInf when $0 < r_{max} < 1$. When $r_{max} = 0$ the BCInf collapses into the standard KF measurement updated. The first state, which is directly observed, has a reduction in standard deviation from $\sigma_1 = 1$ to about 0.3. The not directly measured state uncertainty is unchanged which is to be expected since there is no prior correlation between the observed and non-observed states (i.e. $\rho = 0$). In contrast when $r_{max} = 1$ the BCInf filter matches CI and hence an unknown amount of any admissible correlation between $\hat{\mathbf{x}}$ and y is assumed. As a result the observed state uncertainty is only reduced to $\sigma_1 = 0.6$ which is about half of what it was for the EKF case. The not directly measured state, however, has an inflation in uncertainty to $\sigma_2 = 1.1$ which is consistent with the uncertainty trade-off discussion inherent to the CI fusion method. As r_{max} is reduced from 1 to 0 there is further reduction in uncertainty for the observed state as well as elimination of the uncertainty inflation for the non-observed state.

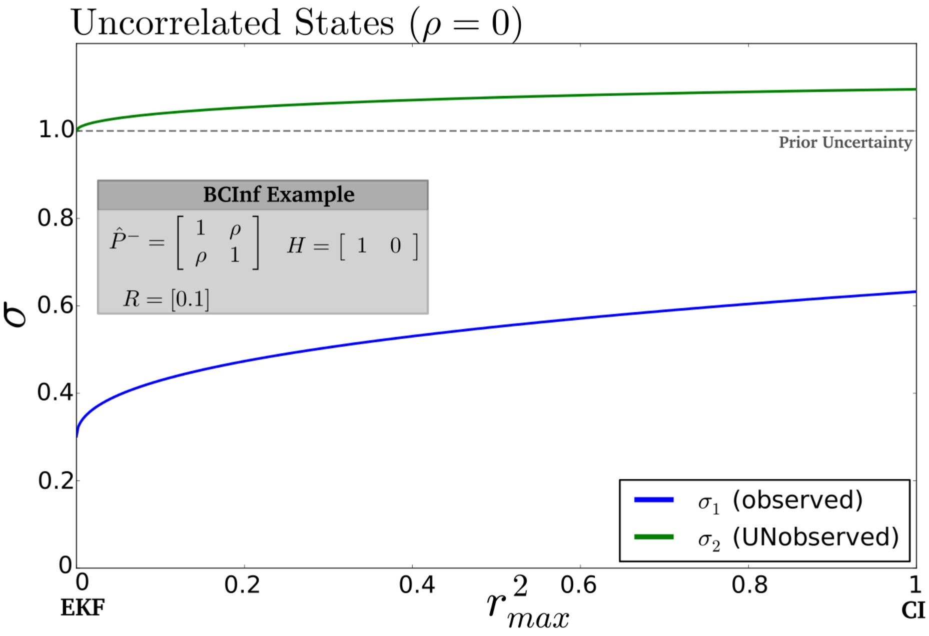


Figure 4.5: Example illustrating effect of r_{max} on both observed and not directly measured states.

Repeated Measurements The same example is considered where three repeated measurements y are fused with $\hat{\mathbf{x}}^{-}$. The result of this repeated fusion is shown in Figure 4.6 where the single fusion example shown previously is overlaid in faded colors.

The standard KF assumption of independent error between $\hat{\mathbf{x}}^{-}$ and y means that repeated observations should continue to shrink the observed state uncertainty. The non-observed state, where there is no inter-state correlation, would again be unaffected by the observations. This is visible in Figure 4.6 for $r_{max} = 0$. In contrast the CI fusion, namely when $r_{max} = 1$, has no benefit from the repeated measurements. Since any possible unknown correlation may exist between $\hat{\mathbf{x}}^{-}$ and y all derivable benefit is obtained from the single observation and any subsequent repeat observations are inconsequential until R or H changes.

The hybrid case $0 < r_{max} < 1$ is interesting. For example, consider when $r^2_{max} = 0.2$. In this case the repeated observation does reduce the uncertainty of the observed state as compared to the single observation. Consequentially, reduction of uncertainty along the observed state leads to further inflation in uncertainty along the not directly measured state. This illustrates how aggressive aiding along one observed state can be be detrimental to the estimated uncertainty of the non-observed state. Thus we see a hybrid of the properties of the standard KF and CI when $0 < r_{max} < 1$.

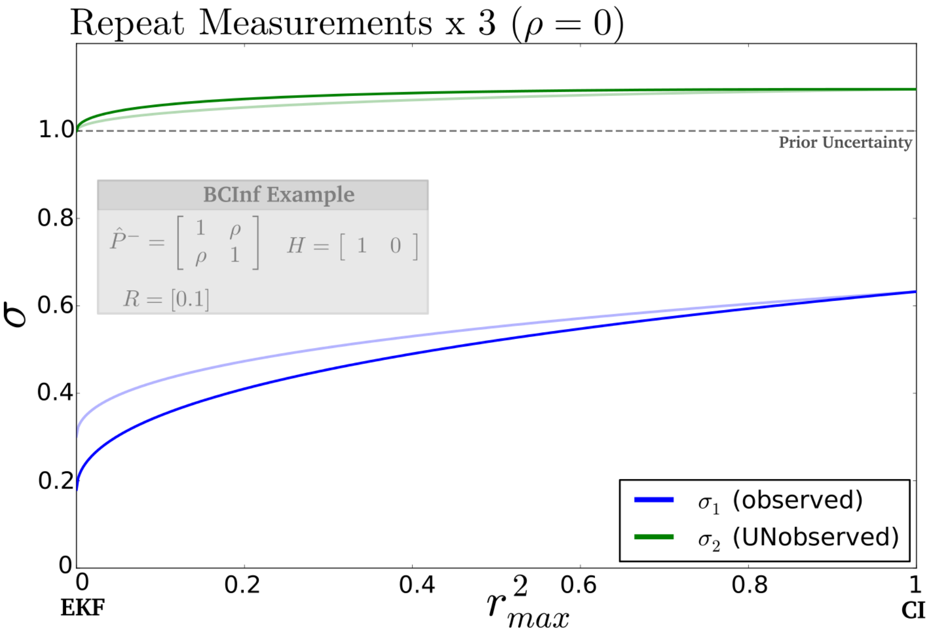


Figure 4.6: Example illustrating effect of repeated measurements, as a function of r_{max}, on both observed and not directly measured states.

Presence of inter-state correlation The presence of inter-state correlation means that direct observations of one state can benefit other non-observed but correlated states. In Figure 4.7 the first example is repeated for a range of *a priori* inter-state correlations (i.e. $\rho \neq 0$). The non-observed state uncertainty σ_2 is reduced to near 0.9 in the KF fusion ($r_{max} = 0$). When $r_{max} = 1$, however, the uncertainty trade-off associated with the non-observed state is still clearly present. Interestingly, there exists a point near $r^2_{max} = 0.3$ where the KF property of derivable benefit to the not directly measured state overtakes the uncertainty inflation property of CI. Therefore for any $r^2_{max} < 0.3$ the BCInf fusion does reduce the uncertainty in both observed and non-observed states.

Cooperative Aiding and Indirect Consequences of BCInf The single fusion step example was sufficient for capturing key properties of BCInf. More complex applications of BCInf may uncover other interesting properties. One of these will be presented in Section 7.2.5 and illustrates an indirect consequence of BCInf for non-observed states when $0 < r_{max} < 1$. In Figure 4.6 it was shown that repeated observations can lead to further inflation in uncertainty for not directly measured states. If BCInf is implemented on all vehicles in a community operating by cooperative aiding, then the choice of r_{max} has significant influence on the amount of

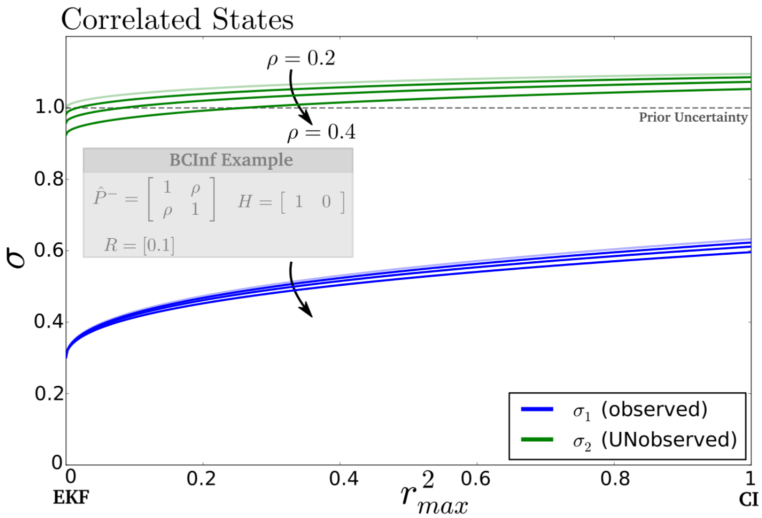


Figure 4.7: Example illustrating effect of r_{max} on both observed and not directly measured states in presence of inter-state correlation.

cooperative aiding that occurs. As r_{max} is reduced from 1 towards 0 the individual cooperative aiding between vehicles become more aggressive. As a result, neighboring vehicles have better solutions which can propagate into more aggressive aiding throughout the community. This is advantageous for the directly measured states, however, without proper guidance[1] inflation in the estimated uncertainty of the non-observed states will be dramatic. Hence as r_{max} is reduced the inflation along the non-observed state will continue to grow. Ultimately, however, there will arrive a point where the KF properties take over and the inflation along not directly measured directions will quickly vanish.

This points to an example where the choice of r_{max} has indirect effects on the overall performance.

Picking Correlation Bound r_{max}

Gaining intuition on the parameters that affect the choice of the bound r_{max} in general can be challenging. In a 2×2 system, however, r_{max} is the same as the absolute value of the correlation coefficient. Therefore studying the evolution of the correlation coefficient in a 2×2 system can be used to draw limited but useful insight for higher-order systems encountered in practice.

The scalar correlation coefficient, defined in Equation 4.36, depends on three parameters:

[1] For example, covariance normalization. Unlike the UAV results section, the automotive and commercial aviation simulation results do not use covariance normalization for CI/BCInf.

the variance of the two states p_x, p_y and their scalar covariance p_{xy}. These three are not entirely independent as proper covariance matrix requirements enforce that $|\rho| \leq 1$. In an estimation application the correlation coefficient will be a time-varying quantity whose evolution will depend on:

- dynamic state model
- process noise specification
- measurement update frequency
- measurement model
- measurement noise specification

Figure 4.8 shows the evolution of the correlation coefficient for a two state example fused using a standard KF. The variability that results from changes to measurement noise and process noise assumptions are depicted. For example, if the measurements are low-noise, then a single measurement results in nearly fully correlation $\rho \approx 1$. That would imply that a BCInf filter would need to assume $r_{max} = 1$ and therefore provide no advantage over CI. In contrast, if noisy measurements are used then $max(|\rho|)$ might not exceed 0.6. In this case a BCInf decentralized fusion may be able to provide improved performance over CI.

In applications where the parameters listed above are predictably fixed then r_{max} can be safely specified based on off-line analysis. There are, however, many applications where some or all of these parameters are time-varying and unknown. The frequency of measurement updates may depend on the vehicle trajectory and the measurement noise may depend on the measurement geometry. This is generally true for EKFs where the dynamic or measurement model may depend on the current estimate. Therefore the evolution of the correlation will vary both in-run and run-to-run. This introduces a challenge in safely specifying r_{max} and perhaps the only bound one can guarantee for a final estimator implementation is that of CI: $r_{max} = 1$.

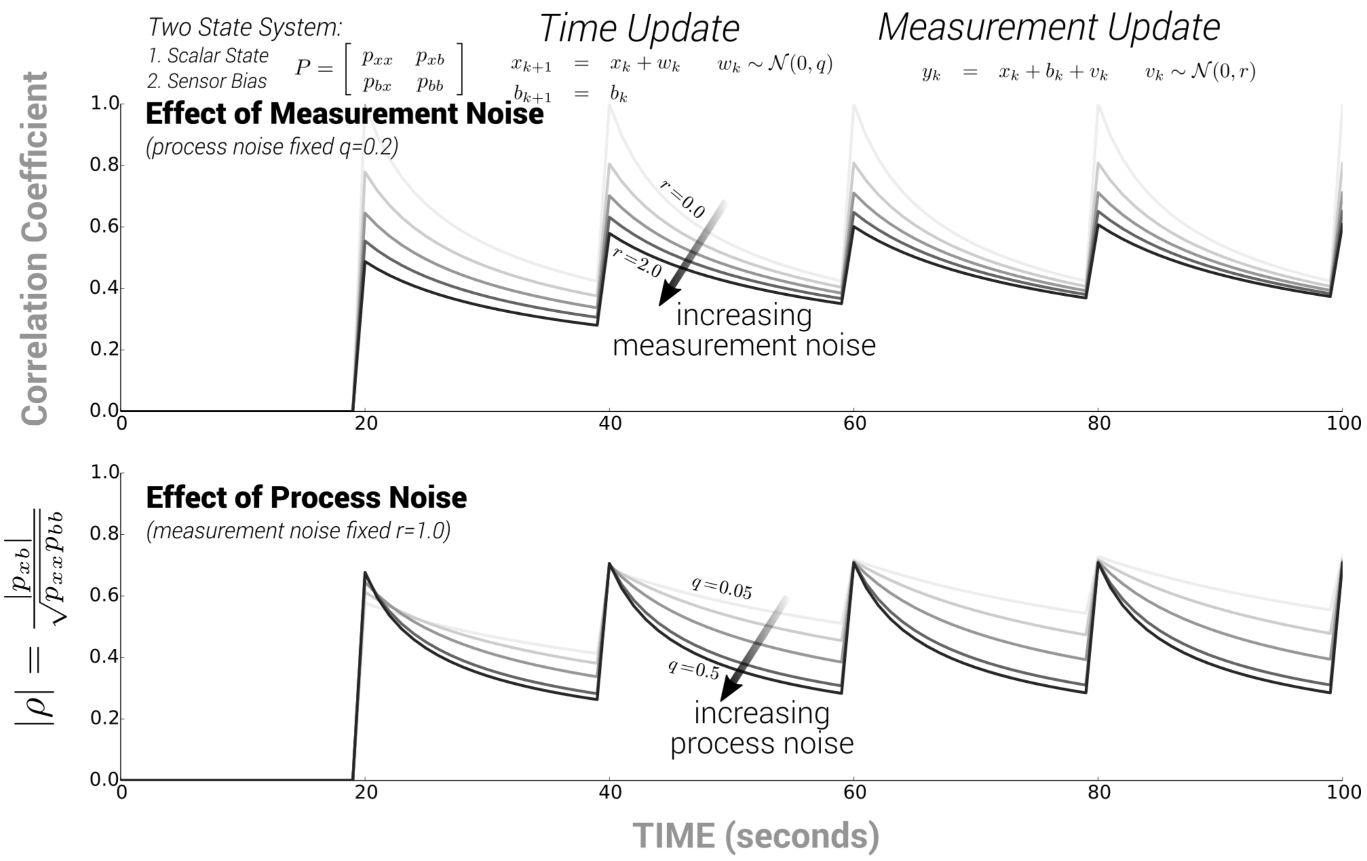


Figure 4.8: Example illustrating how r_{max} can depend on both time and scenario specific parameters like measurement and process noise specifications.

In closing it should be mentioned that there is a circular dependency here. The example in Figure 4.8 shows the evolution of the true correlation coefficient if the fusion is carried out using a KF. The bound that is sought, however, is one which bounds the true correlation for the decentralized BCInf implementation (which will have no knowledge of p_{xy}). The BCInf implementation assumes a bound r_{max} and this bound must be larger than the largest correlation that will result. Hence such analysis involves an iterative process which must be repeated until the assumed r_{max} is a tight upper-bound on the largest correlation observed. Details of two methods used to pick r_{max} for the decentralized cooperative aiding application are presented in Appendix A.

4.5 Conclusion

Three goals were accomplished in this chapter. First, a CI-based recursive filter suitable for decentralized cooperative navigation applications was presented. Second, a unified derivation of KF, CI, and BCInf filters was given. Any of the three estimators can be applied using Equations 4.17-4.19 together with the respective error-statistics in defined in Table 4.1. Finally, a detailed discussion of practical implementation issues important for applying CI/BCInf was presented. The uncertainty trade-off inherent to unknown correlation estimators was described as was the adverse effect of higher-order state vectors. A covariance normalization step was motivated and shown to be important for properly applying CI/BCInf. Lastly, the challenge associated with picking r_{max} when using BCInf filters was demonstrated. Applications with linearized estimators or variable measurement update rates, as is the case for cooperative navigation, are particularly challenging for safely specifying r_{max}. Perhaps the most conservative bound, $r_{max} = 1$ as is the case for CI, is the only safe choice.

The next three chapters will evaluate the cooperative navigation estimators described in this chapter. As part of this, the practical issues described using simple examples here will be visible in the more complex simulation results involving hundreds of vehicles navigating simultaneously, without access to GPS, by way of cooperative aiding.

Here it is noted the commercial flight and automotive application results in Chapters 5 and 7 were completed prior to the discovery of the proposed covariance normalization step. Hence, the decentralized estimators in these chapters follow the implementation described in Section 4.2 which excludes the covariance normalization step. The lessons from these studies helped with the discovery of this step, which was critical for the success of decentralized cooperative navigation for the more complex three-dimensional UAV application. Thus, the proposed covariance normalization step, as described by the implementation in Section 4.4.4, is applied only in the Chapter 6 UAV application results.

Chapter 5

Results: Aviation Applications

5.1 Introduction

In this chapter a scenario of commercial aircraft flying through the national airspace where a large geographic area is affected by a GPS outage is simulated to evaluate the proposed cooperative navigation estimation architecture. Although the sensor measurements are simulated, the aircraft trajectories are based on flight data recorded by air traffic control radars and obtained from the FAA. The first section of this chapter describes and presents the simulation results. A natural question to ask is whether the density of of aircraft in the United States are sufficient to support a cooperative aiding-based backup navigation system. To address this question the second section of this chapter presents an availability study evaluating aircraft densities in the United States. More specifically, flight data trajectories are used to determine the fraction of time that cooperative aiding could be utilized in the event of a GNSS-denial event affecting a large geographic area.

5.2 Simulation Overview

The simulated aircraft are derived from flight trajectories recorded by air traffic control radars. A description of the flight data obtained from the FAA for this analysis is given in Appendix C. The simulation is restricted to the horizontal plane and models the north and east positions and velocities, heading, and the time-varying biases for two accelerometers and a single gyro. An overview of the INS model used is given in Appendix B. While this may be an over-simplification of the real dynamics of an INS, the purpose of the analysis is to show the impact of cooperative measurement exchanges. It is judged that the slight reduction in fidelity of the INS modeling is justified for the sake of clarity demonstrating the benefit of the SPA + CI filter described previously. The flights considered in this simulation are within approximately 200 *nmi* of Minneapolis/St. Paul International Airport (MSP) from 8:00 - 9:00 AM CST (14:00 - 15:00 GMT). There are a total of 298 unique flights with the following user class distribution:

- Commercial: 161
- Air Taxi: 50
- Cargo: 7
- General Aviation: 80

The quality of inertial sensors on board each aircraft is assigned based on their user class. The commercial aircraft are assumed to be equipped with navigation grade INS, whereas the remaining air taxi, cargo, and general aviation aircraft users have tactical grade sensors. Nominally, all aircraft will aid their INS with absolute measurements from GPS. However in this simulation GPS is assumed unavailable and thus the only aiding available to the aircraft is via cooperative methods. Operating without aiding for one hour may be acceptable for navigation grade inertial navigation systems but this will not generally be the case for tactical grade users.

The relative measurements and data-link used are modeled to approximately represent TCAS and ADS-B, respectively. The range measurements are corrupted by Gaussian white noise with a standard deviation of 50 ft and the cooperation range is limited to 14 *nmi*. Each aircraft will broadcast their position and associated position covariance at regular intervals and the measurement updates are limited to every 5 seconds.

Among the 298 unique flights simulated, a single flight is selected for analysis. The selected flight, hereafter referred to as ownship, is a Cessna C414 general aviation aircraft flying from Mitchell, South Dakota to Chicago, Illinois. The trajectories of all the flights around MSP along with the highlighted path of ownship are plotted in Figure 5.1. Ownship is approximately 160 nmi south of MSP at the onset of the GPS denial and has another 36 minutes of flight to exit the GPS denied zone.

The surrounding traffic around ownship creates the opportunity for collaboration. The direct opportunities for collaboration with ownship are drawn in Figure 5.2. The blue lines connecting ownships trajectory to that of surrounding traffic shows points where cooperation is possible (i.e. the traffic falls within the 14 *nmi* cooperation range of ownship). Both spatial and temporal requirements must be met for a possible collaboration. Despite ownship being more than 160 *nmi* away from MSP, this figure illustrates the multiple opportunities for collaboration. However, the derivable benefit from surrounding traffic will depend on factors like the estimator architecture chosen, ownship uncertainty, collaborator uncertainty, and the sensor qualities of each vehicle.

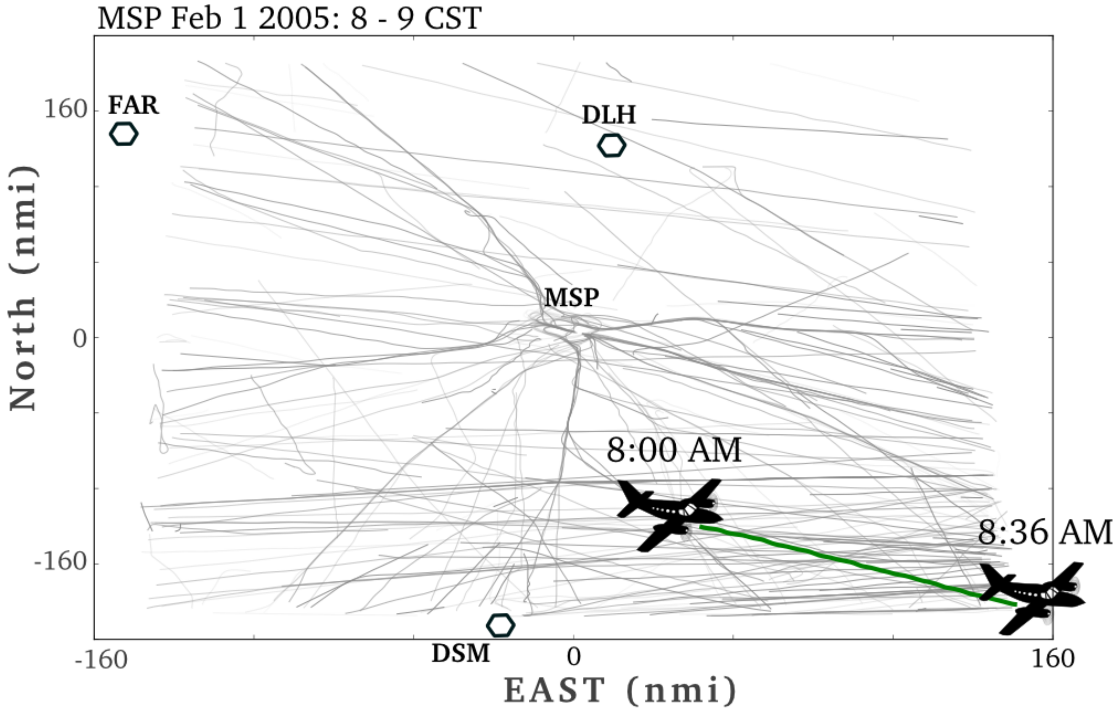


Figure 5.1: Trajectories of aircraft simulated around MSP with the true path of ownship highlighted.

5.2.1 Simulation Results and Analysis

The operational benefit of using the proposed SPA + CI decentralized estimator is evaluated by comparing the performance for several scenarios. First is the INS only case, where ownship operates entirely using an unaided INS. Second is the cooperative navigation case where a centralized estimator is available and used to aid the INS. Although a centralized estimator is impractical in practice, it serves as a good reference in simulation as the performance limit in the extended Kalman filter framework. Between these extremes we evaluate the proposed SPA + CI decentralized estimator performance.

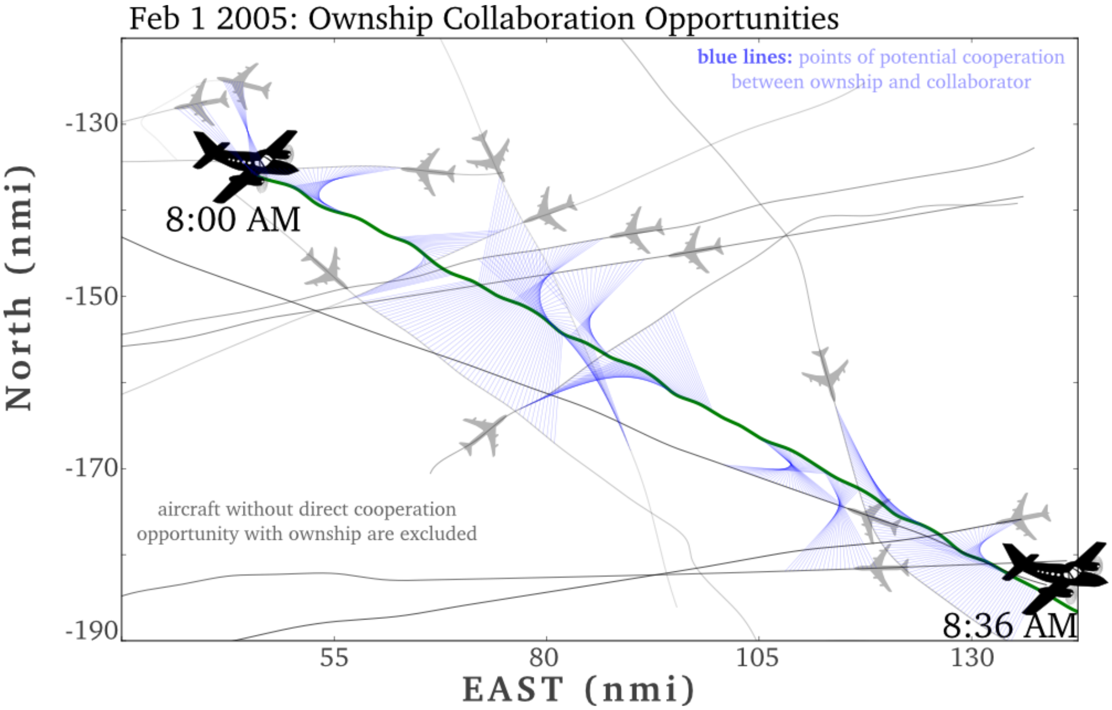


Figure 5.2: Exploded view of ownship trajectory as well as all other aircraft within direct collaboration range of ownship.

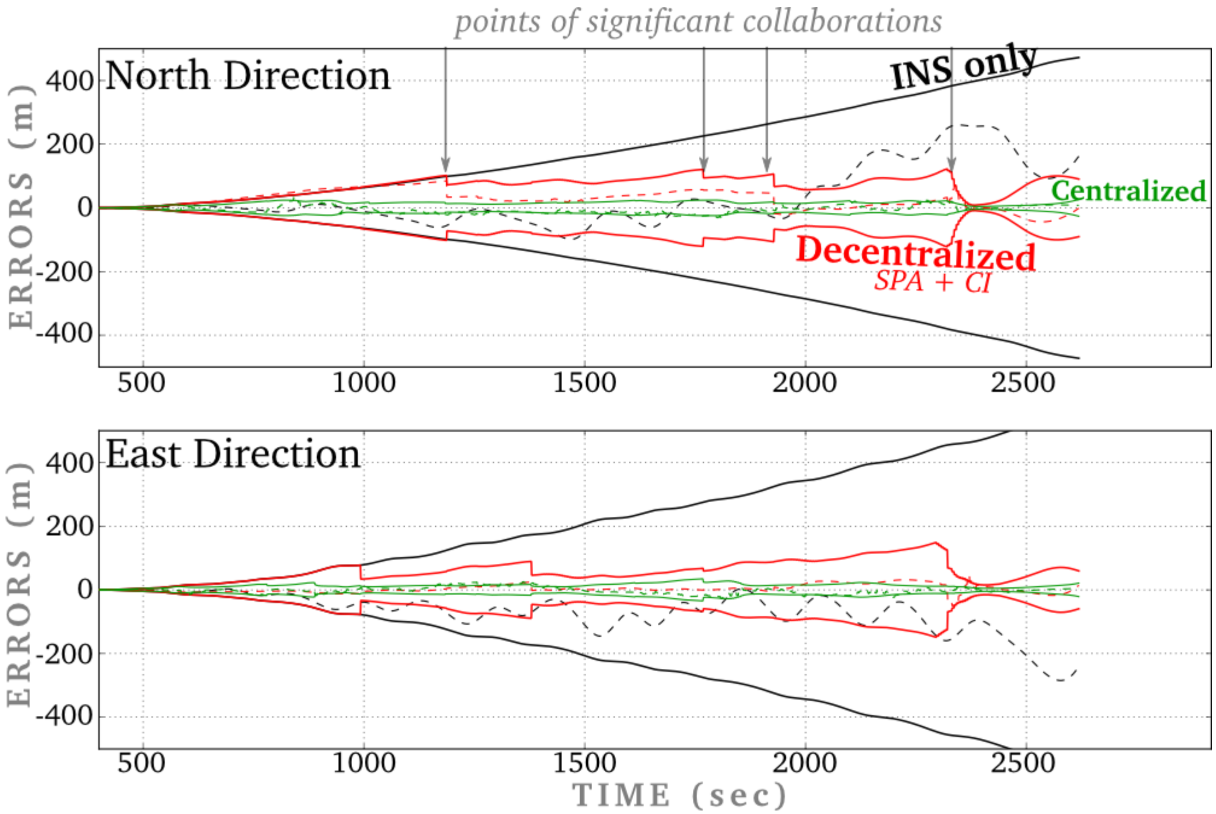


Figure 5.3: Ownship estimated 1-σ bounds for INS only, centralized, and decentralized SPA+CI estimator. The dotted lines are the computed errors for a single realization.

A comparison of all three method for the north and east error states of ownship are plotted in Figure 5.3. The INS only case shows the unaided performance for ownship. The 1-σ bound for North position uncertainty approaches 500 *meters* within 36 minutes. We note that this is optimistic for a tactical grade sensor. However, recall that we have assumed a simplified 2-D dynamics. Any aided solution should outperform the INS only case. In contrast, the centralized estimator represents the limit in performance which is achieved by relaxing bandwidth constraints and modeling all aircraft error states in a single estimator. The 1-σ bound is less than 30 *meters*, more than an order of magnitude improvement from the unaided INS only case. Although the plot is not included, the estimated 1-σ bound for the democratic decentralized estimator is actually smaller than that of the centralized estimator while the actual errors for a single realization are as large as the INS only estimated bounds. This integrity violation and unrealistic 1-σ bound is due to the presence of error loops as well as the neglected correlation between the error states of the aircrafts in the community. In contrast the SPA + CI decentralized estimator 1-σ bound is 100 *meters* and falls between that of the centralized and the INS only case. This is reassuring, since a conservative estimator that correctly accounts for the unmodeled states should have a larger uncertainty than a single estimator that models all states. It is noteworthy that the SPA + CI estimator is not limited by any source selection architecture thus making it applicable to a larger class of cooperative navigation applications.

Next we redraw Figure 5.3 with a tighter scale (exploded view) focusing on the North position state and include the estimated 1-σ bound for all estimators discussed in this work.

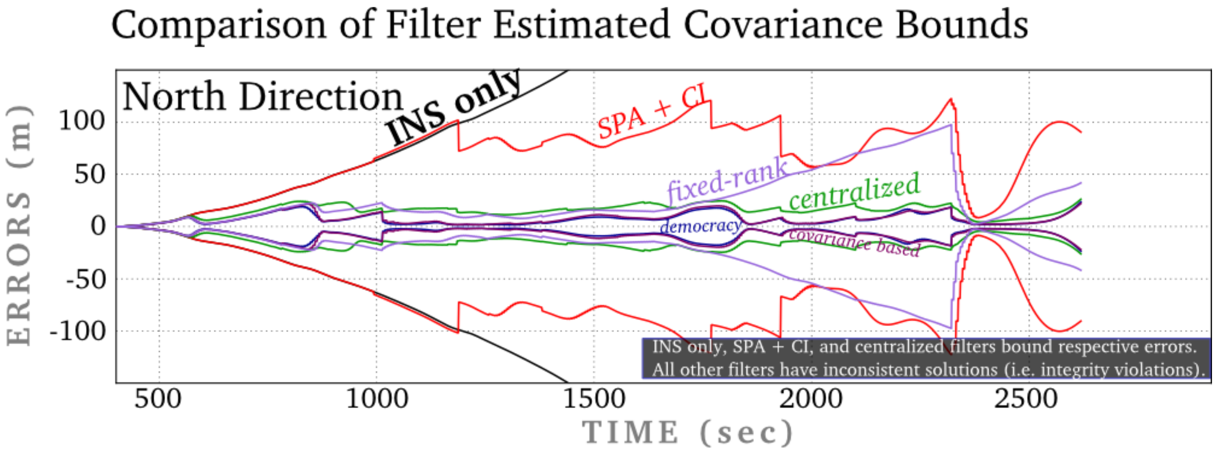


Figure 5.4: Ownship estimated 1-σ bounds for all discussed decentralized estimators. The centralized estimator and INS only result are included for reference.

Figure 5.4 compares the estimated covariance for all forms of decentralized estimators discussed. The centralized and INS only estimator results are also included as reference. The error realizations are excluded for clarity, however as pointed out on the figure the decentralized democracy, fixed-rank, and covariance based estimators are inconsistent, meaning the state errors are not bounded by the estimator covariance.

Several interesting observations are made from this figure. First, all decentralized estimators use measurement updates early on (near $t = 600\ s$), with the exception of SPA + CI. This communicates the conservative nature of SPA + CI since it is designed to assume unknown correlations between ownship and collaborators. In contrast, the other decentralized estimators are operating by neglecting the correlation. This illustrates that when using SPA + CI, collaboration opportunities may occur where no measurement update is executed.

A second observation is the long term similarity between the democratic and covariance based hierarchy results. This type of similarity is predicted as one possibility in [7]. The covariance based hierarchy is argued to behave like a democratic organization in the worst case, or at best like a fixed-rank hierarchy. The particular performance will depend on the number of rank reversals that occur; something unknown in advance due to the dynamic rank assignment.

Lastly we look at the fixed-rank estimator solution. In the fixed-rank hierarchy, each vehicle is assigned a rank depending on the sensor quality, which in this case was assigned based on the flight user class. The estimated covariance seems to largely bound the centralized covariance, however this is only the case for the later half of the simulation. Earlier on the fixed-rank covariance is smaller than that of the centralized. The rigid rank assignment only retards but does not prevent the usage of redundant information. The flow of information for a fixed-rank hierarchy shown previously in Figure 1.2 makes this clear. While there are no error-loops present in a fixed-rank architecture, there exists multiple paths for information to flow from a higher-rank to lower-rank users. To assume this information is uncorrelated is incorrect and this explains the occasional overly confident estimate covariance for the fixed-rank estimator.

Observing the errors and estimated 1-σ bounds for the other position states or other aircraft convey similar conclusions. The SPA + CI decentralized estimator is able to provide consistent estimates without resorting to process-noise tunning or requiring any restrictive source selection scheme.

5.3 Availability

How practical is the scenario described and analyzed in the previous section? Is the density of aircraft in the United States, for example, sufficient to support this? To answer this question we use historical data of flights in the National Airspace System to perform an availability study.

The availability study looks at aircraft densities near representative US airports using air traffic control flight trajectory data. The definition for availability under the cooperative navigation application is assumed to be the fraction of time three or more aircraft, referred to as collaborators, are within the communication range of the ownship aircraft. Details on the method used to compute availabilities are given next.

In navigation applications availability is loosely defined to be the fraction of time that the system is usable for navigation [51]. In the case of cooperative navigation as outlined in this paper, availability will depend on the following four parameters:

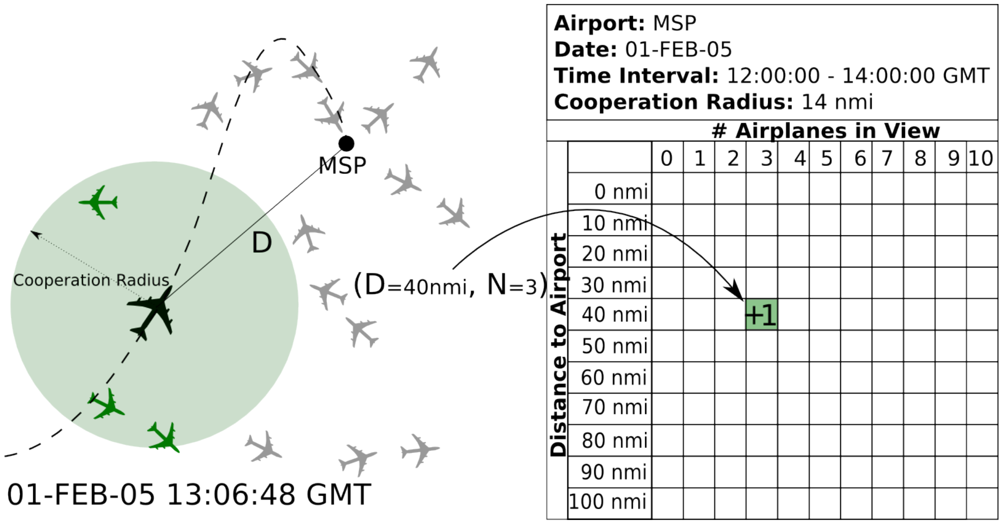
1. ***T:*** time of day,
2. User location, partially specified by:
    - ***P:*** destination (or closest) airport
    - ***D:*** distance to airport
3. ***R:*** maximum sensor/communication range, referred to as cooperation radius
4. ***N:*** number of collaborators required

Now the availability question can be specified in more concrete technical terms as follows: At time T, what is the probability that a user located D miles from airport P will have at least N potential collaborators within R miles of itself? The answer to this question depends on the specific set of parameters under consideration. In fact, the answer varies depending on parameters like time of day or the user location. In this section we use real flight trajectory data from the national air space to answer the above question.

