COOPERATIVE GEOLOCATION USING UAVS WITH GIMBALLING CAMERA SENSORS WITH EXTENSIONS FOR COMMUNICATION LOSS AND SENSOR BIAS ESTIMATION

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

William Wesley Whitacre

May 2010

COOPERATIVE GEOLOCATION USING UAVS WITH GIMBALLING CAMERA SENSORS WITH EXTENSIONS FOR COMMUNICATION LOSS AND SENSOR BIAS ESTIMATION

William Wesley Whitacre, Ph.D.

Cornell University 2010

This dissertation considers the geolocation of a point of interest (POI), i.e., determining the location of a POI in the world, using multiple cooperating uninhabited aerial vehicles (UAVs) with gimballing camera sensors. A square root sigma point information filter (SR-SPIF) is developed to provide a probabilistic estimate of the POI location. The SR-SPIF utilizes the UAV's onboard navigation system to save computation and also takes important properties for numerical accuracy (square root), tracking accuracy (sigma points), and fusion ability (information). The SR-SPIF is general and scales well to any tracking problem with multiple, moving sensors.

In the development of the SR-SPIF, the errors in the navigation system output are assumed to be zero mean. However, in the practical application, there are non zero mean errors (biases), which degrade geolocation accuracy. Therefore, a decentralized approach to simultaneously estimate the biases on each UAV and the unknown POI location is developed. The new decentralized bias estimation approach provides accurate geolocation in spite of sensor biases and further scales well with the number of UAVs.

Communication is an important part of a cooperative geolocation mission and in practice communication losses and delays are inevitable. Therefore, a new method for cooperative geolocation in the presence of communication loss, termed

the predicted information method, is developed from a separable formulation of the extended information filter. The predicted information method is shown to give the exact solution for linear systems when the measurement dynamics are constant or known by all UAVs.

In addition to theoretical developments, extensive experimental flight tests with ScanEagle UAVs have been performed. The experimental flight tests serve two purposes: 1) to develop practical guidelines for geolocation 2) to validate all of the new approaches presented in this dissertation. In addition to the flight tests, a high fidelity, distributed, hardware in the loop simulation test bed was developed and used as further validation of all new approaches.

BIOGRAPHICAL SKETCH

William Whitacre graduated high school in 1999 from Hampshire High School in Romney, WV. After high school he graduated from Virginia Tech in 2004 with a B.S. in Aerospace and Ocean Engineering. Now in 2010 he is graduating from Cornell University with a Ph.D. in Theoretical and Applied Mechanics.

ACKNOWLEDGEMENTS

I would first like to thank my parents, Frank and Mary Whitacre, for their support over the years. Early in life they instilled in me a desire to learn and to understand the world. I would not have been able to make it without them. I would also like to thank my committee, Prof. Mark Campbell, Prof. Hod Lipson, and Prof Richard Rand for their guidance throughout. I would especially like to thank my advisor Prof. Mark Campbell. He was always available to assist me and provide advice both professionally and otherwise. Finally, I would like to thank my friends and colleagues who helped to keep me going throughout this process. In particular, I would like to thank Laura Whitacre for her support in spite of the strain this experience cause, Mike Czabaj for helping to reduce stress through many varied adventures, and Ranajay Ghosh for many enlightening discussions about every topic imaginable.

TABLE OF CONTENTS

Biographical Sketch . . . iii
Acknowledgements . . . iv
Table of Contents . . . v
List of Tables . . . viii
List of Figures . . . ix

1 Introduction 1
1.1 Square Root Sigma Point Information Filter . . . 2
1.2 Sensor Bias . . . 4
1.3 Communication Loss . . . 6
1.4 Experimental and Simulated Flight Data . . . 8

2 Square Root Sigma Point Information Filter for Cooperative Geolocation 9
2.1 Abstract . . . 9
2.2 Introduction . . . 10
2.3 The Cooperative Geolocation Tracking Problem . . . 13
2.4 Square Root, Sigma Point Information Filter (SR-SPIF) . . . 16
2.4.1 SR-SPIF Prediction . . . 18
2.4.2 Conversion from Prediction (n_a) to Update (n_{2a}) . . . 19
2.4.3 SR-SPIF Update . . . 21
2.4.4 Cooperative Fusion . . . 24
2.5 Augmentations to the Geolocation Filter . . . 25
2.5.1 Additional sensor measurements . . . 25
2.5.2 Estimation with Delayed Data . . . 26
2.5.3 Square Root, Interacting Multiple Model Filter (SR-IMM) . . . 32
2.6 The SeaScan UAV . . . 35
2.6.1 Avionics, Inertial/Attitude Estimation, Control Loops . . . 36
2.6.2 Camera and Image Software . . . 37
2.6.3 Sensor Measurement . . . 37
2.7 Flight Results using the SeaScan UAV . . . 40
2.7.1 POI Model . . . 41
2.7.2 Stationary POI . . . 42
2.7.3 Moving POI . . . 43
2.7.4 With and Without POI Altitude Estimate . . . 44
2.7.5 Implementation of the SR-IMM . . . 46
2.8 Conclusions . . . 47

3 An Improved Square Root Sigma Point Information Filter for Cooperative Geolocation 49
3.1 Introduction . . . 49
3.2 SR-SPIF Development . . . 51

3.3 Filter Comparison . . . 57
3.3.1 Presentation as Extended Information Filters . . . 57
3.3.2 Comparison of the SRSPIF Versions . . . 60
3.4 Conclusions . . . 61

4 Decentralized Geolocation and Sensor Bias Estimation 63
4.1 Abstract . . . 63
4.2 Introduction . . . 63
4.3 Cooperative Geolocation . . . 66
4.4 Geolocation with Bias Estimation . . . 71
4.4.1 Centralized Approach . . . 73
4.4.2 Decentralized Approach . . . 74
4.5 Observability Procedure . . . 78
4.6 Validation with Experimental and Simulated Flight Data . . . 81
4.6.1 Experimental and Simulated Flight Test Setup . . . 81
4.6.2 Stationary POI Results . . . 83
4.6.3 Moving Point of Interest . . . 92
4.7 Conclusions . . . 95

5 Cooperative Geolocation in the Presence of Communication Loss 97
5.1 Abstract . . . 97
5.2 Introduction . . . 97
5.3 Decentralized Cooperative Estimation . . . 101
5.4 Separable Extended Information Filter . . . 105
5.5 Predicted Information Method . . . 107
5.5.1 Predicting Information Updates . . . 109
5.5.2 Stability and Performance . . . 112
5.5.3 Maintaining Equivalent Estimates Across Sensor Nodes . . . 113
5.5.4 General Communication Loss Scenarios . . . 114
5.6 Benchmark Methods for Communication Loss in Decentralized Estimation . . . 115
5.6.1 Drop Information . . . 115
5.6.2 Store and Burst . . . 116
5.6.3 Hybrid State Fusion . . . 118
5.7 Cooperative Geolocation Using Flight Test Results . . . 121
5.7.1 Single, Extended Communication Loss . . . 125
5.7.2 Random Losses . . . 131
5.8 Conclusions . . . 134

6 Flight Results from Tracking Ground Targets Using SeaScan UAVs with Gimballing Cameras 136
6.1 Abstract . . . 136
6.2 Introduction . . . 136
6.3 Geolocation with the SeaScan UAV . . . 138

6.3.1 The SeaScan UAV 139
6.3.2 Sensor Bias 141
6.4 Flight Results 143
6.4.1 Stationary POI 145
6.4.2 Moving POI 150
6.5 Conclusions 152

7 Conclusions **155**

A Derivation of Separable Extended Information Filter **161**

B Computing Equivalent State Transition Matrix **164**

C Camera Measurement **165**

Bibliography **167**

LIST OF TABLES

4.1 Test configurations for the HiL simulations. 82

6.1 Flight tests for a stationary POI. 145
6.2 Flight test parameters for the moving POI tests. 150

LIST OF FIGURES

1.1 Overview of the cooperative geolocation problem, showing the tracking Point of Interest (POI) on the ground, with multiple UAVs pointing cameras at the POI. 2
1.2 Single UAV tracking a stationary POI. The blue dots are computed camera line of sight intersections with the ground from two orbits of tracking. 4

2.1 Overview of the cooperative geolocation problem, showing the tracking Point of Interest (POI) on the ground, with multiple UAVs pointing cameras at the POI. 14
2.2 Architecture for a single UAV in the proposed decentralized geolocation estimator for cooperative vehicles tracking points of interest. 16
2.3 Plot of the one sigma uncertainty volume over time for several cases: single UAV, two UAVs cooperating with full communication, and two UAVs cooperating with communication drops, with and without fusion. 31
2.4 Plot of the memory (communication), computation, and performance (time to converge back to the two UAV solution) for the delayed data simulations. 32
2.5 The SeaScan UAV in flight. 36
2.6 Image taken from a flight test of the SeaScan; note that the POI is at the center of the image frame. 40
2.7 Plot of the tracking latitude and longitude errors and uncertainty bounds for a stationary target, using single and two UAVs. 42
2.8 Evolution of two dimensional tracking ellipsoids over time for single (left) and two (right) UAV tracking systems. 43
2.9 UAV trajectories (triangles indicate UAV locations for the estimates shown), POI truth, and estimated POI for single and two UAV examples tracking the POI. 44
2.10 Latitude and longitude estimation errors for the single and multiple UAV cases, and with and without attitude measurements. 45
2.11 Tracking a moving POI using a single UAV, with a single model SR-SPIF and a n_m model SR-IMM estimator. 47

4.1 Single UAV tracking a stationary POI. The blue dots are computed camera line of sight intersections with the ground from two orbits of tracking. 71
4.2 Diagram illustrating the configuration (in projected 2D) of two UAVs (triangle) orbiting above a POI ($*$) performing geolocation. The orbits are parameterized by the relative phase, β, and orbit offset, Δ. 82

4.3 Geolocation error during the first six minutes of the 2007 flight test data with the UAV in a POI centered orbit about a stationary POI and using four different choices of biases to estimate: 1) No biases (No Bias) 2) Camera gimbal bias (GIM) 3) UAV attitude (ATT) 4) UAV attitude and camera gimbal biases (ATT-GIM). 86
4.4 Bias estimates for the 2007 flight test data using a single UAV and a centered orbit around a stationary POI. 87
4.5 Observability of least observable mode over a range of orbit offsets. 89
4.6 Average steady state geolocation error from HiL simulation with orbit offsets ranging from 0 m to 500 m (the orbit radius was 500 m). 90
4.7 Geolocation error averaged over four subsets of the 2007 flight test with three choices of cooperation: Single UAV, Decentralized, and Centralized and also with two choices of biases to estimate: No Bias, and ATT-GIM bias. 91
4.8 Representative trajectory from the HiL city driving simulator and geolocation estimates using a single UAV and two bias estimation choices: No Bias, and ATT-GIM bias. 92
4.9 Geolocation error averaged over four subsets of the HiL moving POI test set with three choices of cooperation: Single UAV, Decentralized, and Centralized and also with two choices of biases to estimate: No Bias, and ATT-GIM bias. 94

5.1 Summary of the PI filter during a complete communication loss of $m-1$ time steps. 110
5.2 Summary of the Hybrid State Fusion method. 120
5.3 Overview of the cooperative geolocation problem, showing the tracking Point of Interest (POI) on the ground, with multiple UAVs pointing cameras at the POI. 122
5.4 Sample moving POI trajectory from the hardware in the loop simulations based on a city driving model. 123
5.5 Estimate error (distance) averaged of 250 tests with experimental flight data for several cases: Full Communication (FC), two variations of Predicted Information method (CI, EI), and the three benchmark methods (DI, HSF, and SB). 126
5.6 Kullback-Leibler Divergence from Full Communication for the two variations of Predicted Information method (CI, EI), and the three benchmark methods (DI, HSF, and SB) when locating a stationary POI, averaged over 5000 tests with experimental flight data. 127
5.7 Kullback-Leibler Divergence from Full Communication for the two variations of Predicted Information method (CI, EI), and the three benchmark methods (DI, HSF, and SB) when locating a maneuvering POI, averaged over 5000 tests with HiL simulation data. . . 128

5.8 Communication required by each of the communication loss methods after an m step communication loss when locating a stationary POI. . . . 129
5.9 Additional memory required by each of the communication loss methods for an m step communication loss when locating a stationary POI. . . . 130
5.10 Steady State Kullback-Leibler Divergence as compared to the Full Communication case for the two variations of Predicted Information method (CI, EI), and the three benchmark methods (DI, HSF, and SB) when locating a stationary POI, averaged over 100 tests with HiL simulation data. . . . 132
5.11 Steady State Kullback-Leibler Divergence as compared to the Full Communication case for the two variations of Predicted Information method (CI, EI), and the three benchmark methods (DI, HSF, and SB) when locating a maneuvering POI, averaged over 100 tests with HiL simulation data.. . . . 132

6.1 The SeaScan UAV. . . . 140
6.2 Single UAV tracking a stationary POI. The blue dots are computed camera line of sight intersections with the ground from two orbits of tracking. . . . 142
6.3 Geolocation errors and bounds using the Athena GuideStar navigation system with a camera field of view or 3.84 deg, an altitude above target of 320 m, and no orbit center offset. . . . 143
6.4 Geolocation errors and slant range to POI from flight test eight using GuideStar. . . . 146
6.5 Uncertainty volumes over a range of orbit offsets using the Helmsman avionics system. . . . 147
6.6 Uncertainty volumes over a range of orbit offsets using the GuideStar avionics system. . . . 147
6.7 Uncertainty volumes over a range of camera field of view levels using the Helmsman guidance system. . . . 148
6.8 Uncertainty volumes over a range of camera field of view levels using the GuideStar navigation system. . . . 148
6.9 Geolocation errors and slant range to POI from flight test eleven using the Helmsman navigation system. . . . 151
6.10 Uncertainty volumes over a range of orbit offsets using the Helmsman navigation system. . . . 152
6.11 Uncertainty volumes over a range of orbit offsets using the GuideStar navigation system. . . . 152
6.12 Uncertainty volumes over a range of camera field of view levels using Helmsman. . . . 153
6.13 Uncertainty volumes over a range of camera field of view levels using GuideStar. . . . 153

CHAPTER 1

INTRODUCTION

The central problem addressed in this dissertation is the cooperative geolocation of a point of interest (POI) using multiple uninhabited aerial vehicles (UAVs) with gimballing camera sensors. The conceptual scenario for cooperative geolocation is shown in Figure 1.1. Geolocation is the process of using sensor data to develop statistical estimates of a Point of Interest (POI) on the ground. Each UAV, based on its position and orientation, points the camera (through a gimballing payload mount inside the UAV) at the POI on the ground. While the aircraft is moving (navigation and attitude), and the POI is potentially moving, the camera gimbals must adjust their angles such that the POI always remains within the field of view of the camera. The objective of geolocation is then to estimate the position (2D or 3D) of the POI from the aircraft, gimbal, and camera measurements. Complicating this problem are uncertainties in the aircraft position and orientation, gimbal angles, camera specifications and measurements, and disturbances such as turbulence and engine vibrations.