5.3.1 Mathematical Description

The analysis procedure is described by referring to Figure 5.5. The availability analysis starts by fixing three parameters: airport P, time interval T, and cooperation radius R. These are defined in the header of the table shown in Figure 5.5. The remaining two parameters will be allowed to vary discretely. Distance to airport, D can take on values 0 to 100 nmi in 10 nmi increments. Number of airplanes in view, N, can take integer values from 0 to 10. This defines the two-dimensional state space shown in Figure 5.5.

In an iterative manner, each aircraft flying within 100 nmi of airport P during the time interval T are analyzed. At each time epoch, the distance to the destination airport (rounded to tenths place) and the number of aircraft within sensor/communication range are recorded, (D, N). For example in Figure 5.5, the shown state is ($D = 40$ nmi,$N = 3$). The two-dimensional state space table is used to count the number of occurrences of each discrete (D, N) state.



Airport: MSP
Date: 01-FEB-05
Time Interval: 12:00:00 - 14:00:00 GMT
Cooperation Radius: 14 nmi

# Airplanes in View	0	1	2	3	4	5	6	7	8	9	10
0 nmi											
10 nmi											
20 nmi											
30 nmi											
40 nmi				+1							
50 nmi											
60 nmi											
70 nmi											
80 nmi											
90 nmi											
100 nmi											

Figure 5.5: Visual snapshot in time of the procedure for populating the state space counting table, used for computing availability.

The counting table is converted into a joint probability mass function (PMF) by interpreting the joint probabilities as relative frequencies of occurrences. Each entry in the joint PMF corresponds to $p_{_{DN}} = \mathrm{P}(D = d,\ N = n)$. The resulting joint PMF is used to compute useful statistics for understanding the availability near a given airport over the specified time interval and assumed sensor/communication range.

The joint PMF can be used to compute the probability that 3 or more collaborators will be available at each discrete distance from the airport. This is accomplished by using Bayes' rule and the law of total probability via a summing procedure:

$$P(N \geq 3|D = d) \quad = \quad \frac{1}{P(D = d)} \sum_{n=3}^{\infty} p_{_{DN}}(N = n, D = d)$$

where $P(D = d)$ is the probability that an aircraft has a distance to airport d. This can be obtained through a marginalization procedure on the original joint PMF:

$$P(D = d) = \sum_{n=0}^{\infty} p_{_{DN}}(N = n, D = d)$$

Plotting the probability $P(N \geq 3|D = d)$ over the entire set of discrete distances D for a variety of time intervals provides a measure of the availability that exists near an airport.

The purpose of the availability study is to quantify the probability that there will be sufficient aircraft for utilizing cooperative navigation in the current national airspace. If it is rare for an aircraft to encounter neighboring aircraft within their mutual cooperation range then cooperative navigation would not be a practical solution. In this section we show that the prospective availability is often very high, and identify factors which are most important to the availability of cooperative navigation. Availability trade-studies at several major airports in the United States are shown next. But before that, a definition and justification of the baseline availability parameters are presented.

Baseline Availability Parameters

The number of aircraft required for cooperative navigation is $N \geq 3$, since three collaborating aircraft is the minimum required to independently estimate the position of ownship aircraft. More than 3 collaborators would be necessary if a nuisance parameter was augmented with the position states to be estimated. However as was shown earlier in simulation, even a single collaborator can be used to aid the on board INS solution. Later we show the effects of relaxing this requirement on availability.

The baseline cooperation range under consideration is $R = 14$ nmi horizontal range and $\pm 10,000 ft$ in altitude. This corresponds to the maximum range required, and altitude recommended, to be reliably covered by the TCAS MOPS [38]. Many commercial TCAS systems cover horizontal distances much farther than the required minimum, but this provides a baseline for the achievable performance under current hardware requirements.

Case Studies

The baseline percent availability of aircraft near Minneapolis-Saint Paul International Airport (MSP) is shown in Figure 5.6. The maximum availability occurs closest to the airport. At distances 0 to 20 nmi from MSP, the percent availability is relatively high, between $70 - 100\%$, for the majority of the day and early night. At a distance of 30 nmi or farther out, the availability never exceeds 40% and diminishes quickly. Flights during late nights (00:00 - 02:00 CST) have zero probabilities of simultaneously observing 3 or more planes within the 14 nmi cooperation range at MSP.

Flight User Class

The user class distribution for the national airspace flight data is shown in Figure C.1. Flights with a *commercial* user class make up for the majority of the flights and outnumber all other user class flights combined. Here we again plot the baseline percent availability of aircraft near MSP, however the analysis was restricted to only use *commercial* flights. This is shown in Figure 5.7 and can be directly compared with Figure 5.6 which included all flights recorded by air traffic. There is an approximate $10 - 30\%$ drop in availability at distances 0 to 20 nmi

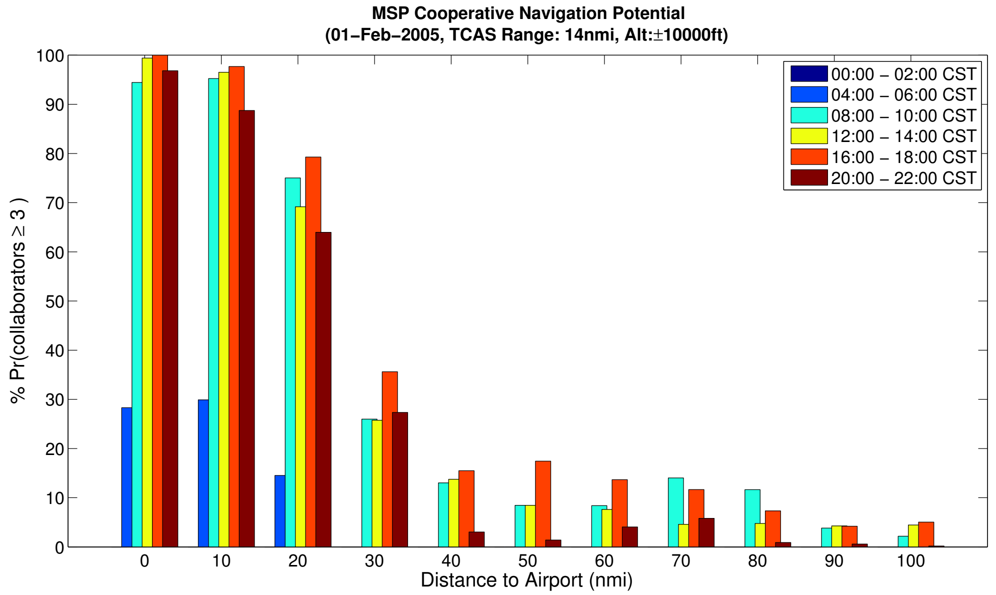


Figure 5.6: MSP availability, nominal ownship range (14 nmi) and altitude (± 10k ft)

from MSP for the majority of the day and early night. What is more noticeable is the near zero availability at distances beyond 20 nmi. Furthermore, late nights (00:00 - 02:00 CST) and early mornings (04:00 - 06:00 CST) both have zero probabilities of simultaneously observing 3 or more planes within the 14 nmi cooperation range at MSP. This suggests that both spatially and temporally, flights from particular user classes may be clustered. Thus, the collaboration opportunities between various user classes may have skewed distributions.

Increasing Horizontal and Vertical Cooperation Radius

The baseline availability in the MSP is less than 20% at distances beyond 30 nmi from the airport, and this is true for all hours of the day. Next we present the effect of increasing the horizontal cooperation range by approximately 55% to 22 nmi. The availability plot is shown in Figure 5.8. The previously poor availability beyond 30 nmi from the airport is greatly improved during daytime hours. No longer is there any time interval with entirely zero availability. An approximate 40% gain in availability is observed during the early morning hours at distances 0 to 20 nmi from the airport. Yet at distances beyond 30 nmi, the availability remains at zero for the late night and very low for the early morning hours. In summary, increasing the cooperation range can greatly improve the the availability for certain time and distance intervals, while having little effect on other intervals where traffic is relatively sparse and thus unaffected by the

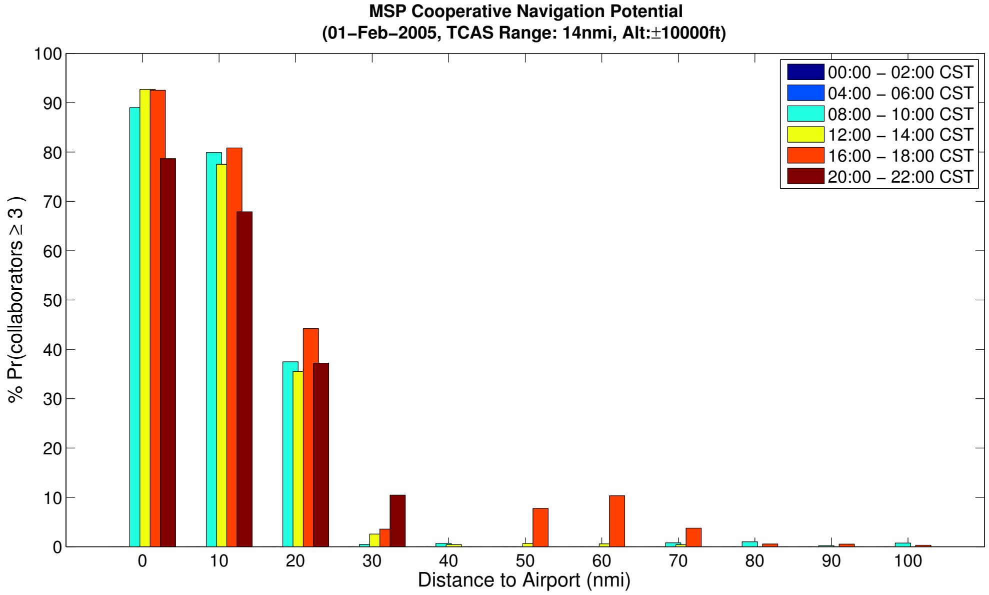


Figure 5.7: MSP availability where only flights with commercial user class are included, nominal ownship range (14 nmi) and altitude (± 10k ft)

cooperation range increase. The biggest gains are during the daytime and evening hours, while there was only minor gain to the late night early morning availability.

Increasing the baseline vertical cooperation altitude by 50% to $\pm 15,000\ ft$ had little change on the baseline availability.

Aircraft Location

Many of the same trends identified for operations in the vicinity of MSP are qualitatively visible at other locations. Naturally the quantitative results will vary depending on location.

MSP is ranked as the 10^{th} busiest airport in the United States based on 2005 FAA enplanement data. Figure 5.9 shows the baseline availability for Memphis International Airport (MEM), the 34^{th} busiest airport. Interestingly, the late night and early morning hours both have excellent availability near the airport, higher than 70% and largely comparable to the day time availabilities. This in contrast to MSP, which had zero availability for the late night hours (00:00 - 02:00 CST). Memphis International Airport is the SuperHub of Federal Express (FedEx), which operates one of the largest aircraft fleets worldwide. In fact, many of these cargo flights take place during late night and early morning hours when traditional passenger flights are rare. The signature of these cargo flights on the MEM availability is visible from this

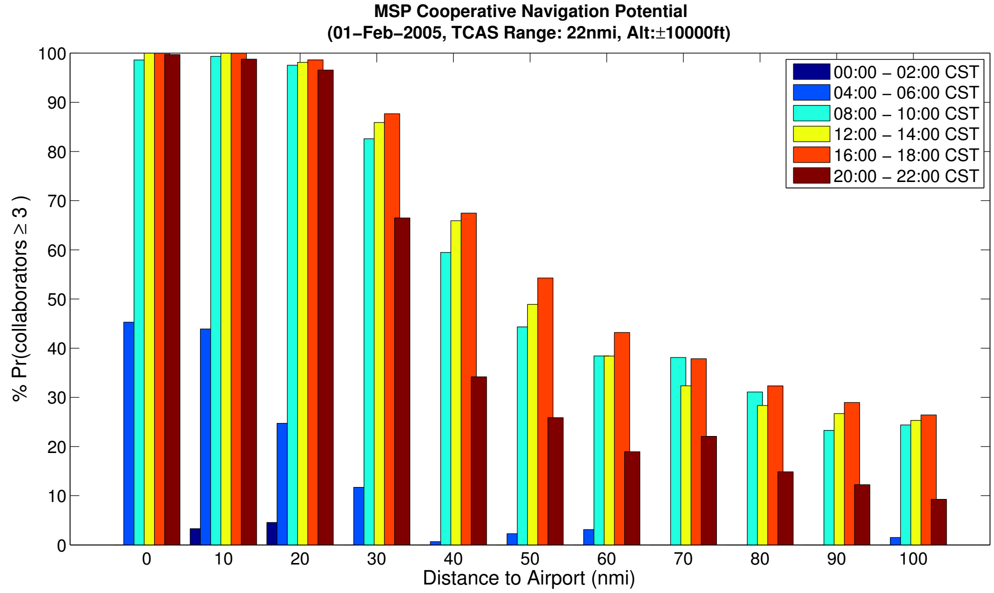


Figure 5.8: MSP availability, extended cooperation range (22 nmi) and baseline attitude (± 10k ft)

analysis.

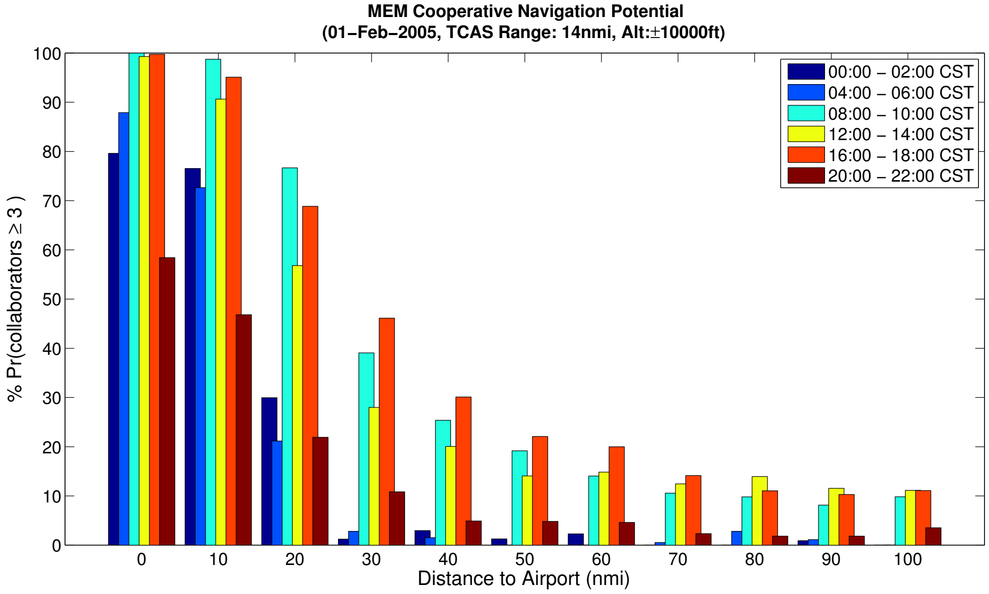


Figure 5.9: MEM availability, nominal ownship range (14 nmi) and altitude (± 10k ft)

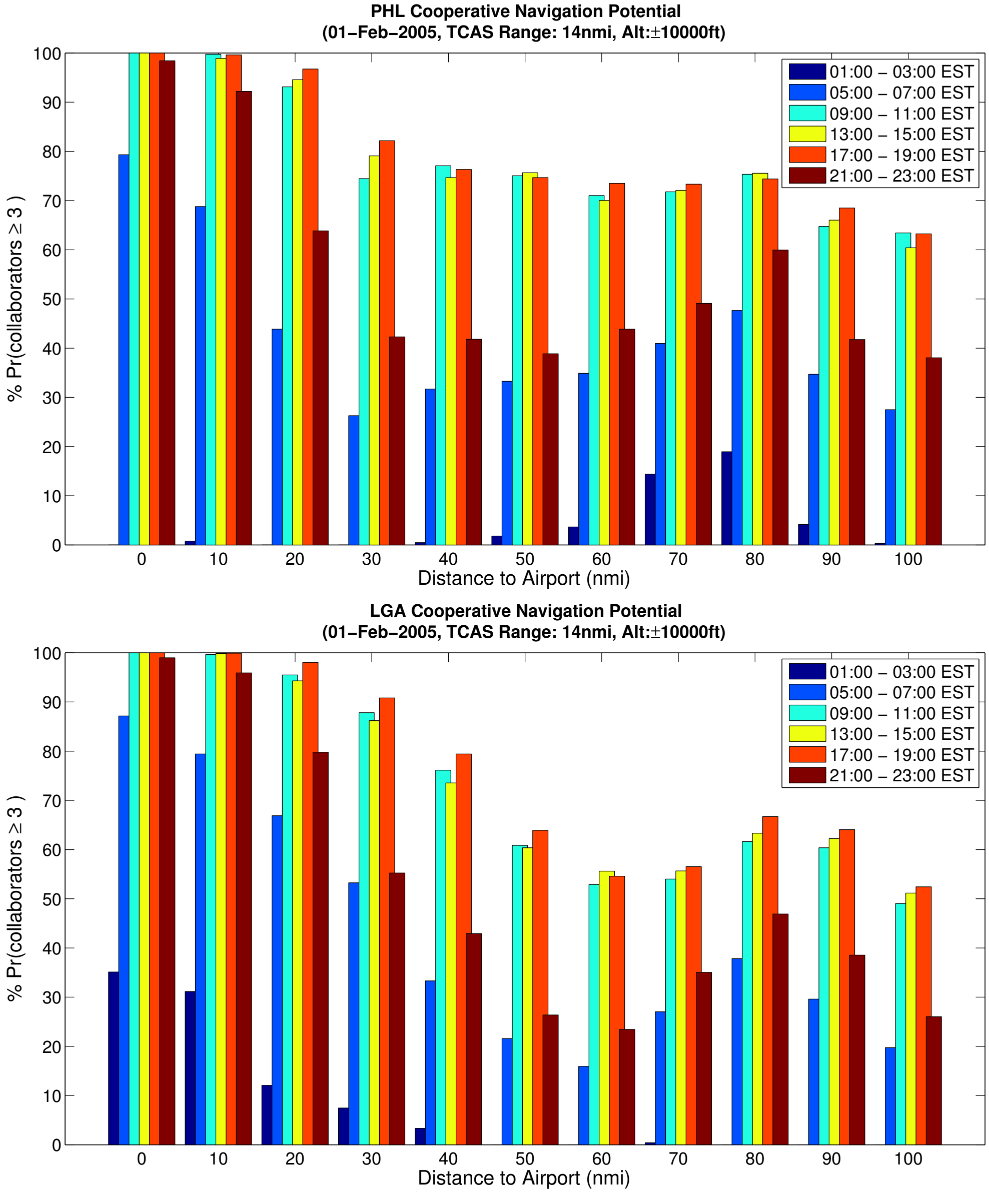


Figure 5.10: PHL (top) and LGA (bottom) availability, nominal ownship range (14 nmi) and altitude (± 10k ft)

Both the baseline MSP and MEM availability exhibited greatly diminished availability beyond 30 nmi from the airport. This turns out to be a trend but not the rule. Two interesting examples are Philadelphia International Airport (PHL) and Laguardia Airport (LGA). As shown in Figure 5.10, PHL and LGA have relatively high availability even at distances well beyond 50 nmi. A possible explanation for this availability profile around PHL is that numerous airways pass above the airport, thus introducing a different traffic density pattern. Similarly, LGA is neighbored by two other major airports: John F. Kennedy International Airport (JFK) to the southeast, and Newark Liberty International Airport (EWR) to the southwest. This again creates a unique traffic density pattern as the traffic from nearby airports become visible.

To end the discussion of location, we acknowledge the shortcomings in the single *distance to airport* aircraft location parameter. The airports are the departure and landing location of any aircraft, and that makes them attractive location anchors for the availability analysis. Yet, the traffic density near any particular airport will depend not only on the radial distance from the airport, but also largely on the angle of arrival. Current airways, largely defined by DME and VOR navigation aids, will be where most of the traffic is located. Therefore, the density will not be uniform around any particular distance from the airport. Nevertheless, the single location parameter is attractive for a first cut at visualizing the availability and quantifying the potential for using cooperative navigation under current traffic trends.

Number of Collaborators Required

Until now the availability analysis required a minimum of three aircraft be simultaneously within the cooperation range of ownship aircraft. Only then would the set of relative measurements be used to update the navigation state. This is a conservative requirement; there exists derivable benefits with even one aircraft in view. As demonstrated earlier, this is the case when using cooperative navigation to aid the on board inertial navigation system. Relaxing the number of collaborators required to only 1 aircraft, we revisit MSP availability to see the changes. Recall the baseline MSP availability never exceeded 20% beyond a distance of 30 nmi, shown in Figure 5.6. The new availability under the relaxed number of collaborators requirement is shown in Figure 5.11. The availability increases substantially and is much more uniform in both distance and time. Even the late night and early morning hours have greater than 30% availability at distances as far out as 20 nmi.

Conclusion

Availabilities between $70 - 100\%$ are observed at major airports like LGA and MSP in an area with a 30 nmi radius around the airport during morning to evening hours. While insufficient if considered exclusively, cooperative navigation has the potential to act as part of the complete

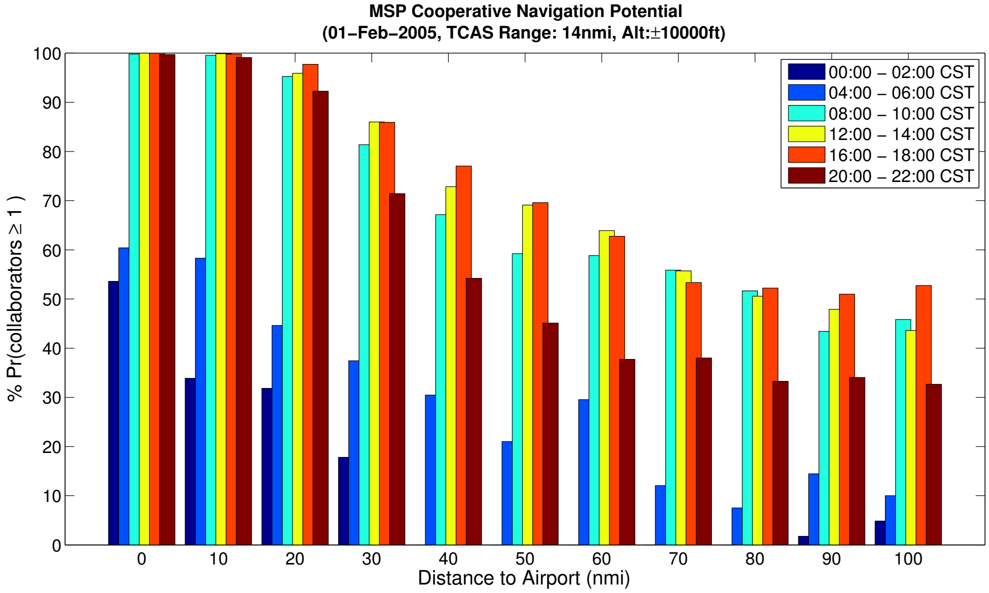


Figure 5.11: MSP availability under relaxed number of collaborators requirement ($N \geq 1$), nominal ownship range (14 nmi) and altitude (± 10k ft)

alternative positioning, navigation, and timing (A-PNT) solution in the case of a GPS outage. The on board INS aided by cooperative navigation would serve to reduce the error growth rate and thereby enable longer periods of operation under nominal separations.

5.4 Conclusion

This chapter examined the feasibility of cooperative navigation using relative range measurements and information sharing as a means to limit inertial navigation system error growth in the absence of absolute measurements. The first section presented a simulated case study of aircraft using ADS-B/TCAS to cooperatively aid their on board INS during a 60 min GPS outage in the national airspace. Cooperative aiding based on the centralized implementation resulted in an order of magnitude reduction in position uncertainty as compared to the INS only case. A decentralized SPA + CI decentralized estimator yields conservative results and has performance between that of the centralized and the INS only case. In this section it was also shown that EKF-based decentralized estimators, regardless of source selection schemes applied, result in inconsistent solutions. Hence the advantage of the SPA + CI decentralized estimator, which is flexible in that it places no restrictions on the size of the community or the source selection protocol used by the community, was demonstrated. Lastly, an availability study of cooperative navigation as proposed in this work around representative US airports was performed. Availabilities between 70-100% were common at major airports like LGA and MSP in a 30 nmi radius around the airport during morning to evening hours.

Chapter 6

Results: Unmanned Aerial Vehicles Applications

6.1 Introduction

In this chapter we analyze an experimental scenario where a community of 7 UAVs are employed to cover a 1 km^2 geographic area. Using the theory developed in Chapter 4, cooperative navigation is used to aid their on-board dead reckoning system during a 5-minutes GPS outage. An overview of the attitude and heading reference system (AHRS) and airspeed-based dead reckoning systems designed and implemented on board each UAV will be given followed by the study results. Three scenarios involving none, one, and two high flyer UAVs are investigated as are both centralized and decentralized estimator architectures. The necessity of utilizing the covariance normalization as well as the challenge of applying BCInf, both of which were discussed earlier in Section 4.4, will be shown as part of the decentralized estimator results. A preliminary analysis of integrity issues will also be presented. More specifically we will discuss the impact of misrepresentative position broadcasts on the rest of the cooperating community.

As discussed in Section 2.4 of this thesis, current small UAVs depend heavily on the availability of GPS measurements. By pulling together all previously described cooperative navigation and correlated data fusion concepts, this chapter shows how both centralized or decentralized cooperative navigation implementations can be used to overcome small UAV vulnerability to GPS outages.

6.2 Flight Data

Consider a scenario where 7 UAVs are employed to cover a 1 km^2 geographic area. The flights were carried out using three airframes from the same runway between 2011 − 2012 by the University of Minnesota UAV Lab. This study is based on playing-back the logged sensor data for each flight, which can be downloaded freely from [54]. To support the scenario under consideration, the flight data was temporally shifted to the same 10-minute time span and spatially shifted to simulate the wide-area coverage. The ground tracks and the altitude profile for the first 8 minutes of all 7 flights are shown in Figures 6.1 and 6.2, respectively. The sensor measurements logged for each flight are the same as those listed in Table 2.1, and additionally a 50 Hz INS/GPS solution computed by the flight computer on-board is logged and serves as the reference attitude and position solution for each flight.

Each UAV is assumed equipped with cross-ranging radio modems. In a protocol similar to ADSB-out requirements, each UAV broadcasts its estimated location and associated covariance at 1 Hz intervals in the play-back simulation. Additionally, the cross-ranging capability enables inter-UAV range measurements with an assumed $1-\sigma$ ranging accuracy of 5 $meters$. For purposes of this case study the range between UAVs were determined using their logged GPS position and subsequently corrupted with additive noise. All other measurements used are exactly the experimentally collected sensor data. Stated differently, all the sensor data used in this study are from actual flight tests. The only simulated data are the cross-range measurements which was based on GPS position corrupted by an error model.

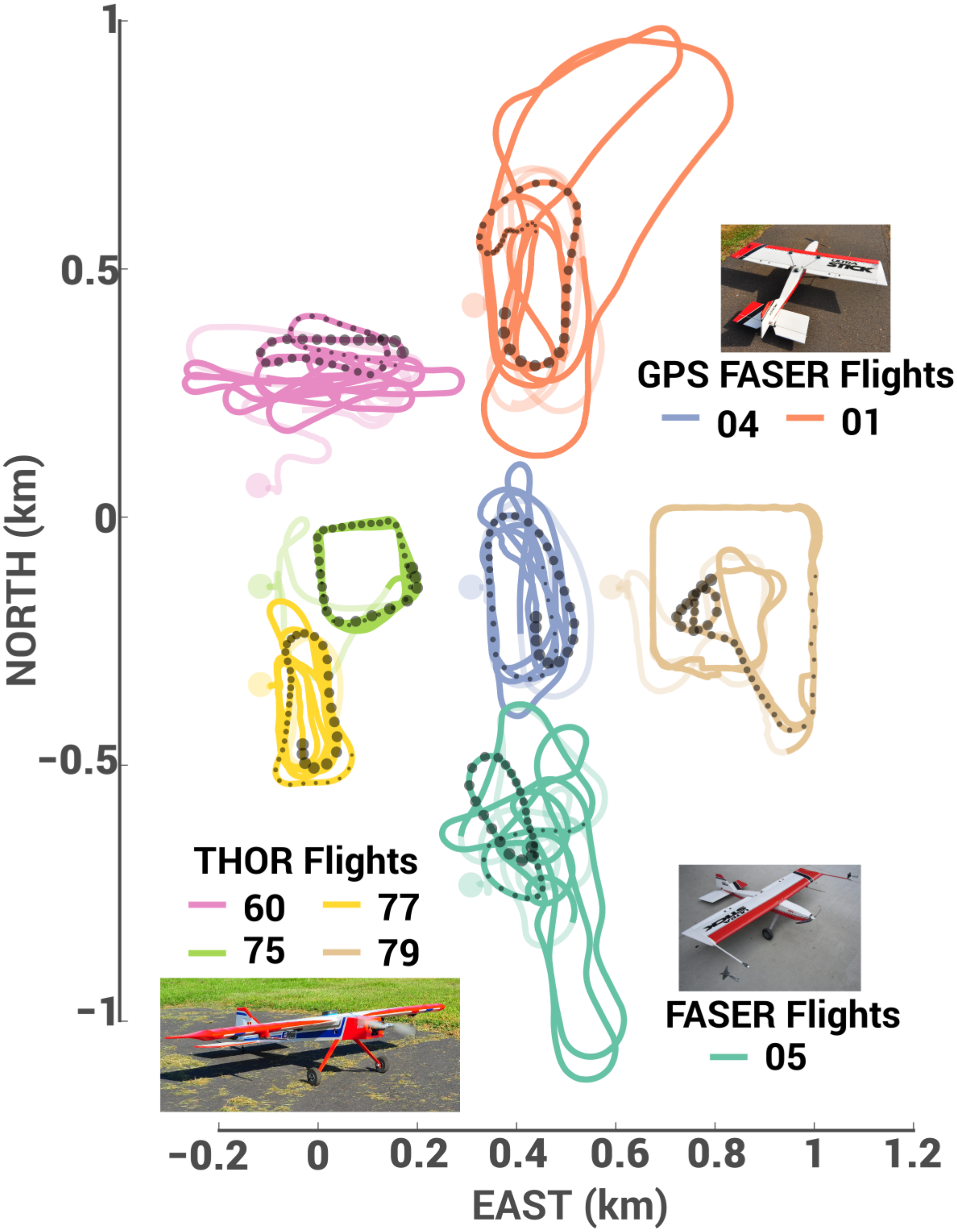


Figure 6.1: Ground track for 7 UAV flights.

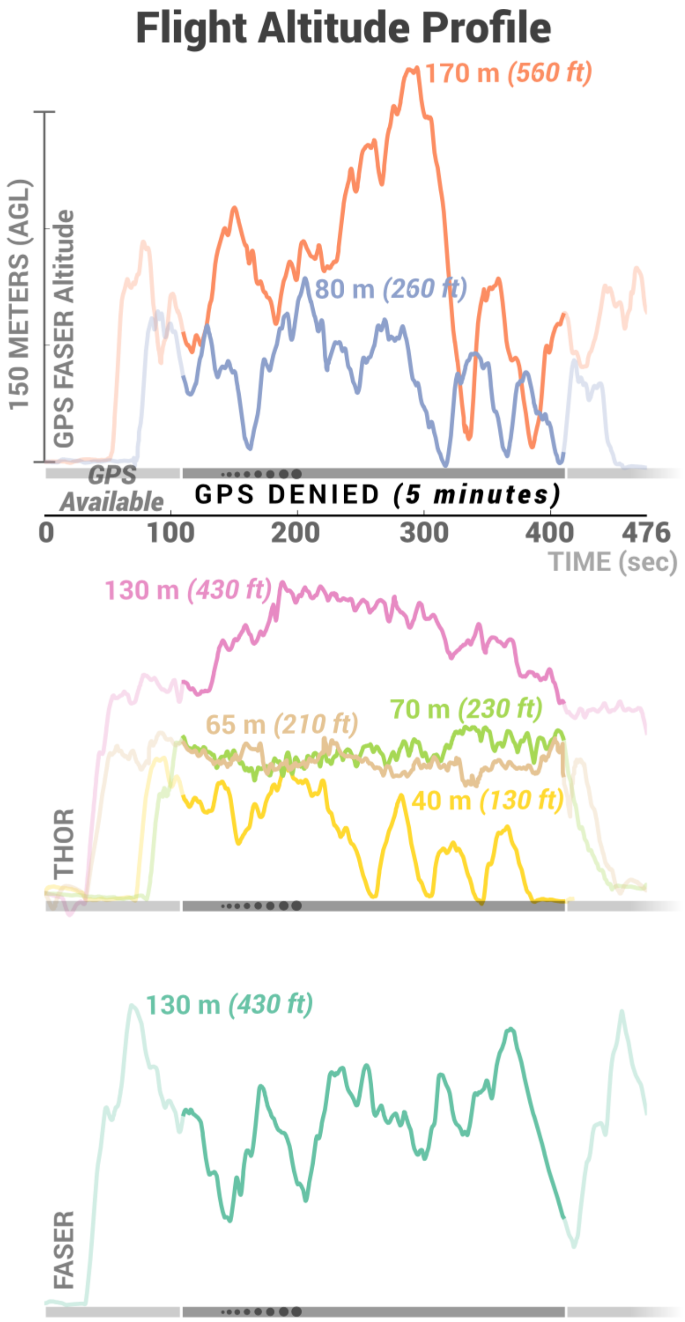


Figure 6.2: Altitude profile for 7 UAV flights.

6.3 Scenarios

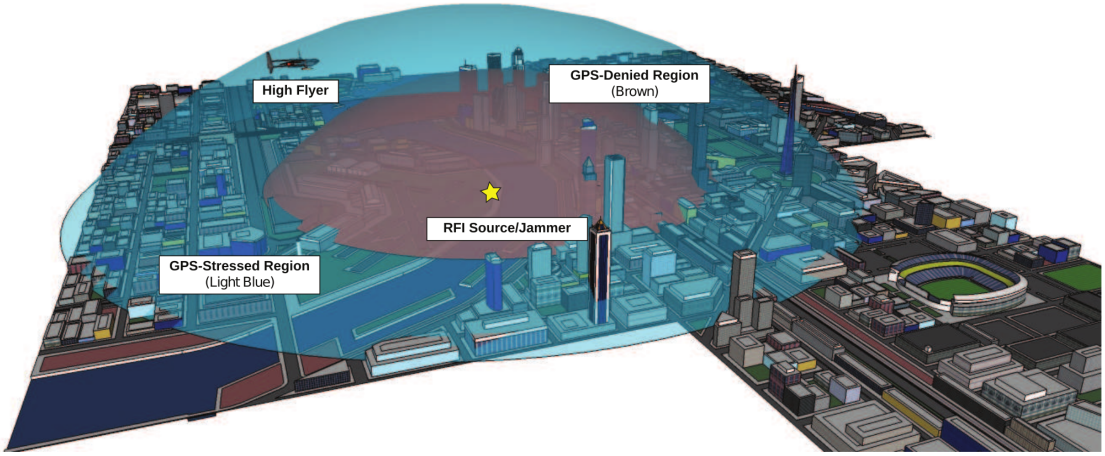


Figure 6.3: Illustration of GPS-denied scenario where large geographic area affected by radio-frequency interference [1].

The effect of radio-frequency interference on GPS in a given geographical area is to either make it difficult for GPS receivers to track the signal or prevent them from tracking it all together. An illustration of this is shown in Figure 6.3. Unless special measures are taken, in a GPS-stressed environment conventional receivers will not be able to track the signal properly. The result can be an intermittent GPS position solution or inaccurate position solutions. In a GPS-denied environment GPS is rendered completely unusable. Receivers, regardless of their type, will not be able to track the GPS signal. If the interference source cannot be located and eliminated, then the only way for a small UAV to operate in a GPS-denied environment is to revert to a non-GPS, alternate or backup navigation system like cooperative aiding.

A GPS-denied navigation scenario is considered, where a 5-minute GPS outage is simulated by withholding the GPS measurements from $t \in [110, 410\ sec]$ to the UAVs. As shown by the altitude profiles in Figure 6.1, the onset of the GPS outage occurs after all flights have reached operating altitude. The initial GPS measurements provide observability to the error states, like *wind* and *yaw-error*, which are critical to improving the quality of the subsequent dead reckoning when GPS is lost. At the onset of the GPS-outage each UAV can coast through the outage using the on-board dead reckoning system. However, without any external aiding, the position estimate will drift. This case will be considered and serves as a baseline worst-case performance. In order to reduce the rate of error-growth, we consider several scenarios based on cooperative aiding.

Three possible scenarios will be compared:

1. **No High Flyers:** all 7 UAVs are in the GPS-denied bubble and equally affected by the GPS-outage and use cooperative navigation (cross-ranging and broadcasting current position estimate and covariance) to aid on-board dead reckoning.

2. **One High Flyer:** One UAV is flying above the GPS-denied bubble and maintains GPS-lock during the outage, while all 6 other UAVs do not. The 6 UAVs without GPS use cooperative navigation to aid dead reckoning.

3. **Two High Flyers:** Two UAVs maintain GPS-lock during the outage, while 5 other UAVs use cooperative navigation to aid dead reckoning.

In this work the High-Flyer (HF) is assumed to maintain GPS-lock. This could, for example, be realized by sophisticated vector tracking GPS receivers on board that have enhanced robustness to radio frequency interference [55]. Alternatively, the HF UAVs might be equipped with a higher-grade inertial navigation system which enable significantly reduced growth-rate of position errors during the 5-minute GPS-outage.

The aiding inter-UAV measurements and information exchange introduces correlation between the error-states of the UAVs. Handling this correlation is among the challenges of making cooperative navigation a reality. Two classes of implementations, centralized and decentralized, will be considered in this work. However, before discussing each implementation and their associated implications, we first describe the on-board cascaded AHRS and airspeed-based dead reckoning system designed and implemented for the UAVs.

6.4 On-board Navigation System

A block diagram of the navigation system mechanized for each UAV is shown in Figure 6.4. The navigation system includes a 9-state AHRS cascaded with an 11-state airspeed-based dead reckoning navigation system. The AHRS attitude solution is used to resolve the airspeed measurement into the North-East-Down coordinates. The resolved airspeed equations are integrated to yield position.

6.4.1 Attitude Heading Reference System

Knowledge of a vehicle orientation or attitude with respect to a reference navigation frame is important for the on-board guidance, navigation, and control systems. The sensors and algorithms that mechanize this capability are collectively known as an Attitude Heading Reference

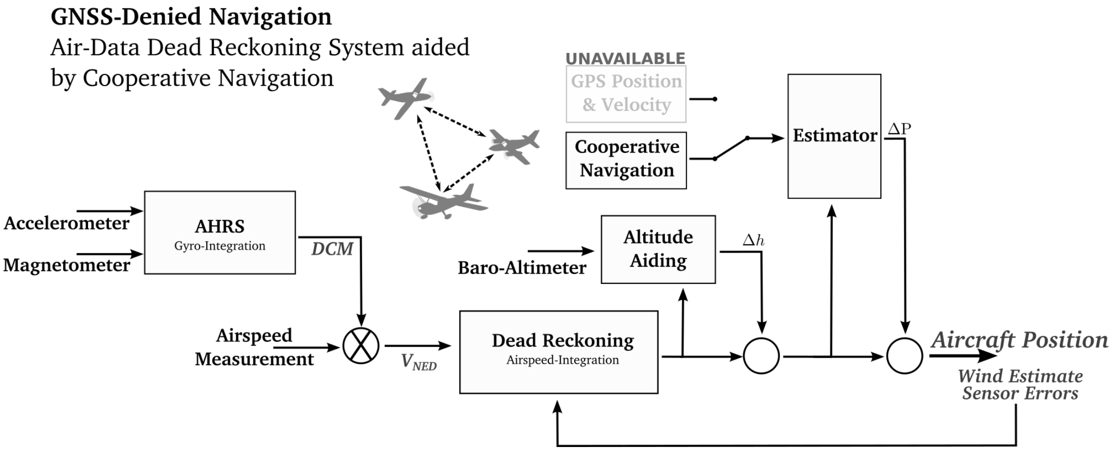


Figure 6.4: Block digram of UAV navigation system operating in GNSS-denied environment.

System (AHRS). An on-board dead reckoning navigator, like an Inertial Navigation System (INS), project body-axes sensor measurements into the reference navigation frame and subsequently integrates to determine the vehicle position. Therefore, the accuracy of the on-board dead reckoning navigation system is heavily dependent on the accuracy of the AHRS attitude estimates. It is not uncommon, for example the GPS-aided INS in [5], that both attitude and position estimates are mechanized into a single estimator. Theoretically this is the proper method since there exists coupling between the position and attitude estimates. Furthermore, proper knowledge of the position-attitude coupling will enable indirect observations of one quantity to provide observability on the error states of other quantities. For example, in the aided-INS framework, GPS position and velocity measurements can provide observability on attitude angles like yaw, pitch, and roll.

In contrast, separating the AHRS and dead reckoning estimators will, in general, be suboptimal due to the lost inter attitude-position error-state correlations. Naturally the advantage of indirect observations on correcting correlated error-states will be lost as well. There are, however, certain practical advantages to the cascaded AHRS-dead reckoning implementation which can trump the disadvantages discussed. The most notable advantage is modularity. The AHRS and dead reckoning can be designed by two separate teams or implemented using separate hardware. Furthermore, aided-dead reckoning navigators typically depend on GNSS-measurements whereas the AHRS has no requirement for GNSS [1] . Therefore the AHRS initialization time

[1] Higher quality AHRS systems use spatially dependent reference models, like earth magnetic field, gravitational vector, or earth rotation rate and therefore require a position estimate. However, this dependence is much less sensitive than the dependence of aided-dead reckoning on aiding position measurements from GNSS

can be significantly shorter than that of the aided-dead reckoning, and any loss of aiding measurements like GPS would be inconsequential.

The AHRS design was inspired by the work in [56, 57, 5] and is implemented using an Extended Kalman Filter (EKF). The AHRS determines a set of attitude parameters which specify the orientation of the aircraft with respect to a given frame of reference. Possible attitude representation include Euler angles, quaternions, or Direction Cosine Matrix (DCM). While all are valid and interchangeable, each representation has strengths and limitations. For this work we chose 3-2-1 Euler angle representations for the AHRS attitude states and DCM tilt-error representations for the AHRS attitude error-states.

The dynamic equations relate the x, y, and z body rotation-rate measurements (p, q, r) to time rate-of-change of the Euler angles, namely yaw, pitch, and roll (ψ, θ, ϕ). An on-board 3-axis gyro aligned with the aircraft body-axes will provide the body rotation-rate measurements. Hence, given this mapping the following Euler angle dynamic equations can be numerically integrated to provide yaw, pitch, and roll angles:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \frac{1}{\cos\theta} \begin{bmatrix} \cos\theta & \sin\theta\sin\phi & \sin\theta\cos\phi \\ 0 & \cos\theta\cos\phi & -\sin\phi\cos\theta \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{6.1}$$

Using the initial attitude estimates (ψ, θ, ϕ) and the 50 Hz gyro measurements (p, q, r), Equation 6.1 is numerical integrated to determine the Euler angles of the UAV. Since we are designing for a traditional fixed-wing aircraft (i.e. non-aerobatic maneuvers), the expected pitch angles will be far from the singularities in Equation 6.1 at $\theta = \pm 90°$.

Without any external aiding, sensor errors like time-varying bias or noise will cause the attitude errors to grow with time. A 9 error-state EKF was mechanized to model these errors and facilitate 1 Hz aiding measurements from accelerometers and magnetometers. The 9 error-states are:

- 3 Direction Cosine Matrix (DCM) *tilt* errors
- 3 time-varing gyro bias states
- 3 time-varying accelerometer bias state

Details of error-state dynamics and the design of the aiding measurements, particularly handling the effects of magnetic field disturbances and UAV accelerations can be found in Appendix D.

6.4.2 Airspeed-Based Dead Reckoning

The choice of an airspeed-based dead reckoning navigator over traditional low-cost INS shall be motivated. This is important because dealing with low-cost airspeed sensors is challenging and introduces the necessity of handling varying wind speeds. In contrast, the self-contained INS is unaffected by wind disturbances. This choice was motivated by three factors. First, the single integration-step from velocity to position results in a slower position error growth of airspeed-based dead-reckoning relative to the INS over extended unaided operations [58]. Second, the decentralized cooperative navigation data fusion algorithms investigated, namely CI and BCInf, degrade in aiding performance as the number of not directly observed states increase. This has to do with the conservative data-fusion that occurs when unknown correlations exist (between collaborating UAV error-states) and was discussed in Section 4.4.2. The airspeed-based dead reckoning system has fewer error states than traditional low-cost INS filters, thus making it better suited for CI/BCInf-based aiding. Lastly, the airspeed-based dead reckoning navigator was suited for a cascaded implementation where the AHRS could easily be separated from the dead reckoning operation. In this manner the impact of lost GPS aiding measurements would be restricted to the UAV position estimates and not affect the AHRS solution.

At the heart of the dead reckoning navigation system is integrating the UAV airspeed measurement. The dead reckoning navigator designed for general aviation aircraft in [58] was adapted and extended to UAV applications with low-cost sensors and increased dynamics. The dead reckoning system is based on integrating aircraft NED velocity estimates and maintains aircraft position using latitude (Λ), longitude (λ), and geometric altitude (h_g) (LLA). Estimates of the aircraft velocity are obtained by projecting the pitot probe airspeed measurement into the NED frame by means of the aircraft attitude, as determined by the on-board AHRS. The UAV velocity, with respect to the surrounding air, can be expressed in the body-axes as:

$$V_{body} = \begin{bmatrix} u & v & w \end{bmatrix}^T \tag{6.2}$$

It is assumed that the pitot probe is aligned with the UAV x-body axis, therefore u is the on-board airspeed measurement. Without further information, v and w are zero, implying zero angle-of-attack and sideslip angle. Error-states for all three axes will be introduced later and hence capture variations about these nominal values. At each epoch the yaw, pitch, and roll (ψ, θ, ϕ) estimate, as output by the AHRS, are used to form the transformation matrix R_{body}^{NED} which maps from the body axis to the NED frame [57]. This is commonly known as the *inverse direction cosine matrix* and is used to form the aircraft NED-velocity estimate:

$$V_{NED} = R_{body}^{NED} V_{body} + \begin{bmatrix} W_{NS} & W_{EW} & 0 \end{bmatrix}^T \tag{6.3}$$

Notice that V_{body} is with respect to the surrounding air mass. In contrast, V_{NED} is with respect to the Earth since estimates of the horizontal wind speeds are included in Equation 6.3. Both horizontal wind components will be estimated online. The dynamic model relating the estimated NED velocity and the LLA coordinates are:

$$\dot{\Lambda} = \frac{V_N}{R_{NS} + h} \tag{6.4}$$

$$\dot{\lambda} = \frac{V_E}{(R_{EW} + h)\cos\Lambda} \tag{6.5}$$

$$\dot{h}_g = -V_D \tag{6.6}$$

where R_{NS} and R_{EW} are the earth's north-south (meridian) and east-west (prime vertical) radii of curvature [58]. The LLA-rate equations are the basis for the UAV dead reckoning navigator. Errors in the terms defining Equations 6.4-6.6 will naturally degrade the dead reckoning performance. Error-states will be introduced as part of an 11-state estimator to model the dead reckoning errors and enable aiding measurements to be fused by means of the estimator measurement update equations.

State	Description
$\Delta N, \Delta E, \Delta D$	North-East-Down position error
W_{NS}, W_{EW}	North-South and East-West wind
$\Delta\psi, \Delta\theta$	Yaw-error and pitch-error
$\Delta u, \Delta v, \Delta w$	Body-axes velocity measurement errors
Δh	offset between *geometric* and *pressure* altitude (e.g GPS vs baro altitude)

Table 6.1: List of 11 error-states for dead reckoning system.

A list of the 11 error-states are given in Table 6.1. The error-states modeled fall into four categories:

1. **Position errors:** maintained by NED error-states
2. **Wind estimates:** 2 horizontal wind components
3. **Attitude errors:** corrections to attitude estimates from AHRS
4. **Body-axes airspeed errors:** pitot probe errors and non-zero angle-of-attack or side slip angle.

Although the position state vector is represented in LLA coordinates, the choice of NED position error-states requires defining the associated dynamics model.

$$\Delta\dot{N} = \Delta V_N \tag{6.7}$$

$$\Delta\dot{E} = \Delta V_E \tag{6.8}$$

$$\Delta\dot{D} = \Delta V_D \tag{6.9}$$

where the three V_{NED} components were defined previously in Equation 6.3. In order to apply position corrections to the LLA positions the NED position errors are converted into LLA corrections [58]:

$$\Delta\Lambda = \frac{\Delta N}{R_{NS}} \tag{6.10}$$

$$\Delta\lambda = \frac{\Delta E}{R_{EW}\cos\Lambda_0} \tag{6.11}$$

where Λ_0 is the latitude of the NED-frame origin.

Term	Description	σ	Time Constant (*seconds*)
	Wide-band Additive noise		
w_u, w_v, w_w	Noise on body-axes airspeed measurements	$(2, 1, 1)$ m/s	
w_ψ, w_θ	Noise on AHRS attitude estimates	$(2, 1)$ deg	
	Driving Noise for Error-States Modeled as First-Order Markov Process		
$w_{d_{W_{NS}}}, w_{d_{W_{EW}}}$	Wind states	5 m/s	60
w_{d_ψ}, w_{d_θ}	Correlated attitude uncertainty	$(7.6, 2.0)$ deg	$(35, 10)$
$w_{d_u}, w_{d_v}, w_{d_w}$	Body-axes airspeed errors	2 m/s	10
w_{d_h}	Altitude offset	4 m	35

Table 6.2: List of 13 process-noise terms for dead reckoning system.

Dynamical models for the error-states are used to propagate the error-state covariance. The error-states *means* are unchanged during the time-update step. As is done in the EKF, propagating the covariance requires a linear state-space model of the form:

$$\begin{aligned}
\mathbf{x} &= \begin{bmatrix} \Delta N & \Delta E & \Delta D & W_{NS} & W_{EW} & \Delta\psi & \Delta\theta & \Delta u & \Delta v & \Delta w & \Delta h \end{bmatrix}^T \\
\mathbf{w} &= \begin{bmatrix} w_u & w_v & w_w & w_\psi & w_\theta & w_{d_{W_{NS}}} & w_{d_{W_{EW}}} & w_{d_\psi} & w_{d_\theta} & w_{d_u} & w_{d_v} & w_{d_w} & w_{d_h} \end{bmatrix}^T \\
\dot{\mathbf{x}}(t) &= F(t)\mathbf{x}(t) + G(t)\mathbf{w}(t) \qquad (6.12)
\end{aligned}$$

where $\mathbf{x}(t)$ is the 11×1 error-state vector and $\mathbf{w}(t)$ is the 13×1 process-noise vector. Matrices $F(t)$, $G(t)$ are the linearized dynamics and input matrices formed by the Jacobian of the error-state dynamic equations with respect to $\mathbf{x}$ and $\mathbf{w}$, respectively. The process noise vector accounts for the uncertainties in both sensor measurements and attitude estimates. Table 6.2 describes the terms that make up the process noise vector. All other error states are modeled as a time-varying bias possibly corrupted by additive wideband-noise.

The wide-band noise and Markov model parameters presented in Table 6.2 were determined by characterizing multiple flights and using the logged INS/GPS navigation solution as a truth reference. The error modeling followed the procedure described in Section 3.5 of [59]. This process involved computing an error signal and finding a suitable mathematical description for the error signal and the associated error statistics. For example, the difference between the INS/GPS and AHRS estimated yaw angle was characterized and the numerical values shown in Table 6.2 were determined for wide-band noise statistics and the Markov model time constant and driving noise statistics. A similar procedure was used for pitch angle, altitude, airspeed, and wind error-states.

Note about Wind States

The linearized dynamics UAV model in Equation 6.15 includes two states for horizontal wind, W_{NS} and W_{EW}. In the decentralized filter each UAV will maintain on-board estimates of the wind experienced locally. Wind is a spatially varying quantity and hence it is reasonable that each UAV estimates the local wind. The centralized filter collects the states of all UAVs, including the wind estimates. In this case the wind estimate of each UAV is really the local variation experienced by the UAV. There is, however, a global average wind which mutually affects all UAVs. Without access to absolute information this global wind estimate is unobservable and the position estimate of the entire community will drift at a rate equal to the global wind. Excluding the global wind state is neglecting this global drift.

The flight data used in the subsequent analysis is from UAV flights that occurred at different times and days which have been temporally shifted to the same time. Since the data has been collected as such there is no mutual global wind term. This term is therefore neglected in the results that follow. This means that the centralized estimated uncertainty is not capturing the effect of a global wind drift which would affect a community of UAVs which are indeed flying concurrently. It does, however, model the local wind variations experienced by the UAVs and the utility of inter-UAV range measurements to estimate these states.

Aiding Measurements

Under nominal conditions, GPS position and velocity measurements are used by each UAV to arrest the error-growth of the dead reckoning solution. The GPS measurements provide observability to the error states, like *wind* and *yaw-error*, which are critical to improving the quality of the dead reckoning solution between updates. The innovation for the GPS position and velocity measurements are formed by:

$$\tilde{\mathbf{y}}_{pos} = \begin{bmatrix} \Lambda \\ \lambda \\ h_g \end{bmatrix}_{GPS} - \begin{bmatrix} \hat{\Lambda} \\ \hat{\lambda} \\ \hat{h}_g \end{bmatrix} \tag{6.13}$$

$$\tilde{\mathbf{y}}_{vel} = \begin{bmatrix} V_N \\ V_E \\ V_D \end{bmatrix}_{GPS} - \begin{bmatrix} \hat{V}_N \\ \hat{V}_E \\ \hat{V}_D \end{bmatrix} \tag{6.14}$$

where the caret symbol denotes the current state estimate. When available, the GPS measurement errors are assumed uncorrelated over time and hence standard EKF measurement update equations from Table 3.1 are used for the measurement update.

The AHRS and dead reckoning navigation system was mechanized for all 7 UAV flights. Subsequent discussions of cooperative navigation will be about the dead reckoning system. The dead reckoning error-state vector and associated linearized dynamic model for the i^{th} UAV is:

$$\dot{\mathbf{x}}_i(t) = F_i(t)\mathbf{x}_i(t) + G_i(t)\mathbf{w}_i(t) \tag{6.15}$$

where $\mathbf{x}_i$ is the 11×1 error-state vector and $\mathbf{w}_i$ is the associated 13×1 process-noise vector for the i^{th} vehicle. The associated 11×11 error-state covariance matrix for the i^{th} UAV is $P_{ii} = E\{\Delta\mathbf{x}_i\Delta\mathbf{x}_i^T\}$.

Consider a second vehicle, UAV j. In the absence of any collaboration between the UAVs, the inter-vehicle correlation matrix, P_{ij}, would simply be zero. However, once collaboration is used, the error-states for UAV i and j will become correlated, thus making $P_{ij} \neq 0$.

6.5 Flight Test Results

The effect of concept of operations (i.e. number of high flyers present) and cooperative aiding architecture (i.e. centralized vs decentralized filters) on the navigation performance of the 7 UAVs through an extended 5-minute GPS outage will be studied. The three scenarios described in Section 6.3 will be compared: no high flyer, one high flyer, and two high flyers. In all scenarios the coasting or unaided-dead reckoning results will be overlaid to serve as a worst case baseline. In contrast, the centralized EKF navigation performance for each of the three scenarios will define the best case or globally optimum performance. This is because the centralized filter models the error states of all 7 UAVs and can compute and maintain the exact[2] inter-UAV correlations. In this manner the optimal or covariance trace-minimizing fusion of the cooperative aiding measurements are computed and applied.

The navigation performance in the decentralized implementation for each scenario will fall somewhere between coasting and centralized. The missing knowledge of inter-UAV error correlations requires that alternative data-fusion algorithms like CI or BCInf be applied in order to maintain consistent estimates. Without knowledge of the inter-UAV correlations, the computed fusion gains for the cooperative aiding will be suboptimal. Furthermore, again due to missing knowledge of correlations, the *a posteriori* covariance computed on board will again be inflated or conservative. Therefore it is natural to expect the decentralized implementation to be worse than what is achievable with a single centralized filter. The goal, however, is to find decentralized implementations that both maintain consistent estimates and are as close a possible to the best case centralized performance.

The horizontal position-estimate performance of a single UAV, THOR 75, will be used to exemplify the community's performance. Finally, an average performance for the entire community of UAVs will be presented to illustrate collective benefit from cooperative navigation. Since the intent is to see the improvement from cooperative aiding, high flyers will be excluded from any average performance results.

In the forthcoming navigation performance error plots, the solid lines are the estimated (i.e by the on-board estimator) $1-\sigma$ bounds and the dotted lines are the errors, as compared with the logged reference ground-truth system which, in this case, was an INS/GPS solution.

6.5.1 Collective Outage (No High Flyer)

In the collective outage scenario all 7 UAVs are equally affected by the 5-minute GPS outage. Their similar avionics explain the similar performance degradation over these 5 minutes. The loss of access to absolute measurements means that the navigation solution for all 7 UAVs will drift. However, the Cooperative aiding in this scenario will demonstrate the attainable reduction in error growth-rate through the fusion of inter-UAV range measurements. The horizontal position error history for this scenarios is shown in Figure 6.5.

Coasting By the end of the *5-minute* outage, there is an approximate 800 meters of uncertainty in the vehicle position. This is the unaided performance of the on-board dead reckoning

[2] *Exact* really defines the reference or *to the best of our knowledge* performance. Linearization errors as well as error-model deficiencies are two reasons the *truth* may differ from the *reference* solution.

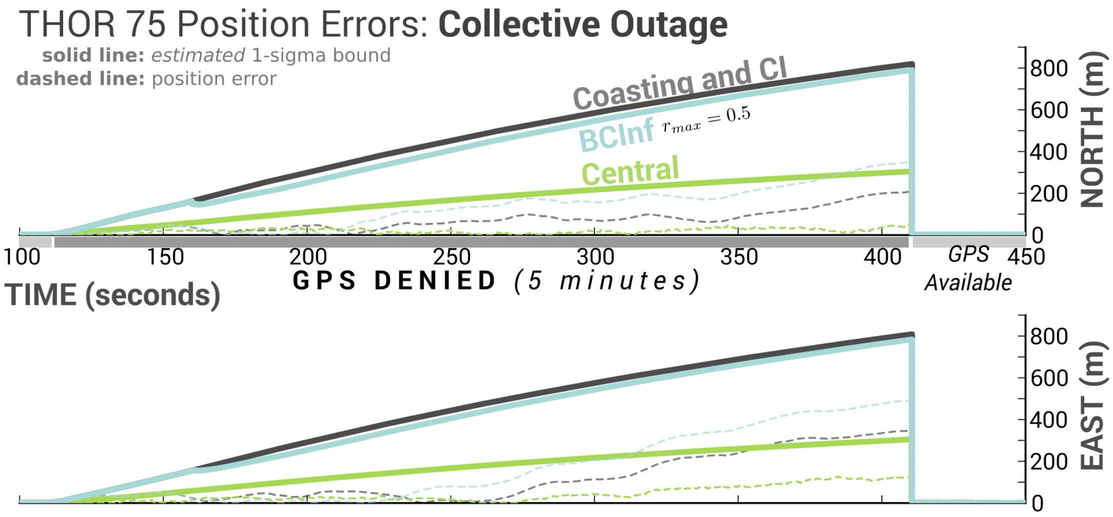


Figure 6.5: THOR 75 position errors for coasting and centralized filter during collective GPS outage.

system. Of course, the initial 110*seconds* where GPS was available was critical to provide initial observability on the error-states like *yaw-bias* or *wind estimates*. The coasting performance defines a worst-cast performance for handling the GPS-outage.

Cooperative Aiding: Centralized Filter The performance of cooperative navigation based on a centralized estimator shows a more than 50% improvement as compared to the unaided dead reckoning. There is only 300 *meters* of position uncertainty by the end of the 5-minute GPS outage. Despite all UAVs losing access to GPS, the cross-ranging and information exchange enabled a significant reduction in the rate of error-growth in the dead reckoning system for each UAV. A centralized filter, however, has significant communication and computation requirements, and thus is not as flexible as a decentralized implementation.

Decentralized Filters: Covariance Normalization

In Section 4.4.3 a covariance normalization step was demonstrated to be important to conducting a proper uncertainty trade-off for unknown correlation data fusion filters like CI and BCInf. In Section 6.5.3 a comparison is made between including and excluding the normalization step. For the UAV flight results, unless otherwise noted, all CI and BCInf results will have included the covariance normalization step. Here we describe the choice of normalization for the UAV application.

The proposed normalizing of the covariance as part of the CI/BCInf uncertainty trade-off

Dead Reckoning Error-State	**Goal Uncertainty**
Horizontal Position ($\Delta N, \Delta E$)	$max(3 \text{ m}, \ f(\Delta t))$ where Δt is duration of GPS-denied operation.
Altitude (ΔD)	5 m
Wind (W_{NS}, W_{EW})	8 m/s
Yaw Bias ($\Delta\psi$)	10°
Pitch Bias ($\Delta\theta$)	5°
Airspeed Bias ($\Delta u, \Delta v, \Delta w$)	5 m/s
Altitude Offset (Δh)	5 m

Table 6.3: UAV goal error-state uncertainties used for covariance normalization step in decentralized cooperative aiding implementations.

was accomplished by defining matrix T_{goal}. As specified in Equation 4.66, this diagonal matrix is populated using the goal or desired uncertainties σ_{goal} for each state. By Equations 4.64-4.65, matrix T_{goal} is used to normalize P and H, respectively, and the normalized matrices are used in the uncertainty trade-off optimization step to determine the optimal ω. The goal uncertainties used are specified in Table 6.3. The reasoning behind these goal uncertainties will be given next.

The decentralized cooperative aiding is applied when there is loss of access to GPS measurements. During this time, the error-states can be separated into two categories: One where the goal uncertainty is time-dependent and the second group of states where the goal uncertainty is largely constant. The time-dependent goal uncertainties are primarily the position uncertainties. In other words, the goal position uncertainty after 1 minute of operating without GPS will be different than the goal after 10 minutes of cooperative-aiding operation. In contrast, the goal uncertainty for states like wind estimates, attitude and airspeed bias does not include this time dependency. These error states are modeled as first-order Markov processes and the goal uncertainty for all time can be approximated using the steady-state Markov model uncertainty. This was the guiding principle used to define the goal uncertainties for the non-position error states shown in Table 6.3.

The goal uncertainties for horizontal position are specified in terms of a time-dependent model $f(\Delta t)$. The model specifies an acceptable rate of error-growth for vehicles operating with only cooperative aiding. Naturally, this model would differ for vehicles with different on-board sensor qualities. Furthermore, it is difficult to know *a priori* the availability of cooperative aiding measurements but that would likely also affect the model parameters. In this application, a single model was chosen for all UAVs. The model choice was guided by the principle that the decentralized cooperative aiding implementation should result in performance half-way between unaided operation and the attainable performance using the centralized cooperative aiding implementation. By this principle, the goal uncertainties were specified in four steps:

1. Define model $f(\Delta t)$ in terms of general parameters.

2. Use unaided dead reckoning estimated $1-\sigma$ bounds for North and East errors to fit $f(\Delta t)$ twice. Average the parameters determined using North and East errors to specify a single set of parameters for the unaided dead reckoning case. This gives $f_{unaided}(\Delta t)$

3. Repeat the fitting procedure in Step 2 using the centralized cooperative aiding implementation estimated $1-\sigma$ bounds. This determines $f_{centralized}(\Delta t)$.

4. Specify the goal uncertainty for the decentralized filters by averaging unaided dead reckoning and centralized implementation models: $f_{decentralized}(\Delta t) = \frac{1}{2}(f_{unaided}(\Delta t) + f_{centralized}(\Delta t))$

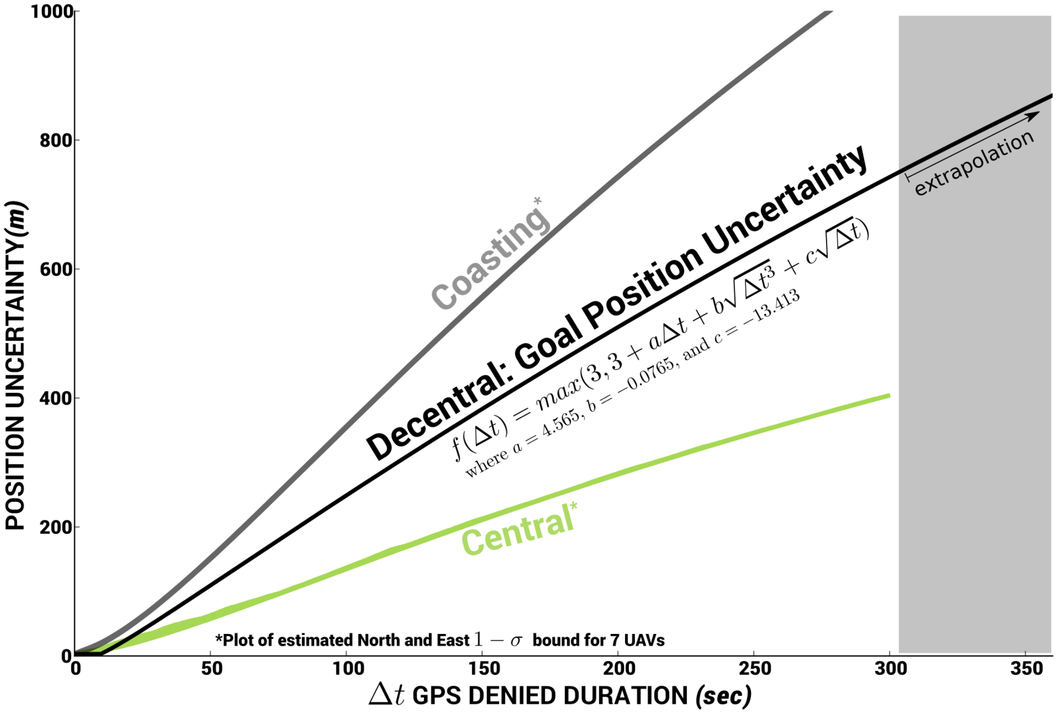


Figure 6.6: Goal position uncertainty for decentralized cooperative aiding implementations, used for CI/BCInf normalization steps.

This process was carried out for the UAV application Collective Outage (No High Flyer) scenario. The model specified is:

$$f(\Delta t) = 3m + a\Delta t + b\sqrt{\Delta t^3} + c\sqrt{\Delta t} \tag{6.16}$$

where a, b, c are general parameters. In general, it would be reasonable to impose that $a, b, c \geq 0$. In this case, however, this requirement was relaxed to get better a better fit. This had two implications, however. The fit has a valid input range and cannot be safely extrapolated beyond the original data. So, for example, if the fit is done using an outage of 5 *minutes*, unexpected results (e.g. decreasing uncertainty) could occur if the fit was applied to an application that went far beyond 5 minutes. Additionally, small Δt could result in negative $f(\Delta t)$. To handle these issues the goal uncertainty used was specified using $max(3\text{ m}, f(\Delta t))$, as shown in Table 6.3, and any extrapolation was checked. By this procedure the choice of parameters was $a = 4.565$, $b = -0.0765$, and $c = -13.413$ based on a fit of 5-minute outage and by additionally averaging over 7 separate flights. The fit is useful until about $\Delta t = 15$ *min* after which the extrapolation becomes unreasonable. The result of this procedure including the final specification of the goal uncertainty for ΔN and ΔE is shown in Figure 6.6.

In some applications the described off-line analysis for determining time-dependent goal uncertainties may not be possible. In these cases one may use the system performance specifications together with engineering judgment to specify a time-dependent goal uncertainty. Systems testing should reveal how the true system performance compares with the specified goal uncertainties. This knowledge in turn can be used to update the goal uncertainty model. It is important to remember, however, that no choice of goal uncertainties will change the underlying system performance as specified by the system model, measurements available, and the assumption of unknown correlations. Reasonable goals uncertainty specifications simply facilitate reasonable uncertainty trade-offs as part of the CI/BCInf measurement updates.

Now we return to showing decentralized results for the collective outage (no high flyer) case.

Cooperative Aiding: Decentralized Filter (CI) As shown in Figure 6.5 the performance for the decentralized CI filter was the same as the coasting performance. The conservative nature of the CI data fusion algorithm resulted in no aiding. This reflects the findings in [60] that CI-based decentralized implementations offer little benefit when a uniform quality of users makeup the collaborating community. Stated differently in [25], unknown correlation data fusion algorithms like CI exploit differences in shapes of the covariances of the quantities being fused. Therefore, the fusion of similarly shaped covariances are worst case scenarios for CI since there is little benefit derived from the fusion.

Cooperative Aiding: Decentralized Filter (BCInf) If no cooperative aiding occurs, as was the case in the decentralized CI implementation, then clearly there will be no inter-UAV correlation and the CI estimator assumption of *any possible* (unknown) correlation is too conservative. In contrast, the standard EKF measurement update assumption of independence would also be incorrect after a single cooperative aiding. The BCInf estimator attempts to bridge this gap. By assuming an unknown, but bounded correlation, the BCInf estimator is able to transition from CI to EKF.

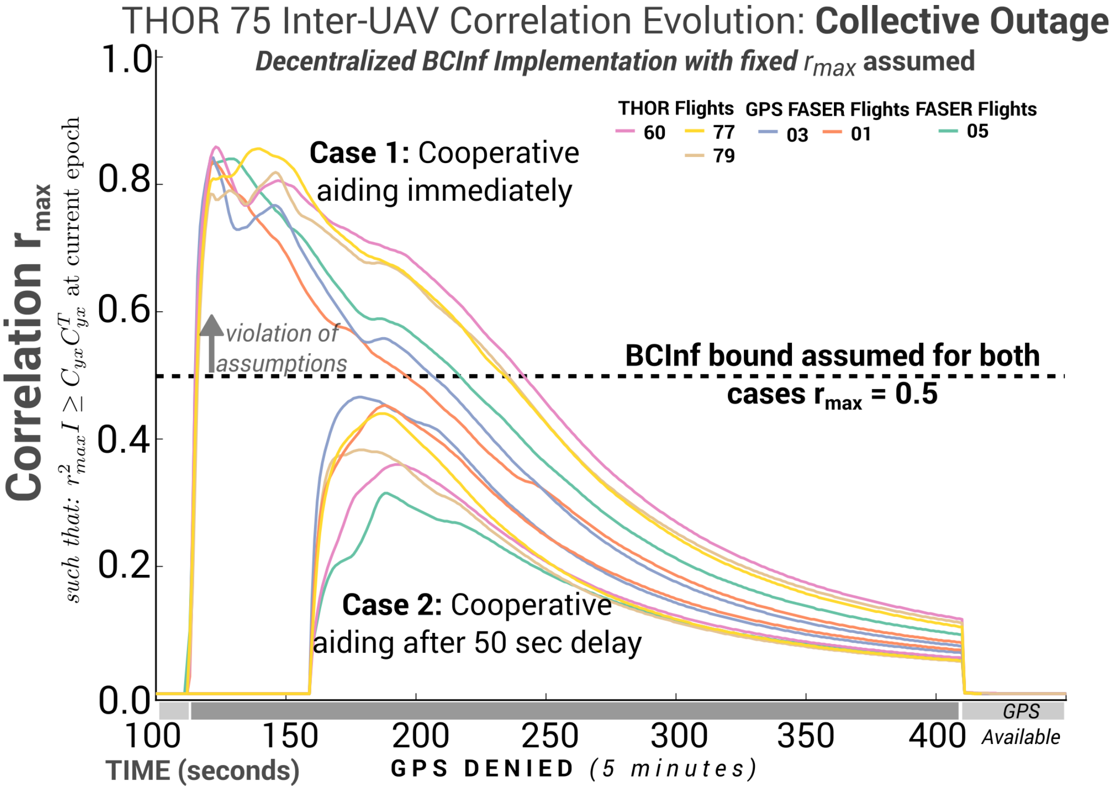


Figure 6.7: Inter-UAV correlation evolution for THOR 75 under decentralized BCInf implementation.

The assumed bound on the inter-UAV correlation was set to $r_{max} = 0.5$. This choice was based on the off-line analysis of the true correlations discussed in Appendix A. This iterative process required assuming a bound r_{max} and then analyzing the true correlation evolution and ensuring the bound was not violated. A snapshot of the analysis results where it was assumed $r_{max} = 0.5$ are shown in Figure 6.7. Two cases are shown in this figure. In the first case, cooperative aiding is applied immediately after the loss of GPS whereas in the second case there is a 50 *seconds* delay. There are six lines plotted for each case. Each line represents the inter-UAV correlation between THOR 75 and one of the six other UAVs.

As part of the analysis it was decided to operate in coasting mode for 50 *seconds* prior to starting the decentralized BCInf cooperative aiding. This is advantageous as it reduces the maximum inter-vehicle correlations that form in the community. If instead cooperative aiding is used immediately after the loss of GPS, the size of the inter-vehicle correlation can grow very large (e.g. $r_{max} = 0.8$) as shown in *Case 1* of the figure. By delaying the cooperative aiding the size of the maximum correlation is significantly reduced. As discussed in Section 4.4.5, r_{max} depends on the relative size of ownship uncertainty, collaborator uncertainty, and measurement noise and this reflects the variability of r_{max} depending on the frequency or onset of measurement updates. Despite the tighter bound made possible by the delayed onset of cooperative aiding the performance gain from knowledge of this bound on the correlation is modest. As is visible in Figure 6.5, there is little reduction in position uncertainty compared to the unaided coasting case. Furthermore, the time evolution of the true correlations plotted in Figure 6.7, and their sensitivity to scenario-specific conditions like update rates, illustrates the challenge of specifying a bound r_{max}. Thus, as was discussed previously in Section 4.4.5, designing a decentralized BCInf estimator that would be valid over a range of scenarios is challenging.