In other research related to using vision on UAVs, several groups are using vision systems for navigation, such as for road following [20] or obstacle avoidance [34]. The latter implements a bearings-only Simultaneous Localization and Mapping (SLAM) algorithm to localize both the vehicle and obstacles and navigate using only a low cost inertial measurement unit and a monocular camera. Ridley *et al* [51] and Grocholsky *et al* [23] have implemented UAV systems with cameras using decentralized fusion (information filtering) concepts. Several other groups have implemented gimballing camera systems on UAVs [25, 26], a few with initial target tracking results [49, 63, 15]. Kaaniche [31] *et al* present an interesting traffic

Figure 1.1: Overview of the cooperative geolocation problem, showing the tracking Point of Interest (POI) on the ground, with multiple UAVs pointing cameras at the POI.

surveillance approach with UAVs using a graph cut formulation and a verification step. Stepanyan and Hovakimyan [56] demonstrate visual tracking of one aircraft with another aircraft using only a monocular camera. In a more closely related work Vercauteren and Wang [62] develop a Sigma Point Information Filter.

1.1 Square Root Sigma Point Information Filter

A square root, sigma point information filter (SR-SPIF) is developed in [11] and presented in Chapter 2 to solve the cooperative geolocation problem. The square root formulation is used to maintain numerical integrity in real time. The sigma point formulation is used for its accuracy with nonlinear dynamics and nonlinear measurement equations. An information form is used for ease in fusing measurements from other UAVs. The SR-SPIF presented in Chapter 2 incorporates uncertainty in the UAV states with a combined state (POI state) and parameter (UAV state) formulation with a non-standard update, which utilizes the onboard naviga-

tion system to save computation. Chapter 3 presents an improved SR-SPIF which incorporates the UAV state uncertainty as a non-additive, nonzero mean noise. This formulation saves both computation and communication. Both versions of the SR-SPIF have the following important properties:

Decentralized: Each vehicle has its own geolocation estimator, and then communicates only necessary information to the other vehicles. This minimizes memory and communication, and enables network robustness.

Information Form: An information form is used to 1) minimize the amount of information shared between vehicles, 2) simplify the multiple vehicle fusion problem, and 3) simplify the problem of delayed data (from communication drop-outs). For single vehicle tracking the use of the information form leads to a slight increase in the computational burden. However, when performing tracking with cooperative vehicles, the information form can significantly reduce the amount of computation.

Sigma Points: Sigma points are used to develop statistical linearizations of the dynamics, which have been shown to be more accurate than the traditional Extended Kalman Filter (or the Extended Information Filter) [27]. The use of sigma points requires a small increase in computation compared to the Extended Kalman or Information Filters. However, it is proposed here that the increase in tracking performance with the nonlinear measurement is justification for this increase in computational cost.

Square Root: A square root version of the estimator is used for its numerical accuracy in real time implementation. The square root implementation requires very little additional complexity when used in conjunction with sigma points. The square root version is equivalent in computation to the Square

Root Sigma Point Filter [6], which was shown to work in real time at 20Hz for an UAV aerodynamic model estimator [9].

1.2 Sensor Bias

In the development of the SR-SPIF, the errors in the estimate of the UAV state are assumed to be zero mean, white, and Gaussian, which is not accurate in the practical case, for two reasons: correlated outputs of the navigation filter and biases in the outputs. Consider Figure 1.2, which shows a series of Sensed Points of Interest (SPOIs), which is defined as the line of sight intersection of the camera with the ground as computed based on the estimates of the UAV state (NAV, ATT, and GIM). A total of 2000 SPOIs are plotted for two orbits around a stationary POI from a flight test of the ScanEagle UAV on March 18, 2006 [67]. Figure 1.2 shows that the SPOI moves in a roughly circular pattern around the true POI location. The period of this oscillation corresponds directly to the UAVs orbit about the POI and is due to nonzero mean errors (biases) in the UAV state estimate.

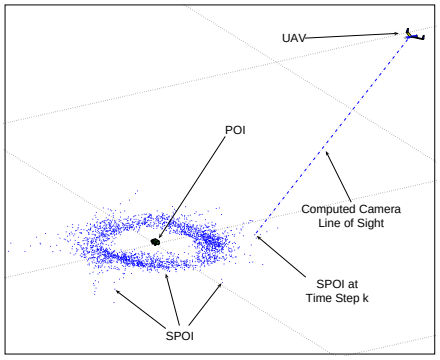


Figure 1.2: Single UAV tracking a stationary POI. The blue dots are computed camera line of sight intersections with the ground from two orbits of tracking.

Sensor biases were shown in Refs. [10] and [67] to be a significant source of error for geolocation using the ScanEagle UAV and were compensated for by augmenting

the output of the estimator with additional uncertainty, based on empirical data. However, this did nothing to improve the estimate itself, but simply improved estimator consistency. Biases have also been shown to be a problem for estimation in other works. In one of the first treatments [21], Friedland showed that biases could be estimated efficiently in a linear system by partitioning the state. More recently in Refs. [32] and [54], bias estimation in a radar tracking context with multiple targets was addressed. Bias estimation was further considered in Refs. [37] and [45], where multiple sensors were used to track multiple targets in a centralized formulation. Bias estimation has also been considered in a least squares estimation context by Dogancay in Ref. [16].

In Chapter 4, an approach is developed to jointly estimate the sensor biases and the unknown POI state in a decentralized manner, while using the solution from the onboard navigation system to save significant computation. The decentralized formulation allows the UAVs to share information on only the POI state, and model only their local biases, saving computation as well as communication, and moreover, giving geolocation accuracy comparable to the centralized case. Further, this decentralized approach fits nicely into the decentralized data fusion paradigm [17, 59] and allows for effective cooperation not only among UAVs with potentially different biases but different sensors altogether. A numerical observability analysis procedure is also developed in Chapter 4 and applied, which gives a meaningful measure of the degree of observability and also gives insight into the effects of UAV flight path on observability. The new decentralized approach is validated using both experimental flight data and high fidelity hardware in the loop simulations.

1.3 Communication Loss

Communication is an important component of the cooperative estimation process, and recent research has focused on the effects of digital communication in both control and estimation. Delchamps [14] presented a seminal work describing the effect of quantization on controller performance. More recently, significant work has been done on the effect of network communication between sensors, controllers, and system plants [60, 69, 52, 70]. In addition to quantization effects, the cooperative estimation system must be robust to communication losses and delays. For example, in autonomous underwater vehicle applications, communication is particularly limited in reliability as well as bandwidth. In [1], Akyildiz *et al* characterize the underwater communication channel noting that communication is lost or delayed frequently due multi-path and obstructions.

The problem of cooperative estimation in the presence of communication loss is considered in Chapter 5. Communication loss refers to a situation in which the sensor nodes are unable to communicate with each other for a period of time that is unknown a priori. It is assumed here that the communication losses are symmetric and known by each sensor node. Communication delay is a related problem and refers to the situation in which sensor nodes must communicate over a medium which induces potentially unknown and varying delays in the data transmission; for example, communication over an ad hoc wireless network. It is assumed here that the delays are not symmetric between nodes and there is no confirmation of receipt provided from the receiving node. In this case, any time a sensor node sends data, it will have no knowledge of how long before the data is received at the other end. Also, there is no guarantee that the data will be received at a sensor node in the order it was generated. Under this scenario, it is the responsibility of

the receiver to make effective use of the data received, even if the data is both late and out of order.

The delayed data problem was explored in the Kalman filtering domain by Larsen *et al.* in Ref. [35], where the focus was on estimation with a combination of fast and slow sensors. In Ref. [30], Julier and Uhlmann developed an approach algebraically equivalent to the results of Larsen. In a centralized estimation framework, Bar-Shalom developed an exact solution for out-of-sequence measurements[3]. Nettleton and Durrant-Whyte [44] examined the delayed data problem in the information filtering domain, which is more amenable to decentralized sensor fusion. This approach is theoretically equivalent to centralized estimation in linear systems, but requires large memory and bursty communication. None of the existing methods address the communication loss (or delay) problem in the nonlinear decentralized estimation context. These cases arise when the POI dynamics or measurements are nonlinear, which is common in the UAV tracking problem [11].

In Chapter 5, a new method is developed for decentralized estimation using multiple UAVs communicating over a lossy network. The new method, termed the Predicted Information (PI) method, is developed as an approximation to the Separable Extended Information Filter (SEIF), which is an alternate form of the Extended Information Filter (EIF) derived in Appendix A. The basic concept of the PI method is to predict the information matrix updates during the communication loss to more easily update the estimator when communication is re-established. Two variations are presented that trade accuracy with computation, memory, and communication load. The first variation estimates the information matrix updates over time, while the second variation uses a piecewise constant approximation to

the information matrix updates. Although the delayed data problem is not specifically addressed here, the PI method can be used with delayed data but requires minor modifications.

1.4 Experimental and Simulated Flight Data

Each of the three theoretical developments in this dissertation are validated using experimental flight data using the ScanEagle UAV. Flight tests were performed on October 6, 2004 [11], March 18, 2006 [67], and March 16, 2007 [57]. In addition to using the flight tests for validation, the 2006 flight test was used for a parametric study of the effect of various flight parameters on geolocation accuracy [67]. The results of the study are included in Chapter 6.

A high fidelity hardware in the loop (HiL) simulation test bed for distributed tracking was also developed. The distributed tracking test bed was used for validation of both the bias estimation and communication loss algorithms. Further, the simulation test bed was used in the experimental demonstrations of a flight test of cooperative tracking. [58]

CHAPTER 2

SQUARE ROOT SIGMA POINT INFORMATION FILTER FOR COOPERATIVE GEOLOCATION

This Chapter provides a reproduction of the journal article developing the Square Root Sigma Point Information Filter.

2.1 Abstract

A cooperative tracking approach for uninhabited aerial vehicles with camera based sensors is developed and verified with flight data. The approach utilizes a square root sigma point information filter, which takes important properties for numerical accuracy (square root), tracking accuracy (sigma points), and fusion ability (information). Important augmentations to the filter are also developed for delayed data, by estimating the correlated processes, and moving targets, by using multiple models in a square root interacting multiple model formulation. The final form of the algorithm is general and scales well to any tracking problem with multiple, moving sensors. Flight data using the SeaScan UAV is used to verify the algorithms for stationary and moving targets. Cooperative tracking results are evaluated using multiple test flights, showing excellent results.

2.2 Introduction

Tracking of stationary and moving targets with multiple sensors is a problem of wide reaching importance with obvious and abundant military applications. Other missions include search and rescue [12], factory surveillance [50], and even animal migration [53]. It is well known that data fusion techniques can significantly increase the information (decrease the uncertainty) of the target being tracked based on multi-sensor measurements [17]. Challenges arise, however, when the sensors themselves are moving. Such a problem exists with multiple uninhabited aerial vehicles (UAVs) tracking ground targets. Enabled by digital off the shelf cameras, UAV systems are now being developed with visual cameras on board. The use of such off the shelf systems can lead to missions involving a fleet of UAVs tracking and classifying multiple targets on the ground. These UAVs are very complex systems which require the integration of several hardware components (camera, UAV, GPS, attitude sensors) and software components (camera image processing, inner loop and path planning control, and estimation software) in order to develop realistically accurate estimates of the object being tracked.

In other research related to using vision on UAVs, several groups are using vision systems for navigation, such as for road following [20] or obstacle avoidance [34]. The latter implements a bearings-only Simultaneous Localization and Mapping (SLAM) algorithm to localize both the vehicle and obstacles and navigate using only a low cost inertial measurement unit and a monocular camera. Ridley *et al* [51] and Grocholsky *et al* [23] have implemented UAV systems with cameras using decentralized fusion (information filtering) concepts. Several other groups have implemented gimballing camera systems on UAVs [25, 26], a few with initial target tracking results [49, 63, 15]. Kaaniche [31] *et al* present an interesting traffic

surveillance approach with UAVs using a graph cut formulation and a verification step. Stepanyan and Hovakimyan [56] demonstrate visual tracking of one aircraft with another aircraft using only a monocular camera. In a more closely related work Vercauteren and Wang [62] develop a Sigma Point Information Filter.

Most of these works typically only address a subset of the important requirements for vision tracking systems for production UAVs, namely 1) scalability to a modest number of UAVs, 2) robustness to communication loss and bandwidth limitations, and 3) numerical stability and efficiency in real time implementation for nonlinear, decentralized tracking.

This paper details the theoretical development of a distributed, cooperative estimation methodology for multiple UAVs which track stationary and moving ground targets with on board cameras which meets all of the above requirements. Key attributes of the proposed cooperative geolocation tracking estimator are:

Decentralized: Each vehicle has its own geolocation estimator, and then communicates only necessary information to the other vehicles. This minimizes memory and communication, and enables network robustness.