In the collective outage (no high flyer) cooperative aiding was only advantageous when used in a centralized implementation. Due to relatively uniform performance degradation, the decentralized implementations were unable to take advantage of the presence of neighboring collaborates. Next we add a single high flyer to the community of UAVs.

6.5.2 One High Flyer

In this scenario one UAV, in this case FASER 05, maintained GPS access throughout the entire flight. All other UAVs relied on cooperative aiding (both from the high flyer and from each other) to aide their dead reckoning system during the 5-minute GPS outage. The horizontal position error history for this scenario is shown in Figure 6.8.

Cooperative Aiding: Centralized Filter The centralized filter illustrates how the GPS-quality position from high flyer-FASER 05 was propagated to low flyer-THOR 75 by means of cooperative aiding. Hence, the presence of a single UAV with high quality position estimates enables comparable position accuracy to be obtained by other UAVs. The ability of the centralized filter to model all the UAVs and their inter-state correlation is the factor that enables the low flyers to maintain position errors on the order of tens of meters for the entire 5-minute duration of GPS-outage.

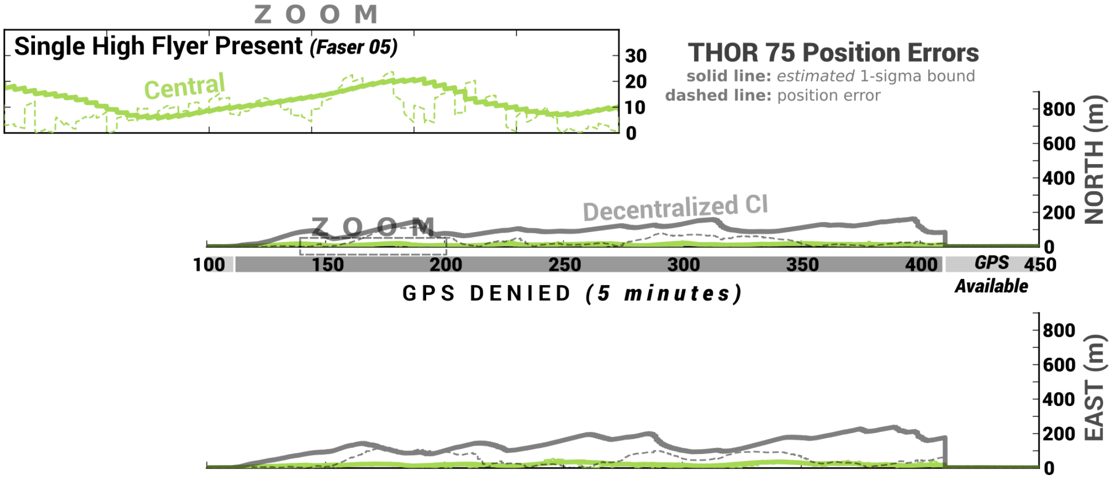
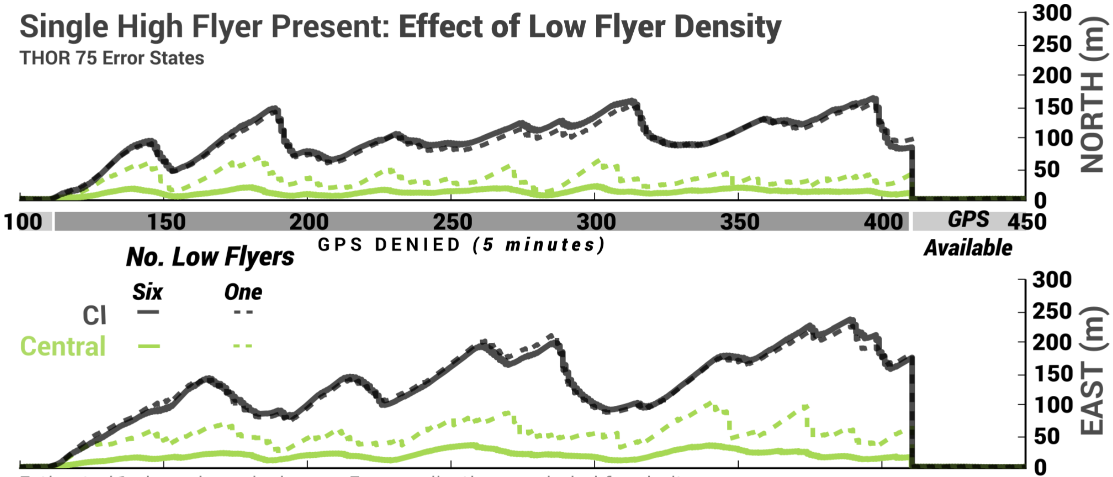


Figure 6.8: THOR 75 position errors for centralized and decentralized (CI) filters during single high flyer (FASER 05) scenario.

Cooperative Aiding: Decentralized Filter (CI) In contrast to the collective outage scenario where the decentralized estimator was largely ineffective, here we observe how CI enables THOR 75 to pass through 5-minutes of GPS outage with less than 200 *meters* of uncertainty. This may be an order of magnitude worse than the centralized filter, but the performance gain is significant when considering the flexible nature of the decentralized CI filter. Each UAV only models ownship error-states and there is no requirement for a centralized processing location. Inter-UAV correlation is unknown and yet CI enables cooperative aiding to be applied and consistent position estimates are maintained.

Low Flyer Density The analysis thus far considered the performance gain to low flyers due to the presence of a single high flyer (FASER 05). The analysis was solidified by considering the performance of a single low flyer, namely THOR 75. There are, however, five other low flyers present in the community. It is interesting to ask what advantage is served by the presence of these other low flyers? Is the performance gain to THOR 75 solely due to the presence of the high flyer or do the other low flyers add value as well? The answer to this question has direct design implications for it guides whether resources should be used to increase the community density or whether quality (i.e. high vs low flyer) is more important that quantity.

To answer this question the performance of THOR 75 in the single high flyer scenario, shown previously in Figure 6.8, was repeated in a scenario where only THOR 75 (low flyer) and FASER 05 (high flyer) were present. This was compared with the original THOR 75 results where the community comprised of 1 high flyer and 6 low flyers. The effect for both the centralized EKF



Estimated 1-sigma bounds drawn. Error realizations excluded for clarity.

Figure 6.9: THOR 75 position errors under two low flyer densities for centralized and decentralized (CI) filters during single high flyer (FASER 05) scenario.

implementation as well as the decentralized CI implementation are plotted in Figure 6.9. In the centralized implementation loss of low flyer density is clearly detrimental to the performance. The $<$ 10 m accuracy attainable in a community of 1 high flyer + 6 low flyer inflates to an average 50 m of uncertainty when only 1 high flyer + 1 low flyer are present. In contrast, there no change in performance for the decentralized CI implementation. It was previously shown in the collective outage scenario that vehicles with similar performance are of little benefit to each other in the decentralized implementation. Hence it makes sense that the advantage afforded to low flyers in the single high flyer scenario is largely due to the high flyer and little benefit comes from neighboring low flyers. There may be, however, a scenario where increased low flyer density is advantageous for the decentralized implementation. In this UAV analysis the cooperation range for the UAVs was assumed to be 5 km, which is sufficient for all UAVs to collaborate directly. If instead the cooperation range was less, for example 250 m, then the presence of other low flyers may serve to transmit the high quality high flyer information to other low flyers not directly within cooperation range with the high flyer.

6.5.3 Necessity of Normalizing CI

The decentralized implementation single high flyer scenario serves as a chance to compare and contrast the standard CI filter and the proposed weighting step. The decentralized performance shown previously in Figure 6.8 included the proposed covariance normalization step. Meaning, as part of the uncertainty trade-off step of CI, the on-board dead reckoning covariance matrix

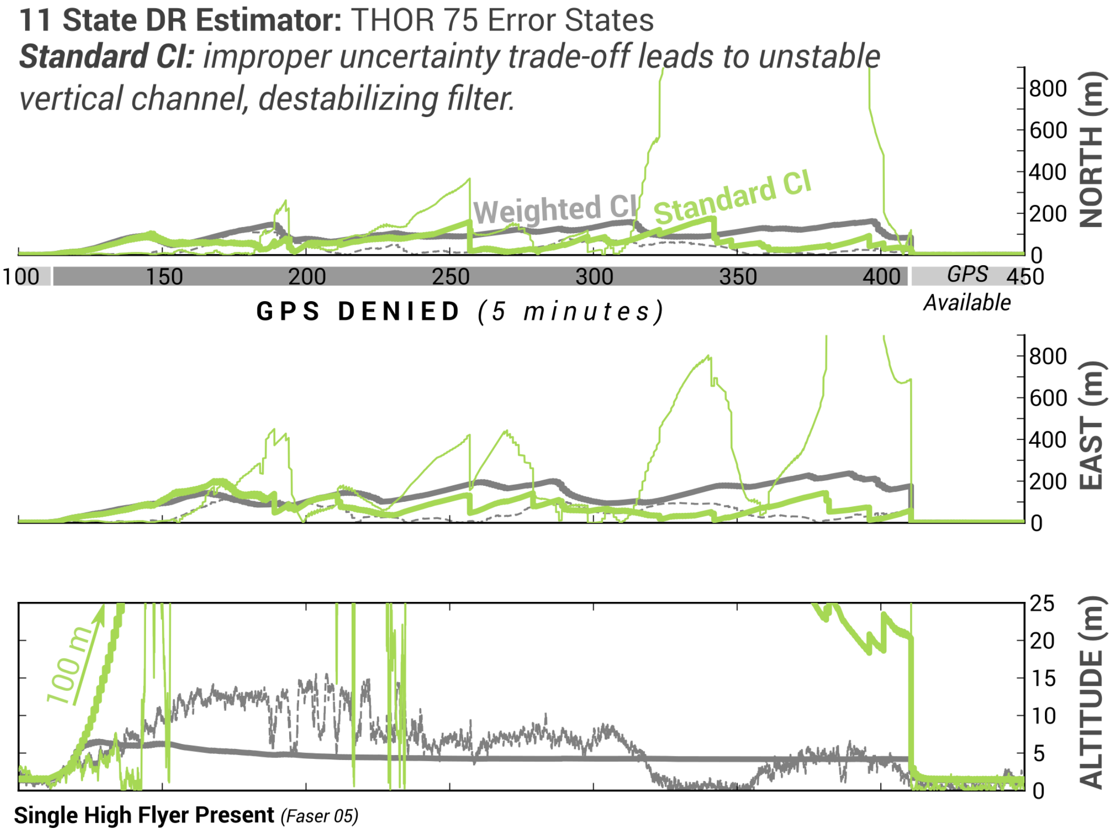


Figure 6.10: Comparison of standard CI and proposed weighted CI for THOR 75 position errors in decentralized (CI) filter during single high flyer (FASER 05) scenario.

was normalized in order to conduct a meaningful trade-off. Here we overlay those results with the standard CI implementation where there is no normalization to guide the uncertainty trade-off. The position errors that result for THOR 75 in this decentralized cooperative aiding scenario is shown in Figure 6.10.

The standard CI estimator diverges and the estimated uncertainty bound are not reflective of the growing errors. The source of this divergence is an unstable vertical channel. The altitude errors are plotted in the bottom section of Figure 6.10. The altitude error grows to nearly 100 m, in contrast to the proposed weighted CI altitude error which never exceeds 15 m. Such significant altitude errors lead to poor linearizations for the cooperative aiding measurements and finally lead to estimator divergence. To explain what is happening several variations of this scenario are considered. In this case it is not standard CI itself that causes divergence, but unguided uncertainty trade-off indirectly leads to poorly bounded altitude errors.

In the dead reckoning estimator the altitude state is geometric altitude. Therefore, GPS altitude observations are direct observations of this state. Baro-altitude observations, however, are an observations of pressure altitude and a state h_{offset} was introduced to capture the difference between geometric and pressure altitude. In this manner, the baro observation is the sum of geometric altitude and h_{offset}. This prelude is important to understanding why the vertical channel went unstable.

When GPS measurements are lost decentralized cooperative aiding is used to fuse inter-UAV range measurements. The range measurements are converted into horizontal range and provide observations of ΔN and ΔE error states. Necessarily, the fusion of these partial state-vector observations leads to inflation in uncertainty along the unobserved directions. This phenomena was described in Section 4.4.1. Without any guidance in the uncertainty trade-off, the standard CI filter leads to significant and repeated inflation in the estimated covariance of h_{offset}. As the uncertainty in the h_{offset} error-state grows, the utility of the baro measurement in arresting the vertical errors decreases because h_{offset} is responsible for connecting pressure altitude to the geometric-altitude state. In this manner the vertical channel is left unchecked, leading to linearization errors and finally to estimator divergence.

This demonstrates how the unguided uncertainty trade-off of standard CI can have unintended consequences for other estimator states seemingly unrelated to the measurement fusion. One may argue, however, that it is possible to decouple the dead reckoning horizontal and vertical channels and simply use the baro-altimeter for the vertical channel. In this case, how does the standard CI perform compared to the proposed weighted CI?

This exact comparison was conducted. The original 11-state dead reckoning estimator implemented on each UAV was modified to remove 3 vertical error-states (ΔD, Δw, and h_{offset}) resulting in an 8-state dead reckoning estimator. Altitude estimates are derived directly from the baro-altimeter measurements. The single high flyer scenario was rerun for this case and the standard CI and proposed weighted CI position errors plotted for comparison in Figure 6.11.

Immediately visible is that the estimator divergence issue has been resolved. The standard CI estimator errors are bounded by the estimator uncertainty. During the first 50 *seconds* of GPS denied operation the standard CI and proposed weighted CI position errors and estimated uncertainty are very similar. However, as time progresses, the standard CI estimated horizontal position uncertainty becomes slightly larger than the proposed weighted CI. In order to understand what is happening it is useful to zoom onto the standard CI estimated covariance. Unlike the relatively smooth position uncertainty of the proposed weighted CI solution, the standard CI position uncertainty is affected by large-uncertainty growth during time-updates followed by aggressive reduction of uncertainty during measurement updates. For example, near $t = 260$ *seconds*, the time-update rate of error growth for standard CI is near 100 m/s whereas the

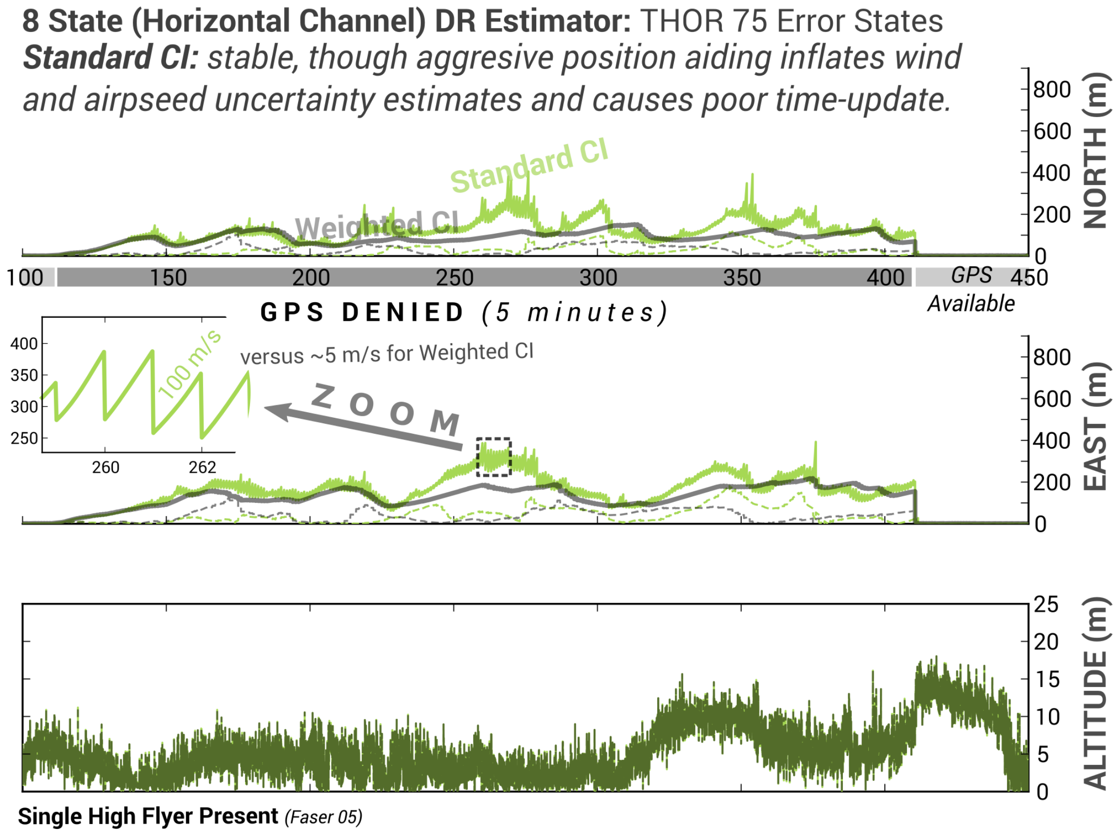


Figure 6.11: Comparison of standard CI and proposed weighted CI for THOR 75 position errors in modified 8-state decentralized (CI) dead reckoning system during single high flyer (FASER 05) scenario.

proposed weighted CI is only near 5 m/s. What has happened can be explained by visiting other error-states, like wind or airspeed bias error states.

The wind estimates and associated estimated uncertainty for the same 8-state dead reckoning implementation is shown in Figure 6.12. This figure explains the unreasonable rate of position uncertainty growth for the standard CI implementation. In order to aggressively reduce position uncertainty by way of the cooperative aiding range measurements, the unguided uncertainty trade-off of standard CI causes unreasonable inflation in wind-estimate uncertainty. So while there is an immediate reduction in overall (i.e. trace of covariance matrix) uncertainty at the time of the measurement update, the subsequent time-updates see rapid growth in position uncertainty due to the significant uncertainty in wind. In contrast, the guided uncertainty trade-off of the weighted CI is conducted in a manner where the goal uncertainties in Table 6.3

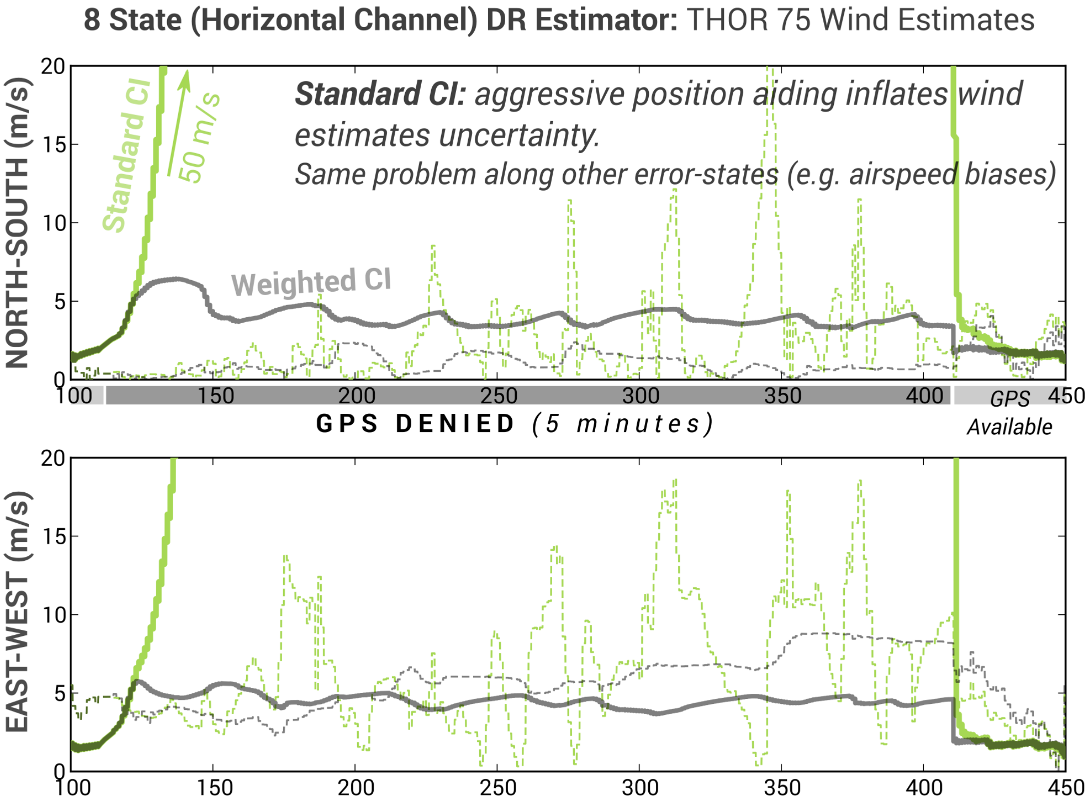


Figure 6.12: Comparison of standard CI and proposed weighted CI for THOR 75 wind state estimates for 8-state dead reckoning system single high flyer (FASER 05) scenario.

prevent undesirable trade-offs between position uncertainty and other error states. Therefore, the presence of extra unobserved error-states in addition to unguided trade-offs is still a problem for standard CI.

The challenge associated with extra unobserved states was described in Section 4.4.2. This was part of the motivation for choosing an airspeed-based dead reckoning estimator as opposed to a standard INS which can have 15 or 21 error-states. It is possible to do state reduction on the airspeed based dead reckoning estimator down to 4 states: two horizontal position error states and two horizontal wind estimates. This was implemented and again the same single high flyer scenario was run and the position errors for THOR 75 computed and plotted in Figure 6.13. The conclusion is unchanged from 8-state implementation. Although the position uncertainty is largely the same, the time-update is even worse, as is visible from the *thick* covariance bound, than it was for the 8-state implementation in Figure 6.12. Unlike the 8-state

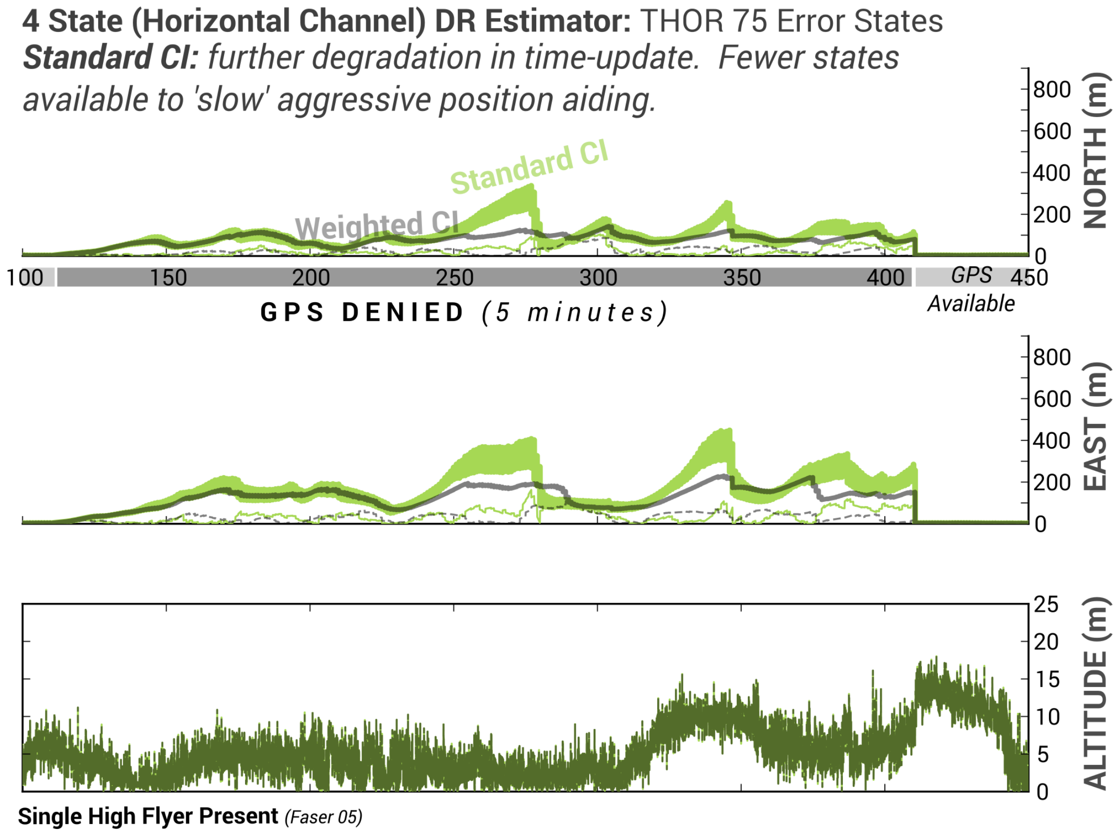


Figure 6.13: Comparison of standard CI and proposed weighted CI for THOR 75 position errors in minimal 4-state decentralized (CI) dead reckoning sytem during single high flyer (FASER 05) scenario.

estimator where there were multiple other states like attitude or airspeed bias with potential to slow the aggressive position aiding, the 4-state estimator is only position and wind. Therefore, the standard CI estimator will conduct the trade-off in favor of position and at the expense of wind uncertainty until finally their estimated uncertainty values near one another. There is no way for the standard CI estimator to differentiate between 10 m of position uncertainty and 10 m/s of wind uncertainty. In contrast, the weighted CI is able to use goal uncertainties to conduct the trade-off in favor the desired levels of uncertainty.

It is noteworthy that the 11-state, 8-state, and 4-state implementations of the proposed weighted CI filter, as shown in Figures 6.10, 6.11, and 6.13, were all very similar. This shows how the guided uncertainty trade-off is less sensitive to the number of states present. Having demonstrated the benefit of the proposed normalization step, the decentralized cooperative aiding in the following UAV scenarios will continue to use it.

6.5.4 Two High Flyers

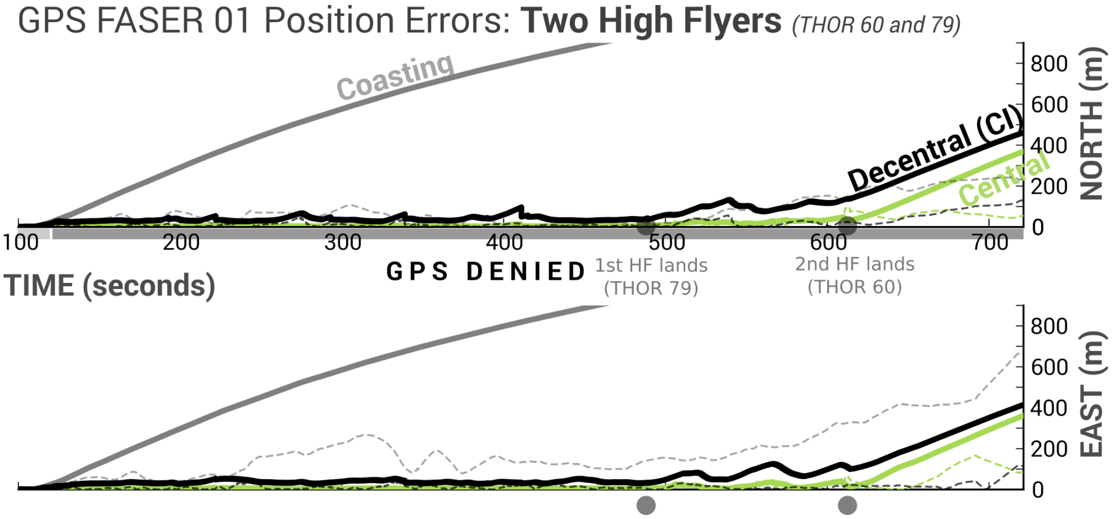


Figure 6.14: GPS FASER 01 position errors operating with two high flyers present but without GPS until landing. Centralized and decentralized (CI) filters performance plotted and coasting performance shown as reference.

In this scenario two UAVs, THOR 60 and THOR 79, maintained GPS access throughout the entire flight. As before, the on-set of the GPS outage for the 5 other low flyer-UAVs is at $t = 100$ *seconds*. However, in this scenario there is no return of GPS for the low flyers. The low flyers must rely on cooperative aiding from the high flyers and from each other to complete their mission and land.

The chosen high flyers, THOR 79 and THOR 60, themselves land at $t = 489$ and $t = 614$, respectively. One low flyer is still airborne after the second high flyer has landed, it is therefore interesting to observe the performance degradation as the high flyers land and become unavailable for aiding.

To facilitate this comparison, the navigation performance for GPS FASER 01, which lands at $t = 721$ (almost two minutes after the second high flyer has landed) is presented in this section and are plotted in Figure 6.14. To facilitate visual comparison, the y-axis is kept at the same scale as previous error-plots. However, the independent time axis is longer as it runs until landing for GPS FASER 01.

Loss of High Flyers This scenario is interesting as it shows the degradation in performance as the high flyers sequentially land and become unavailable for cooperative aiding. The onset of the GPS-outage for low flyers is at time $t = 100$ *seconds*. Until $t = 489$ two high flyers are present and during this time the cooperative aiding is most significant for GPS FASER 01 as is visible in Figure 6.14. The momentary peaks in error-growth (e.g. $t = 345$ or 410) are actually short durations where GPS was truly (i.e. in the experimentally logged flight data) unavailable to one or both of the high flyer aircraft. During these incidents the uncertainty in the high flyer aircraft position begins to grow and the loss of position accuracy propagates to the dependent low flyer aircraft, as is visible here.

At time $t = 489$ the first high flyer (THOR 79) lands. The effect of the loss of a high flyer is most noticeable by observing the decentralized (CI) performance. While the error growth is still bounded, there is a general upward shift in the steady-state uncertainty in GPS FASER 01 caused by the loss of a high flyer.

At time $t = 614$ *sec* the second and last high flyer (THOR 60) lands. At this point any remaining low flyer aircraft can only rely on cooperative aiding from one-another and the situation becomes similar to the collective outage first scenario. The decentralized algorithms had little benefit in the collective outage, and that explains the coasting-like growth in position uncertainty observable for $t > 614$ *seconds* for GPS FASER 01.

The similarity between the centralized and decentralized rate of error-growth for GPS FASER 01 after both high flyers have landed also needs explanation. In this data set all other low flyers have landed either at or before $t = 614$ *sec*. For example, THOR 75, the results of which were shown previously, lands at $t = 476$ *sec*. Therefore, after the second high flyer has landed, GPS FASER 01 is the only UAV left flying. There is no other UAVs for cooperative aiding and the advantage of the centralized over the decentralized implementations is only in the initial conditions and little difference is observed in the rate of error-growth.

This exemplifies one of the drawbacks of cooperative aiding, namely that the availability is limited to where collaborators exist and are within communication and measuring range. A second challenge with cooperative aiding is proper handling of integrity. This will be discussed next.

6.5.5 Integrity Issues

The information sharing inherent to cooperative aiding methods introduces a significant challenge for maintaining integrity. This topic deserves significant attention if cooperative aiding is to be considered a viable backup to GPS. In this section properties of this challenge are highlighted, which in reality are current limitations on the proposed techniques.

As part of the cooperative aiding the main shared quantity is the estimated location and

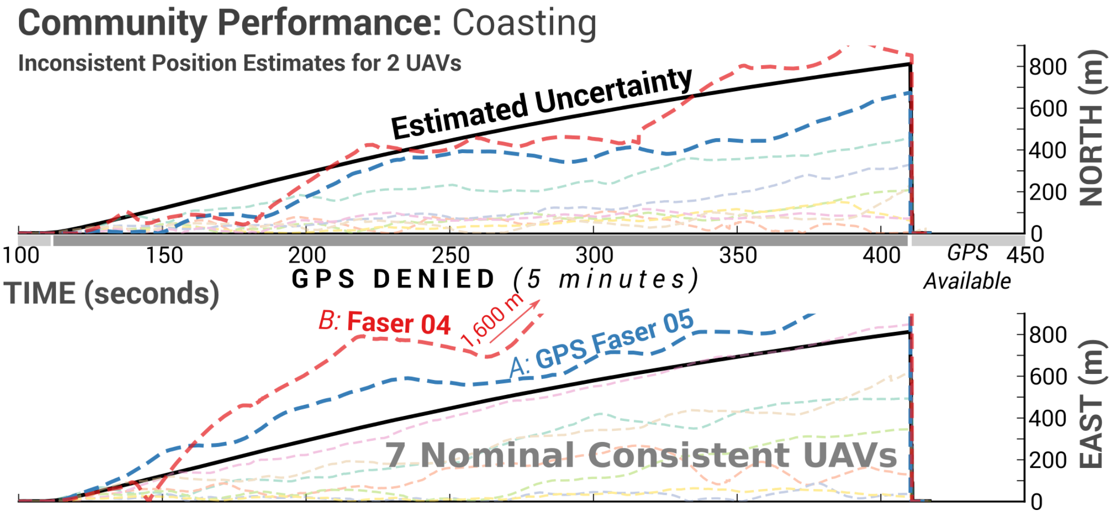


Figure 6.15: Unaided dead reckoning performance for two inconsistent UAVs.

associated uncertainty. If, additionally the relative measurement requires active participation of the potential collaborator, that introduces a second point of interest for integrity checks. Here we focus on the estimated location broadcasts. This broadcast is received by neighboring vehicles and used to form the measurement innovation and mapping matrix necessary for fusing the relative measurement. Therefore, from the perspective of ownship vehicle the incoming measurement is a combination of relative measurement and assumed uncertainty together with the received collaborator position estimate and estimated uncertainty. An inconsistency between the received measurement and measurement uncertainty would be akin to fusing an erroneous measurement, one whose true errors are not reflected by the assumed error statistics. Handling possibly erroneous measurements is common in estimators. What makes this more important in cooperative aiding applications is how erroneous broadcasts from a single vehicle can compromise the navigation integrity of all other vehicles. This can occur through direct receiving of erroneous broadcasts as well as possibly indirectly receiving erroneous broadcasts as the error propagates through the community.

Until this point the community of UAVs was comprised of 7 UAVs. One property of these 7 UAVs that is significant is that the unaided dead reckoning estimates for all 7 UAVs are consistent. This means that the estimated position uncertainty bounds the true errors. This was determined with the advantage of post-processing and with access to a reference GPS-aided INS solution. Originally 10 UAV flights were considered for this work, of which these 7 maintained consistent unaided dead reckoning estimates. Three flights did not do so, two of which will be the focus of this section: GPS FASER 05 and FASER 04.

Consistent unaided estimates are good indications of maintaining consistent estimates when cooperative aiding is used. We begin by plotting in Figure 6.15 the unaided dead reckoning only solution for the nominal 7 UAVs overlaid with the two inconsistent UAV flights. The main inconsistency is along with East direction, where GPS FASER 05 is moderately inconsistent and the inconsistency in FASER 04 is significantly larger as the true errors are nearly an order of magnitude larger than the estimated uncertainty. Notice that the estimated uncertainty for all $7 + 2$ UAVs are largely equivalent during the GPS outage. This is due to the similar quality avionics on board.

The cause of the inconsistent position estimates of these two flights is beyond the scope of this work. Modifications to the dead reckoning estimator error models would be required to capture the true errors observed. Therefore they were excluded from the subsequent cooperative aiding studies shown previously. In this section we sequentially consider including one of the inconsistent flights as part of the cooperative aiding community during the collective outage scenario. The results shown are for the centralized implementation.

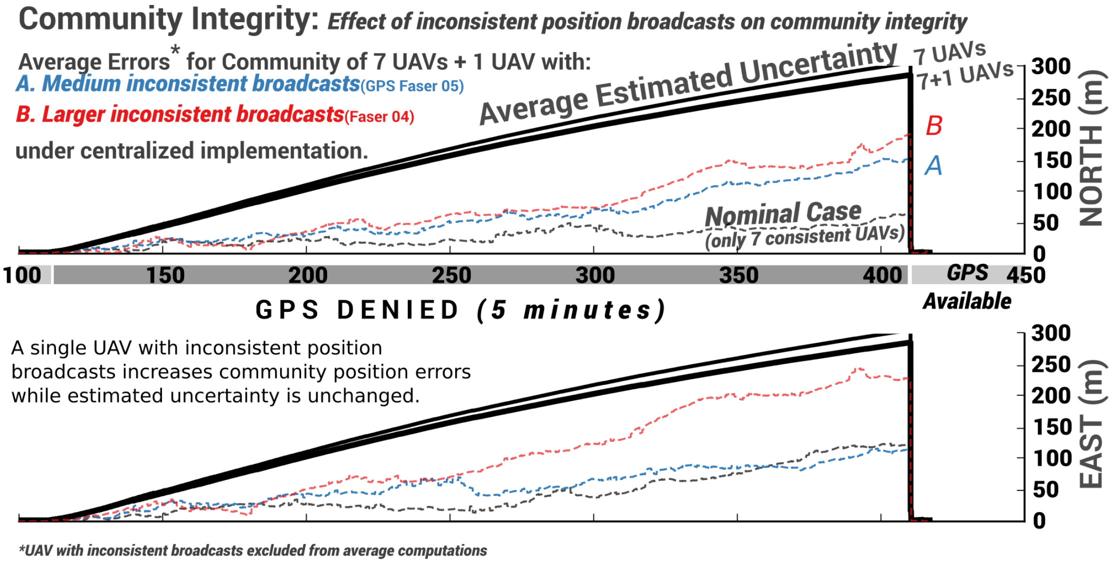


Figure 6.16: Comparison of community average errors when single UAV with inconsistent position broadcasts added to nominal 7 UAV community.

Figure 6.16 shows the average estimated uncertainty for the nominal 7 UAV community as well as an 8 UAV community. This is drawn as a solid line and is computed by averaging the estimated position uncertainty of each UAV in the community. As would be expected the presence of an additional UAV lowers the estimated uncertainty in the community. Without any integrity monitor, there is no way for the community of UAVs to know that the added UAV is broadcasting position estimates which are possibly not bounded by the broadcast uncertainty. The average true position errors in each case is computed and plotted using dashed lines. Notice that the presence of the 8th UAV increases the community position errors. Furthermore, in the average community position errors are larger as the UAV with larger inconsistency, as determined from the unaided results in Figure 6.15, is part of the cooperative aiding community. Thus we observe how a single user with inconsistent position estimates can degrade the position estimate consistency for all vehicles in the community.

It is worthy to note that this example is not a malicious attack on the community integrity. It is simply one member of the community which, due to mis-modeling, the true errors are not reflected by the estimated uncertainty. A more significant challenge may be position broadcasts made with malicious intent to disrupt or spoof the community of vehicles.

6.5.6 Community Performance

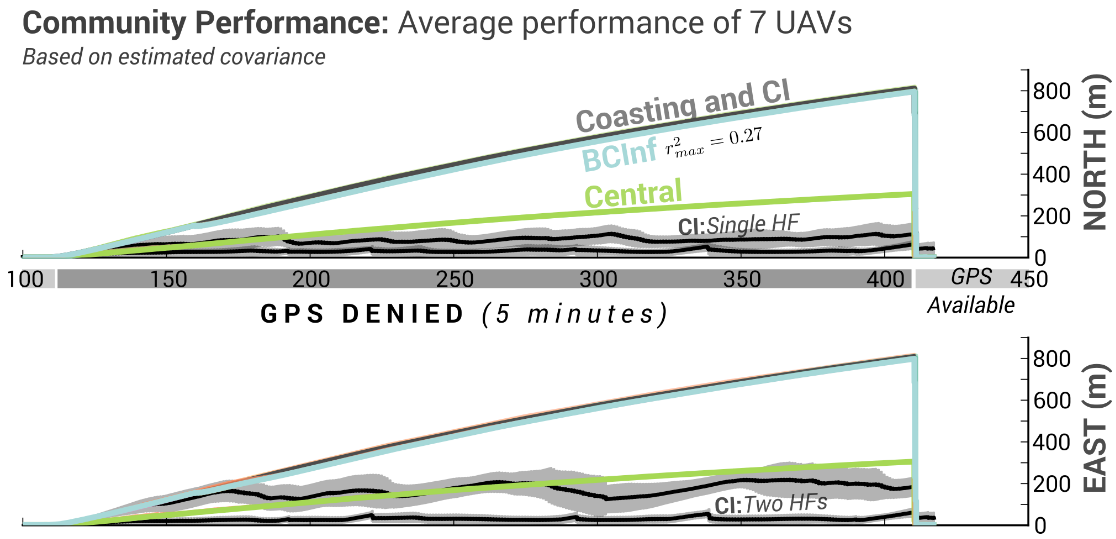


Figure 6.17: Comparison of average performance of all UAVs in community under various CONOPs and filter implementations.

By focusing on the performance of a single vehicle, THOR 75, previous results demonstrated the effect of estimator implementation and concept of operations on cooperative aiding performance. It is desirable that cooperative aiding is of benefit to all UAVs and therefore in this section the average performance for the entire community of UAVs will be compared for the none, one, or two high flyer scenarios under the decentralized cooperative aiding implementations. These are plotted in Figure 6.17. Additionally the coasting and centralized implementations are included to serve as a reference. This plot is based on the estimated covariance of each UAV. The estimated standard deviation is averaged and plotted. Additionally, where there is variability, the standard deviation of the community uncertainty is plotted through shading. Where shading is not visible is because there is little variability from the average performance.

Figure 6.17 shows that the community results are unchanged from what was observed from THOR 75 for the coasting, CI, BCInf, and central implementations during a collective GPS outage. Due to the uniform sensor qualities on board the 3 airframes, even the coasting performance is very similar between UAVs. Before considering the high flyer scenarios, it is worthy to note that the aggressive aiding of the centralized implementation will likely ensure uniform performance across community members, as is seen in this case.

Performance variability is visible in the Single and Two high flyer scenarios. Both these cases are for the decentralized CI-based cooperative aiding implementations and in computing the community performance the high flyers have been excluded. The single high flyer scenario results in performance that is largely superior to the collective outage centralized implementation. This is significant since it suggests that using a decentralized implementation the presence of a single high-quality user can out-perform a centralized implementation. Designing and implementing the computation and bandwidth required to enable a centralized cooperative aiding implementation is significant. This result suggests that such investment is unnecessary and instead a fraction of these resources could be spent on ensuring the presence of several high-quality users in the community.

Choice of High Flyer on Community Performance

Previously the results were shown for the case that FASER 05 was selected as the single high flyer. Here we investigate the variability of results that would have occurred had an alternate UAV been selected as the high flyer. This is done by comparing the average community performance under repeated single high flyer scenarios using the decentralized cooperative aiding scenario where each time a different high flyer is chosen. The average community horizontal position errors for this study are plotted in Figure 6.18. The shading associated with variability in the community performance has been excluded for better readability and simply the average performance is plotted. Note that the vertical axis has been expanded to better separate the results.

Figure 6.18 shows that while the choice of high flyer does affect the community performance, there is no clear superior or inferior choice for high flyer in this case study . For example, while the THOR 79 high flyer scenario has the worst error uncertainty along the North direction, it

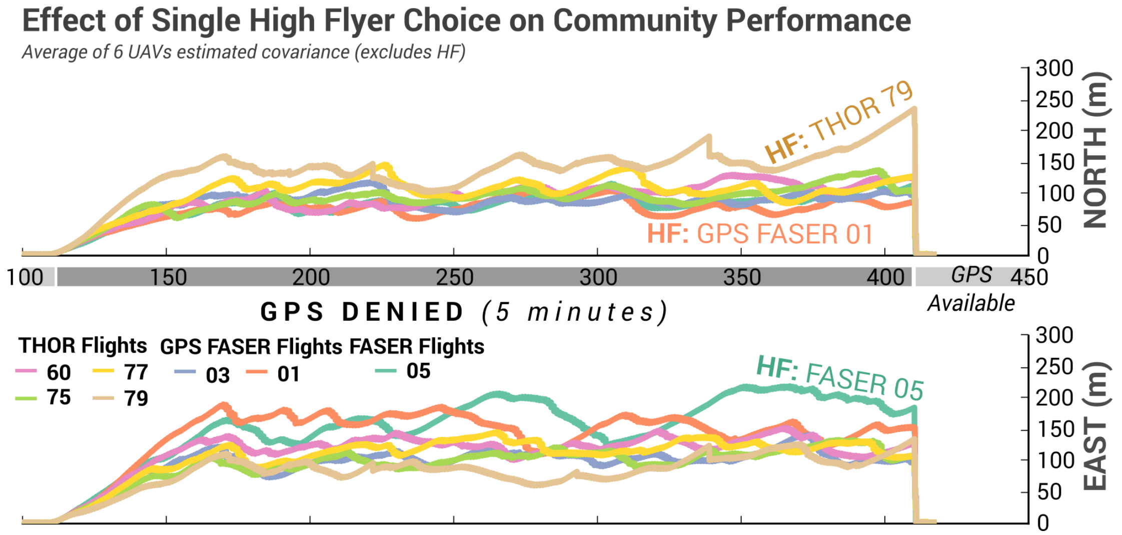


Figure 6.18: Average performance of all UAVs (decentralized CI implementation) under various choices of high flyer UAV.

has the best performance along the East direction. Some of this can be explained by revisiting the original UAV ground tracks in Figure 6.1. For example, THOR 79 is to the East of all other UAVs and thus the line-of-sight between THOR 79 and other UAVs is predominantly along the East-West direction with less variability along the North-South. Hence we observe improved East-West position estimates when THOR 79 is the selected high flyer. A similar analysis can be done for FASER 05 where the line-of-sight is predominantly along the North-South and therefore there is relatively poor aiding to the community along the East-West direction when FASER 05 is the high flyer. This demonstrates how the geometry between vehicles in cooperative aiding can and will affect the results and will be an important consideration in system design.

6.6 Conclusion

This chapter showed how both centralized or decentralized cooperative navigation implementations can be used to overcome small UAV vulnerability to GPS outages. The performance of the networked UAV navigation system was evaluated using experimentally logged flight data from 7 UAVs. Three scenarios involving none, one, and two high flyer UAVs were investigated as were both centralized and decentralized estimator architectures. During a collective outage, where all vehicles lose access to GPS, cooperative aiding based on a centralized architecture was shown to reduce the error-growth rate by more than 50% as compared to the unaided dead reckoning case. The decentralized CI/BCInf filters, however, provided little benefit due to the nearly uniform degradation in navigation performance accuracy in the community of vehicles. In the second and third scenarios, the presence of one or more high flyer UAVs were sufficient to enable decentralized cooperative aiding filters to significantly arrest the rate of error-growth during the GPS outage. Furthermore, the scenario studied in this chapter suggested that the presence of a single high-quality user in a CI-based decentralized implementation can out-perform a centralized implementation where all users are of equipped similarly. Finally, this chapter showed how the covariance normalization step introduced in Section 4.4.3 can be critical for successful application of CI/BCInf-based decentralized estimators.

Chapter 7

Results: Automotive Applications

7.1 Introduction

In this chapter cooperative navigation-based aiding is evaluated in an automotive simulation with a community of 2,000+ low-cost INS users operating in a GNSS-denied zone. Both single-vehicle and community-wide performance are compared. Single vehicle performance is used to illustrate performance trends whereas community-wide performance analyzes the collective benefit of cooperative aiding to all vehicles in the community. The comparison includes both prior art decentralized estimation schemes as well as the proposed CI-based decentralized filter and the centralized filter. The last section of this chapter is a quantitative analysis of the bandwidth requirements for both centralized and the proposed CI-based decentralized filters. This will be anchored to the upcoming dedicated short range communication protocol described earlier in the automotive cooperative navigation concepts of operation of Section 2.5.

7.2 Simulation

The performance of our CI-based cooperative navigation algorithm is evaluated using a simulation. The inputs to the simulation are based on real highway car traffic data obtained from the Next Generation Simulation (NGSIM) program [4]. The data used is vehicle traffic trajectory for an approximate half-mile stretch of U.S. Highway 101 (Hollywood Freeway) in Los Angeles, California. A schematic of the location where the data was collected is shown in Figure 7.1 .

The 10 Hz trajectory data consists of vehicles' position, velocity, and acceleration history derived from strategically placed video cameras. The simulation is based on data from 7:50 am - 8:05 am where a community of 2,169 unique vehicles traveled the one-direction 5-lane highway.

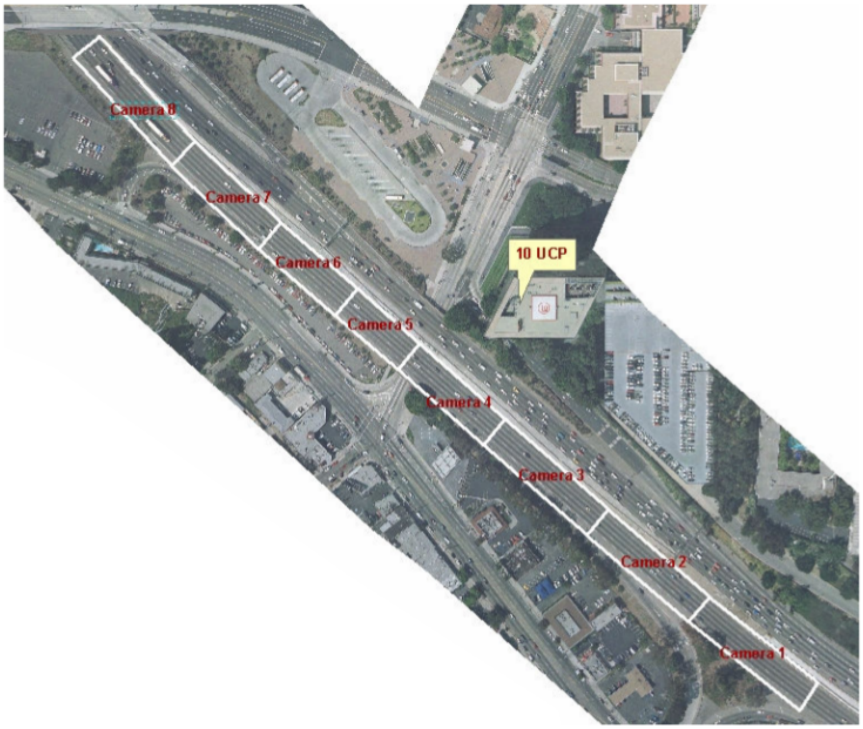


Figure 7.1: Schematic of location where data was collected, taken from [4].

Of these vehicles, 96% are automobiles and the remainder was composed of 53 trucks and 30 motorcycles. Simulated body-axis inertial measurements were generated for all the vehicles by inverting the locally level mechanization of the INS equations. The inertial measurements are corrupted by error models consistent with tactical and automotive grade sensor given in [61] and [5].

7.2.1 Simulation Setup

Each vehicle is equipped with an inertial navigation system, radar, and a wireless modem for broadcasting their own state vector and covariance matrix at regular intervals. Unless otherwise specified, all automobiles and motorcycles have low-cost consumer grade inertial sensors and trucks have tactical grade inertial sensors. The range-only radar installation assumed is depicted in Figure 7.2. Subsequent references to cross-track and in-track errors are in accordance with this figure. Only the closest vehicle inside each of the four radar sweeps will be visible to ownship. The range errors are zero mean and have a variance of $R = 0.25\ m^2$. Radar complexities like target detection and tracking were neglected, therefore the assumed range and accuracy were

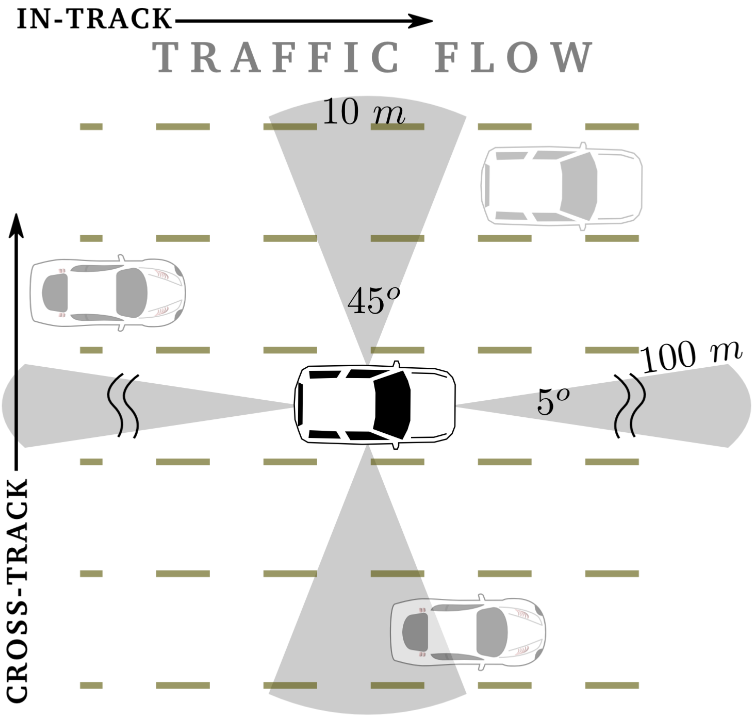


Figure 7.2: Nominal vehicle range-only radar installation assumed.

selected to be less capable than what is commercially available, like [3].

The half-mile stretch of highway is assumed to be a GNSS denied zone. Hence each vehicle enters the half-mile stretch of highway with a very accurate state estimate, but thereafter relies entirely on cooperative navigation to reduce the INS error growth. A snapshot of the simulation at 24 seconds past 7:50 am is shown in Figure 7.3 with a further exploded view in Figure 7.4. The goal is to study the performance of cooperative navigation as a means to limit the INS error growth during an extended unavailability of absolute measurements, like from GPS. Towards this end, we perform several trade-studies to compare the performance gains achieved when centralized and decentralized cooperative navigation algorithms are used to aid a low-cost INS in GNSS denied situations. We also highlight the performance of the CI-based estimator developed here and compare it to prior work which used EKFs and source selection.

First, the results from a single vehicle in the community will be analyzed to understand the performance trends. The path of the single vehicle, vehicle 125, is highlighted in Figure 7.3. Then, the performance of the entire community (i.e. all vehicles in the cooperative navigation system) will be analyzed.

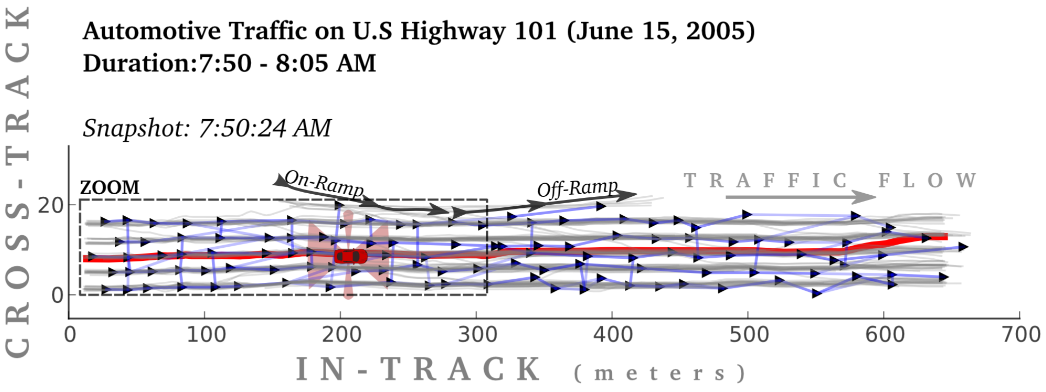


Figure 7.3: Snapshot of automotive traffic flow along 5-lane highway.

One note should be made about the CI/BCInf results presented in this section. Both CI and BCInf results presented in this section do not take advantage of the covariance normalization step proposed in Section 4.4.3. It is believed that including the covariance normalization step may slightly improve these results.

7.2.2 Case I: Single Vehicle Performance

Vehicle 125 was selected to illustrate the performance of a sample vehicle in a cooperative navigation community. We compare the performance of two groups of estimators. First, the traditional EKF based decentralized estimators which utilize source selection to mitigate the effect of inter-vehicle correlation. Second, the proposed CI-based decentralized estimator is compared in relation to the INS-only and the centralized estimator.

Looking at a single error realization can be misleading, hence navigation system performance is commonly evaluated using the estimate covariance. The estimate covariance conveys the performance distribution that can be expected over many realizations. However, we are operating with linearized estimators and therefore we must establish whether the estimator covariance is capable of bounding the true error over multiple realizations. If it cannot bound the error, this may represent an integrity violation. This is particularly important if we evaluate the traditional EKF based decentralized estimators where correlation information is neglected. Therefore, 50 Monte Carlo runs were conducted for each implementation, and the estimator covariance is overlaid on top of the position error realizations.

The position errors of traditional decentralized EKF using a democratic, fixed-rank, and covariance-based source selection architecture are plotted in the top, middle, and bottom of

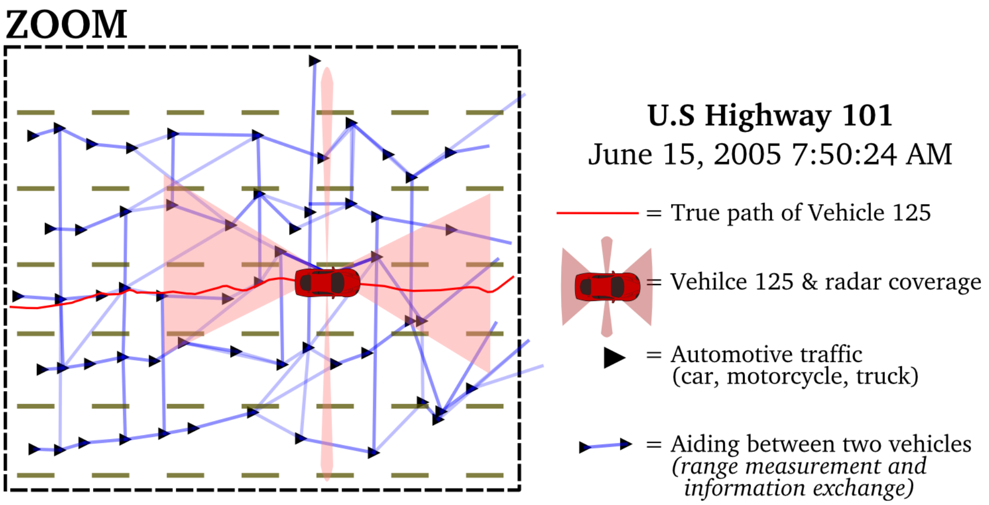


Figure 7.4: Exploded view of trajectories and simulated cooperative navigation. Cooperative aiding only occurs between vehicles that fall within respective radar coverage.

Figure 7.5. Immediately obvious is the fact that all three are inconsistent estimators, meaning that the estimator covariance fails to capture the true estimate errors. For example, the democratic organization is seen to be overly optimistic. The unrestrictive cooperation with neglected correlation causes improper measurement fusion and filter divergence in both in-track and cross-track directions. In contrast, the fixed-rank estimator has nearly no measurement updates in the cross-track direction, but still suffers from divergence in the in-track direction. Only 6 trucks (i.e. tactical grade INS users) are present during the time interval when vehicle 125 is traveling. Since vehicle 125 has a consumer grade INS (i.e lower rank solution) this explains why there are very few updates in the fixed-rank hierarchy. Lastly, the dynamic rank assignment of the covariance-based hierarchy leads to results between that of either extreme: unrestricted democracy or rigid fixed-rank.

Next we evaluate the proposed CI-based decentralized estimator in relation to INS-only and centralized estimators. The position errors are plotted for these three implementations in Figure 7.6. Firstly, it is noticeable that the estimator covariance for all three cases largely bounds the error realizations. This lends legitimacy to future comparisons based on estimator covariance values. The consumer-grade INS-only results in a position error standard deviation of approximately 15 meters after 45 seconds of travel. When standard highway lanes are approximately 3.5 meters wide, the INS-only systems leaves much to be desired. In contrast, the centralized filter performance (assuming unlimited communication bandwidth and computational capabilities) achieves sub-meter performance in the same duration by aggressively aiding the INS.

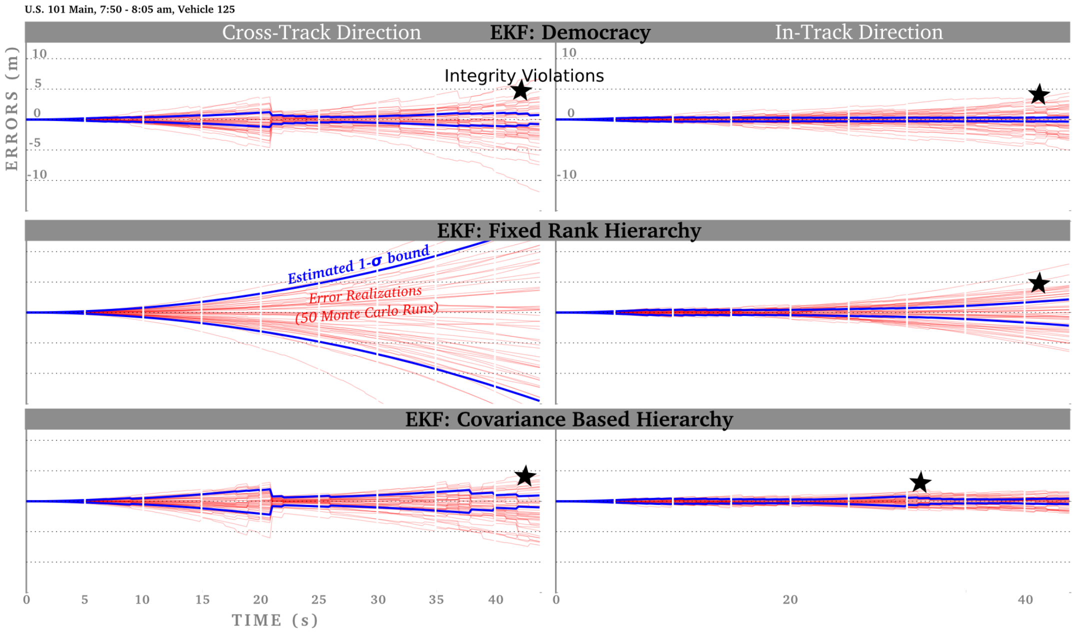


Figure 7.5: Performance of vehicle 125 using traditional decentralized estimators.

The proposed CI-based decentralized estimator performance is partially between the two extremes. A key to understanding the estimator performance is to recall that it fuses information conservatively (i.e. it overbounds or inflates measurement and *a priori* covariances). Thus, all things being equal, it will tend to discount more measurements than an EKF. Firstly, near time $t = 20$ *sec*, there is the first significant aiding to the INS of vehicle 125. The source of this update is a vehicle which enters the highway from a ramp, and hence has a significantly smaller covariance then other vehicles which have traveled for a much longer duration in the GNSS-denied zone. The second significant aiding occurs at about $t = 40$ *sec* and is along the in-track direction. Interestingly, this aiding in the in-track direction is coupled with an inflation in uncertainty in the perpendicular direction. This counter-intuitive inflation in uncertainty is a by product of the conservative CI fusion which operates under unknown correlation. While this is inherent to the method, the effect of the trade-off becomes pronounced in applications where the measurements are partial (e.g. range only) and do not include the entire state vector.

7.2.3 Case II: Community Performance

To compare the community performance, the estimated $1 - \sigma$ bound for each of the $2,169$ vehicles was collected as a function of *true distance traveled* along the highway. The mean and

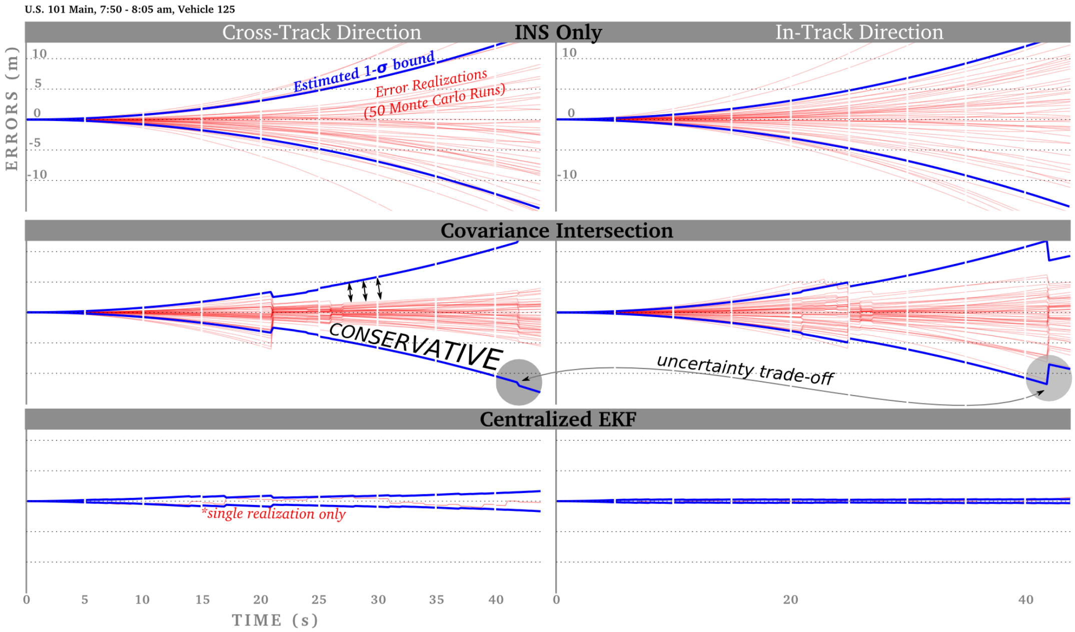


Figure 7.6: Performance of a single vehicle in the cooperative navigation community (vehicle 125).

standard deviation was calculated and then plotted, as shown in Figure 7.7. Shading was used in place of error bars to improve readability. In the single vehicle performance plots, multiple runs for a single vehicle were evaluated. Here, multiple vehicles in a single run are compiled into a single plot. The in-track performance was about 20% better than the cross-track performance. However, the relative performance between estimators were similar for both in-track and cross-track directions, hence we plot only the in-track community performance in Figure 7.7.

Immediately clear is the significant advantage afforded by the centralized estimator. A community of 2,169 vehicles, of which more than 97% of the vehicles have consumer grade INS on board, is able to achieve an order of magnitude improvement using a centralized cooperative-aiding filter compared to the INS-only case. In the case of the CI-based decentralized estimator the average community performance did improve compared to the INS-only case, but the collective improvement is modest, an approximate 20% improvement. It is clear that all vehicles did not uniformly benefit. Those in favorable geometries or coming within collaborating range with a more accurate neighboring vehicle, as was described for vehicle 125 and the merging collaborator, found improvement. However, for many vehicles, the conservative fusion by CI resulted in selecting $\omega \approx 1$, which as described earlier, means the relative measurements were

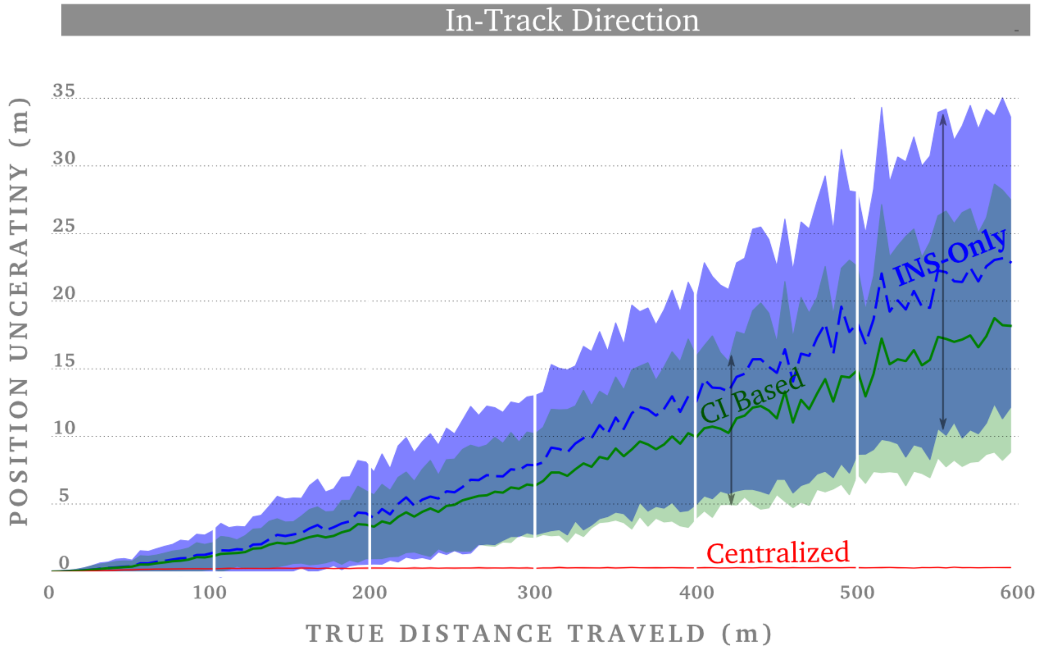


Figure 7.7: Performance of a cooperative navigation community of 2,169 vehicles along in-track direction.

discarded. This is a by-product of unfavorable geometries, the availability of only partial information (range-only measurements), and the conservative fusion under unknown correlation by CI.

The performance limitations imposed by unfavorable geometries are not specific to CI. In satellite-based navigation this concept is known as dilution of precision (DOP) or in other navigation contexts it may be referred to as poor estimator observability [5]. The conservative nature of the CI-based decentralized estimator only magnifies this problem. When measurements along independent directions (for example, in-track and cross-track) are scarce, then the reduced weighting or occasional discarding of measurements by CI further reduces geometric observability. The assumption of unknown correlation must therefore only be used in relevant applications.

7.2.4 CI and Community Composition

When the non-cooperating community is composed of a mix of navigation solution qualities, the advantages of CI become more apparent. The different qualities can be either due to sensor qualities, access to external measurements, or initial conditions. This is graphically depicted

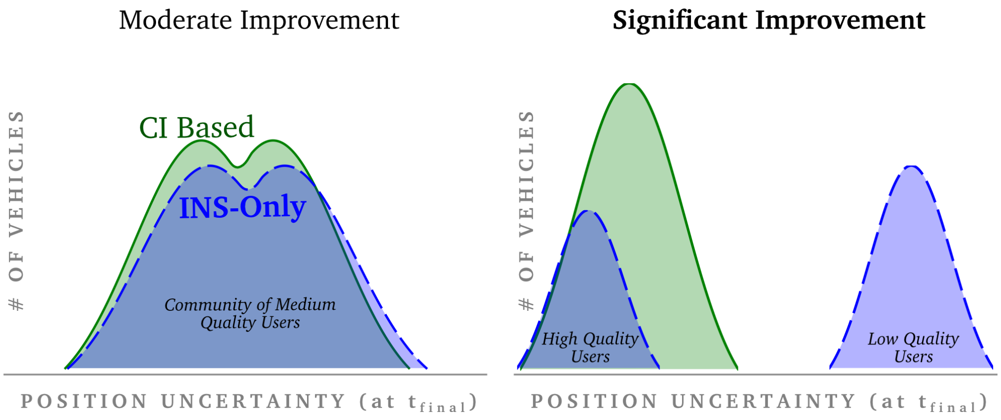


Figure 7.8: Comparison of final position uncertainty for large community of users, illustrating when CI is most advantageous.

in Figure 7.8. On the left-hand side, a community of users with only small variation in user quality is shown to benefit only moderately from CI. The simulation results presented in Figure 7.7 reflects this case, as the majority of users are of consumer-grade quality. Furthermore, the difference between tactical-grade and consumer-grade users is very moderate over the 30-45 seconds each vehicle travels.

The right-hand side figure shows a community with two groups of users: high and low quality users. The difference in user quality is significant, and therefore, the cooperative aiding using CI affords significant advantage to the community. This idea is confirmed by repeating the simulation of 2,169 vehicles, but assigning every-other vehicle with high and low quality sensors. The high and low quality assignment was accomplished by using navigation-grade and tactical-grade INS specifications. A more realistic alternative to the navigation-grade INS specification would be tactical-grade users equipped with additional sensors enabling the higher quality solution. As before, the community performance is evaluated by collecting the estimated $1-\sigma$ bound for each vehicle and plotting the mean and standard deviation as a function of true distance traveled, as shown in Figure 7.9.

Compared to the INS-only case, the CI-based decentralized solution provides more than 50% reduction in community uncertainty. This is a great improvement compared to the results for the relatively uniform community in Figure 7.7 where only 20% reduction was achieved.

The advantage afforded by the proposed CI-based decentralized estimator is best understood considering that without knowledge of inter-vehicle correlation, and with small communication and computational requirements, the high quality user information was used to improve the navigation solution of the entire community.

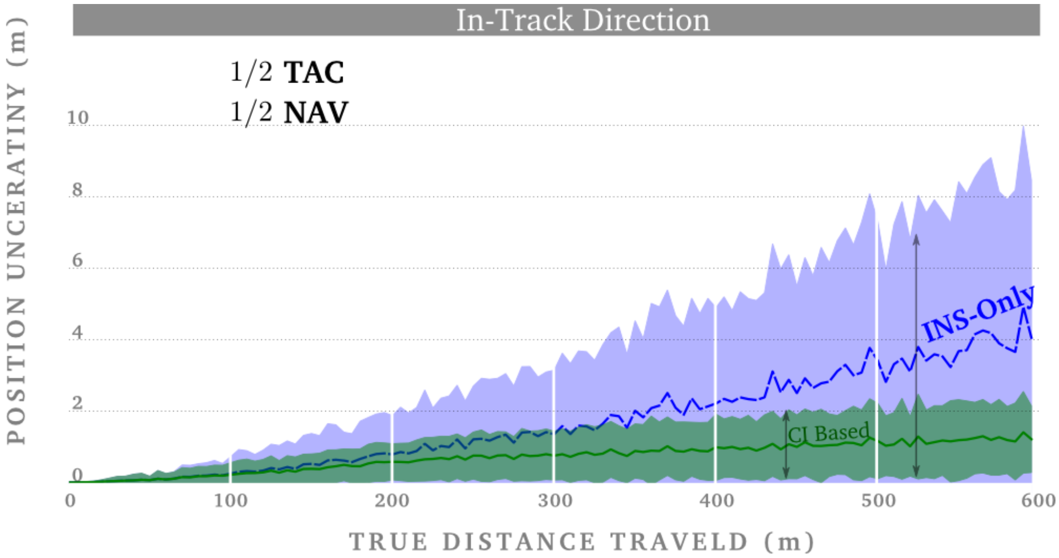


Figure 7.9: Performance of cooperative navigation community with both high and low quality users present.

7.2.5 BCInf Properties and Performance

The CI-based decentralized estimator had modest improvement gains as compared to the INS-only case. This was shown as part of the single vehicle performance in Figure 7.6 as well as the community average performance in Figure 7.7. Here we consider applying the BCInf filter by assuming knowledge on the size of possible inter-vehicle correlation. In this manner the cooperating aiding measurements will be used more aggressively as compared to the standard CI and hopefully improve the performance.