Simplified Prediction: Only the target dynamics are used in the prediction portion of the estimator, thus allowing half of the estimator to scale very well with the number of vehicles.

Information Form: An information form is used to 1) minimize the amount of information shared between vehicles, 2) simplify the multiple vehicle fusion problem, and 3) simplify the problem of delayed data (from communication drop-outs). For single vehicle tracking the use of the information form leads to a slight increase in the computational burden. However, when performing

tracking with cooperative vehicles, the information form can significantly reduce the amount of computation.

Sigma Points: Sigma points are used to develop statistical linearizations of the dynamics, which have been shown to be more accurate than the traditional Extended Kalman Filter (or the Extended Information Filter) [27]. The use of sigma points requires a small increase in computation compared to the Extended Kalman or Information Filters. However, it is proposed here that the increase in tracking performance with the nonlinear measurement is justification for this increase in computational cost.

Square Root: A square root version of the estimator is used for its numerical accuracy in real time implementation. The square root implementation requires very little additional complexity when used in conjunction with sigma points. The square root version is equivalent in computation to the Square Root Sigma Point Filter [6], which was shown to work in real time at 20Hz for an UAV aerodynamic model estimator [9].

Multiple Model: A multiple model form is used to more accurately track maneuvering targets.

In addition to the theoretical developments, the work here is validated using flight test data for tracking a moving target using the SeaScan UAV both cooperatively and noncooperatively, demonstrating excellent results. The SeaScan [61, 24] is a long endurance (24+hours) UAV developed by the Insitu Group, and is now a key component in Boeing's UAV strategic plan.

The paper is outlined as follows. The next section gives an overview of the geolocation problem, followed by the decentralized estimation architecture, square root estimator with extensions for delayed data and multiple model implementa-

tion. The subsequent section details the SeaScan UAV and its components. The final section includes the flight test results for geolocation tracking of stationary and moving objects.

2.3 The Cooperative Geolocation Tracking Problem

Figure 2.1 shows a conceptual scenario for the geolocation problem. The aircraft, based on its position and orientation, points the camera (through a gimballing payload mount inside the UAV) at the Point of Interest (POI) on the ground which shows up as a projection on a screen. While the aircraft is moving (navigation and orientation), and the POI is potentially moving, the camera gimbals must adjust their angles to point at the POI. The objective of geolocation is then to estimate the position (2D or 3D) of the POI from the aircraft, gimbal, and camera measurements. Geolocation also requires the camera to remain directed at the POI while the UAV and POI are moving, or else the measurement would not exist. Complicating this problem are uncertainties in the aircraft position and orientation, gimbal angles, camera specifications and measurements, and disturbances such as turbulence and engine vibrations. Each UAV then communicates information to every other UAV to fuse together a cooperative estimate of the POI.

Building a centralized estimator on each UAV, including UAV and gimbal models for each UAV and a target model, would require a very high load of computation, memory, and communication. Fortunately, most UAVs use an inertial measurement system which includes estimators that provide statistics (estimates and covariances) for both the UAV attitude and navigation. As described in [10] for one vehicle, these UAV attitude and navigation estimators can be used to simplify

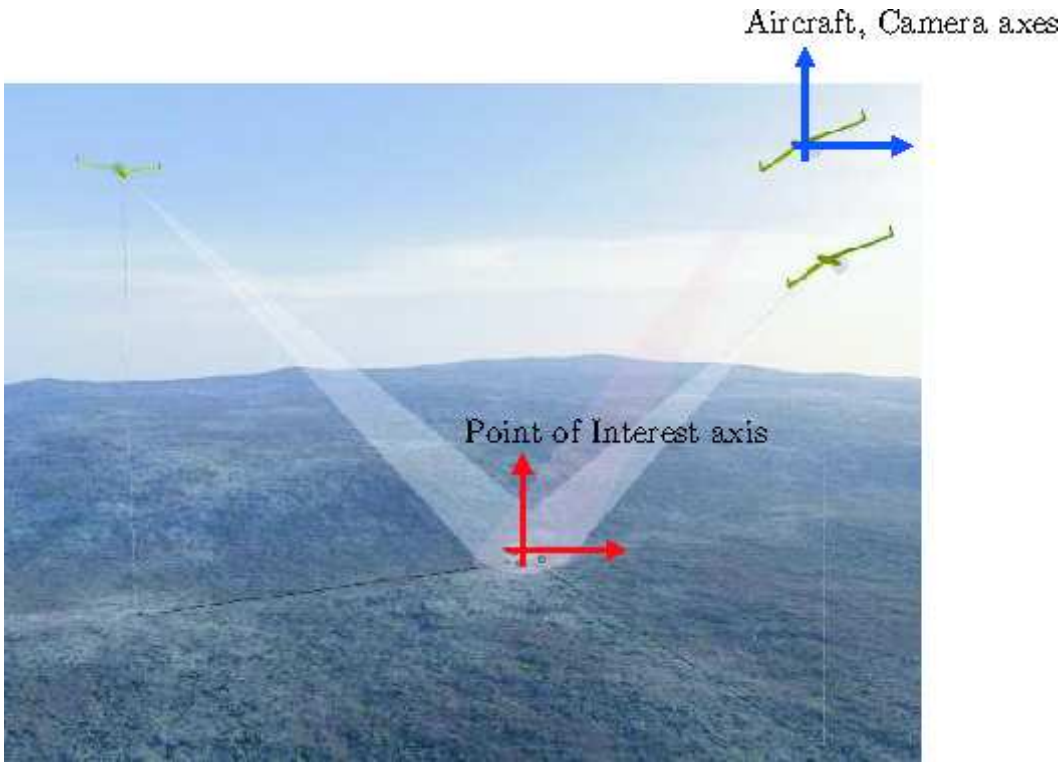


Figure 2.1: Overview of the cooperative geolocation problem, showing the tracking Point of Interest (POI) on the ground, with multiple UAVs pointing cameras at the POI.

the geolocation tracking estimator. In this case, the dynamics for the single UAV, single POI tracking problem are written formally as

$$\dot{\mathbf{x}}_{\text{POI}} = f_{\text{POI}}(\mathbf{x}_{\text{POI}}, \mathbf{w}_{\text{POI}}) \tag{2.1}$$

$$\mathbf{z}_{\text{SCR}} = g_{\text{SCR}}(\mathbf{x}_{\text{POI}}, \begin{bmatrix} \mathbf{x}_{\text{NAV}} \\ \mathbf{x}_{\text{ATT}} \\ \mathbf{x}_{\text{GIM}} \end{bmatrix}, \mathbf{v}_{\text{SCR}}) \tag{2.2}$$

where $\mathbf{x}_{\text{POI}}$ is the POI state vector, $\mathbf{x}_{\text{NAV}}, \mathbf{x}_{\text{ATT}}, \mathbf{x}_{\text{GIM}}$ are the UAV navigation (NAV), attitude (ATT), and turret gimbal (GIM) states and $\mathbf{z}_{\text{SCR}}$ is the location of the POI in the camera screen (SCR). Estimates of the NAV, ATT, and GIM states are assumed to be provided from other on-board estimators or measurements in the form $\mathcal{N}(\hat{\mathbf{x}}_{(\cdot)}, P_{(\cdot)})$.

Consider now an estimator designed to provide the statistics of the POI state $(\hat{\mathbf{x}}_{\text{POI}}, P_{\text{POI}})$. The prediction step is based on (2.1), and as such, only requires the POI model. The update step, based on (2.2), requires a more complicated least squares solution using all of the available statistics (POI,NAV,ATT,GIM). The update step, in this case, is similar in context to that of a fully centralized Geolocation estimator (taking into account all random variables), but not producing estimates for the aircraft or gimbal states (thus being less complicated, and less sensitive to model errors/tuning of these other estimators).

The cooperative UAV tracking problem uses the same POI dynamics in (2.1), and is not a function of any of the UAV vehicle states. Therefore, the POI prediction scales very well with the number of cooperating UAVs. The update, however, must scale with measurements and statistics from each UAV. The measurements in this case are written for N UAVs tracking the same POI as:

$$\begin{bmatrix} \mathbf{z}^1_{\text{SCR}} \\ \vdots \\ \mathbf{z}^{\text{N}}_{\text{SCR}} \end{bmatrix} = \begin{bmatrix} g^1_{\text{SCR}}(\mathbf{x}_{\text{POI}}, \begin{bmatrix} \mathbf{x}^1_{\text{NAV}} \\ \mathbf{x}^1_{\text{ATT}} \\ \mathbf{x}^1_{\text{GIM}} \end{bmatrix}, \mathbf{v}^1_{\text{SCR}}) \\ \vdots \\ g^{\text{N}}_{\text{SCR}}(\mathbf{x}_{\text{POI}}, \begin{bmatrix} \mathbf{x}^{\text{N}}_{\text{NAV}} \\ \mathbf{x}^{\text{N}}_{\text{ATT}} \\ \mathbf{x}^{\text{N}}_{\text{GIM}} \end{bmatrix}, \mathbf{v}^{\text{N}}_{\text{SCR}}) \end{bmatrix} \tag{2.3}$$

Note that the POI state, $\mathbf{x}_{\text{POI}}$, is common in the measurements across all UAVs.

Figure 2.2 shows the proposed decentralized architecture of the Geolocation estimator in a block diagram with $i_{(\cdot)}$, $I_{(\cdot)}$ representing the information about the POI that is shared between vehicles. In this case, the model prediction step has been reduced to only propagating the POI model (thus, no aircraft or gimbal models are required). The least squares (LS) update step is more complicated

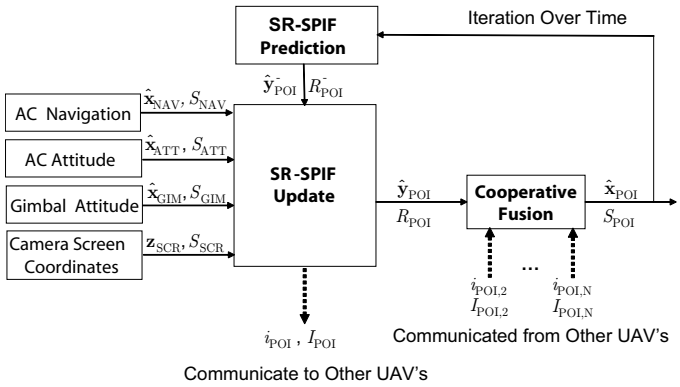


Figure 2.2: Architecture for a single UAV in the proposed decentralized geolocation estimator for cooperative vehicles tracking points of interest.

because of the stochastic dependency on the aircraft and gimbal states.

2.4 Square Root, Sigma Point Information Filter (SR-SPIF)

A square root, sigma point information filter is developed in this work to solve the cooperative geolocation problem. The square root formulation is used to maintain numerical integrity in real time. The sigma point formulation is used for its accuracy with nonlinear dynamics and nonlinear measurement equations. An information form is used for ease in fusing measurements from other UAVs. This section presents the base estimator with the simplified prediction step, while the next section details the cooperative fusion of estimates from multiple vehicles and the handling delayed data, followed by the interactive multiple model implementation. As background, details on information and sigma-point filtering can be found in [27], [50], [6], and [42].

The continuous dynamic model in Equations 2.1, 2.2 is first discretized to form

$$\begin{aligned} \mathbf{x}_{\mathrm{POI},k+1} &= \mathbf{x}_{\mathrm{POI},k} + \Delta T f_{\mathrm{POI}}(\mathbf{x}_{\mathrm{POI},k}, \mathbf{w}_k) \qquad (2.4) \\ &= f_{D\mathrm{POI}}(\mathbf{x}_{\mathrm{POI},k}, \mathbf{w}_k) \end{aligned}$$

$$\mathbf{z}_{\mathrm{SCR},k+1} = g_{\mathrm{SCR}}(\mathbf{x}_{\mathrm{POI},k+1}, \begin{bmatrix} \mathbf{x}_{\mathrm{NAV},k+1} \\ \mathbf{x}_{\mathrm{ATT},k+1} \\ \mathbf{x}_{\mathrm{GIM},k+1} \end{bmatrix}, \mathbf{v}_{\mathrm{SCR},k+1}) \qquad (2.5)$$

where $\mathbf{x}_{\mathrm{POI},k} \in \mathbb{R}^{n_{\mathrm{POI}} \times 1}$. The process and sensor noises, $\mathbf{w}_k \in \mathbb{R}^{n_w \times 1}, \mathbf{v}_{\mathrm{SCR},k+1} \in \mathbb{R}^{n_{\mathrm{SCR}} \times 1}$ are uncorrelated, zero mean, white processes with covariance $P_{w,k}, P_{\mathrm{SCR},k+1}$.

Now consider an augmented state vector defined as,

$$\mathbf{x}_{a,k} = \begin{bmatrix} \mathbf{x}_{\mathrm{POI},k} \\ \mathbf{w}_k \end{bmatrix} \qquad (2.6)$$

where $\mathbf{x}_{a,k} \in \mathbb{R}^{n_a}$, and $n_a = n_{\mathrm{POI}} + n_w$.

The initial augmented state estimate and square root covariance are assumed to be,

$$\begin{aligned} \hat{\mathbf{x}}_{a,0} &= E[\mathbf{x}_{a,0}] \qquad (2.7) \\ &= \begin{bmatrix} \mathbf{x}_{\mathrm{POI},0} \\ 0_{[n_w,1]} \end{bmatrix} \end{aligned}$$

$$\begin{aligned} S_{a,0} &= \begin{bmatrix} \sqrt{P_{\mathrm{POI},0}} & 0 \\ 0 & \sqrt{P_{w,0}} \end{bmatrix} \qquad (2.8) \\ &= \begin{bmatrix} S_{\mathrm{POI},0} & 0 \\ 0 & S_{w,0} \end{bmatrix} \end{aligned}$$

where $0_{[n_w,1]}$ is a column vector of zeros with length n_w. The covariances are

factored using $P_{(\cdot)} = S_{(\cdot)}^T S_{(\cdot)}$, where $S_{(\cdot)}$ is an upper right triangular square root covariance matrix.