Picking Correlation Bound r_{max}

BCInf assumes knowledge on the allowable size of correlation as defined in Equation 4.39. Appendix A describes two procedures for picking r_{max} in a justifiable manner. The first procedure, namely using Monte Carlo runs, is used here. In this case 50 Monte Carlo runs of the cooperative aiding scenario were used. These are the same as was used to generate the results in Figure 7.6 with the exception that BCInf-based decentralized estimators were implemented in the community. The consistency analysis, however, focused on Vehicle 125 and the results of the consistency check are shown in Table 7.2.5. Although all other vehicle results were not systematically checked, several individual vehicle checks showed results similar to that of Vehicle 125. The choice of $r_{max}^2 = 0.15$ was selected as the smallest correlation bound able to maintain consistent estimates in both cross-track and in-track directions.

	r^2_{max}	Cross-Track (%)	In-Track (%)	Heading (%)	No. of Runs
(CI)	1	100	100	100	54
	0.5	100	100	98	52
	0.25	100	100	98	50
	0.15	**100**	**96**	**100**	**51**
	0.05	85	70	100	27
	0.03	70	40	100	20
	0.01	28	20	100	50
	0.005	45	0	100	20
	0.001	30	15	100	20
	0.0005	20	5	100	20
(EKF)	0	10	2	88	51

Table 7.1: Results of consistency check for Vehicle 125 in order to pick r_{max}.

Delving further into the results of Table 7.2.5 is insightful for understanding the effect of r_{max} on the decentralized-BCinf performance. Figure 7.10 plots the estimated uncertainty at t_f for vehicle 125 for various choices of r_{max}. Position, velocity, and heading states are included in the plot. The advantage of the inter-vehicle cooperative aiding increases as the bound r_{max} is tightened (i.e. decreased from r_{max} towards zero). The benefit to the velocity states follows closely the benefit to the position states. A simple example paralleling this more complex simulation was presented in Figure 4.5. Both position and velocity follow closely the expected results for an observed state. The direct relation between position and velocity means that the velocity state benefits in a manner similar to the position states despite using only position observations. The heading state in Figure 7.10 has a dramatic inflation in uncertainty which is not reflected in the simple example of Figure 4.5. The cause of this dramatic inflation in uncertainty was discussed in Section 4.4.5. As r_{max} is reduced, there is increased amounts of cooperative aiding throughout the community. As a result, the additional measurement updates lead to further inflation in uncertainty along the unobserved states like heading.

BCInf Community Performance

The choice of r_{max} was used to evaluate the community performance in Figure 7.11. As before this figure plots the average estimated $1-\sigma$ bound for each of the $2,169$ vehicles as a function of *true distance traveled* along the highway. The CI results are the same as those shown previously in Figure 7.7 but are included to show the performance gain from using BCInf. As a result of

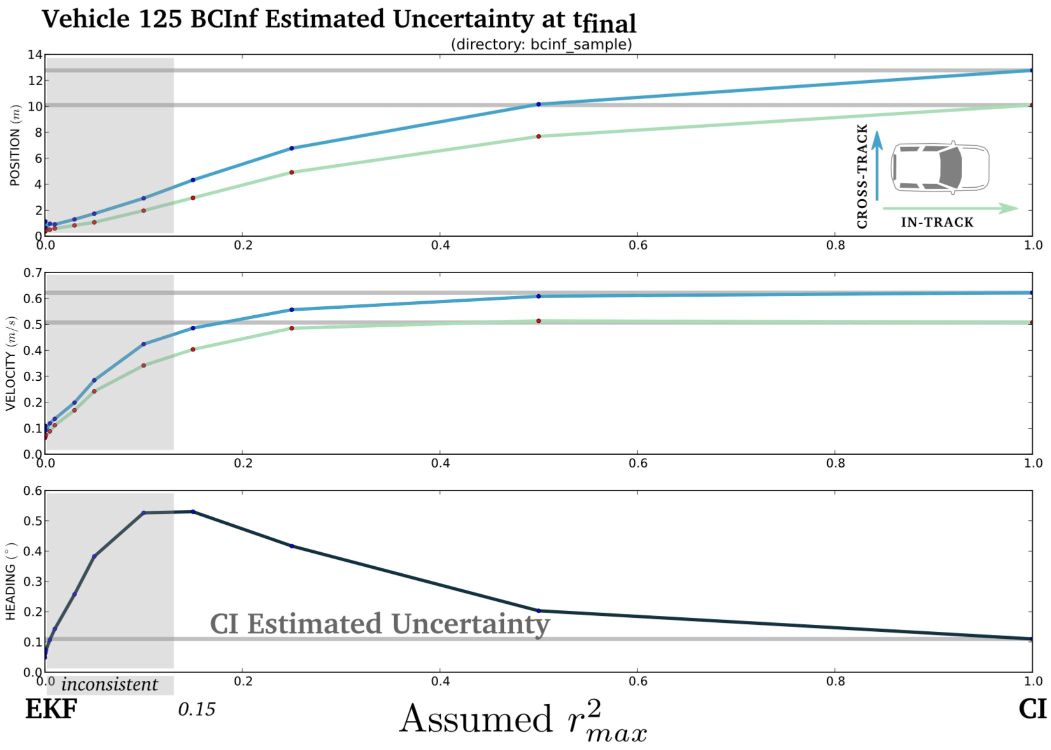


Figure 7.10: Estimated uncertainty in position and heading for Vehicle 125 as a function of r_{max} choice.

the assumed knowledge on the limited size of the unknown correlation between vehicles there is a more than 50% average reduction in in-track position uncertainty for the entire community. Though it is not shown, the performance gain along the cross-track direction is more modest, closer to 30%.

7.3 Bandwidth Requirement Analysis

In view of the dramatic improvement afforded by a centralized cooperative navigation scheme, one wonders whether barriers to its immediate use (i.e communication bandwidth) will cease to be an issue in the near future. To this end we do a quantitative evaluation of bandwidth requirements. We anchor our analysis on the IEEE 802.11p wireless standard described in Section 2.5.

The proposed decentralized filter only requires communication for measurement updates. The time update procedure is entirely local to the user. As previously described, at a minimum

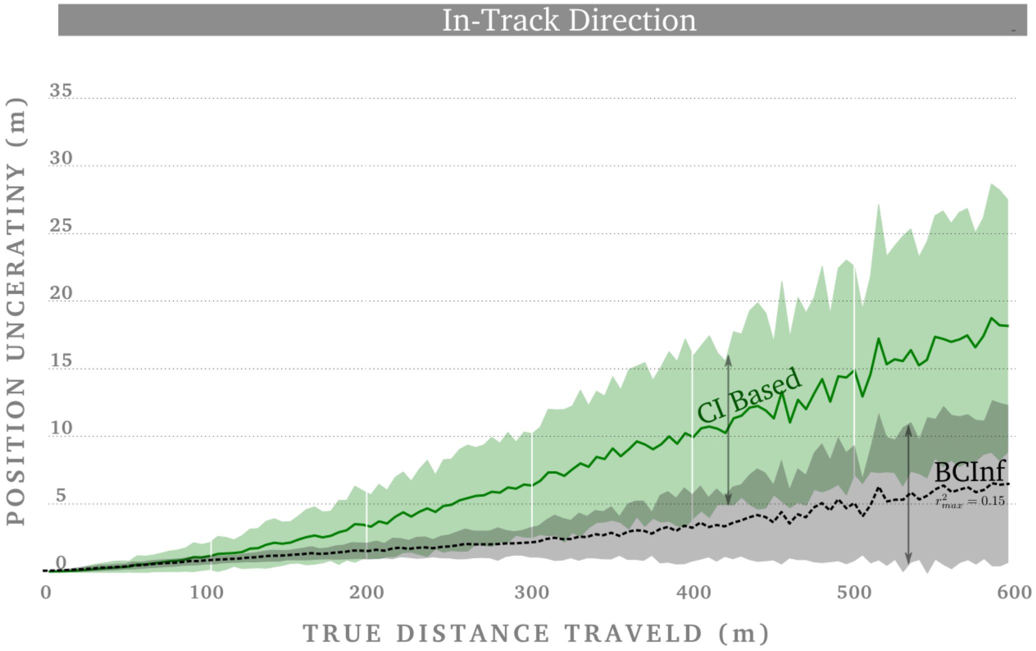
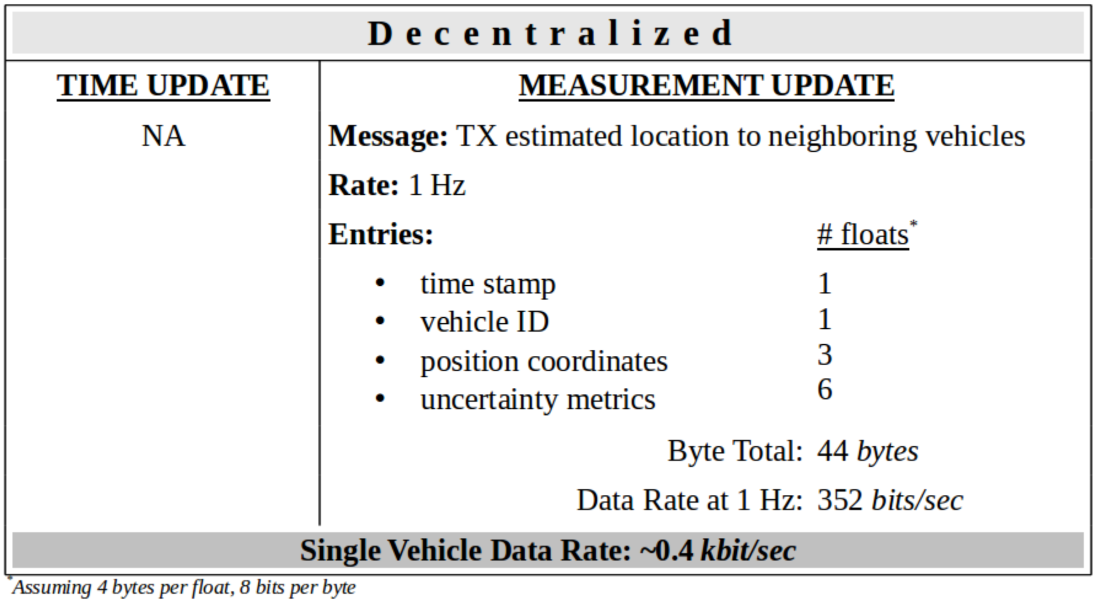


Figure 7.11: Comparison of CI and BCInf performance of a cooperative navigation community of 2,169 vehicles along in-track direction.

each vehicle must broadcast their current estimated position and associated uncertainty at regular intervals to enable cooperative aiding. We assume 3D positions and the covariance matrix as the uncertainty metric. The symmetric nature of the 3 × 3 position covariance means only 6 entries are transmitted. As detailed in Table 7.2, the raw-data bandwidth requirement for a single vehicle would be about 0.4 $kbit/sec$. This is the transmit (TX) requirement for a single vehicle. The transmit will be accompanied by a receive (RX) from neighboring vehicles. We'll assume there are 4 neighboring collaborators, hence bringing the total TX and RX bandwidth requirement to $(1 + 4) \times 0.4$ $kbit/sec$ or 2 $kbit/sec$. We have not accounted for any start and stop bits, however, the data requirements fall well within the average 6 $Mbit/sec$ capability of DSRC.

The centralized implementation is more involved. A naive implementation would require each vehicle to transmit all on board measurements (inertial, absolute, and relative) to the central filter for processing, and thereafter receive the corrected state estimate. In this case the high-rate inertial measurements (e.g. 50 Hz) could clearly be a problem. There is a significantly more efficient processing method for the centralized implementation where all high-rate processing is done on board the local vehicle and only low-rate covariance updates and measurement



Decentralized

TIME UPDATE	MEASUREMENT UPDATE
NA	**Message:** TX estimated location to neighboring vehicles **Rate:** 1 Hz

Entries:	# floats*
time stamp	1
vehicle ID	1
position coordinates	3
uncertainty metrics	6
Byte Total:	44 *bytes*
Data Rate at 1 Hz:	352 *bits/sec*

Single Vehicle Data Rate: ~0.4 *kbit/sec*

**Assuming 4 bytes per float, 8 bits per byte*

Table 7.2: Decentralized implementation broadcast requirements for a single vehicle.

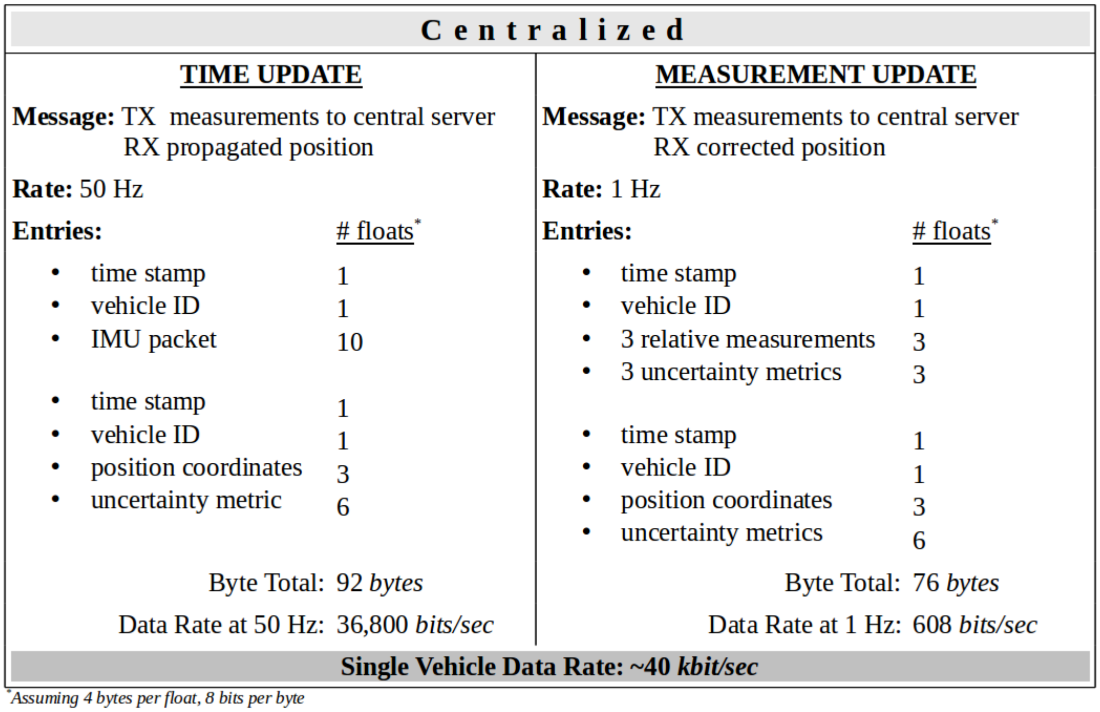
update corrections are done on the central processor (for example, computational complexity reduction techniques in [27]). This discussion and issues like data association and tracking for the automotive radar are beyond the scope of this work. For demonstration purposes, however, we present a basic data-link analysis for the naive central implementation.

The basic procedure for the centralized filter was graphically illustrated previously in Figure 2.1. In summary, the time update step requires:

1. Users transmit IMU data at 50 Hz to central processing station.
2. Central filter propagates community covariance and state.
3. Users receive current state and uncertainty metric from central filter.

The measurement update step:

1. Users transmit relative measurements and uncertainty metric to central processing station at 1 Hz.
2. Central filter computes and applies correction to community navigation states.
3. Users receive *a posteriori* state and associated uncertainty metric from central filter.



Centralized			
TIME UPDATE		**MEASUREMENT UPDATE**	
Message: TX measurements to central server RX propagated position		**Message:** TX measurements to central server RX corrected position	
Rate: 50 Hz		**Rate:** 1 Hz	
Entries:	# floats*	**Entries:**	# floats*
• time stamp	1	• time stamp	1
• vehicle ID	1	• vehicle ID	1
• IMU packet	10	• 3 relative measurements	3
		• 3 uncertainty metrics	3
• time stamp	1		
• vehicle ID	1	• time stamp	1
• position coordinates	3	• vehicle ID	1
• uncertainty metric	6	• position coordinates	3
		• uncertainty metrics	6
Byte Total:	92 *bytes*	Byte Total:	76 *bytes*
Data Rate at 50 Hz:	36,800 *bits/sec*	Data Rate at 1 Hz:	608 *bits/sec*
Single Vehicle Data Rate: ~40 *kbit/sec*			

Assuming 4 bytes per float, 8 bits per byte

Table 7.3: Centralized implementation broadcast requirements for a single vehicle.

Although the central filter may model navigation and sensor error states, only navigation states need to be broadcast back to the users. As before we assume a 3D position and the associated position covariance matrix. A modern calibrated consumer-grade IMU, like the Vector Nav VN100 is used to size the IMU message packet [62]. As detailed in Table 7.3, a time update would require 37 *kbit/sec* and the measurement update step, assuming 3 relative measurements, requires less than 1 *kbit/sec*. Hence, the total bandwidth requirement for a single vehicle in two-way communication with the central server is 40 *kbit/sec* . This requirement should be multiplied by the number of vehicles near a V2I transponder for the centralized system.

It is plausible that transponder antennas are installed with variable spacing to cover a given geographic area, all synchronized to the central processor. We'll assume a single transponder is providing coverage to the 1/2 mile stretch of U.S. Highway 101 presented earlier in Figure 7.1. The traffic density was analyzed and 124 was the median number of vehicles present in this section of highway, as shown in Figure 7.12. Hence, multiplying the *per vehicle* TX/RX requirement by 124 brings the total bandwidth requirement for the naive central implementation to just under 5 *Mbit/sec*. The nearness to limits of expected DSRC capabilities warrants including additional details in the analysis or seeking a more efficient implementation.

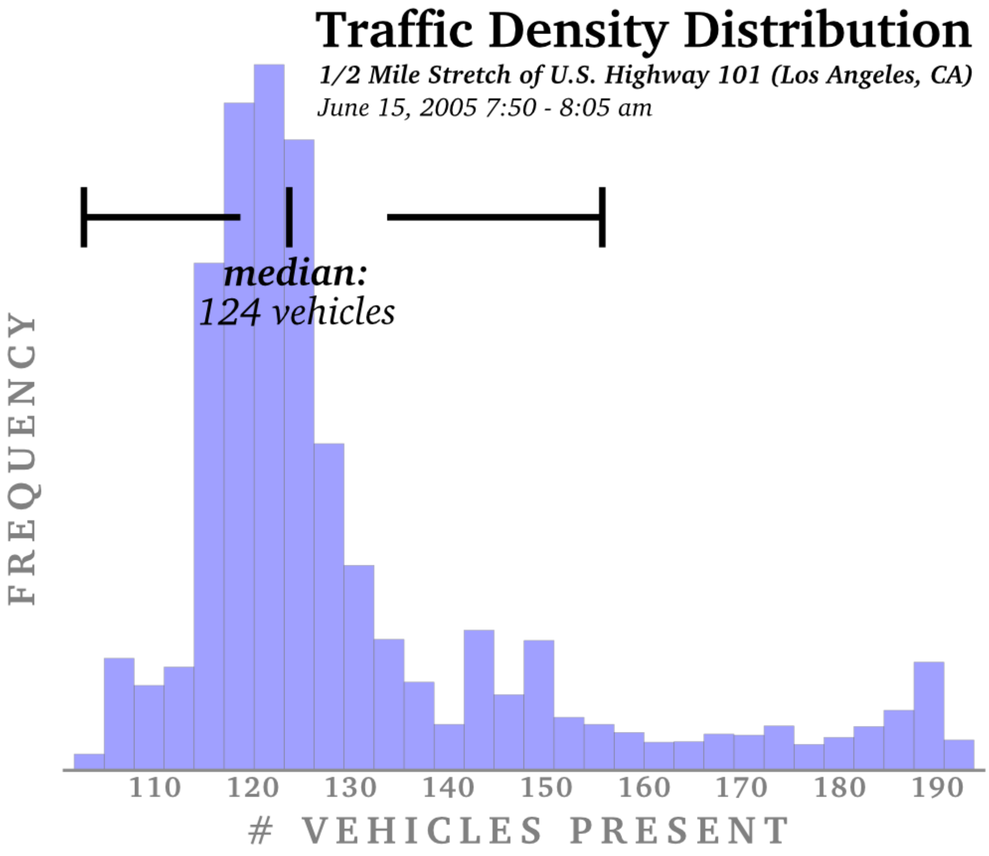


Figure 7.12: Traffic density during Wednesday morning on 5-lane major highway.

This simple analysis shows that our proposed decentralized estimator falls comfortably within modern connectivity capabilities. A naive implementation of the high-performance centralized estimator is also achievable, but it was demonstrated to be burdensome, nearing the bandwidth limits.

7.4 Conclusion

This chapter showed how a large and dynamic community of automotive users operating under limited communication bandwidth can use cooperative navigation to add utility to a low-cost INS on board each vehicle when GNSS is unavailable. The best demonstrated performance was using an EKF-based centralized estimator architecture where an order of magnitude improvement was attained over the non-cooperative case. Low-cost INS users maintained meter-level accuracy for a half-mile stretch of highway using only relative measurements and cooperative information exchange to arrest error growth. The CI-based decentralized estimator overcame traditional challenges of handling unknown correlation, like requiring source selection. The conservative nature of the decentralized estimator introduced limitations where partial information, like range-only measurements, can be insufficient to apply updates under unknown correlation. This caused the derivable benefit in a community of users to be modest, about 20%, depending on the inter-vehicle geometry and the state of neighboring users. If bounds on the size of the unknown correlation between vehicles are assumed, it was shown that BCInf could be applied and the derivable benefit improves to nearly 50% reduction in position uncertainty. It was shown, however, that a community composed of both high and low quality users implementing the CI-based decentralized estimator can derive significantly greater benefit. This is important as it does not require the challenging task of assuming knowledge bounding the size of the unknown correlations, as is the case for applying BCInf.

The last section of this chapter presented the bandwidth requirements of centralized and decentralized architectures in relation to available data links. Preliminary analysis showed both approaches fall within available bandwidth resources of upcoming navigation and communication utilities such as DSRC and have the potential to be feasible solutions in the near future.

Chapter 8

Conclusion

This work demonstrated how networked vehicles can operate in GPS-stressed or denied environment by way of cooperative aiding. Three applications were analyzed: commercial aviation, UAVs, and automotive traffic. The centralized implementation was based on a single EKF estimator while the decentralized implementation was based on unknown-correlation data fusion algorithms CI and BCInf. By applying CI/BCInf not only was their utility to enabling cooperative aiding measurements evaluated, strengths and limitations of these estimators were also uncovered. These conservative data fusion techniques are of little utility when there are nearly uniform accuracies present in the community. On the other hand, it was shown how unrestrictive decentralized implementations can utilize these estimators to share superior navigation accuracy through the entire community. As part of this work performance features particular to CI/BCInf were uncovered and discussed at length. Finally, an important covariance normalization step is proposed and demonstrated to be important for properly conducting the uncertainty trade-off inherent to CI/BCInf.

There is a trend of improved connectivity between systems and therefore barriers to the utility of cooperative navigation are quickly falling. There are, however, certain topics which must be addressed in order to make cooperative navigation feasible. Four topics that are important are handling delayed information, fault detection and integrity, covariance sharing, and less conservative correlated data fusion estimators. Each topic will be briefly described.

The UAV application case study presented in this work was largely based on playing back of experimental data. Transitioning these results to a real-time implementation must be able to handle information delay. Both centralized and decentralized implementations must handle the transit and processing times. This may required extending the message broadcasts to include velocity information so that short-term extrapolations can be carried out on the position broadcasts. The proper handling of delayed information is particularly important in large and dynamic communities, like the commercial aviation and automotive example, where the coupling of unknown correlations and measurement delays may lead to interesting phenomenon.

The effect of misrepresentative position broadcasts was illustrated as part of the UAV application case study. Both centralized and decentralized implementations must be able to detect and isolate faulty information. It is likely that what works for the centralized case, where there are an abundance of information which can be used to cross-compare results, will not be suited for the decentralized case where there are unknown correlations and each vehicle is only modeling ownship error-states. This, in turn, may impose requirements on the minimum number of collaborators required for decentralized fusion while ensuring solution integrity.

The position broadcasts in this work assumed the position covariance was also available. In 2-dimensional applications this is a 2×2 symmetric matrix whereas in 3-dimensional problems this is a 3×3 symmetric matrix. The evolving message standards like ADS-B for manned and unmanned aviation or DSRC for automotive applications may define a scalar uncertainty parameters. Hence, the decentralized implementation must be able to operate with scalar uncertainty representations. Methods to maintain consistency, while limiting the sacrifice of performance, when using scalar uncertainty representations in cooperative navigation should be investigated.

Lastly, this work showed that decentralized implementations based on CI/BCInf were of little utility if all users are of similar sensing and navigation performance. In order to be effective, these estimators require one or more users be equipped with higher quality position estimates. The centralized implementations, however, demonstrates that even a community of homogeneous users can significantly benefit from cooperative aiding. The conservative nature of the decentralized correlated data fusion filters leads to the current results. Alternative techniques, possibly based on partial correlation information, may help extract better performance from the decentralized estimators. Though the decentralized implementation will understandably be worse than the centralized, the goal is to reduce the performance gap.

References

[1] Demoz Gebre-Egziabher and Brian Taylor. Impact and Mitigation of GPS-Unavailability on Small UAV Navigation, Guidance and Control. Technical Report Report #2012-2, November 2012.

[2] I. Gresham, A. Jenkins, R. Egri, C. Eswarappa, N. Kinayman, N. Jain, R. Anderson, F. Kolak, R. Wohlert, S.P. Bawell, J. Bennett, and J.-P. Lanteri. Ultra-wideband radar sensors for short-range vehicular applications. *Microwave Theory and Techniques, IEEE Transactions on*, 52(9):2105–2122, Sept 2004.

[3] Robert Bosch GmbH. LRR3: 3rd generation Long-Range Radar Sensor. http://www.bosch-automotivetechnology.com/media/db_application/downloads/pdf/safety_com/en_com/lrr3_datenblatt_de_com009.pdf, 2009. [Online Datasheet; accessed 14-April-2013].

[4] Cambridge Systematics, Inc. Summary Report: NGSIM U.S. 101 Data Analysis (7:50 a.m. to 8:05 a.m.). http://www.ngsim-community.org/, December 2005. [Online; accessed 15-February-2013].

[5] Scott Gleason and Demoz Gebre-Egziabher. *GNSS Applications and Methods*. Artech House, July 30 2009.

[6] Chris Pocock. UAV Crash in Korea Linked To GPS Jamming . http://www.ainonline.com/aviation-news/ain-defense-perspective/2012-06-01/uav-crash-korea-linked-gps-jamming, June 1 2012. [Online; accessed 1-January-2013].

[7] G. F. Gobbini. *Relative Navigation by means of Passive Ranging*. PhD thesis, Massachusetts Institute of Technology, 1981.

[8] Thomas H. Kerr and Leonard Chin. A Stable Decentralized Filtering Implementation for JTIDS Relnav. In *IEEE PLANS*, 1980.

[9] The Analytic Sciences Corporation. *Applied Optimal Estimation*. The MIT Press, 5 1974.

[10] W. R. Fried and R. Loeliger. Principles, System Configuration and Algorithm Design of the Inertially Aided JTIDS Relative Navigation Function. *Navigation: Journal of The Institute of Navigation*, 26(3):224–236, 1979.

[11] H. J. Rome and J. S. Stambaugh. Evaluation of the Accuracy of Community Relative Navigation Organization Concepts. *Navigation: Journal of The Institute of Navigation*, 24(2):168–180, summer 1977.

[12] E. E. Pentecost and A. R. Stubberud. Synthesis of Computationally Efficient Sequential Linear Estimators. *IEEE Transactions on Aerospace and Electronic Systems*, 3(2):242–249, March 1967.

[13] M. M. Shah. *Sub-optimal Filtering Theory for Interacting Control Systems*. PhD thesis, Cambridge University, England, 1971.

[14] L. Chin and A. Fam. Global stability/local optimality trade-offs in a class of decentralized integrated communication-navigation filters. In *Decision and Control, 1983. The 22nd IEEE Conference on*, volume 22, pages 347–352. IEEE, 1983.

[15] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In I. J. Cox and G. T. Wilfong, editors, *Autonomous Robot Vehicles*, volume 8, pages 167–193. Springer New York, 1990.

[16] J. A. Castellanos, J.D. Tardos, and G. Schmidt. Building a global map of the environment of a mobile robot: the importance of correlations. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, volume 2, pages 1053–1059 vol.2, Apr 1997.

[17] S. J. Julier and J. K. Uhlmann. A Non-divergent Estimation Algorithm in the Presence of Unknown Correlations. In *Proceedings of the American Control Conference*, June 1997.

[18] L.C. Carrillo-Arce, E.D. Nerurkar, J.L. Gordillo, and S.I Roumeliotis. Decentralized multi-robot cooperative localization using covariance intersection. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1412–1417, Nov 2013.

[19] P. O. Arambel, C. Rago, and R. K. Mehra. Covariance Intersection Algorithm for Distributed Spacecraft State Estimation. In *Proceedings of the American Control Conference*, pages 4398–4403, June 2001.

[20] Julio Cesar Bolzani de Campos Ferreira and Jacques Waldmann. Covariance intersection-based sensor fusion for sounding rocket tracking and impact area prediction. *Control Engineering Practice*, 15(4):389–409, 2007.

[21] S. J. Julier and J. K. Uhlmann. Simultaneous Localisation and Map Building Using Split Covariance Intersection. In *Proceedings of the 2001 IEEE/RSJ*, pages 1257–1262, Oct. 29 - Nov. 03 2001.

[22] Hao Li and Fawzi Nashashibi. Cooperative multi-vehicle localization using split covariance intersection filter. In *Intelligent Vehicles Symposium*, pages 211–216. IEEE, 2012.

[23] U.D. Hanebeck, K. Briechle, and J. Horn. A tight bound for the joint covariance of two random vectors with unknown but constrained cross-correlation. In *Multisensor Fusion and Integration for Intelligent Systems, 2001. MFI 2001. International Conference on*, pages 85–90, 2001.

[24] S. Reece and S. Roberts. Robust, low-bandwidth, multi-vehicle mapping. In *Information Fusion, 2005 8th International Conference on*, volume 2, pages 8 pp.–, July 2005.

[25] S.J. Julier. Estimating and exploiting the degree of independent information in distributed data fusion. In *Information Fusion, 2009. FUSION '09. 12th International Conference on*, pages 772–779, July 2009.

[26] A. Bahr, M. R. Walter, and J. J. Leonard. Consistent Cooperative Localization. pages 3415–3422, May 12-17 2009.

[27] Esha D Nerurkar and Stergios I Roumeliotis. Power-SLAM: a linear-complexity, anytime algorithm for SLAM. *International Journal of Robotics Research*, 30(6):772–788, 2011.

[28] M.E.L.D.L.H. James Llinas. *Handbook of Multisensor Data Fusion: Theory and Practice, Second Edition*. Electrical Engineering & Applied Signal Processing Series. Taylor & Francis, 2008.

[29] S.I Roumeliotis and George A Bekey. Distributed multirobot localization. *Robotics and Automation, IEEE Transactions on*, 18(5):781–795, Oct 2002.

[30] K. Y. K. Leung. *Cooperative Localization and Mapping in Sparsely-Communicating Robot Networks*. PhD thesis, University of Toronto, 2012.

[31] S. Lo and P. Enge. Study of Two Potential Alternative Position Navigation and Timing (APNT) Technologies for Aviation. In *Proceedings of Institute of Navigation International Technical Meeting*. Institute of Navigation, Jan 2011.

[32] Federal Aviation Administration. *Automatic Dependent Surveillance - Broadcast (ADS-B) Out Performance Requirements to Support Air Traffic Control (ATC) Service: Final Rule*. 14 CRF Part 91. Department of Transportation, May 28 2010.

[33] Fact Sheet – Automatic Dependent Surveillance-Broadcast. http://www.faa.gov/news/fact_sheets/news_story.cfm?newsid=7131, May 27 2010. [Online; accessed 1-January-2013].

[34] RTCA Special. *Minimum Operational Performance Standards for 1090 MHz Extended Squitter Automatic Dependent Surveillance – Broadcast (ADS-B) and Traffic Information Services – Broadcast (TIS-B)*. RTCA, Inc., 2009.

[35] Presentation on and Proposal for Handling of Time Registration, as presented by Bob Grappel of MIT Lincoln Lab. http://adsb.tc.faa.gov/WG4_Meetings/ASSAP_MOPS/Meeting03.htm, May 2006. [Online; accessed 1-January-2013].

[36] Federal Aviation Administration. Automatic Dependent Surveillance-Broadcast (ADS-B) Out equipment performance requirements. http://www.ecfr.gov/cgi-bin/text-idx?c=ecfr&SID=46d2c751c3d0a240734a71d9c70450a0&tpl=/ecfrbrowse/Title14/14cfr91_main_02.tpl, 2013. [Online; accessed 1-January-2013].

[37] ADS-B Aviation Rulemaking Committee. Report From the ADS-B Aviation Rulemaking Committee to the Federal Aviation Administration. http://www.faa.gov/nextgen/implementation/programs/adsb/media/arcReport2008.pdf, September 2008. [Online; accessed 1-January-2013].

[38] RTCA Committee-185Am. *Minimum Operational Performance Standards for TCAS II Airborne Equipment*. RTCA, Inc., 1997.

[39] Demoz Gebre-Egziabher. Rpv/uav surveillance for transportation management and security. *University of Minnesota Intelligent Transportation Systems Institute Report*, (CTS 08-27), 2008.

[40] Zhen Zhu, S. Roumeliotis, J. Hesch, H. Park, and D. Venable. Architecture for asymmetric collaborative navigation. In *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION*, pages 777–782, April 2012.

[41] J. R. Wilson. Stealth Sneaks into UCAV. *Aerospace America*, 51(6):28–33, June 2013.

[42] C.M. Haissig. Military formation flight as a model for increased capacity in civilian airspace. In *Digital Avionics Systems Conference, 2004. DASC 04. The 23rd*, volume 1, pages 1.C.4–1.1–9 Vol.1, Oct 2004.

[43] J. K. Kuchar. Update on the Analysis of ACAS Performance on Global Hawk. Technical Report SCRSP WG A/WP A10-04, Aeronautical Surveillance Panel, May 1-5 2006.

[44] Bradley D Farnsworth and David WA Taylor. High precision narrow-band rf ranging. In *Proceedings of the 2010 International Technical Meeting of The Institute of Navigation*, pages 161–166, 2001.

[45] ENSCO Inc. High-Precision 2.4 GHz DSSS RF Ranging. http://www.ensco.com/userfiles/file/Products_Services_PDF/05_GPS-Denied/High-Precision-DSSS-RF-Ranging-ENSCO-Geolocation-Navigation-Tech-Paper.pdf. [Online; last accessed 24-June-2014].

[46] Sagetech Corporation. UAS Unmanned Aircraft & ADS-B. http://www.sagetechcorp.com/ads-b-technology/uas-unmanned-aircraft-ads-b.cfm. [Online; last accessed 31-May-2014].

[47] J. Wenger and S. Hahn. Long range and ultra-wideband short range automotive radar. In *Ultra-Wideband, 2007. ICUWB 2007. IEEE International Conference on*, pages 518–522, Sept 2007.

[48] U.S. Department of Transportation. Intelligent Transportation Systems - Dedicated Short Range Communications. http://www.its.dot.gov/DSRC/. [Online; last accessed 11-July-2013].

[49] A. Paier, R. Tresch, A. Alonso, D. Smely, P. Meckel, Y. Zhou, and N. Czink. Average downstream performance of measured ieee 802.11p infrastructure-to-vehicle links. In *Communications Workshops (ICC), 2010 IEEE International Conference on*, pages 1–5, 2010.

[50] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press New York, 1970.

[51] M. Kayton and W. R. Fried. *Avionics Navigation Systems*. John Wiley & Sons, Inc, 1997.

[52] Lingji Chen, P.O. Arambel, and R.K. Mehra. Estimation under unknown correlation: covariance intersection revisited. *Automatic Control, IEEE Transactions on*, 47(11):1879–1882, 2002.

[53] D. Simon. *Optimal State Estimation*. John Wiley & Sons, Inc, 2006.

[54] Mokhtarzadeh, H. Reference Position and Attitude together with Raw Sensor Data from Seven Small UAV Flights during 2011-12. http://hdl.handle.net/11299/165567, Sept 2014. [Online; University of Minnesota Digital Conservancy].

[55] S. Bhattacharyya. *Performance and Integrity Analysis of the Vector Architecture of GNSS-Receivers.* PhD thesis, Department of Aerospace Engineering & Mechanics, University of Minnesota, Twin Cities Campus, 2012.

[56] D. Gebre-Egziabher, R.C. Hayward, and J.D. Powell. Design of multi-sensor attitude determination systems. *Aerospace and Electronic Systems, IEEE Transactions on*, 40(2):627–649, April 2004.

[57] Jay Farrell. *Aided navigation: GPS with high rate sensors.* McGraw-Hill New York, 2008.

[58] Demoz Gebre-Egziabher, C.O. Boyce, J David Powell, and Per Enge. An inexpensive dme-aided dead reckoning navigator. *Navigation*, 50(4):247–263, 2003.

[59] Z. Xing. *Over-bounding Integrated INS/GNSS Output Errors.* PhD thesis, Department of Aerospace Engineering & Mechanics, University of Minnesota, Twin Cities Campus, 2010.

[60] Hamid Mokhtarzadeh and Demoz Gebre-Egziabher. Cooperative inertial navigation. *Navigation, Journal of the Institute of Navigation*, 2014. Accepted for Publication: March 2014.

[61] P.D. Groves. *Principles of GNSS, inertial, and multi-sensor integrated navigation systems.* GNSS technology and applications series. Artech House, 2008.

[62] VectorNav. VN -100 User Manual. Technical Report Rev 1.2.8.

[63] Computer Sciences Corporation. Attitude Systems Operation and J.R. Wertz. *Spacecraft Attitude Determination and Control.* Astrophysics and Space Science Library : a series of books on the recent developments of space science and of general geophysics and astrophysics. Reidel, 1978.

[64] Analog Devices. Triaxial Inertial Sensor with Magnetometer ADIS16400/ADIS16405 - User Manual. Technical Report Rev B, 2009.

[65] Haim Baruh. *Analytical dynamics.* WCB McGraw-Hill Boston, 1999.

Appendix A

Methods to Pick Correlation Bound

Section 4.4.5 discussed the challenges of picking the correlation bound r_{max} for BCInf estimators. Here two practical offline analysis methods are proposed for picking r_{max}. The first is a Monte Carlo method and the second is based on keeping track of true correlations.

A.1 Monte Carlo Method

The goal is to find the smallest bound r_{max} for which the BCInf estimator results are still consistent. This means that the estimator covariance is capable of bounding the true error over multiple realizations. The following steps are used to determine the correlation bound r_{max}:

1. Repeated Monte Carlo runs are conducted for each choice of r_{max} used in the decentralized BCInf implementation.

2. The consistency of the results for each run is checked:

   - Run is declared consistent if the error realizations are bounded by the estimated $2-\sigma$ bound for all estimator error-states.

3. The percentage of consistent runs are compiled for each choice of r_{max}.

4. The final choice of r_{max} is picked by selecting the smallest value which still satisfies the desired consistency requirements.

This technique depends on the ability to run large numbers of Monte Carlo simulations and may be computationally burdensome. An alternative method is proposed based on keeping track of the true correlations that evolve as the decentralized BCInf implementation is used.

A.2 Maintaining True Correlations

A modified centralized filter which models all states present in the problem (e.g. the states for all vehicles), is run in parallel to the decentralized BCInf implementation. The centralized filter is used only to maintain the true inter-state correlations that evolve but are not maintained in the decentralized filter. Hence the centralized filter is used to conduct a covariance analysis on the decentralized BCInf implementation. In this manner P_{xy} is maintained and the matrix of correlation coefficients C_{yx} will be known and a bounding r_{max} can be computed in the manner discussion in Section 4.3.4.

The centralized filter model given in Equation 3.5 is used, where as before the system and stochastic matrices are in the block diagonal form defined by Equation 3.6. The time-update covariance propagation is unchanged. The measurement update, however, will be different. The centralized filter has no measurement update of its own. Instead the centralized filter keeps track of all the correlations that form as a result of the decentralized BCInf measurement updates. This is done in a sequential manner for all the measurements which are fused as part of the decentralized BCInf systems.

A general decentralized measurement updated by vehicle i using scalar relative measurements (e.g. range) from vehicle j will be of the form:

$$\begin{aligned} \hat{\mathbf{x}}_i^+ &= \hat{\mathbf{x}}_i^- + K\tilde{y}_{ij} \\ &= \hat{\mathbf{x}}_i^- + K(\mathbf{H}_i\tilde{\mathbf{x}}_i^- + \mathbf{H}_j\tilde{\mathbf{x}}_j^- + v) \qquad \text{(A.1)} \end{aligned}$$

where K is measurement gain computed by the decentralized BCInf estimator on board ownship and $\tilde{y}_{ij}$ is the measurement innovation defined previously in Equation 3.10. Reference to the current epoch has been dropped for simplicity and as before tilde is used to represent the difference between the true and estimated quantity. Equation A.1 can be written exclusively using error states. This is done by subtracting both sides from the true ownship state $\mathbf{x}_i$:

$$\begin{aligned} \mathbf{x}_i - \hat{\mathbf{x}}_i^+ &= \mathbf{x}_i - \left(\hat{\mathbf{x}}_i^- + K(\mathbf{H}_i\tilde{\mathbf{x}}_i^- + \mathbf{H}_j\tilde{\mathbf{x}}_j^- + v)\right) \\ \tilde{\mathbf{x}}_i^+ &= (I - K\mathbf{H}_i)\tilde{\mathbf{x}}_i^- - K\mathbf{H}_j\tilde{\mathbf{x}}_j^- - Kv \qquad \text{(A.2)} \\ & \qquad \text{(A.3)} \end{aligned}$$

Equation A.2 shows how the error state update of vehicle i depends on *a priori* error states of both vehicles i and j, as well as measurement noise v.

A decentralized implementation would be unable to maintain both $\tilde{\mathbf{x}}_i^+$, $\tilde{\mathbf{x}}_j^+$ as well as their correlations. However, by running a centralized filter in parallel one can update the covariance based on the update defined in Equation A.2. To facilitate this update Equation A.2 is rewritten to be in terms of the complete centralized state vector modeling all vehicles:

$$\begin{aligned}\begin{bmatrix}\tilde{\mathbf{x}}_1\\ \vdots\\ \tilde{\mathbf{x}}_i\\ \vdots\\ \tilde{\mathbf{x}}_j\\ \vdots\\ \tilde{\mathbf{x}}_N\end{bmatrix}^+ &= \left(\mathbf{I} - \begin{bmatrix} & & & O & & & \\ \mathbf{0} & \dots & KH_i & \dots & KH_j & \dots & \mathbf{0}\\ & & & & & & \\ & & O & & & & \end{bmatrix}\right)\begin{bmatrix}\tilde{\mathbf{x}}_1\\ \vdots\\ \tilde{\mathbf{x}}_i\\ \vdots\\ \tilde{\mathbf{x}}_j\\ \vdots\\ \tilde{\mathbf{x}}_N\end{bmatrix}^- + \begin{bmatrix}\mathbf{0}\\ \vdots\\ K\\ \vdots\\ \mathbf{0}\\ \vdots\\ \mathbf{0}\end{bmatrix} v\\ &= \mathbf{K}_X\mathbf{X}^- + \mathbf{K}_R v \end{aligned} \tag{A.4}$$

where $\mathbf{I}$ is an $N \cdot n$ identity matrix and $\mathbf{K}_X, \mathbf{K}_R$ are defined to simplify notation. Matrix $\mathbf{K}_X$ is an $N \cdot n$ square matrix. If the measurement of vehicle i is a scalar measurement (e.g. range), then v is simply a scalar and matrix $\mathbf{K}_R$ is of dimension $(N \cdot n) \times 1$.

It is useful to reflect on Equation A.4 prior to presenting the associated covariance update. The $n \times 1$ matrix gain K is the gain computed by vehicle i as part of the decentralized BCInf fusion of the range measurement to vehicle j. Therefore Equation A.4 reflects an update to the error states of vehicle i using a combination of *a priori* error states for both vehicle i and j. This is different than the measurement update equations for a centralized filter. Fusing a range measurement in a centralized filter implementation would lead to updates to both vehicle i and j error states. Equation A.4, reflects the decentralized implementation where the update computed locally only updates the states of ownship. All other vehicle states are modeled in order to maintain the true covariance and analyze the effect of the decentralized updated. Having defined the measurement update, the true community covariance can be propagated:

$$\mathbf{P}^+ = \mathbf{K}_X\mathbf{P}^-\mathbf{K}_X^T + v\mathbf{K}_R\mathbf{K}_R^T \tag{A.5}$$

In conclusion a single centralized estimator run in parallel to the decentralized BCInf implementation can be used to maintain the true correlations that form. The covariance must be updated by Equation A.5 for each inter-vehicle decentralized measurement update that takes place in the community. In this manner the true correlations will be known and r_{max} can be selected to find the tightest bound on the correlations. This process will be iterative until the assumed decentralized BCInf r_{max} upper bounds the true correlations that form.

Appendix B

2D INS Equations

A 2D flat-earth 11-state INS model is developed here. This 11-state model is used for the Chapter 7 automotive simulation results. An 8-state variant of this 2-D model, where the three sensor null shift terms are excluded, was used for the commercial aviation simulation presented in Section 5.2. Table B.1 lists the states modeled as well as the initial standard deviations assumed for the simulation.

		Initial Standard Deviation
Error State	**Description**	(consumer, tactical, navigation)
N	North position	0
E	East position	0
V_N	North velocity	0
V_E	East velocity	0
ψ	heading	0
n_{ax}	x-accel null shift	$(1 \times 10^{-3}, 5 \times 10^{-4}, 5 \times 10^{-6})\ g$
d_{ax}	x-accel bias drift	$(1.2 \times 10^{-3}, 5 \times 10^{-5}, 2.5 \times 10^{-5})\ g$
n_{ay}	y-accel null drift	*same as x-axis*
d_{ay}	y-accel bias drift	*same as x-axis*
n_g	z-gyro null shift	$(0.05, 1.7 \times 10^{-3}, 8 \times 10^{-4})\ \circ/s$
d_g	z-gyro bias drift	$(180, 15, 3 \times 10^{-3})\ \circ/hr$

Table B.1: INS states modeled for each vehicle

The nonlinear dynamic model for position and heading are:

$$\begin{aligned}\ddot{N}(t) &= \cos\psi(t)a_x(t) - \sin\psi(t)a_y(t) \\ \ddot{E}(t) &= \sin\psi(t)a_x(t) - \cos\psi(t)a_y(t) \\ \dot{\psi}(t) &= \omega(t) \qquad \text{(B.1)}\end{aligned}$$

where a_x are a_y are the accelerometer measurements in the body axes, and ω is the vertical gyro measurement. The measured quantities are naturally the sum of the truth and sensor errors. A constant null-shift (n) and a bias drift (d), modeled as a first order Markov process, is the assumed sensor error model. Error model equations for a general sensor axis u, are:

$$\begin{aligned}\Delta u(t) &= n + d(t) + w(t) \qquad \text{(B.2)} \\ \dot{d}(t) &= -\frac{1}{\tau}d(t) + w_1(t) \qquad \text{(B.3)}\end{aligned}$$

where τ is assumed 100 and 300 *seconds* for the consumer accelerometers and gyro, 60 and 100 *seconds* for the tactical grade accelerometers and gyro, and 3600 for both the navigation grade accelerometers and gyro. The noise statistics for the sensor error models are selected to match the initial standard deviations in Table B.1.

The state propagation is completed by first-order Euler integration of Equations B.1. The covariance propagation requires the linearized version of the state space model, as defined in Equation 3.2. The linearized system matrix for a single vehicle is:

$$\begin{aligned}\mathbf{F}_i &= \\ &\left[\begin{array}{ccccc|cccccc} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & a_1 & c\hat{\psi} & c\hat{\psi} & -s\hat{\psi} & -s\hat{\psi} & 0 & 0 \\ 0 & 0 & 0 & 0 & a_2 & s\hat{\psi} & s\hat{\psi} & c\hat{\psi} & c\hat{\psi} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{\tau_a} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{\tau_a} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{\tau_g} \end{array}\right] \\ a_1 &= -\hat{a}_x \sin\hat{\psi} - \hat{a}_y \cos\hat{\psi} \\ a_2 &= -\hat{a}_x \cos\hat{\psi} - \hat{a}_y \sin\hat{\psi}\end{aligned}$$

where the $\hat{hat}$ is used to denote the best current estimate, or the corrected sensor quantity. For clarity, shorthands for *sin* and *cos* have been used and the time dependence of terms like heading or sensor measurements are excluded. The associated input mapping matrix:

$$\mathbf{G}_i \quad = \quad \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ c\hat{\psi} & -s\hat{\psi} & 0 & 0 & 0 & 0 \\ s\hat{\psi} & c\hat{\psi} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

where the driving noise vector is $\mathbf{w}_i(t) = [w_{ax}, w_{ay}, w_g | w_{1ax}, w_{1ay}, w_{1g}]^T$. The process noise is defined as $\mathbf{Q}_i = E\left\{\mathbf{w}_i(t)\mathbf{w}_i^T(t)\right\}$. Finally, the covariance propagation is completed using Equations 3.4.

Appendix C

FAA Flight Data

National airspace data was obtained through two separate data requests to the FAA. The data covers a 24-hour period on February 1, 2005 and represents flights in the National Airspace System's domestic airspace. The first data set came from the Federal Aviation Administration's Enhanced Traffic Management System (ETMS) database, and includes all flights operating under the *commercial flight* user class. The second data set come from the FAA's Traffic Flow Management System (TFMS) database, and includes flights from all possible user classes, excluding military flights. These user classes are: *Commercial*, *General Aviation*, *Cargo*, *Air Taxi*, and *Other*. It should be noted that both ETMS and TFMS databases only have records for flights which operated under instrument flight rules (IFR) or where handled by air traffic control in a radar environment.

It was expected that the second (all user classes) data set be all encompassing and should include all the commercial flights within the first (commercial only) data set. While this was true for about $17,500$ commercial flights, it was discovered an additional $7,500$ commercial flights were not present. To resolve this discrepancy we merged the two data sets. The merged data set was used in the analyses presented in this paper. The total number of unique flights in this data set for each user class is shown in figure C.1.

The combined records are stored as a single comma separated value (CSV) file. Once duplicate line entries have been removed, there are $3,980,742$ lines of flight record data. Each line corresponds to a radar hit for a unique flight and the sampling interval is approximately 60 seconds. A sample snippet of the data format is shown:

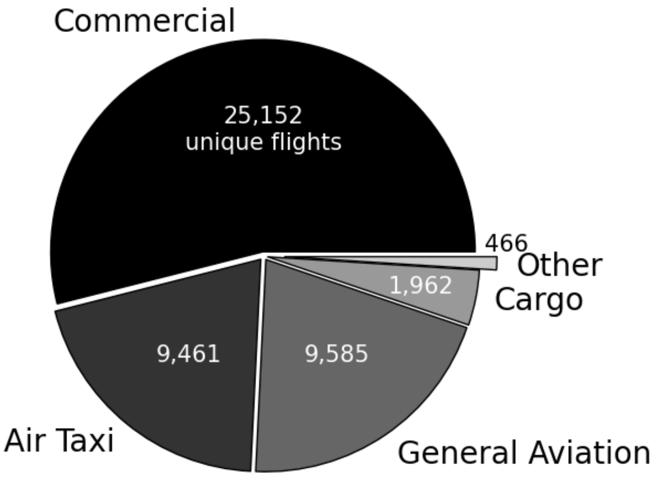


Figure C.1: Number of unique flights for each user class in the complete data set. There was a total of $46,626$ unique flights.

```
AIRCRAFT_ID,ETMS_FLIGHT_INDEX,UNIQUE_FLIGHT_ID,DATE,DEPT_APRT,DEPT_DATE_GMT,DEPT_TIME_GMT,ARR_APRT,
    ARR_DATE_GMT,ARR_TIME_GMT,USER_CLASS,PHYSICAL_CLASS,ACFT_TYPE,RADAR_HIT_DATE_GMT,
    RADAR_HIT_TIME_GMT,LATITUDE,LONGITUDE,ALTITUDE
AAH441,102156,20050201663062,01-FEB-05,OAK,20050201,15:44:00,HNL,20050201,20:54:28,Commercial,Jet,
    B737,20050201,15:45:09,37.75,-122.28333,20
AAH441,102156,20050201663062,01-FEB-05,OAK,20050201,15:44:00,HNL,20050201,20:54:28,Commercial,Jet,
    B737,20050201,15:46:28,37.75,-122.4,43
AAH441,102156,20050201663062,01-FEB-05,OAK,20050201,15:44:00,HNL,20050201,20:54:28,Commercial,Jet,
    B737,20050201,15:47:28,37.71666,-122.5,68
AAH441,102156,20050201663062,01-FEB-05,OAK,20050201,15:44:00,HNL,20050201,20:54:28,Commercial,Jet,
    B737,20050201,15:48:28,37.68333,-122.6,93
AAH441,102156,20050201663062,01-FEB-05,OAK,20050201,15:44:00,HNL,20050201,20:54:28,Commercial,Jet,
    B737,20050201,15:49:28,37.65,-122.7,112
AAH441,102156,20050201663062,01-FEB-05,OAK,20050201,15:44:00,HNL,20050201,20:54:28,Commercial,Jet,
    B737,20050201,15:50:28,37.61666,-122.83333,139
```

Appendix D

Attitude Heading Reference System

Here the details of a stand-alone AHRS system designed for low-cost remotely piloted or unmanned aircraft where attitude estimates are reliably available without dependence on GNSS-measurements is described. The AHRS design was inspired by the work in [56, 57, 5] and is implemented using an Extended Kalman Filter (EKF). First the system architecture and the attitude dynamic model are presented followed by details of the aiding measurements. The importance of both proper error-modeling and the handling of disturbances like aircraft acceleration and magnetic-field disturbances will be presented.

D.1 System Architecture

Attitude parameters specify the orientation of one coordinate frame relative to a reference coordinate frame. In aerospace applications, this commonly a specification of body-frame orientation with respect to the navigation-frame. A local geodetic or tangent plane, like the North-East-Down (NED)frame is the common navigation reference-frame used when specifying vehicle attitude [57]. The AHRS presented specifies the orientation of the aircraft body-frame with respect to the local NED frame.

Attitude Parameterization	Notation	Notes
Direction Cosine Matrix	R_n^b	3×3 rotation matrix *(nav to body)*
3-2-1 Euler Angles	Yaw ψ, Pitch θ, Roll ϕ	Set of ordered rotations from navigation to body frame

Attitude Parameterization	Notation	Notes
Quaternions	$\mathbf{q} = [q_1, q_2, q_3, q_4]$, $\mathbf{q}\mathbf{q}^T = 1$	Math is advantageous, but reduced physical intuition

A list of common attitude parameterizations are given in Table 1. Known relations exist between each parameterization. A discussion of the advantageous and disadvantages of these parameterizations (and others) can be found in Chapter 12 of [63]. In this work Euler angles are chosen as the attitude parameterization. The same parameterization can be used for the attitude error-states, however this does not need to be the case. Since switching between parameterizations is possible, there is no requirement to use one parameterization for both attitude and attitude error-states. As will later be described, this work will use attitude error-states derived from the Direction Cosine Matrix (DCM) parameterization for the Kalman Filter fusing the aiding measurements.

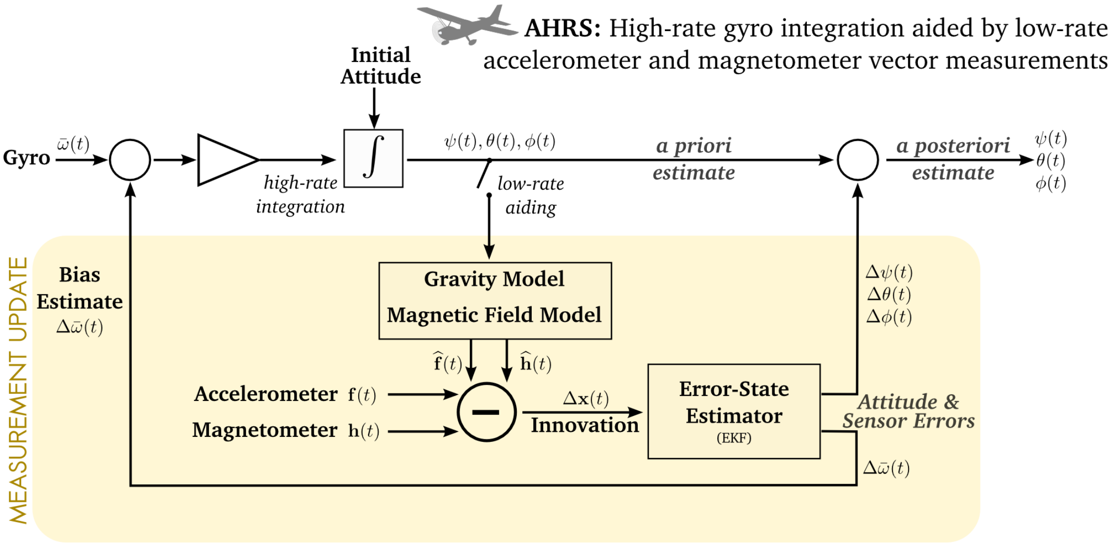


Figure D.1: AHRS Block Diagram

A high level block diagram of the AHRS is shown in Figure D.1. The top portion of the block diagram defines the open-loop or unaided operation of the AHRS. The attitude estimates in this case are based on high-rate integration of gyro measurements. The lower-half of the block diagram shows the aiding measurement mechanization used to arrest error-growth caused by numerical-integration errors and gyro-measurement errors (e.g. noise and time-varying bias). Notice the innovations are based on vector accelerometer and magnetometer measurements. This is like the work in [57]. An alternative formulation is to directly compute yaw, pitch, and roll predictions based on the accelerometer and magnetometer measurements and differencing with the *a priori* estimates to form innovations [56]. One advantage of this technique is there is no requirement for reference gravity or magnetic-field models. This type of aiding mechanization might be called *loose-integration*, were we have borrowed terminology from the aided-INS literature [5]. This work, however, mechanizes the aiding measurements at the level of accelerometer and magnetometer vector measurements.

A challenge of applying these aiding measurements are aircraft accelerations and magnetic field disturbances. Non-gravitational accelerations are indistinguishable from aircraft tilt angles and introduced errors into the aiding measurement. Similarly, magnetic-field disturbances caused by on-board electronics will superimpose with the Earth magnetic field and again corrupt the aiding measurements. Both of these effects introduce (unknown) time-correlated errors into the aiding measurements and therefore the standard Kalman Filter assumption of *white* measurement noise is violated. Albeit with degraded performance, a combination of proper error-modeling and estimator tunning will enable utility of the aiding measurements even for high-dynamics small UAVs.

D.2 AHRS Dynamic Equations

In this section we focus on the AHRS Euler angle attitude state dynamics. The error-states will be described later when aiding from accelerometer and magnetometer measurements will be considered. For open-loop operation, as shown in the top-half of Figure D.1, error-states are unnecessary.

The dynamic equations relate the x, y, and z body rotation-rate measurements $\bar{\omega}_{nb}^{b} = [\omega_x(t), \omega_y(t), \omega_z(t)]^T$ to time rate-of-change of the Euler angles, namely yaw ψ, pitch θ, and roll ϕ. The subscript on $\bar{\omega}_{nb}^{b}$ indicates the rotation-rate of the body-frame with respect to the NED-frame, and the superscript is the frame of expression. This is closely related to the on-board 3-axis gyro measurement $\bar{\omega}_{ib}^{b}(t) = [p(t), q(t), r(t)]$ with the exception that gyro measurements are with respect to the inertial frame. Thus the on-board gyro measurement must be corrected by the knowledge of the inertial rotation-rate of the navigation frame $\bar{\omega}_{in}$:

$$\bar{\omega}_{nb}(t) = \bar{\omega}_{ib}(t) - \bar{\omega}_{in}(t) \tag{D.1}$$

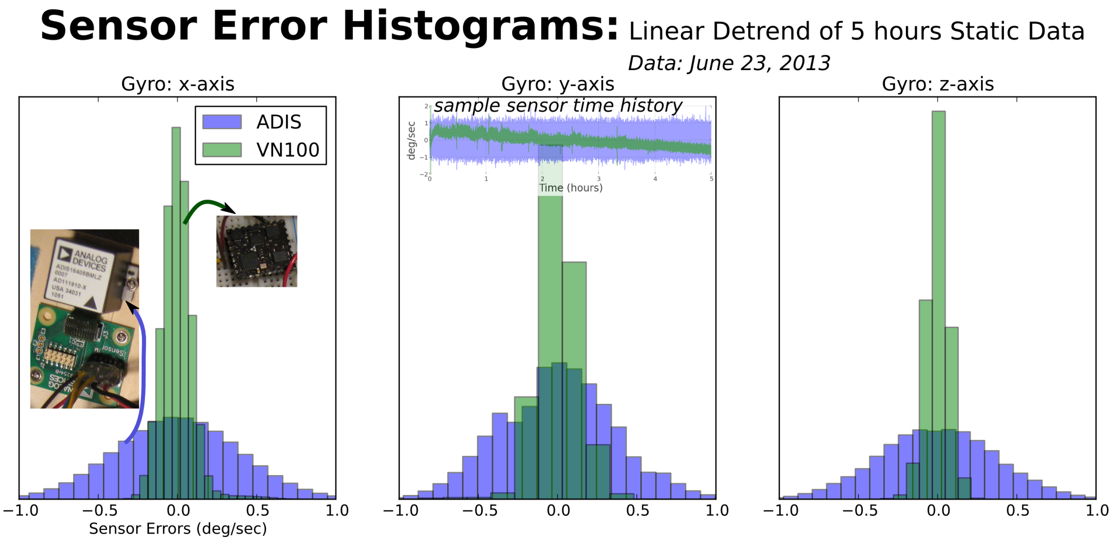
where all terms should be expressed in a common frame, for example the body-frame.

D.2.1 Transport Rate

The rotation-rate of the NED frame $\bar{\omega}_{in}$ is commonly known as the transport-rate and is given by [57]:

$$\bar{\omega}_{in}^{n} = \begin{bmatrix} (\dot{\lambda} + \omega_{ie}) \cos\phi \\ -\dot{\phi} \\ -(\dot{\lambda} + \omega_{ie}) \sin\phi \end{bmatrix} \tag{D.2}$$

where λ, ϕ are the latitude and longitude of the NED-frame origin and ω_{ie} is the Earth angular rate (specified in WGS84). As is clear from Equation D.2 the transport-rate depends both on the position and velocity of the NED-frame origin. The velocity dependence is negligible for conventional aircraft traveling below the speed of sound and thus the dominant term is the Earth angular rate $\omega_{ie} \approx 15/3600$ or 0.004 deg/s. In precision applications, neglecting ω_{ie} can lead to 10+ deg of error in attitude over the course of 60 min of gyro integration. In contrast, the noise level on low cost consumer-grade gyros can be significantly larger than this. Five hours of static gyro measurements were collected for two two commercial IMUs, the VectorNav VN-100 [62] and the Analog Devices ADIS 16405 [64], and error histograms were plotted as shown in Figure D.2. The consumer-grade gyros have noise-levels two orders of magnitude larger than ω_{ie}. Therefore, for low-cost sensor applications it is safe to neglect the transport-rate altogether and assume $\bar{\omega}_{nb}(t) \approx \bar{\omega}_{ib}(t)$.



Sensor Error Histograms: Linear Detrend of 5 hours Static Data

Data: June 23, 2013

Figure D.2: Histogram of noise levels on consumer-grade gyros.

D.2.2 Rotation Rate to Euler Rate

An on-board 3-axis gyro aligned with the aircraft body-axes will provide the body rotation-rate measurements $\bar{\omega}^b_{ib} = [p(t), q(t), r(t)]^T$. In this section we define the mapping from rotation rate to yaw, pitch, and roll rate. Given this mapping and initial attitude estimates, the dynamic equations can be numerically integrated to provide yaw, pitch, and roll states.

The 3-2-1 rotation sequence implies that the rotation from NED-frame to aircraft body-frame is achieved by rotating by an ordered sequence of rotations: ψ about the 3^{rd} or down-axis, θ about the intermediate 2^{nd} or y-axis, and finally ϕ rotation about the 1^{st} or x-body axis. In this sequence the roll-rate is equivalent to the x-body gyro measurement p. The pitch-rate and yaw-rate, however, must be rotated from intermediate reference frames to align with the y and z body gyro measurements, q and r. This is implemented in Equation $D.3$.

$$\begin{aligned}\begin{bmatrix} p \\ q \\ r \end{bmatrix} &= \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \left(\begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \right) \\ &= \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\theta\cos\phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \end{aligned} \tag{D.3}$$

The inverse mapping, however, is the desired quantity. Thus, by inverting and simplifying the above equation we get:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \frac{1}{\cos\theta} \begin{bmatrix} \cos\theta & \sin\theta\sin\phi & \sin\theta\cos\phi \\ 0 & \cos\theta\cos\phi & -\sin\phi\cos\theta \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{D.4}$$

Using the current attitude estimates $\psi(t_k), \theta(t_k), \phi(t_k)$ and gyro measurements $p(t_k), q(t_k), r(t_k)$ Equation $D.4$ can be discretized and numerical integrated to determine the Euler angles at time t_{k+1}.

Unaided AHRS attitude estimates can be suitable for tens of seconds with several degrees of accuracy, assuming that the AHRS was properly initialized. Where by proper initialization it implies the initial attitude and sensor error estimates (e.g. bias) are accurate. This is even true for low-cost systems. A demonstrative example based on flight data published in [5] is shown in Figure D.3. Unaided attitude estimates over several minutes are plotted in light-blue and can be compared with the reference pitch and roll angles in green. Over the first minute the unaided and reference solutions largely overlay and it is difficult to distinguish between them. However as the flight continues to evolve the unaided estimates continue to drift and the attitude errors reach tens of degrees.

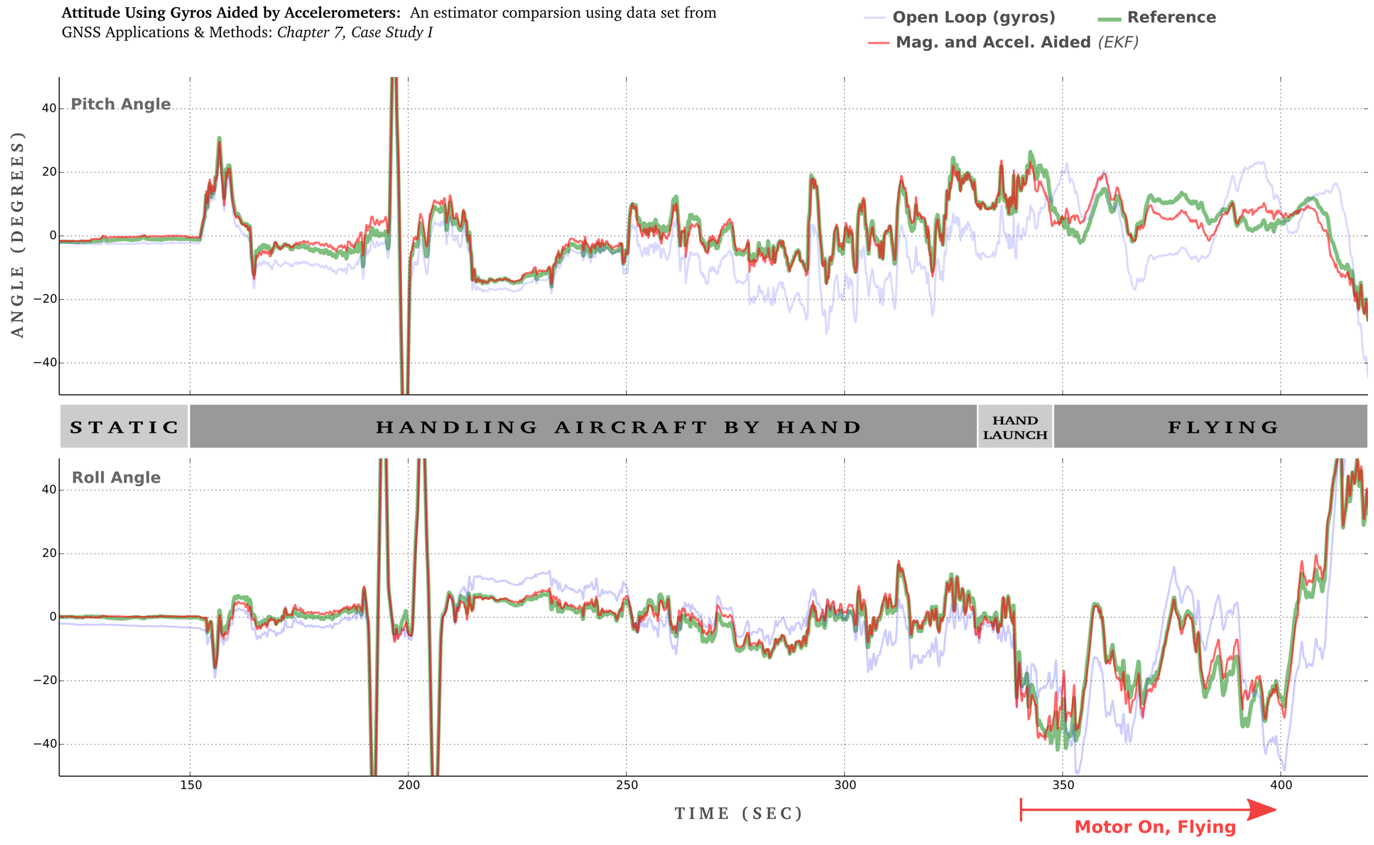


Figure D.3: Example Pitch and Roll AHRS performance comparing unaided and aided AHRS with reference attitude

This motivates the requirement for aiding measurements to arrest the attitude errors. For this purpose an Extended Kalman Filter (EKF) will be mechanized to fuse the aiding measurements. Before detailing the aiding measurements and the EKF mechanization, however, the attitude and sensor error-states are defined.

D.3 Error-States

In this section we present the AHRS error-states and their associate dynamics. A total of 6 error-states will be used:

- 3 gyro bias terms ($\mathbf{b_g}$)
- 3 small-angle attitude errors ($\bar{\rho}$)

The work in [57] also includes 3 accelerometer bias terms ($\mathbf{b_a}$). Although its inclusion was tested, further analysis leads us to believe its inclusion has limited utility and may even be detrimental in high-dynamic UAV applications. Later we provide a brief discussion of why.

D.3.1 Gyro Error Model

The gyro rotation-rate sensor measurements are modeled as the true rotation-rate $\bar{\omega}_{ib}^{b}(t)$ corrupted by a time-varying bias and additive wide-band noise. More precisely:

$$\mathbf{y_g}(t) = \bar{\omega}_{ib}^{b}(t) + \mathbf{b_g}(t) + \mathbf{w_g}(t) \tag{D.5}$$

where $\mathbf{y_g}$ is the measured gyro output, $\bar{\omega}_{ib}^{b}$ is the true body-axes rotation-rate which is then corrupted by the time-varying bias $\mathbf{b_g}$ and additive wide-band noise $\mathbf{w_g}(t)$. Given an estimate of the time-varying bias, the measured rotation-rate can be corrected:

$$\hat{\bar{\omega}}_{ib}^{b}(t) \quad = \quad \mathbf{y_g}(t) - \hat{\mathbf{b}}_\mathbf{g}(t) \tag{D.6}$$

Equation D.6 will be the corrected gyro measurements used.

The relationship between the gyro sensor error-states and the error in estimated rotation-rate must be determined to facilitate the error-covariance propagation:

$$\begin{aligned}
\Delta\bar{\omega}_{ib}^{b}(t) &= \bar{\omega}_{ib}^{b}(t) - \hat{\bar{\omega}}_{ib}^{b}(t) \\
&= \bar{\omega}_{ib}^{b}(t) - (\mathbf{y_g}(t) - \hat{\mathbf{b}}_\mathbf{g}(t)) \\
&= -\Delta\mathbf{b}_g(t) - \mathbf{w_g}(t) \qquad \text{(D.7)}
\end{aligned}$$

where the model for the gyro measurement $\mathbf{y_g}(t)$ in Equation D.5 has been used.

D.3.2 Attitude Errors

An attitude error-state parameterization derived from DCM small-angle rotations serve as a suitable error-state parameterization choice for *tight integration* of aiding measurements [57]. These are small-angle rotations which account for the difference between the estimated and true rotation matrices that transforms from the body frame to the navigation frame (NED). This error model will be derived first.

Consider a body to NED frame rotation matrix $R_b^n(\psi, \theta, \phi)$, formed using Euler angles of the vehicle. Equation D.8 defines the inverse transformation, commonly referred to as DCM, as a 3-2-1 sequence of single-axis rotations. The transpose of the DCM defines R_b^n.

$$\begin{aligned} R_n^b &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & s_\phi \\ 0 & -s_\phi & c_\phi \end{bmatrix} \begin{bmatrix} c_\theta & 0 & -s_\theta \\ 0 & 1 & 0 \\ s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} c_\psi & s_\psi & 0 \\ -s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} && \text{(D.8)} \\ R_b^n &= (R_n^b)^T && \text{(D.9)} \end{aligned}$$

where s_α, c_α are short-hands for $\sin\alpha, \cos\alpha$ for general angle α. It is useful to also present the mapping from R_b^n back to Euler angles [57].

$$\begin{aligned} \psi &= atan2\left(R_b^n[2,1], R_b^n[1,1]\right) && \text{(D.10)} \\ \theta &= -atan\left(\frac{R_b^n[3,1]}{\sqrt{1-(R_b^n[3,1])^2}}\right) && \text{(D.11)} \\ \phi &= atan2\left(R_b^n[3,2], R_b^n[3,3]\right) && \text{(D.12)} \end{aligned}$$

where $R_b^n[i,j]$ refers to ith row, jth column of the matrix (one-based indexing) and $atan2(y,x)$ is standard four-quadrant inverse tangent. Equations D.10-D.12 will be used later to extract the corrected Euler angles attitude representation from an updated rotation matrix.

When R_b^n is formed using the true vehicle Euler angles in Equation D.9, then the transformation will accurately rotate a vector from the vehicle body frame to the true-NED frame. A second estimated rotation matrix can be formed using estimated Euler angles $\widehat{R}_b^{\hat{n}}(\hat{\psi}, \hat{\theta}, \hat{\phi})$ as indicated by the overhead *hat*. The estimated rotation will transform from the vehicle body-axis to the estimated NED-frame. Mathematically:

$$\widehat{R}_b^{\hat{n}} = R_n^{\hat{n}} R_b^n \tag{D.13}$$

where a third rotation matrix, $R_n^{\hat{n}}$ defines the transformation to rotate the true NED-frame to the estimated NED-frame. This transformation captures the errors in the estimated rotation matrix and can be formed using three single-axis small-angle rotations $\epsilon_N, \epsilon_E, \epsilon_D$. The first two components ϵ_N, ϵ_E are referred to as tilt errors and the last component ϵ_D is called yaw error

[57]. The three attitude error components will be collected into a vector $\bar{\rho}$:

$$\bar{\rho} = [\epsilon_N, \epsilon_E, \epsilon_D]^T \tag{D.14}$$

and will be part of the AHRS error-states estimated by the aiding measurements.