Next, $2n_a + 1$ sigma points are defined,

$$\begin{aligned}\mathcal{X}_{a,0} &= \left[\begin{array}{ccc} \hat{\mathbf{x}}_{a,0}, & \hat{\mathbf{x}}_{a,0} \cdot \mathbf{1}_{[1,n_a]} + \sigma_f S_{a,0}, & \hat{\mathbf{x}}_{a,0} \cdot \mathbf{1}_{[1,n_a]} - \sigma_f S_{a,0} \end{array}\right] \\ &= \left[\begin{array}{c} \mathcal{X}_{\text{POI},0} \\ \mathcal{X}_{w,0} \end{array}\right], \end{aligned} \tag{2.9}$$

where σ_f is a scaling for the distance of the sigma points from the mean, and $\mathbf{1}_{[1,n_a]}$ is a $1 \times n_a$ row vector of ones. A set of associated weights is then defined, which can be used to find the sample mean/covariance:

$$\begin{aligned} W_m^0 &= \frac{\sigma_f^2 - n_a}{\sigma_f^2} \\ W_c^0 &= \frac{\sigma_f^2 - n_a}{\sigma_f^2} + 3 - \frac{\sigma_f^2}{n_a}, \\ W &= W_m^j = W_c^j = \frac{1}{2\sigma_f^2}, \;\; j = 1, \cdots, 2n_a \end{aligned} \tag{2.10}$$

where m, c denote mean and covariance respectively.

The SR-SPIF algorithm requires three steps: SR-SPIF Prediction, Conversion from Prediction (n_a) to Update (n_{2a}), and SR-SPIF Update.

2.4.1 SR-SPIF Prediction

The SR-SPIF prediction step propagates each of the $2n_a + 1$ sigma points through the nonlinear dynamics (2.11) and evaluates the sample mean (2.12). The predic-

tion step is based on the POI model given in (2.4).

$$\mathcal{X}^{j-}_{\text{POI},k+1} = f_{D\text{POI}}(\mathcal{X}^{j}_{\text{POI},k}, \mathcal{X}^{j}_{w,k}), \qquad j = 0, \cdots, 2n_a \tag{2.11}$$

$$\hat{\mathbf{x}}^{-}_{\text{POI},k+1} = \sum_{j=0}^{2n_a} W^j_m \mathcal{X}^{j-}_{\text{POI},k+1}, \tag{2.12}$$

$$\mathcal{X}^{c0-}_{\text{POI},k+1} = \mathcal{X}^{0-}_{\text{POI},k+1} - \hat{\mathbf{x}}^{-}_{\text{POI},k+1} \tag{2.13}$$

$$\mathcal{X}^{c-}_{\text{POI},k+1} = \left[\mathcal{X}^{1-}_{\text{POI},k+1} - \hat{\mathbf{x}}^{-}_{\text{POI},k+1}\ , \cdots, \mathcal{X}^{2n_a-}_{\text{POI},k+1} - \hat{\mathbf{x}}^{-}_{\text{POI},k+1}\right] \tag{2.14}$$

The square root covariance for the POI states is formed through an orthogonalization and Cholesky update for the 0th sigma point, as its weight W^0_c could be negative.

$$S^{-}_{\text{POI},k+1} = up\{\sqrt{W} \cdot orth\{(\mathcal{X}^{c-}_{\text{POI},k+1})^T\}, (\mathcal{X}^{c0-}_{\text{POI},k+1})\sqrt{|W^0_c|}, sgn(W^0_c)\} \tag{2.15}$$

With $S^{-}_{\text{POI},k+1}$ being upper triangular and only representing the POI states, the square root information matrix can be calculated with modest computation,

$$R^{-}_{\text{POI},k+1} = orth\left\{\left(S^{-}_{\text{POI},k+1}\right)^{-T}\right\} \tag{2.16}$$

where $Y_{(\cdot)} = R^T_{(\cdot)} R_{(\cdot)}$ defines the information matrix $Y_{(\cdot)}$ and its upper right triangular, square root information factor $R_{(\cdot)}$.

2.4.2 Conversion from Prediction (n_a) to Update (n_{2a})

Converting to a form for the update step requires additional information based on the dependency of the output equation on the aircraft navigation, attitude, and gimbal states. Specifically, the augmented state vector $\mathbf{x}_{a,k}$ from the prediction step must now be further augmented for the update step to include the NAV, ATT, and GIM states as well as the sensor noise. The secondary augmented state

vector is then written as

$$\mathbf{x}_{2a,k} = \begin{bmatrix} \mathbf{x}_{a,k} \\ \mathbf{v}_{\mathrm{SCR},k} \\ \mathbf{x}_{x2,k} \end{bmatrix} \tag{2.17}$$

where the secondary states are defined as $\mathbf{x}_{x2,k} = \left[\mathbf{x}^T_{\mathrm{NAV},k+1}, \mathbf{x}^T_{\mathrm{ATT},k+1}, \mathbf{x}^T_{\mathrm{GIM},k+1}\right]^T$. The dimension of the secondary augmented vector is $\mathbf{x}_{2a,k} \in \mathbb{R}^{n_{2a} \times 1}$, where $n_{2a} = n_a + n_{\mathrm{SCR}} + n_{x2}$.

As shown in Fig. 2.2, the statistics available for the update from the aircraft include estimate and square root covariance for the aircraft navigation $(\hat{\mathbf{x}}_{\mathrm{NAV},k+1}, S_{\mathrm{NAV},k})$, the aircraft attitude $(\hat{\mathbf{x}}_{\mathrm{ATT},k+1}, S_{\mathrm{ATT},k+1})$, and the gimbal pointing $(\hat{\mathbf{x}}_{\mathrm{GIM},k+1}, S_{\mathrm{GIM},k+1})$. It is assumed that square root information factors for each variable are also available, either as the output of a square root information filter, or through the modest computation of inverting the square root covariance. Thus, the secondary state estimates and square root covariance/information factors are written formally as

$$\hat{\mathbf{x}}^-_{x2,k+1} = \begin{bmatrix} \hat{\mathbf{x}}_{\mathrm{NAV},k+1} \\ \hat{\mathbf{x}}_{\mathrm{ATT},k+1} \\ \mathbf{z}_{\mathrm{GIM},k+1} \end{bmatrix} \tag{2.18}$$

$$S^-_{x2,k+1} = \begin{bmatrix} S_{\mathrm{NAV},k+1} & 0 & 0 \\ 0 & S_{\mathrm{ATT},k+1} & 0 \\ 0 & 0 & S_{\mathrm{GIM},k+1} \end{bmatrix} \tag{2.19}$$

and $R^-_{x2,k+1} = orth\left\{(S^-_{x2,k+1})^{-T}\right\}$.

The centralized sigma points are now updated (in the case of POI from the previous sigma points in Equations 2.11,2.13, and 2.14) and defined (in the case

of SCR and the secondary states) as shown in (2.20).

$$\begin{bmatrix} \mathcal{X}^{c0-}_{\text{POI},k+1} \\ \mathcal{X}^{c0-}_{\text{SCR},k+1} \\ \mathcal{X}^{c0-}_{x2,k+1} \end{bmatrix} = \begin{bmatrix} \mathcal{X}^{c0-}_{\text{POI},k+1} \\ 0_{[n_{\text{SCR}},1]} \\ 0_{[n_2,1]} \end{bmatrix} \tag{2.20}$$

$$\begin{bmatrix} \mathcal{X}^{c-}_{\text{POI},k+1} \\ \mathcal{X}^{c-}_{\text{SCR},k+1} \\ \mathcal{X}^{c-}_{x2,k+1} \end{bmatrix} = \begin{bmatrix} \mathcal{X}^{c-}_{\text{POI},k+1} & \mathcal{X}^{c0-}_{\text{POI},k+1} \cdot \mathbf{1}_{[1,2n_{\text{SCR}}]} & \mathcal{X}^{c0-}_{\text{POI},k+1} \cdot \mathbf{1}_{[1,2n_{x2}]} \\ 0 & \begin{bmatrix} +S_{\text{SCR},k+1} & -S_{\text{SCR},k+1} \end{bmatrix} & 0 \\ 0 & 0 & \begin{bmatrix} +S^{-}_{x2,k+1} & -S^{-}_{x2,k+1} \end{bmatrix} \end{bmatrix}$$

The propagated sigma points are calculated by adding the mean, or

$$\begin{bmatrix} \mathcal{X}^{0-}_{\text{POI},k+1} \\ \mathcal{X}^{0-}_{\text{SCR},k+1} \\ \mathcal{X}^{0-}_{x2,k+1} \end{bmatrix} = \begin{bmatrix} \mathcal{X}^{c0-}_{\text{POI},k+1} \\ \mathcal{X}^{c0-}_{\text{SCR},k+1} \\ \mathcal{X}^{c0-}_{x2,k+1} \end{bmatrix} + \begin{bmatrix} \mathbf{x}^{-}_{\text{POI},k+1} \\ 0_{[\text{SCR},1]} \\ \mathbf{x}^{-}_{x2,k+1} \end{bmatrix},$$

$$\begin{bmatrix} \mathcal{X}^{-}_{\text{POI},k+1} \\ \mathcal{X}^{-}_{\text{SCR},k+1} \\ \mathcal{X}^{-}_{x2,k+1} \end{bmatrix} = \begin{bmatrix} \mathcal{X}^{c-}_{\text{POI},k+1} \\ \mathcal{X}^{c-}_{\text{SCR},k+1} \\ \mathcal{X}^{c-}_{x2,k+1} \end{bmatrix} + \begin{bmatrix} \mathbf{x}^{-}_{\text{POI},k+1} \\ 0_{[\text{SCR},1]} \\ \mathbf{x}^{-}_{x2,k+1} \end{bmatrix} \cdot \mathbf{1}_{[1,2n_{2a}]} \tag{2.21}$$

Finally, the sigma point weights are updated for the augmented state vector with n_{2a} states in the system. As shown in (2.10), only the central (0) sigma point weights must be updated

$$W^0_m = \frac{\sigma_f^2 - n_{2a}}{\sigma_f^2}, \;\; W^0_c = \frac{\sigma_f^2 - n_{2a}}{\sigma_f^2} + 3 - \frac{\sigma_f^2}{n_{2a}} \tag{2.22}$$

2.4.3 SR-SPIF Update

The SR-SPIF update is similar to an update on the square root information form. First, the state sigma points are propagated through the nonlinear measurement

equation (2.5), and sample mean and measurement sigma points are calculated.

$$\begin{aligned}
\mathcal{Z}^{j}_{\mathrm{SCR},k+1} &= g_{\mathrm{SCR}}(\mathcal{X}^{j-}_{\mathrm{POI},k+1}, \mathcal{X}^{j-}_{x2,k+1}, \mathcal{X}^{j-}_{\mathrm{SCR},k+1}), \quad j = 0, \cdots, 2n_{2a} \\
\hat{\mathbf{z}}_{\mathrm{SCR},k+1} &= \sum_{j=0}^{2n_{2a}} W^{j}_{m} \mathcal{Z}^{j}_{\mathrm{SCR},k+1} \\
\mathcal{Z}^{c0}_{\mathrm{SCR},k+1} &= \mathcal{Z}^{0}_{\mathrm{SCR},k+1} - \hat{\mathbf{z}}_{\mathrm{SCR},k+1} \\
\mathcal{Z}^{c}_{\mathrm{SCR},k+1} &= \left[\begin{array}{c} \mathcal{Z}^{1}_{\mathrm{SCR},k+1} - \hat{\mathbf{z}}_{\mathrm{SCR},k+1} \end{array} , \cdots, \mathcal{Z}^{2n_{2a}}_{\mathrm{SCR},k+1} - \hat{\mathbf{z}}_{\mathrm{SCR},k+1} \right]
\end{aligned} \tag{2.23}$$

Next, the stochastic linearization of the nonlinear measurement is found using the state-measurement cross covariance,

$$P_{\mathrm{POI}x2,\mathrm{SCR}} = W \cdot \begin{bmatrix} \mathcal{X}^{c-}_{\mathrm{POI},k+1} \\ \mathcal{X}^{c-}_{x2,k+1} \end{bmatrix} \left(\mathcal{Z}^{c}_{\mathrm{SCR},k+1}\right)^{T} + \tag{2.24}$$

$$W^{c}_{0} \cdot \begin{bmatrix} \mathcal{X}^{c0-}_{\mathrm{POI},k+1} \\ \mathcal{X}^{c0-}_{x2,k+1} \end{bmatrix} \left(\mathcal{Z}^{c0}_{\mathrm{SCR},k+1}\right)^{T} \tag{2.25}$$

$$C_{\mathrm{SCR},k+1} = \left(P_{\mathrm{POI}x2,\mathrm{SCR}}\right)^{T} \begin{bmatrix} P^{-1}_{\mathrm{POI},k+1} & 0 \\ 0 & P^{-1}_{x2,k+1} \end{bmatrix} \tag{2.26}$$

where $P^{-1}_{\mathrm{POI},k+1}$ and $P^{-1}_{x2,k+1}$ are computed as

$$\begin{aligned}
P^{-1}_{\mathrm{POI},k+1} &= \left(R^{-}_{\mathrm{POI},k+1}\right)^{T} R^{-}_{\mathrm{POI},k+1} \\
P^{-1}_{x2,k+1} &= \left(R^{-}_{x2,k+1}\right)^{T} R^{-}_{x2,k+1}
\end{aligned} \tag{2.27}$$

The square root information state and gain are then calculated,

$$i_{k+1} = \begin{bmatrix} R^{-}_{x2,k+1} \hat{\mathbf{x}}^{-}_{x2,k+1} \\ R_{\mathrm{SCR},k+1} \left(\nu_{k+1} + C_{\mathrm{SCR},k+1} \begin{bmatrix} \hat{\mathbf{x}}^{-}_{\mathrm{POI},k+1} \\ \hat{\mathbf{x}}^{-}_{x2,k+1} \end{bmatrix} \right) \end{bmatrix} \tag{2.28}$$

$$I_{k+1} = R_{\mathrm{SCR},k+1} C_{\mathrm{SCR},k+1} \tag{2.29}$$

where ν_{k+1} is the measurement innovation and is given by

$$\nu_{k+1} = \mathbf{z}_{\mathrm{SCR},k+1} - \hat{\mathbf{z}}_{\mathrm{SCR},k+1} \tag{2.30}$$

The square root information state and gain are subsequently used for two purposes: 1) on-board the current UAV in its SR-SPIF geolocation estimator, and 2) communicating to the other UAVs for fusion into their SR-SPIF geolocation estimators.