The general definition of R_n^b in Equation D.8 as a sequence of three single-axis rotations can be used to form $R_n^{\hat{n}}$. To do so, the small-angle rotations $\epsilon_D, \epsilon_E, \epsilon_N$ will replace the sequence of yaw, pitch, and roll angle single-axis rotations, respectively. The small-angle approximations $\cos\epsilon \approx 1$ and $\sin\epsilon \approx \epsilon$ will be applied. Unlike general single-axis rotation sequences, the small-angle assumption implies that the sequence of application of the corrective rotations does not matter.

$$\begin{aligned} R_n^{\hat{n}} &\approx \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \epsilon_N \\ 0 & -\epsilon_N & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -\epsilon_E \\ 0 & 1 & 0 \\ \epsilon_E & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \epsilon_D & 0 \\ -\epsilon_D & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= (I - [\bar{\rho}\times]) \end{aligned} \tag{D.15}$$

where $[\bar{\rho}\times]$ is the 3×3 *skew-symmetric* matrix formed using the 3×1 tilt-error vector:

$$[\bar{\rho}\times] = \begin{bmatrix} 0 & -\epsilon_D & \epsilon_E \\ \epsilon_D & 0 & -\epsilon_N \\ -\epsilon_E & \epsilon_N & 0 \end{bmatrix} \tag{D.16}$$

The matrix in Equation D.15 rotates true NED-frame to the estimated NED-frame. In the AHRS mechanization, however, the inverse corrective rotation matrix is of interest, namely $R_{\hat{n}}^n$.

$$\begin{aligned} R_{\hat{n}}^n &= (R_n^{\hat{n}})^{-1} = (R_n^{\hat{n}})^T \\ &\approx (I + [\bar{\rho}\times]) \end{aligned} \tag{D.17}$$

Therefore the true body to NED frame transformation can be expressed in terms of the estimated transformation and attitude errors:

$$R_b^n = (I + [\bar{\rho}\times])\, \widehat{R}_b^{\hat{n}} \tag{D.18}$$

As will be described next Equation D.18 can be used to update or correct the estimated body to NED-frame rotation matrix.

Applying Attitude Corrections Equation D.13 expressed the estimated body to NED-frame rotation matrix as a function of the true rotation matrix and an corrective rotation formed $R_n^{\hat{n}}$ by $\bar{\rho}$. All the terms required to represent attitude errors as well as apply corrections to attitude estimates, are now present. The attitude corrections update will be done in three steps:

1. Compute attitude errors $\bar{\rho}$ using EKF measurement update.
2. Form estimated rotation matrix $\widehat{R}_b^{\hat{n}}$ by Equation D.9 using *a priori* Euler angle estimates $(\hat{\psi}^-, \hat{\theta}^-, \hat{\phi}^-)$.
3. Form corrective rotation matrix $R_{\hat{n}}^n$ using Equation D.17 and apply corrective rotation to estimated rotation matrix using Equation D.18:

$$\begin{aligned} \widehat{R}_b^n &= R_{\hat{n}}^n \widehat{R}_b^{\hat{n}} \\ &= (I + [\bar{\rho}\times]) \widehat{R}_b^{\hat{n}} \end{aligned} \tag{D.19}$$

4. Extract corrected or *a posteriori* Euler angles $(\hat{\psi}^+, \hat{\theta}^+, \hat{\phi}^+)$ from $\widehat{R}_b^n$ using Equations D.10-D.12.

D.4 Error-State Dynamics

D.4.1 Attitude Error

A continuous-time dynamic model for the attitude error-states $\bar{\rho}$ is presented. The derivation is based on taking derivatives of Equation D.19 with respect to time. That will require knowing the rate-of-change of the body to navigation frame rotation matrix. Therefore, the expression for $\dot{R}_b^n$ is presented first and thereafter used to determine $\dot{\bar{\rho}}$.

The definition of a derivative is the starting point for deriving $\dot{R}_b^n$:

$$\dot{R}_b^n = \lim_{\Delta t \to 0} \frac{R_b^n(t + \Delta t) - R_b^n(t)}{\Delta t} \tag{D.20}$$

where the time argument specifies the particular set of Euler angles used to form the transformation matrix. If the time interval is small, it is possible to represent $R_b^n(t + \Delta t)$ as a function of $R_b^n(t)$ and a small-angle rotation over the interval Δt. Here we assume the NED navigation frame orientation is fixed and instead the orientation of the body-frame is time-dependent. The body-frame for a given time is made explicit by the functional argument.

$$\begin{aligned} R_b^n(t + \Delta t) &= R_{b(t+\Delta t)}^n \\ &= R_{b(t)}^n R_{b(t+\Delta t)}^{b(t)} \end{aligned} \tag{D.21}$$

where $R_{b(t+\Delta t)}^{b(t)}$ is a small-angle rotation over the time-interval to effectively undo the rotations. Here we apply the small angle rotation matrix defined in Equation D.17:

$$R_{b(t+\Delta t)}^{b(t)} \approx I + \Delta t[\bar{\omega}_{nb}^b \times] \tag{D.22}$$

where $\Delta t \bar{\omega}_{nb}^b$ is vector of small-angle rotations of the body-frame with respect to the navigation frame, expressed in the body-axes, and can be formed using the gyro measurements.

Combining Equations D.20, D.21, and D.22 will finalize the expression for $\dot{R}_b^n$:

$$\begin{aligned}\dot{R}_b^n &= R^n_{b(t)}[\bar{\omega}^b_{nb}\times] \\ &= R^n_{b(t)}[(\bar{\omega}^b_{ib} - \bar{\omega}^b_{in})\times] \\ &\approx R^n_{b(t)}\Omega^b_{ib} \end{aligned} \tag{D.23}$$

where $\Omega^b_{ib} = [\bar{\omega}^b_{ib}\times]$ is formed using the gyro measurements. The approximation is due to the neglected transport rate which as discussed previously is justified for consumer-grade sensor applications.

The attitude-error dynamics $\dot{\bar{\rho}}$ are determined by taking derivatives of Equation D.18 with respect to time:

$$\begin{aligned}\dot{R}_b^n &= (I + [\bar{\rho}\times])\dot{\widehat{R}}{}_b^n + [\dot{\bar{\rho}}\times]\widehat{R}_b^n \\ R_b^n\Omega^b_{ib} &= (I + [\bar{\rho}\times])\widehat{R}_b^n\widehat{\Omega}^b_{ib} + [\dot{\bar{\rho}}\times]\widehat{R}_b^n \\ [\dot{\bar{\rho}}\times] &= (I + [\bar{\rho}\times])\widehat{R}_b^n\Delta\Omega^b_{ib}\widehat{R}_n^b \\ &\approx \widehat{R}_b^n\Delta\Omega^b_{ib}\widehat{R}_n^b \end{aligned} \tag{D.24}$$

where $\Delta\Omega^b_{ib} = \Omega^b_{ib} - \hat{\Omega}^b_{ib}$ are the gyro errors and the expression of R_b^n in terms of the estimate and attitude errors was used in the second to third step. Multiplicative errors (between $[\bar{\rho}\times]$ $\Delta\Omega^b_{ib}$ were dropped, and hence the linear approximation. Although in matrix form, Equation D.24 can be expanded and written in equivalent vector form:

$$\begin{aligned}\dot{\bar{\rho}} &= \widehat{R}_b^n\Delta\omega^b_{ib} \\ &= -\widehat{R}_b^n\Delta\mathbf{b}_g - \widehat{R}_b^n\mathbf{w}_g \end{aligned} \tag{D.25}$$

where the gyro error model derived in Equation D.7 replaced $\Delta\omega^b_{ib}$. This finalizes the linear attitude-error dynamic model.

D.4.2 Sensor Errors

The 3-element time-varying gyro bias is modeled as 3 first-order Markov processes with time-constant $\tau_g = 300$ s and steady-state variance $\sigma_{b_g} = 0.5$ deg/s. The error-model choice and specifications were motivated by the generic automotive/consumer-grade error model in [5]. The dynamic model is:

$$\Delta\dot{\mathbf{b}}_g(t) = F_g\Delta\mathbf{b}_g(t) + \mathbf{w}_{d_g}(t) \tag{D.26}$$

where $F_g = -\frac{1}{\tau_g}\mathbf{I_{3\times 3}}$ and $\mathbf{w}_{d_g}$ is the 3×1 driving noise vector with variance $\frac{2\sigma^2_{b_g}}{\tau_g}$.

D.4.3 Combined Error Dynamics

The dynamic model for the 9-state AHRS error-states will now be compiled using the equations presented. The goal is to form a continuous linear state-space model of the form:

$$\Delta\dot{\mathbf{x}} = F\Delta\mathbf{x} + G\mathbf{w} \tag{D.27}$$

where $\Delta\mathbf{x}^T = [\bar{\rho}^T, \mathbf{b_g}^T]$ is the 6×1 error-state vector and $\mathbf{w}^T = [\mathbf{w_{d_g}}^T, \mathbf{w_g}^T]$ is the 6×1 driving noise vector.

Combining the above equations into this form results in the following dynamic and input mapping matrices:

$$F = \begin{bmatrix} \mathbf{0} & -\widehat{R}_b^n \\ \mathbf{0} & F_g \end{bmatrix} \tag{D.28}$$

$$G = \begin{bmatrix} \mathbf{0} & -R_b^n \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \tag{D.29}$$

where each block specified is a 3×3 matrix.

D.5 Aiding Measurements

Attitude errors accumulated from integrating of the AHRS attitude dynamic model can be arrested via aiding measurements. Additionally, these aiding measurements help estimate the time-varying gyro bias which may change as the aircraft experiences temperature changes or due to other effects like acceleration or motor-induced vibrations.

In this work aiding measurements strictly based on an accelerometer and magnetometer triad will be considered. Other possible aiding sources include heading derived from GNSS-velocity measurements, GNSS-aided INS attitude estimates, multi-GNSS antenna attitude estimates, or camera-based measurements. While all have potential to improve the attitude solution, there are two reasons we restrict attention to only accelerometer and magnetometer-based aiding.

1. Simple and robust: all required sensors are part of single inertial measurement unit (with exception of airspeed which is used for centripetal acceleration corrections).
2. Independent of GNSS availability: attitude estimate is not affected by loss or degraded GNSS signals.

Hence we are accepting a reliable medium-quality solution over brittle or complex high-quality attitude estimates.

D.5.1 Direct Attitude Estimates

Well-known relations exist between Euler angles and vector accelerometer and magnetometer measurements. In both cases the expected vector quantity in the navigation (NED) frame is known and the unknown attitude translates the body frame sensor measurements to the known NED-frame quantities.

Accelerometer: Pitch and Roll The gravitational acceleration in the NED frame is $\mathbf{g}^n = [0, 0, 1]^T$ g's whereas the sensor measures the gravitational acceleration in the body-frame or $\hat{\mathbf{g}}^b = [f_x, f_y, f_z]^T$. If the sensor output is nominally m/s^2, then it is divided by -9.81 m$/s^2$ so that both magnitude and sign match the reference gravitational acceleration when body and NED-frame are aligned. At this point sensor noise, bias, and any non-gravitational acceleration of the vehicle is being neglected. The two quantities are related by the R_n^b:

$$\begin{aligned} \hat{\mathbf{g}}^b &= R_n^b \mathbf{g}^n \\ \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} &= \begin{bmatrix} * & * & -s_\theta \\ * & * & s_\phi c_\phi \\ * & * & c_\phi c_\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{aligned} \tag{D.30}$$

where the first two columns of R_n^b are inconsequential and hence excluded. From Equation D.30 estimates of the vehicle pitch and roll angle are readily obtained:

$$\hat{\theta} = -\sin^{-1}(f_x) \tag{D.31}$$

$$= tan^{-1}\left(\frac{-f_x}{\sqrt{f_y^2 + f_z^2}}\right) \tag{D.32}$$

$$\hat{\phi} = tan^{-1}\left(\frac{f_y}{f_z}\right) \tag{D.33}$$

The second expression for pitch is equivalent but advantageous as the units accelerometer units cancel. In practice a four-quadrant arctangent should be used $tan^{-1}\left(\frac{a}{b}\right) = atan2(a, b)$. If the sign on the gravitational acceleration vector $\mathbf{g}^n$ were different, then the signs on Equations D.33 would change.

The quality of estimates of pitch and roll formed using equations D.32 andD.33 are evaluated using flight data published in [5] and plotted in Figure D.4. Although noisy, both pitch and roll estimates track the reference attitudes nicely when the aircraft is static or being handled on the ground. During flight, however, the pitch estimate performance degrades and the roll-angle estimate becomes useless. This is primarily caused by the non-gravitational acceleration experienced by the aircraft, something which was not accounted for in the derived equations.

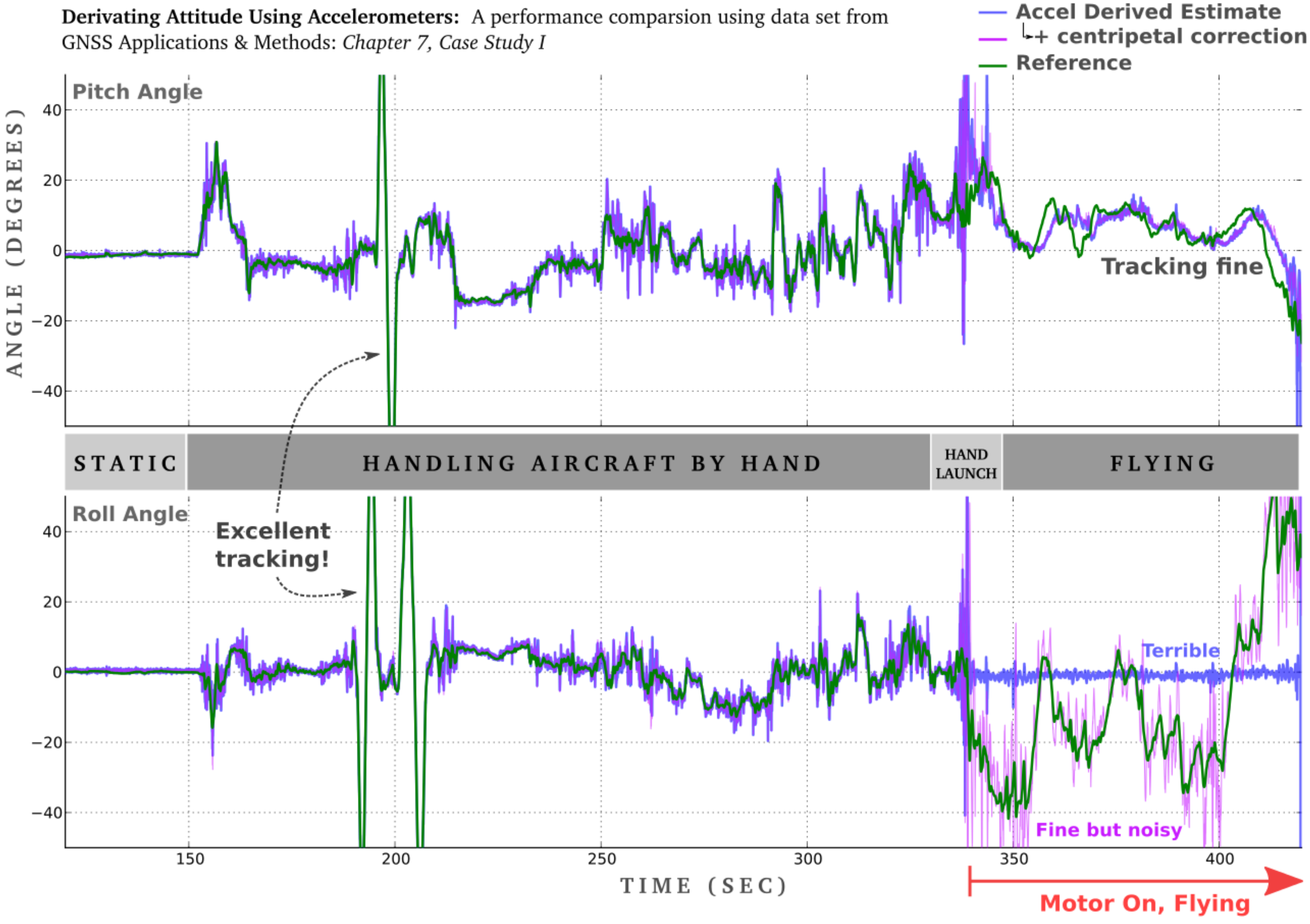


Figure D.4: Quality of pitch and roll estimates derived from accelerometers

The 3-axis accelerometer measures the specific force ($\mathbf{a}_{ib}^b - \mathbf{g}^b$) experienced by the aircraft and cannot distinguish between aircraft accelerations and gravitational acceleration. The specific force measurement model is:

$$\mathbf{y}_{\mathbf{a}}(t) = \mathbf{a}_{ib}^b(t) - \mathbf{g}^b(t) + \mathbf{b}_{\mathbf{a}}(t) + \mathbf{w}_{\mathbf{a}}(t) \tag{D.34}$$

where $\mathbf{b}_a(t)$, $\mathbf{w}_{\mathbf{a}}(t)$ are the sensor bias and noise, respectively. The desired quantity is the measured gravitational vector in the vehicle body-frame. The measured accelerometer output can be used to form an estimated gravitational vector:

$$\hat{\mathbf{g}}^b(t) = -\left(\mathbf{y}_{\mathbf{a}}(t) - \hat{\mathbf{a}}_{ib}^b(t) - \hat{\mathbf{b}_{\mathbf{a}}}(t)\right) \tag{D.35}$$

where $\hat{\mathbf{a}}_{ib}^b(t)$ and $\hat{\mathbf{b}}_a$ are the estimated aircraft non-gravitational acceleration and sensor bias vector, respectively. Without substantial extra effort, there is little information about the expected aircraft accelerations $\mathbf{a}_{ib}^b(t)$. One major component of this acceleration, namely centripetal acceleration, can be approximated and corrected for. As shown in Figure D.4, this single correction restores the utility of the accelerometer-derived roll angle, even during flight.

Centripetal Correction Assume $\mathbf{v}_{ib}^b$ is the inertial velocity vector of the aircraft, expressed in the aircraft body-frame. The acceleration vector is found by taking a single derivative of $\mathbf{v}_{ib}^b$ with respect to time in an inertial frame. Alternatively, the derivative may be taken in the (moving) body-frame by applying the transport theorem [65].

$$\mathbf{a}_{ib}^b(t) = \mathbf{a}_{translation}^b(t) + \bar{\omega}_{ib}^b(t) \times \mathbf{v}^b \tag{D.36}$$

The translational acceleration is unknown. However, the second term, known as centripetal acceleration, can be approximated. The gyro measurement forms the first part, $\bar{\omega}_{ib}(t)$. The velocity vector of the aircraft is approximated by assuming the airspeed measurement v_{ias} is aligned with the aircraft x-body axis $\mathbf{v}^b = [v_{ias}, 0, 0]^T$. In this manner the centripetal acceleration is

$$\begin{aligned} \hat{\mathbf{a}}_{ib}^b(t) &\approx \bar{\omega}_{ib}^b(t) \times \mathbf{v}^b \qquad \text{(D.37)} \\ &= \begin{bmatrix} 0 \\ v_{ias} r \\ -v_{ias} q \end{bmatrix} \qquad \text{(D.38)} \end{aligned}$$

where q and r are the gyro measurements along the y and z axis, respectively. If estimates of the gyro bias are available, naturally they should be applied prior to applying Equation D.38. As shown in Figure D.4, this correction is critical to deriving meaningful roll-estimates from the accelerometer during flight. Naturally, estimate accuracy is degraded by unmodeled accelerations, non-zero angle-of-attack or side-slip, and by the noise on the pitot-probe airspeed measurement.

Magnetometer: Yaw A similar procedure to the accelerometer case can be used to derive magnetic-heading from the magnetometer measurement. $\hat{\mathbf{h}}^b = [h_x, h_y, h_z]^T$. First the body-axis measurement are leveled or projected into the horizontal plane as $\hat{\mathbf{h}}^{level}$.

$$\begin{aligned} \hat{\mathbf{h}}^{level} &= R_b^{level}\hat{\mathbf{h}}^b \\ \begin{bmatrix} h_x^{level} \\ h_y^{level} \\ h_z^{level} \end{bmatrix} &= \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix} \begin{bmatrix} h_x \\ h_y \\ h_z \end{bmatrix} \end{aligned} \tag{D.39}$$

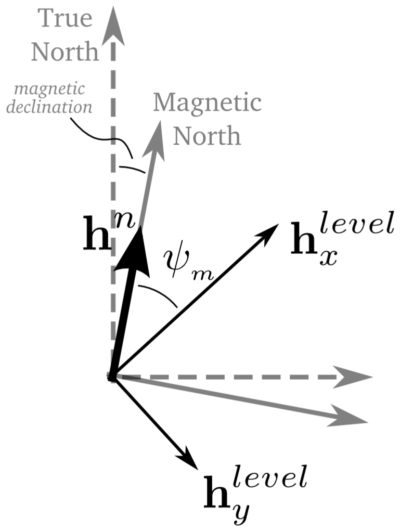
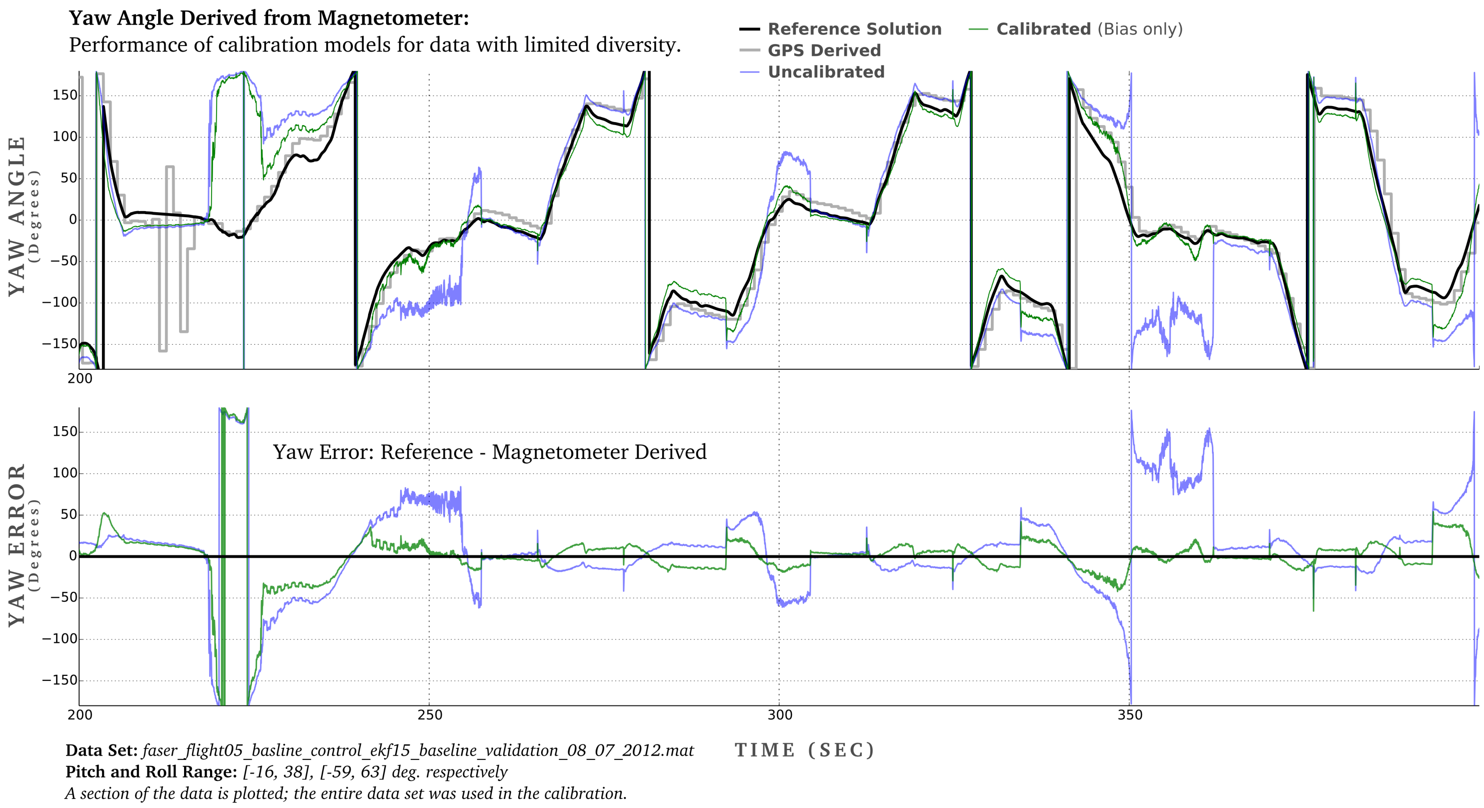


Figure D.5: Yaw from magnetometer measurements

Then the yaw angle defines the final rotation to align the leveled measurements with the reference magnetic north. This is drawn in Figure D.5 and the resulting yaw angle is given in Equation D.40.

$$\hat{\psi} = -\tan^{-1}\left(\frac{h_y^{level}}{h_x^{level}}\right) \tag{D.40}$$

The flight data in [5] used previously does not include magnetometer measurements. Therefore the performance of the magnetometer-derived yaw is demonstrated using flight data collected by the U of MN UAV Lab and plotted in Figure D.6. Plotted against the reference yaw



Yaw Angle Derived from Magnetometer:
Performance of calibration models for data with limited diversity.

Data Set: *faser_flight05_basline_control_ekf15_baseline_validation_08_07_2012.mat*
Pitch and Roll Range: *[-16, 38], [-59, 63] deg. respectively*
A section of the data is plotted; the entire data set was used in the calibration.

Figure D.6: Yaw Estimate Comparison

estimate are:

1. Yaw derived from GPS
2. Yaw derived from uncalibrated magnetometer
3. Yaw derived from calibrated magnetometer

The heading derived from GPS is based on the NED-velocity measurements:

$$\psi = \tan^{-1}\left(\frac{v_{east}}{v_{north}}\right) \tag{D.41}$$

where v_{north} and v_{east} are the inertial velocities measured by the GPS receiver. Therefore such a heading is only accurate when the aircraft is moving, and furthermore, this angle is really with respect to the aircraft velocity vector and not necessarily the x-body axis.

The yaw estimate from the uncalibrated magnetometer is most interesting. During long sections of the flight the heading angle tracks the reference heading. However, there are significant errors during select portions of the flight, for example near $t = 225$ or 300 s. These deviations are due to magnetic-field disturbances caused by the on-board electronics. The most pronounced and extended deviation at $t = 225$ is during take-off and all the magnetic-field induced by the current flowing to the the control surfaces washes out the magnetic-field of the earth. Thus the term *uncalibrated* is used loosely to both account for sensor calibration errors and unmodeled magnetic-field disturbances.

In post-processing, a set of three magnetometer bias estimates $\mathbf{b}_h$ were determined and subtracted from the body-axis measurements. The yaw estimates were then derived using the corrected measurements and plotted as the "Calibrated (Bias only)" result. The estimates are significantly improved compared to the uncalibrated case, however, the errors still occasionally reach near 45°.

In summary, Equations D.40, D.32, D.33 can be used to directly determine yaw, pitch, and roll estimates. These equations will be used to initialized the AHRS by providing the initial conditions to the gyro-integration. However, the aiding measurements for the EKF will be based on the sensor vector measurements.

D.5.2 Indirect Aiding Measurements

Indirect aiding measurements based on the accelerometer and magnetometer measurements will be presented in the form of the standard EKF measurement update:

$$\Delta\mathbf{y} = H\Delta\mathbf{x} + \mathbf{v} \tag{D.42}$$

where $\Delta\mathbf{y} = \mathbf{y} - \hat{\mathbf{y}}$ is the measurement innovation, $\Delta\mathbf{x}^T = [\bar{\rho}, \Delta\mathbf{b}_g]$ is the error-state vector, and $\mathbf{v}$ is the zero-mean measurement noise vector with covariance R. In this manner the standard

EKF equations can be used to compute the Kalman gain, update the error-state and associated error-covariance:

$$\begin{aligned} K &= P^{-}H^{T}(HP^{-}H^{T}+R)^{-1} && \text{(D.43)} \\ \Delta\mathbf{x}^{+} &= \Delta\mathbf{x}^{-}+K\Delta\mathbf{y} && \text{(D.44)} \\ P^{+} &= =(I-KH)P^{-}(I-KH)^{T}+KRK^{T} && \text{(D.45)} \end{aligned}$$

The *a priori* attitude error-state $\bar{\rho}^{-}$ will always be zero. This is because the *a posteriori* attitude error will immediately be applied to the attitude estimates as described in Section D.3.2 and therefore reset to zero for the next measurement update.

First we details the aiding measurements from the accelerometer and second the magnetometer.

D.5.3 Accelerometer

The aiding measurement obtainable from the accelerometer is based on knowledge of the earth gravitational field vector in the NED-frame. Specifically

$$\mathbf{g}^{n}=\begin{bmatrix} 0 \\ 0 \\ 9.81 \end{bmatrix} m/s^{2} \tag{D.46}$$

The accelerometer measurements on-board the aircraft can be used to form $\hat{g}^{b}$, an estimate of the gravitational acceleration measured in the body frame, as shown in Equation D.35. To compare the reference and estimated gravitational acceleration requires a common frame of reference. This is achieved using the estimated vehicle attitude and either the reference gravity $\mathbf{g}^{n}$ is rotated into the body-frame or the estimated $\hat{g}^{b}$ is rotated into the NED frame. Both methods have been explored and their is no significant advantage to either. In this work the NED frame mechanization was selected and will be presented.

The estimated vehicle attitude is used to form the body to NED transformation matrix and subsequently rotate the estimated gravitational acceleration into the NED frame:

$$\hat{\mathbf{g}}^{n}(t)=\widehat{R}_{b}^{n}(t)\hat{\mathbf{g}}^{b} \tag{D.47}$$

Both reference and estimated gravitational vectors are expressed in the NED frame. The difference between them defines the accelerometer measurement innovation Δy_{g}^{n}:

$$\begin{aligned} \Delta y_{g}^{n} &= \mathbf{g}^{n}-\hat{\mathbf{g}}^{n}(t) \\ &= \mathbf{g}^{n}-\widehat{R}_{b}^{n}(t)\hat{\mathbf{g}}^{b} && \text{(D.48)} \end{aligned}$$

Translating the innovation into error-state corrections requires casting the innovation equation into the general linear form of Equation D.42 in terms of the error-states. To do so requires first to represent $\widehat{R}_b^n$ in terms of the attitude error-states by the transpose of Equation D.18. Second, the gravitational vector $\hat{\mathbf{g}}^b$ in Equation D.35 is expanded using the assumed accelerometer measurement model in Equation D.34:

$$\begin{aligned}\hat{\mathbf{g}}^b &= \mathbf{g}^b - \Delta\mathbf{a}_{ib}^b - \Delta\mathbf{b}_a - \mathbf{w}_a \\ &= \mathbf{g}^b - \bar{\mathbf{w}}_a \end{aligned} \tag{D.49}$$

where $\Delta\mathbf{a}_{ib}^b$ and $\Delta\mathbf{b}_a$ are errors in the non-gravitational acceleration correction and the accelerometer bias estimate. Without additional external measurements, there is no way to observe these quantities. Hence moving forward we will lump them with the additive noise $\mathbf{w}_a$ into a new noise term $\bar{\mathbf{w}}_a$.

Replacing both terms, the innovation equation is:

$$\begin{aligned}\Delta\mathbf{y}_g^n &= \mathbf{g}^n - (I - [\bar{\rho}\times])R_b^n(\mathbf{g}^b - \bar{\mathbf{w}}_a) \\ &= R_b^n\bar{\mathbf{w}}_a + [\bar{\rho}\times]\mathbf{g}^n + [\bar{\rho}\times]R_b^n\bar{\mathbf{w}}_a \\ &= -[\mathbf{g}^n\times]\bar{\rho} + R_b^n\bar{\mathbf{w}}_a \end{aligned} \tag{D.50}$$

where the general mathematical transformation $[\mathbf{a}\times]\mathbf{b} = -[\mathbf{b}\times]\mathbf{a}$ was used and multiplicative error terms were dropped. The innovation in Equation D.48 together with the linear model in Equation D.50 define the aiding measurement based on the accelerometer. Therefore if $\Delta\mathbf{x} = [\bar{\rho}, \Delta\mathbf{b}_g]$, then the linear measurement mapping matrix is:

$$H = \begin{bmatrix} -[\mathbf{g}^n\times] & \mathbf{0}_{3\times3} \end{bmatrix} \tag{D.51}$$

Since $\mathbf{g}^n$ has only a non-zero down component, the mapping matrix H makes clear that aiding accelerometer measurement only observes the first two elements of the attitude error-vector $\bar{\rho}$ and that no observability is afforded for the yaw-error ϵ_D. This motivates the necessity for the magnetometer aiding measurements. Prior to ending, however, a brief discussion of the lumped error term $\bar{\mathbf{w}}_a = \Delta\mathbf{a}_{ib}^b - \Delta\mathbf{b}_a - \mathbf{w}_a$ is povided.

Corrections for the centripetal acceleration were presented in Equation D.38. While this was demonstrated to be important, clearly there will be other acceleration errors which are not modeled and hence a non-zero $\Delta\mathbf{a}_{ib}$. Similarly, a low-cost accelerometer outputs will include biases that may vary based on sensor conditions like temperature or vibration. Using off-line calibration look-up tables may minimize the impact of biases, however, the unmodeled variation will again introduce non-zero bias errors $\Delta\mathbf{b}_a$. The additive nature of both errors makes it difficult to separate their contributions. Furthermore, without additional information, there is little utility in their combined effect. For example, estimates of the unmodeled non-gravitational acceleration at the last epoch $\Delta\mathbf{a}_{ib}(t)$ may be irrelevant at the measurement update 1 second later. The adverse effect of including such terms as error states was confirmed and hence their exclusion is recommended for this application.

D.5.4 Magnetometer

The aiding measurement from the magnetometer is conceptually the same as was presented for the accelerometer. The aiding measurement is mechanized in the NED frame as shown in Figure D.5. Magnetic-field models of the Earth, like the International Geomagnetic Reference Field (IGRF) can be used to look-up the local magnetic field for a particular location:

$$\mathbf{h}^n = \begin{bmatrix} h_N \\ h_E \\ h_D \end{bmatrix} \tag{D.52}$$

Following the same procedure as was done for the accelerometer, the difference between the reference and estimated magnetic field in the NED frame, formed using the magnetometer measurement $\hat{\mathbf{h}}^b$, defines the measurement innovation:

$$\begin{aligned} \Delta y_h^n &= \mathbf{h}^n - \hat{\mathbf{h}}^n(t) \\ &= \mathbf{h}^n - \widehat{R}_b^n(t)\hat{\mathbf{h}}^b \end{aligned} \tag{D.53}$$

and the associated error model will be:

$$\Delta\mathbf{y}_h^n = -[\mathbf{h}^n\times]\bar{\rho} + R_b^n\bar{\mathbf{w}}_h \tag{D.54}$$

where the same procedure to form the accelerometer error-model in Equation D.50 was used. The term $\bar{\mathbf{w}}_h$ is a collective error term comprised on measurement noise, magnetic-field disturbances, and residual errors not captured by offline calibration of the magnetometer.

Expanding Equation D.54 will illustrate the general aiding measurement is a function of all three attitude-error parameters:

$$\Delta\mathbf{y}_h^n = -\begin{bmatrix} 0 & -h_D & h_E \\ h_D & 0 & -h_N \\ -h_E & h_N & 0 \end{bmatrix}\begin{bmatrix} \epsilon_N \\ \epsilon_E \\ \epsilon_D \end{bmatrix} + R_b^n\bar{\mathbf{w}}_h \tag{D.55}$$

This assumes precise knowledge of $\mathbf{h}^n$ being available. In practice, however, this temporal and spatial dependent quantity is only approximately known. Furthermore, magnetic-field disturbances will adversely affect the effectiveness of this measurement. Therefore, it is advantageous to restrict the aiding to only the single channel of yaw-error. This is done using Equation D.55 and setting ϵ_N and ϵ_E equal to zero:

$$\begin{bmatrix} \Delta y_N \\ \Delta y_E \\ \Delta y_D \end{bmatrix} = -\begin{bmatrix} h_E \\ -h_N \\ 0 \end{bmatrix} \epsilon_D + R_b^n \bar{\mathbf{w}}_h \tag{D.56}$$

From Equation D.56 the third row has no dependence on ϵ_E and the first row has a weak dependence since the coefficient h_E is near zero for most locations. Therefore, the final aiding measurement is taken only from the middle row:

$$\Delta y_E = h_N \epsilon_D - T R_b^n \bar{\mathbf{w}}_h \tag{D.57}$$

where $T = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$.

This previous procedure is similar to having used the estimated pitch and roll angle to level the magnetometer measurements, and then forming an aiding measurement based on the leveled output. In both cases the uncertainty associated with the leveling process are neglected and an aiding measurement only dependent on heading-error is derived.