An orthogonalization is performed to find the updated square root information matrix for the POI states $R_{\text{POI},k+1}$,

$$T_{k+1}\begin{bmatrix} R_{\text{POI},k+1} & R_{\text{POI},x2,k+1} \\ 0 & R_{x2,k+1} \end{bmatrix} = orth\left\{\begin{bmatrix} \begin{bmatrix} R^{-}_{\text{POI},k+1} & 0 \\ 0 & R^{-}_{x2,k+1} \end{bmatrix} \\ I_{k+1} \end{bmatrix}\right\} \tag{2.31}$$

Finally, the POI state estimate, $\hat{\mathbf{x}}_{\text{POI},k+1}$, and square root covariance, $S_{\text{POI},k+1}$, are calculated, (2.32) and (2.33), to be used both for the next sigma points and for the geolocation tracking outputs.

$$\begin{aligned} \hat{\mathbf{x}}_{\text{POI},k+1} \;=\; & \begin{bmatrix} R^{-1}_{\text{POI},k+1} & -R^{-1}_{\text{POI},k+1}R_{\text{POI},x2,k+1}R^{-1}_{x2,k+1} \end{bmatrix} \cdot \\ & T^{T}_{k+1}\begin{bmatrix} R^{-}_{\text{POI},k+1}\hat{\mathbf{x}}^{-}_{\text{POI},k+1} \\ i_{k+1} \end{bmatrix} \end{aligned} \tag{2.32}$$

$$\begin{bmatrix} S_{\text{POI},k+1} & S_{\text{POI},x2,k+1} \\ 0 & S_{x2,k+1} \end{bmatrix} = orth\left\{\begin{bmatrix} R^{-1}_{\text{POI},k+1} & -R^{-1}_{\text{POI},k+1}R_{\text{POI},x2,k+1}R^{-1}_{x2,k+1} \\ 0 & R^{-1}_{x2,k+1} \end{bmatrix}\right\} \tag{2.33}$$

The calculations in (2.32) and (2.33) are computationally moderate because they are required for the POI states only, or the first n_{POI} states of the updated state and covariance, and the inversions are of upper right triangular matrix factors.

The sigma points are then re-calculated for the prediction step as in (2.9) and the process repeats.

2.4.4 Cooperative Fusion

Fusion of estimates from multiple UAVs tracking the same POI is now a straightforward integration of the information state and gains received from the other UAVs into the update equations (2.31)-(2.33). First, the information gain is decomposed into partitions associated with the POI and secondary states of each UAV. For the j^{th} UAV, this is defined as

$$[I_{k+1}]^j = \begin{bmatrix} [I_{\text{POI},k+1}]^j & [I_{x2,k+1}]^j \end{bmatrix} \tag{2.34}$$

Equations 2.31-2.33 are then rewritten in a form for cooperative fusion over N UAVs tracking the POI as shown in (2.35),(2.36), and (2.37).

$$T_{k+1}\begin{bmatrix} R_{\text{POI},k+1} & \overline{R}_{\text{POI},x2,k+1} \\ 0 & \overline{R}_{x2,k+1} \end{bmatrix} = orth\left\{\left[\begin{bmatrix} R^-_{\text{POI},k+1} & 0 & 0 & 0 \\ 0 & [R^-_{x2,k+1}]^1 & 0 & 0 \\ & & \ddots & \\ 0 & 0 & 0 & [R^-_{x2,k+1}]^N \end{bmatrix} \begin{bmatrix} [I_{\text{POI},k+1}]^1 & [I_{x2,k+1}]^1 & 0 & 0 \\ \vdots & 0 & \ddots & \\ [I_{\text{POI},k+1}]^N & 0 & 0 & [I_{x2,k+1}]^N \end{bmatrix}\right]\right\} \tag{2.35}$$

$$\hat{\mathbf{x}}_{\text{POI},k+1} = \begin{bmatrix} R^{-1}_{\text{POI},k+1} & -R^{-1}_{\text{POI},k+1}\overline{R}_{\text{POI},x2,k+1}\overline{R}^{-1}_{x2,k+1} \end{bmatrix} T^T_{k+1} \begin{bmatrix} R^-_{\text{POI},k+1}\hat{\mathbf{x}}^-_{\text{POI},k+1} \\ [i_{k+1}]^1 \\ [i_{k+1}]^2 \\ \vdots \\ [i_{k+1}]^N \end{bmatrix} \tag{2.36}$$

$$\begin{bmatrix} S_{\text{POI},k+1} & \overline{S}_{\text{POI},x2,k+1} \\ 0 & \overline{S}_{x2,k+1} \end{bmatrix} = orth\left\{\begin{bmatrix} R^{-1}_{\text{POI},k+1} & -R^{-1}_{\text{POI},k+1}\overline{R}_{\text{POI},x2,k+1}\overline{R}^{-1}_{x2,k+1} \\ 0 & \overline{R}^{-1}_{x2,k+1} \end{bmatrix}\right\} \tag{2.37}$$

2.5 Augmentations to the Geolocation Filter

Three augmentations to the SR-SPIF geolocation filter are considered here because of their importance to the final solutions: 1) Additional sensor measurements, 2) Delayed communication transmissions, and 3) Multiple model tracking. Each of these is described in the following subsections.

2.5.1 Additional sensor measurements

In the development of the SR-SPIF, the measurement of the target was assumed to be the location of the target image in the camera screen (SCR). Frequently, a reasonable estimate of the POI altitude, z_h, is available (Ocean vessels, tracking a ground target with a terrain map). When the altitude is provided as an estimate, it can simply be appended to the list of measurements with the corresponding measurement equation as

$$\begin{bmatrix} \mathbf{z}_{\text{SCR}} \\ z_h \end{bmatrix} = \begin{bmatrix} g_{\text{SCR}}(\mathbf{x}_{\text{POI}}, \begin{bmatrix} \mathbf{x}_{\text{NAV}} \\ \mathbf{x}_{\text{ATT}} \\ \mathbf{x}_{\text{GIM}} \end{bmatrix}, \mathbf{v}_{\text{SCR}}) \\ g_h(\mathbf{x}_{\text{POI}}, \mathbf{v}_h) \end{bmatrix} \tag{2.38}$$

In the current application, the inclusion of an altitude measurement can significantly improve the tracking performance especially for highly maneuverable targets because, when a single UAV using a camera tracks the POI, the states of the POI are instantaneously unobservable. Adding an estimate of the POI altitude, or using two UAVs to cooperatively track a POI, improves observability of the POI states and therefore improves geolocation performance (convergence and

steady state). It is noted that additional sensor measurements, such as laser or radar measurements, can be addressed in a similar manner.

2.5.2 Estimation with Delayed Data

It is important in cooperative estimation schemes that all measurement data is time stamped, and fused at the appropriate time. This implies that when data is delayed, either because of a communication lag, communication blackout, or problem with the camera or estimator, the cooperative estimation algorithms must robustly handle each situation to provide accurate tracking results.

When considering delayed data problems, solutions differ over three important metrics: tracking accuracy, memory storage (communication), and computation. The simplest method to address delayed data is to simply not fuse any information that is not communicated; therefore, fusion only occurs when communication is working. This approach uses little memory storage and communication, but is low performing, especially with lossy communication systems. Therefore, it should only be used in systems with robust communications.

A second approach is to spawn a new information filter when there is a communication drop, and then communicate all information in a small burst when communication is working again. This approach, which also has good memory and communication properties, is alluded to in [44] as a common, conservative option. But, as described in [40], this approach yields suboptimal results as compared to the centralized solution. In fact, the results are not conservative, but overly optimistic. This is a result of the inherent coupling between the incoming information streams as a result of the common process noise model (for the POI

states).

A third method addresses the performance shortcomings of the first two. This method stacks all information to be shared in a queue, and guarantees the information is communicated, even if it is shared at a much later time. This method is similar in concept to that described in [44]. When information is received, the on-board system goes back to the time stamp of the incoming data, and iterates the filter up to the current time. This method, if implemented correctly, can match or even exceed the performance the centralized, no loss solution. This is because, with nonlinear dynamics and measurements, one could, in theory, smooth and filter the estimates using the measurement data. The disadvantage of this method is that it requires large amounts of storage, high computation, and at times bursty communication.

A fourth method is proposed and described here. This method attempts to spawn a new, modified information filter when communication is dropped. Specifically, when communication is stopped, a new information filter is spawned, where the initial condition on the information state and gain is zero. The new aspect of this filter is that, while the vehicles are not communicating, the correlated information between the vehicles is estimated.

To understand the method more clearly, consider the case of two UAVs tracking the same POI. In this example, full information (not square root) forms and only the information gain are considered to improve clarity of the discussion; square root versions are easily derived as modified versions of the previous SR-SPIF estimator. At time k, each UAV has communicated with the other such that they have common information on the POI states, Y_k. On-board UAV_1, the geolocation estimator attempts to fuse its own information gain from the measurements, $I_{1,k}$, as well as the information gain that is to be communicated from the second UAV,

$I_{2,k}$. For this case, one can write a discrete Hamiltonian, which represents both the prediction and update steps in matrix form,

$$\begin{bmatrix} X_{n,k+1} \\ X_{d,k+1} \end{bmatrix} = \begin{bmatrix} A_k^{-T}(I_{1,k} + I_{2,k}) & A_k^{-T} \\ A_k^{-T} + QA_k^{-T}(I_{1,k} + I_{2,k}) & QA_k^{-T} \end{bmatrix} \begin{bmatrix} X_{n,k} \\ X_{d,k} \end{bmatrix} \tag{2.39}$$

where A_k is the linearized states dynamics matrix, and Q is the process noise covariance. The updated information matrix is then written as

$$Y_{k+1} = X_{d,k+1}X_{n,k+1}^{-1} = [\bar{Y}_k + \bar{I}_{1,k} + \bar{I}_{2,k}] \cdot [I + Q(\bar{Y}_k + \bar{I}_{1,k} + \bar{I}_{2,k})]^{-1} \tag{2.40}$$

and $\bar{(\cdot)}$ indicates propagation through the system matrix, or $A_k^{-T}(\cdot)A_k^{-1}$. This form clearly shows the different elements of the fusion problem: 1) information is added, 2) information is discounted due to the correlated process noise Q, and 3) information is discounted due to the dynamics $\bar{(\cdot)}$.

Consider the case when there may be a communication drop (or delayed data). Ideally, the information on UAV_2, which is to be communicated to UAV_1, must be processed and stored for eventual communication in the future. Equation 2.40 is then considered on UAV_1, when the information is not received from UAV_2, but separating the primary component of $(\bar{I}_{2,k})$, or

$$Y_{k+1} = [\bar{Y}_k + \bar{I}_{1,k}] \cdot [I + Q(\bar{Y}_k + \bar{I}_{1,k} + \bar{I}_{2,k})]^{-1} + \tag{2.41}$$

$$[\bar{I}_{2,k}] \cdot [I + Q(\bar{Y}_k + \bar{I}_{1,k} + \bar{I}_{2,k})]^{-1} \tag{2.42}$$

The left term is similar to a filter implementation on UAV_1 alone, with the exception of the discounted information from UAV_2: $Q\bar{I}_{2,k}$. The term on the right is similar to that of propagating UAV_2 information, but discounted by all information (a priori, and from UAV_1 and (2)): $Q(\bar{Y}_k + \bar{I}_{1,k} + \bar{I}_{2,k})$. This coupling through the process noise is a key point. With no process noise, $Q = 0$, the information simply adds. But the process noise creates a coupling, which in turn creates difficulties in handling within the fusion process.

The second approach described above, where a new filter is spawned and only the new information is propagated through the system dynamics, leads to a fusion term of the form (on UAV (2), being readied for communication to UAV_1)

$$I_{2PO,k+1} = [\bar{I}_{2,k}] \cdot [I + Q(\bar{I}_{2,k})]^{-1} \tag{2.43}$$

This approach, termed"propagated information only (PO)", is clearly too optimistic in its generation of information, even for small process noise, compared to the true information shown in (2.42).

The approach proposed here attempts to estimate the true UAV (2) information, or the right hand term of (2.42). To do this, UAV_2 must estimate the current information Y_k and the information on UAV_1, $\bar{I}_{1,k}$. This is written as

$$I_{2PD,k+1} = [\bar{I}_{2,k}] \cdot [I + Q(\hat{\bar{Y}}_k + \hat{\bar{I}}_{1,k} + \bar{I}_{2,k})]^{-1} \tag{2.44}$$

This approach is termed "propagated discounted information (PD)." It is assumed that the information gain and fused information on UAV_1 do not change over small periods of time. Thus, the information estimates are written as

$$\begin{aligned} \hat{\bar{Y}}_l &= \bar{Y}_k \\ \hat{\bar{I}}_{1,l} &= \bar{Y}_{1,k} \end{aligned} \qquad \text{where } \; l \in \{k+1, \cdots, k+M\} \tag{2.45}$$

where M is the number of time steps of the communication delay. This is a recursion that enables very little memory or communication to be required, yet is a good approximation of the fusion over small drops in communication. Because these estimates are simply used to "discount" the current information to be communicated, the information fusion process holds for longer periods of delayed data compared to the other methods. This approach typically begins to break down when the information changes due to the nonlinear nature of the dynamics or measurement.

The following process is proposed for handling delayed data. When a communication drop occurs, three filters are started on each UAV. Consider the three filters on UAV_1: 1) a filter using no measurements from the UAV_2 (left side of (2.42) with $\bar{I}_{2,k} = 0$), 2) a filter using an estimate from UAV_2 (left side of (2.42) with $\hat{\bar{I}}_{2,k} = \bar{I}_{2,k-1}$), and 3) a filter preparing discounted UAV_1 information (right side of (2.42) with indices 1,2 switched, and $\hat{\bar{Y}}_k = \bar{Y}_{k-1}, \bar{I}_{2,k} = \hat{\bar{I}}_{2,k-1}$). Filter #1 provides the best estimate based on the current information; filter #2 provides the best filter for eventual fusion with UAV_2 information; and filter #3 provides the best UAV_1 information for eventual communication to UAV_2.

The delayed data options were evaluated using flight data from the SeaScan UAVs (to be described later). The single UAV tracking estimators were used to track a stationary target. At multiple times during the implementation, a communication drop was simulated. Different estimators were evaluated: Single UAV, two UAVs (continuous communication), two UAVs (communication drop, propagated information only for fusion), two UAVs (communication drop, no fusion), and two UAVs (new discounted information method proposed here). This process was repeated over 25 flight data files; all data presented are averages of these results.

Figure 2.3 shows a plot of the one sigma uncertainty volume for each of these cases over time; a delay of $M = 50$ data points occurs at $k = 300$ (normalized for each data file). The single UAV and two UAV (continuous communication) cases present upper and lower bounds on the volume of the uncertainty ellipsoid (inversely proportional to the information). In all cases, when communication is dropped at $k = 300$, the uncertainty volume begins to increase. This is because the tracking has converged to a pseudo-steady state solution before there is a

communication drop. In the no fusion case, at $k = 350$, the volume begins to decrease again, and eventually (~ 100 time steps) converges back to the two UAV solution (continuous communications). In the propagated information only case, at $k = 350$, the volume immediately decreases to far below the level of the two UAV limit, thus showing how it produces overly optimistic results. In addition, a transient is created which takes a long time (> 400 time steps) to damp out. In the proposed discounted information case, at $k = 350$, the volume immediately decreases to a level that is approximately the same as the two UAV limit. This shows that, even with a 50 time step delay, a nonlinear measurement and a moving sensor on a UAV, the proposed discounted information approach works quite well.

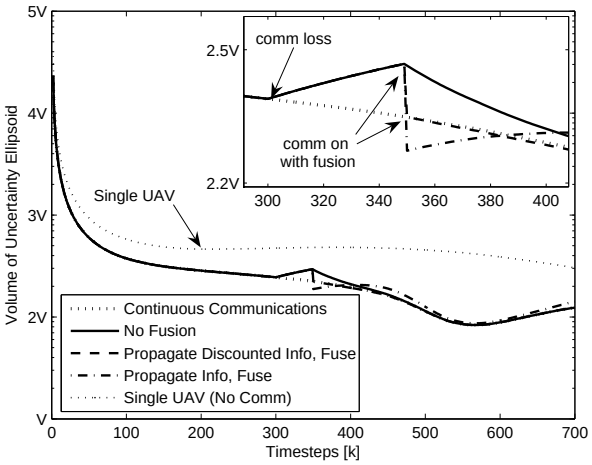


Figure 2.3: Plot of the one sigma uncertainty volume over time for several cases: single UAV, two UAVs cooperating with full communication, and two UAVs cooperating with communication drops, with and without fusion.

Figure 2.4 compares three cases: no fusion, the queuing approach [44], and the proposed, discounted information approach, using the three critical metrics for delayed data problems: 1) the maximum memory storage required (which is also coupled with the required communication), 2) the maximum computation required during one cycle, and 3) the performance, defined here as the settling

time for the volume to return to within 5% of the two UAV solution (continuous communications). As shown, the no fusion case has the best (smallest) memory and best (smallest) computation, but takes a longer time to converge back to the steady state solution; it is noted that this convergence time increases with the delay M. The queuing approach requires increasing levels of memory and computation, but converges immediately back to the original answer. Finally, the discounted information approach is always within 5% of the two UAV solution (continuous communications), and requires only storing the current delayed information matrix and state. There is a slight increase in computation, but it is also nicely distributed across each UAV.

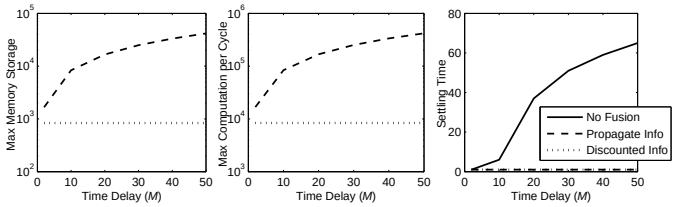


Figure 2.4: Plot of the memory (communication), computation, and performance (time to converge back to the two UAV solution) for the delayed data simulations.

2.5.3 Square Root, Interacting Multiple Model Filter (SR-IMM)

In tracking systems with moving targets, an important algorithm is the Interacting Multiple Model (IMM) filter [4]. The IMM is designed to run n_m tracking estimators with different dynamics models in parallel, and mix the results at each time step to produce n_m new estimates, as well as an estimate of the filter probability. This enables multiple types of targets, or multiple modes of targets, to be explored. As an extension of the SR-SPIF, a square root interacting multiple

model (SR-IMM) filter implementation is developed.

To begin, it is assumed that n_m SR-SPIF geolocation estimators are available using n_m different dynamics models. This is defined as

$$\mathbf{x}^l_{\text{POI},k-1}, S^l_{\text{POI},k-1} \quad \text{for} \;\; l = 1, \cdots, n_m \tag{2.46}$$

Also, available is a Markov chain transition matrix, $\mathcal{P}$, such that the l^{th} row and j^{th} column, $\mathcal{P}^{lj}$ is the probability of switching from model l to model j. This transition matrix is typically chosen using a priori knowledge of the model structure. A mode probability is also defined, ν^l_{k-1}, for $l = 1, \cdots, n_m$

Using the above definitions, a mixing coefficient matrix μ at time step $k-1$ is defined to be

$$\mu^{lj}_{k-1} = \frac{1}{\bar{c}^j}\mathcal{P}^{lj}\nu^l_{k-1}, \quad \text{for} \;\; l, j = 1, \cdots, n_m \tag{2.47}$$

where $\bar{c}^j$ is a normalizing constant $\bar{c}^j = \sum_{l=1}^{n_m}\mathcal{P}^{lj}\nu^l_{k-1}$. The mixing coefficient matrix μ is then used to find n_m probabilistic mixes of the n_m estimates/covariances at time $k-1$. The mixed state estimates are given as

$$\hat{\mathbf{x}}^m_{\text{POI},k-1} = \sum_{l=1}^{n_m}\hat{\mathbf{x}}^l_{\text{POI},k-1}\mu^{lm}_{k-1}, \quad \text{for} \;\; m = 1, \cdots, n_m \tag{2.48}$$

And an orthogonalization is used to find each of the n_m mixed square root covariances

$$T^m_{k-1}S^m_{\text{POI},k-1} = orth\left\{\begin{array}{c} \sqrt{\mu^{1m}_{k-1}}\begin{bmatrix} S^1_{\text{POI},k-1} \\ (\hat{\mathbf{x}}^1_{\text{POI},k-1} - \hat{\mathbf{x}}^m_{\text{POI},k-1})^T \end{bmatrix} \\ \vdots \\ \sqrt{\mu^{n_m m}_{k-1}}\begin{bmatrix} S^{n_m}_{\text{POI},k-1} \\ (\hat{\mathbf{x}}^{n_m}_{\text{POI},k-1} - \hat{\mathbf{x}}^m_{\text{POI},k-1})^T \end{bmatrix} \end{array}\right\} \tag{2.49}$$

with the index, m, running from 1 to n_m. Each of the n_m mixed estimates and square-root covariances at time $k-1$ are then propagated through the appropriate

dynamics and fused with the measurement $\mathbf{z}_{\text{SCR},k}$ at time k using the SR-SPIF. That is,

$$[\hat{\mathbf{x}}^m_{\text{POI},k}, S^m_{\text{POI},k}, \hat{\mathbf{z}}^m_{\text{SCR},k}] = \text{SR–SPIF}\left(\hat{\mathbf{x}}^m_{\text{POI},k-1}, S^m_{\text{POI},k-1}, \mathbf{z}_{\text{SCR},k}\right) \quad \text{for} \quad m = 1, \cdots, n_m \tag{2.50}$$

The mixing probabilities are then updated using measurement likelihoods, or how likely a measurement $\mathbf{z}_{\text{SCR},k}$ is given a particular mixing model. The likelihood is formally written as

$$\Lambda^m_k = \mathcal{P}[\mathbf{z}_{\text{SCR},k} | \hat{\mathbf{x}}^m_{\text{POI},k-1}, S^m_{\text{POI},k-1}] \tag{2.51}$$

This can be approximated as a normal distribution [4]

$$\begin{aligned} \Lambda^m_k &\approx \mathcal{N}[\hat{\mathbf{z}}^m_{\text{SCR},k}; \mathbf{z}_{\text{SCR},k}, P_{\text{SCR},k}] \\ &= \frac{1}{\sqrt{(2\pi)^{n_{\text{SCR}}} |P_{\text{SCR},k}|}} e^{-\frac{1}{2}(\hat{\mathbf{z}}^m_{\text{SCR},k} - \mathbf{z}_{\text{SCR},k})^T P_{\text{SCR},k} (\hat{\mathbf{z}}^m_{\text{SCR},k} - \mathbf{z}_{\text{SCR},k})} \end{aligned} \tag{2.52}$$

where $P_{\text{SCR},k} = (S_{\text{SCR},k})^T S_{\text{SCR},k}$ is the sensor noise covariance. The mixing probabilities are then calculated by normalizing the likelihoods, or

$$\nu^j_k = \frac{1}{c} \Lambda^j_k \bar{c}_j \quad \text{for } j = 1, \cdots, n_m \tag{2.53}$$

where $c = \sum_{j=1}^{n_m} \Lambda^j_k \bar{c}_j$. The process is now repeated starting at (2.47).

During operation, it is typically important to develop a single estimate for use external to the estimator. In this case, the n_m estimates are further mixed to find

a single estimate and square root covariance, as follows.

$$\hat{\mathbf{x}}_{\mathrm{POI},k} = \sum_{j=1}^{n_m} \hat{\mathbf{x}}^j_{\mathrm{POI},k} \nu^j_k \tag{2.54}$$

$$T_{k-1} S_{\mathrm{POI},k-1} = orth \left\{ \begin{array}{c} \sqrt{\nu^1_k} \left[\begin{array}{c} S^1_{\mathrm{POI},k} \\ (\hat{\mathbf{x}}^1_{\mathrm{POI},k} - \hat{\mathbf{x}}_{\mathrm{POI},k})^T \end{array} \right] \\ \vdots \\ \sqrt{\nu^{n_m}_k} \left[\begin{array}{c} S^{n_m}_{\mathrm{POI},k} \\ (\hat{\mathbf{x}}^{n_m}_{\mathrm{POI},k} - \hat{\mathbf{x}}_{\mathrm{POI},k})^T \end{array} \right] \end{array} \right\} \tag{2.55}$$

2.6 The SeaScan UAV

The SeaScan is a long endurance UAV developed by the Insitu Group [61], and is now a key component in Boeing's UAV strategic plan. The SeaScan is unique, as compared to UAVs in other university testbeds, because it:

- has a maximum flight time up to 24 hours
- is very robust, with 15+ years of development and operations in weather reconnaissance [24], crossing the Atlantic [38], and over 25,000 flight hours in combat operations in theater since 2004
- has visual and infrared sensing capability
- has a payload capability for higher level autonomy

Cornell recently has worked with the Insitu Group on developing a payload capability for the SeaScan [9, 8], which has been used for several flight tests. Figure 2.5 shows the SeaScan UAV during flight.

Figure 2.5: The SeaScan UAV in flight.

2.6.1 Avionics, Inertial/Attitude Estimation, Control Loops

The SE555 processor board provides inner loop control and data management. The SE555 board receives data from aircraft sensors and sends commands to control surfaces; flight path characteristics can be determined from pre-programmed or in-flight commands. The air-to-ground communication is a data link used to communicate aircraft status, control, and mission data, as well as to relay messages from payload modules. The avionics-to-payload communication is a data link used to send sensor reports to the payload, and receive commands from the payload. The SE555 board communicates serially with an on-board video gimbal, issuing camera positioning commands as well as aircraft attitude and stabilization data. The video signal bypasses the SE555 board and is sent directly to a ground receiver via an onboard radio frequency link. The SE555 board also communicates

serially with an onboard GPS receiver, receiving differential GPS data from the ground to improve position and velocity solutions. The onboard sensors include roll, pitch and yaw rate gyros; vertical, lateral and longitudinal accelerometers; external temperature sensors; relative pressures of pitot, alpha, beta, gamma for wind axes estimation; and absolute pressures of barometric and manifold.

2.6.2 Camera and Image Software

The centerpiece of SeaScan is a digital video camera integrated into an inertially-stabilized pan / tilt nose turret. The daylight camera has an acuity $\approx 50\%$ better than that of the unaided eye at the telescopic end. It can resolve POIs such as small boats and logs from five miles away. The operator can command the camera to pan back-and-forth for wide-area search, or to remain locked onto a POI while the aircraft maneuvers; the latter mode is used here.

Ground software processes the images from the camera. When the user selects a POI for geolocation, the gimballing turret and ground software attempt to maintain the POI in the center of the frame, from frame to frame. Therefore, the "measurement" of the POI is assumed to be at the center of the image frame, as shown in Fig. 2.6.

2.6.3 Sensor Measurement

The measurement equation, (2.2), is a complicated function of the UAV states and the POI states, which yield the screen coordinates in terms of pixels. For simplicity, it is assumed that the GPS antenna, center of the aircraft, and camera

axes are all located at the same point. The full derivation of the measurement equation is developed in [10], and a summary is presented here.

The sensor measurement is formally defined as the location of the POI in screen coordinates, or PCI,

$$\mathbf{z}_{\mathrm{SCR}} = \begin{bmatrix} x_{\mathrm{S-PCI}} \\ y_{\mathrm{S-PCI}} \end{bmatrix} \tag{2.56}$$

The measurement is converted to camera (CAM) coordinates using the field of view (FOV), as well as the rotations of the aircraft position with respect to the Earth, attitude and gimbal.

$$\mathbf{z}_{\mathrm{SCR}} = \begin{bmatrix} x_{\mathrm{S-PCI}} \\ y_{\mathrm{S-PCI}} \end{bmatrix} \tag{2.57}$$

$$= \begin{bmatrix} y_{\mathrm{C-PCI}}/\lambda_y \\ z_{\mathrm{C-PCI}}/\lambda_z \end{bmatrix} \tag{2.58}$$

$$= \begin{bmatrix} R^{\mathrm{CAM}}_{\mathrm{ECEF}}(2)/\lambda_y \\ R^{\mathrm{CAM}}_{\mathrm{ECEF}}(3)/\lambda_z \end{bmatrix} \left[R^{\mathrm{CAM}}_{\mathrm{ECEF}}(1)\left(\mathbf{x}_{\mathrm{E-POI}} - \mathbf{x}_{\mathrm{E-NAV}}\right)\right]^{-1} \left(\mathbf{x}_{\mathrm{E-POI}} - \mathbf{x}_{\mathrm{E-NAV}}\right)$$

where

$$\lambda_y = \frac{\tan\left(FOV/2\right)}{p_{max-y}}, \lambda_z = \frac{\tan\left(FOV/2\right)}{p_{max-z}}$$

are pixel length scale factors, $p_{max-(y,z)}$ is the maximum camera pixels in the y, z directions. The rotation $R^{\mathrm{CAM}}_{\mathrm{ECEF}}$ is a combination of rotations from the camera (CAM) to aircraft body (ABC) to aircraft in Local level East-North-up (ENU) to the ECEF coordinates,

$$R^{\mathrm{CAM}}_{\mathrm{ECEF}} = \begin{bmatrix} R^{\mathrm{CAM}}_{\mathrm{ECEF}}(1) \\ R^{\mathrm{CAM}}_{\mathrm{ECEF}}(2) \\ R^{\mathrm{CAM}}_{\mathrm{ECEF}}(3) \end{bmatrix} = (R^{\mathrm{ECEF}}_{\mathrm{ENU}} R^{\mathrm{ENU}}_{\mathrm{ABC}} R^{\mathrm{ABC}}_{\mathrm{CAM}})^{-1}$$

where the rotations, $R^{\text{ECEF}}_{\text{ENU}}$, $R^{\text{ENU}}_{\text{ABC}}$, and $R^{\text{ABC}}_{\text{CAM}}$ are given in (2.59), (2.60), and (2.61).

$$R^{\text{ECEF}}_{\text{ENU}} = \begin{bmatrix} -S(y_{\text{E-NAV}}) & -S(x_{\text{E-NAV}})C(y_{\text{E-NAV}}) & C(x_{\text{E-NAV}})C(y_{\text{E-NAV}}) \\ C(y_{\text{E-NAV}}) & -S(x_{\text{E-NAV}})S(y_{\text{E-NAV}}) & C(x_{\text{E-NAV}})S(y_{\text{E-NAV}}) \\ 0 & C(x_{\text{E-NAV}}) & S(x_{\text{E-NAV}}) \end{bmatrix} \tag{2.59}$$

$$R^{\text{ENU}}_{\text{ABC}} = \begin{bmatrix} S(\psi)C(\theta) & C(\phi)C(\psi) + S(\phi)S(\psi)S(\theta) & -S(\phi)C(\psi) + C(\phi)S(\phi)S(\theta) \\ C(\psi)C(\theta) & -C(\phi)S(\psi) + S(\phi)C(\psi)S(\theta) & S(\phi)S(\psi) + C(\phi)C(\psi)S(\theta) \\ S(\theta) & -S(\phi)C(\theta) & -C(\phi)C(\theta) \end{bmatrix} \tag{2.60}$$

$$R^{\text{ABC}}_{\text{CAM}} = \begin{bmatrix} C(p) & -S(p) & 0 \\ S(p) & C(p) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C(t) & 0 & -S(t) \\ 0 & 1 & 0 \\ S(t) & 0 & C(t) \end{bmatrix} \begin{bmatrix} C(s) & -S(s) & 0 \\ S(s) & C(s) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.61}$$

and $\mathbf{x}_{\text{ATT}} = (\phi, \theta, \psi)^T$ are the roll, pitch, and yaw of the aircraft, $\mathbf{x}_{\text{GIM}} = (p, t, s)^T$ are the pan, tilt, and scan of the gimbal, $\mathbf{x}_{\text{E-POI}} = (x_{\text{E-POI}}, y_{\text{E-POI}}, z_{\text{E-POI}})^T$ and $\mathbf{x}_{\text{E-NAV}} = (x_{\text{E-NAV}}, y_{\text{E-NAV}}, z_{\text{E-NAV}})^T$ are the three dimensional POI and NAV states in Earth centered, Earth fixed (ECEF) coordinates and $S(\cdot)$ and $C(\cdot)$ denotes $\sin\cdot$ and $\cos\cdot$ respectively.

Finally, the ECEF coordinates are converted to Latitude, Longitude, and Altitude (LLA) according to

$$\begin{aligned} x_{\text{E-NAV}} &= C(y_{\text{L-NAV}})C(x_{\text{L-NAV}})(z_{\text{L-NAV}} + R_a^2/R_{av}) \\ y_{\text{E-NAV}} &= S(y_{\text{L-NAV}})C(x_{\text{L-NAV}})(z_{\text{L-NAV}} + R_a^2/R_{av}) \\ z_{\text{E-NAV}} &= C(x_{\text{L-NAV}})(z_{\text{L-NAV}} + R_b^2/R_{av}) \end{aligned} \tag{2.62}$$

where R_a, R_b are the equatorial, polar radii, and $R_{av} = \sqrt{R_a^2 C(x_{\text{L-NAV}})^2 + R_b^2 S(x_{\text{L-NAV}})^2}$. The POI and NAV states in the LLA coordinate system then make up the desired state vectors, $\mathbf{x}_{\text{POI}} = \mathbf{x}_{\text{L-POI}} = (x_{\text{L-POI}}, y_{\text{L-POI}}, z_{\text{L-POI}})^T$ and $\mathbf{x}_{\text{NAV}} = \mathbf{x}_{\text{L-NAV}} = (x_{\text{L-NAV}}, y_{\text{L-NAV}}, z_{\text{L-NAV}})^T$.

Figure 2.6: Image taken from a flight test of the SeaScan; note that the POI is at the center of the image frame.

2.7 Flight Results using the SeaScan UAV

Flight tests were performed on 6 Oct 2004 and 18 March 2006. A "truth" was set up using a GPS antenna and receiver in a car near the flight test range. Over five hours of flight tests were recorded over the two days using a variety of POIs. The flight tests evaluated stationary and moving targets, different altitudes of the UAV, and different orbits of the UAV including size and offset. In all tests, the UAV tracked the target using the on-board algorithms and camera software. All telemetry was saved to evaluate estimation and tracking accuracy as a function of the different parameters.

Flight results comparing single and two UAV tracking results (time histories and uncertainty ellipsoid evolution), correlation with UAV trajectories, stationary and moving targets, with and without an attitude estimate, and an IMM filter are presented. Although only a single UAV was used during these test flights, enough test data was taken over repeated trials to allow cooperative estimation using multiple UAVs to be evaluated. For stationary POIs, data over distinct time

blocks were used. For moving targets, data over distinct time blocks for "similar" POI motions were used. An error between the two POIs (for the two time spans) was subtracted from both the second POI and the second UAV position in order to create an appropriate data set. Given that the errors were small (several meters), and the geolocation estimator simply uses, but does not estimate the UAV state, this approach is very realistic compared to the true multiple UAV tracking problem. It is noted that the specific performance data is scaled (by a factor L) due to ITAR restrictions, but the implementation and relative comparisons are still clear.

2.7.1 POI Model

The model used for the geolocation tracking results for all implementations of the SR-SPIF is a general purpose moving target model designed to capture the nonholonomic constraint for a wheeled vehicle. The model is written as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{V} \\ \dot{\phi} \\ \dot{\Omega} \\ \dot{a} \end{bmatrix} = \begin{bmatrix} VS(\phi)/R_e \\ VC(\phi)/(C(x)R_e) \\ 0 \\ a \\ \Omega \\ -\Omega/\rho_\Omega \\ -a/\rho_a \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ w_z \\ 0 \\ 0 \\ w_\Omega \\ w_a \end{bmatrix} \tag{2.63}$$

where V, ϕ, Ω, a are the velocity, heading, turn rate and acceleration respectively, R_e is the radius of the Earth, and $C(\cdot)$ and $S(\cdot)$ denote $\cos(\cdot)$ and $\sin(\cdot)$ respectively. For implementation, this continuous model is discretized using a forward Euler scheme at 10 Hz.

2.7.2 Stationary POI

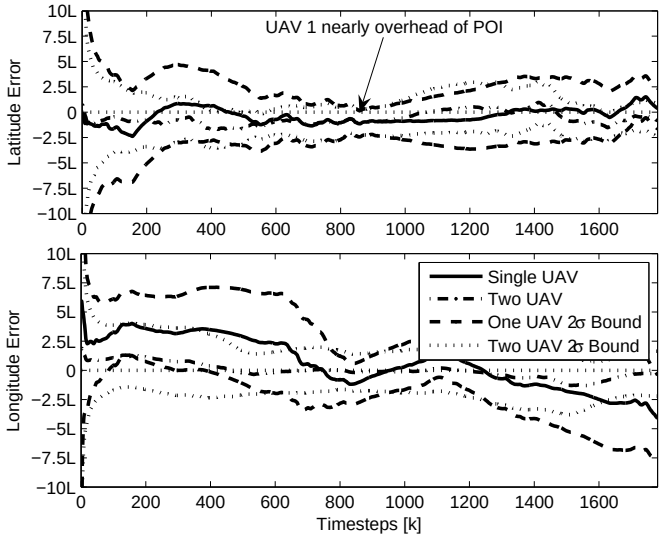


Figure 2.7: Plot of the tracking latitude and longitude errors and uncertainty bounds for a stationary target, using single and two UAVs.

Figures 2.7 and 2.8 show the two dimensional tracking estimation results for a stationary target using UAV orbits with centers off-set from the POI (as opposed to the center of the orbit being directly overhead of the target). A noisy POI altitude estimate is assumed. The errors and uncertainty bounds decrease significantly near $t = 800$ for both the single and two UAV cases. This is a result of one of the UAVs moving nearly overhead of the target, thus producing a very good, high observability measurement of the POI. Notice also that the single and multiple UAV estimates are very similar near $t = 800$ because the tracking is dominated (in terms of accuracy) by only one UAV. The second observation is that, both the errors and the uncertainties of the two UAV case are smaller than the single UAV case. While intuitive, the error reduction is significant. The SeaScan is known to have large bias errors in the attitude estimates [10]. By cooperatively tracking a target, the sensitivities of the tracking system are reduced significantly.

Figure 2.8 shows the evolution of the uncertainty ellipsoids over time. In both

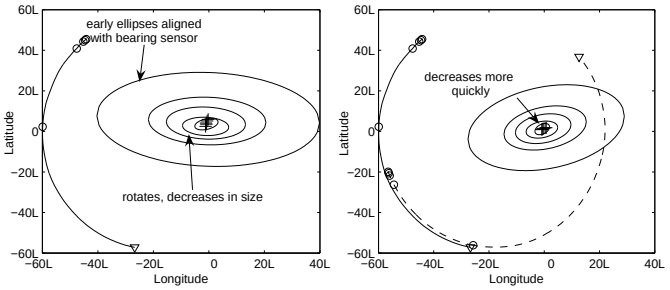


Figure 2.8: Evolution of two dimensional tracking ellipsoids over time for single (left) and two (right) UAV tracking systems.

the single and two UAV cases, the ellipsoids decrease in volume over time. For the single UAV case, the initial ellipsoid is oblong and aligned with the bearing measurement. This makes sense, as the camera is dominated by a near bearing only sensor. As the UAV orbits about the target, the uncertainty ellipsoid rotates along with the trajectory of the UAV. This is a result of the lack of observability of the POI position using a single bearing camera (even with the noisy POI altitude measurement). In the two UAV case, the initial uncertainty ellipsoid is closer to a circular ellipsoid, and the actual shape is a function of the relative spacing (approximately 60 deg in this case). The two UAVs provide bearing measurements from different perspectives, thus improving observability and reducing the uncertainty ellipsoid size equivalently in all directions.

2.7.3 Moving POI

Figure 2.9 shows an example of a geolocation problem using a moving POI. The POI trajectory begins by moving southeast at a constant velocity, slows to a stop, turns left, and increases to a constant speed moving northeast. Figure 2.9 shows the true POI motion, as well as the UAV trajectories. Individual and cooperative geolocation estimates are given at five evenly spaced measurements and the UAV

locations at the time of the measurements are indicated by the triangles. A noisy altitude measurement is assumed. Figure 2.10(a) shows the geolocation error for the moving POI for the single and two UAV cases. In the single UAV cases, the estimate errors (and their uncertainty bounds) increase when the vehicle moves further away from being overhead of the POI. The two UAV case demonstrates a small estimator error over the full time history. This is a result of the two measurements compensating for each others observability problems. Figure 2.10(a) shows the estimation errors with and without cooperative estimation and clearly, the estimation errors are reduced significantly through cooperation.

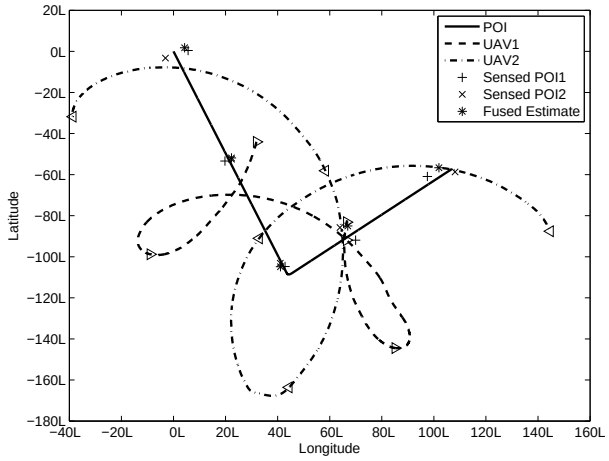
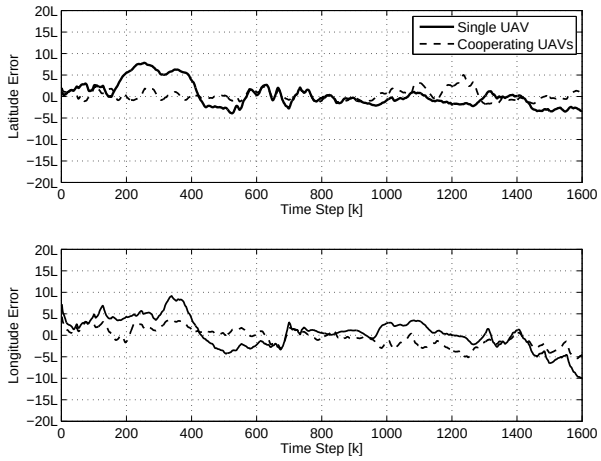


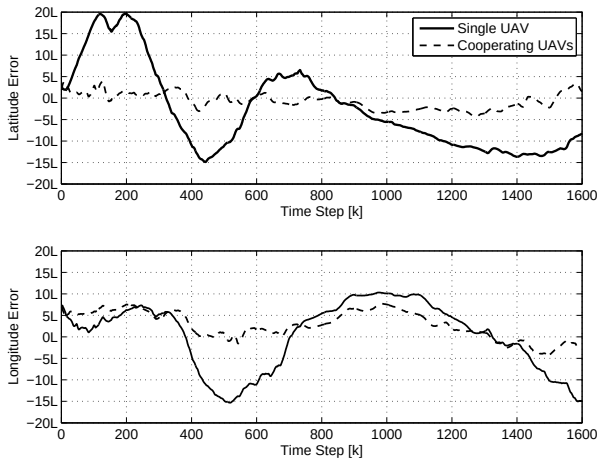
Figure 2.9: UAV trajectories (triangles indicate UAV locations for the estimates shown), POI truth, and estimated POI for single and two UAV examples tracking the POI.

2.7.4 With and Without POI Altitude Estimate

Figure 2.10(b) presents the same moving POI case, but without the use of a POI altitude measurement available. For the single UAV case, by using the POI altitude measurement, the geolocation error appears to be more stable in its convergence than without the measurement, and the error is approximately half the size as



(a) With altitude measurement.



(b) Without altitude measurement.

Figure 2.10: Latitude and longitude estimation errors for the single and multiple UAV cases, and with and without attitude measurements.

when no POI attitude measurement is used; this difference increases with range to the POI. The dependency on altitude measurement decreases in the two UAV case, when there is only a $\approx 10\%$ increase in error due to the lack of POI altitude measurement. It is noted that these trends were similar for other examples such as stationary targets. Errors (and differences) increased when the range to the POI increased, such as when the vehicles were not overhead of the target or were at a

higher altitude.

2.7.5 Implementation of the SR-IMM

The implementation of the SR-IMM estimator used $n_m = 3$ tracking models. The first is a *Setpoint Cruising Model*, which has the same kinematic equations as the *General Cruising Model* (2.63) but with different dynamics equations which attempt to maintain a predetermined speed with random disturbances in heading rate and acceleration. That is

$$\begin{bmatrix} \dot{\Omega} \\ \dot{a} \end{bmatrix} = \begin{bmatrix} -\frac{\Omega}{\rho_\Omega} + w_\Omega \\ -\frac{a}{\rho_a} - k_{ac}(V - V_c) + w_a \end{bmatrix}, \tag{2.64}$$

where k_{ac} is the target's velocity control gain and V_c is the cruising velocity. The *General Turning Model* is identical to the *General Cruising Model* (2.63) except for the turning rate and acceleration, which are given as

$$\begin{bmatrix} \dot{\Omega} \\ \dot{a} \end{bmatrix} \begin{bmatrix} k_\Omega(\Omega^3 - \Omega \cdot \Omega_c^2) + w_\Omega \\ -\frac{a}{\rho_a} + w_a \end{bmatrix}, \tag{2.65}$$

where Ω_c is the turning rate and k_Ω is a control gain to determine how quickly the target reaches the specified turning rate. Note that due to the squaring of Ω_c, both Ω_c and $-\Omega_c$ are equilibria of Ω and convergence depends on the disturbances and the initial sign of the turning rate. The *Stopping Model* is also identical to the *General Cruising Model* (2.63) except for the turning rate and acceleration, which are given as

$$\begin{bmatrix} \dot{\Omega} \\ \dot{a} \end{bmatrix} = \begin{bmatrix} -\frac{\Omega}{\rho_\Omega} + w_\Omega \\ -\frac{a}{\rho_a} - k_{as}V + w_a \end{bmatrix}, \tag{2.66}$$

where k_{as} is the control gain determining how quickly the target is brought to a stop. This is chosen in conjunction with the parameter ρ_a to make the stopping

critically damped so that the velocity is not negative during stopping.

Figure 2.11 shows the results for a single UAV tracking the same moving POI example described earlier, using the SR-SPIF and the SR-IMM. For this example, a noisy measurement of the POI altitude is used. Results show that the single model geolocation estimator increases error when the POI makes a turn, which is a discontinuity in the data. On the other hand, the SR-IMM estimate has a very small error near the corner. While it is clear that the tracking performance with the SR-IMM is better, it is noted that this comes at a cost, namely the required computation and communication transmission scales with the information associated with n_m tracking models.

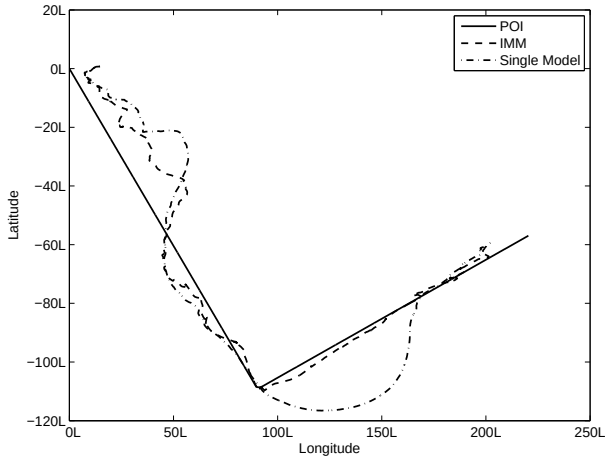


Figure 2.11: Tracking a moving POI using a single UAV, with a single model SR-SPIF and a n_m model SR-IMM estimator.

2.8 Conclusions

A square root, sigma point information filter has been developed and validated using flight test data from an autonomous uninhabited aerial vehicle with an on-

board gimballing camera. The estimator provides numerical accuracy through its square root form, accuracy in the solution through the use of sigma points, and ease in data fusion and delayed data through its information form. A new cooperative data fusion method has been developed and verified, which provides more accurate, discounted information between the vehicles especially during periods of delayed or dropped data. Important requirements in numerical precision, accuracy and memory exist for making this approach work in real time. Validation using flight test data shows excellent, robust results for single/two UAV cases, with and without altitude estimates, and using a multiple model filter. By using two UAVs to track a target, the dependency on an altitude estimate decreases significantly. Single UAV results are also quite sensitive to relative range to the target. Finally, tracking using the multiple model filter provided significantly improved performance as compared to an estimator with single model, with large errors occurring at discontinuities in the data, such as during sharp corners or changes in velocity.

CHAPTER 3

AN IMPROVED SQUARE ROOT SIGMA POINT INFORMATION FILTER FOR COOPERATIVE GEOLOCATION

3.1 Introduction

In Chapter 2, cooperative geolocation is treated as a combined state and parameter estimation problem but without completing the parameter update. The primary drawback of that approach is the complicated form of the cooperative update which requires knowledge of the cooperating UAV statistics. Furthermore, it requires communicating information not only about the POI, but also about the states of each of the UAVs. The approach here is to treat the errors of the onboard navigation system as non-additive noise.

For the development of the improved square root, sigma point information filter (SR-SPIF), the cooperative geolocation of a point of interest (POI) using multiple uninhabited aerial vehicles (UAVs) with camera sensors is defined as follows. First, define the state to be estimated, $\mathbf{x}_{\text{POI},k} \in \mathbb{R}^{n_x}$, to be the state of the POI, with discrete time dynamics governed by

$$\mathbf{x}_{\text{POI},k+1} = \mathbf{f}(\mathbf{x}_{\text{POI},k}, \mathbf{w}_k) \tag{3.1}$$

where the disturbance, $\mathbf{w}_k \in \mathbb{R}^{n_w}$, is zero mean, white, Gaussian noise with covariance $\mathbf{Q}_k$, and the subscript k denotes time step t_k. Assume there are N UAVs with states, $\mathbf{x}^j_{\text{UAV},k+1}$, for $j = 1, \ldots, N$, composed of UAV position, UAV attitude, and camera attitude.

The UAVs are further assumed to have an onboard navigation system and measurements of the camera gimbal angles, which give an estimate of the UAV

state, $\widehat{\mathbf{x}}^j_{\text{UAV},k+1}$. For clarity in the development of the SR-SPIF, a simple model is used here and is given by

$$\mathbf{x}^j_{\text{UAV},k+1} = \overline{\boldsymbol{\psi}}^j_{k+1} + \boldsymbol{\eta}^j_{k+1}, \tag{3.2}$$

where the UAV state estimate error, $\boldsymbol{\eta}^j_{k+1}$, is zero mean, Gaussian, and white with covariance, $\overline{\mathbf{P}}^j_{\text{UAV},k+1}$. This model is known to not be theoretically correct because the statistics are not white, but correlated through the navigation filter. Many times the errors due to autocorrelation are small [39]. Biases may also exist, however, which can have a significant effect on accuracy [67]. A decentralized approach to bias estimation is presented in Chapter 4, as a way of improving geolocation accuracy.

Measurements of the POI are made on each UAV using an onboard vision system which maintains the pixel location of the POI in the camera screen. Mathematically, this vision measurement is modeled as

$$\begin{aligned} \mathbf{z}^j_{k+1} &= \mathbf{g}_{\text{SCR}}(\mathbf{x}_{\text{POI},k+1}, \mathbf{x}^j_{\text{UAV},k+1}) + \mathbf{v}^j_{\text{SCR},k+1} \\ &= \mathbf{g}_{\text{SCR}}(\mathbf{x}_{\text{POI},k+1}, \overline{\mathbf{x}}^j_{\text{UAV},k+1} + \boldsymbol{\eta}^j_{k+1}) + \mathbf{v}^j_{\text{SCR},k+1} \end{aligned} \tag{3.3}$$

where the vision system noise, $\mathbf{v}^j_{\text{SCR},k+1}$, is zero mean, white, Gaussian noise with covariance $\mathbf{P}^j_{\text{SCR},k+1}$. Note that the camera measurement function in Equation 3.3 is a complicated nonlinear function of the POI state and the UAV state; a detailed development is given in Ref. [10] with a summary provided in Appendix C.

Here the output of the navigation system is treated as the mean of a nonzero mean noise, $\mathbf{v}^j_{\text{UAV},k+1} = \overline{\mathbf{x}}^j_{k+1} + \boldsymbol{\eta}^j_{k+1}$, where the change of variables is for clarity as the variable v is used here for noise terms. This leads to a measurement function of the form

$$\mathbf{z}^j_{k+1} = \mathbf{h}^j(\mathbf{x}_{\text{POI},k+1}, \mathbf{v}^j_{k+1}) = \mathbf{g}_{\text{SCR}}(\mathbf{x}_{\text{POI},k+1}, \mathbf{v}^j_{\text{UAV},k+1}) + \mathbf{v}^j_{\text{SCR},k+1} \tag{3.4}$$

where the combined noise vector, $\mathbf{v}_{k+1}^{j} \in \mathbb{R}^{n_v}$ is given by

$$\mathbf{v}_{k+1}^{j} = \begin{bmatrix} \mathbf{v}_{\text{UAV},k+1}^{j} \\ \mathbf{v}_{\text{SCR},k+1}^{j} \end{bmatrix}, \tag{3.5}$$

with mean $\overline{\mathbf{v}}_{k+1}^{j}$, given by

$$\overline{\mathbf{v}}_{k+1}^{j} = \begin{bmatrix} \overline{\mathbf{x}}_{\text{UAV},k+1}^{j} \\ \mathbf{0} \end{bmatrix}. \tag{3.6}$$

Given the new formulation of the measurement, a Square Root, Sigma Point Information Filter (SR-SPIF) can be developed to solve this cooperative geolocaiton problem and it is detailed in the next section.

3.2 SR-SPIF Development

Consider an augmented state vector defined as,

$$\mathbf{x}_k^a = \begin{bmatrix} \mathbf{x}_{\text{POI},k} \\ \mathbf{w}_k \\ \mathbf{v}_k \end{bmatrix} \tag{3.7}$$

where $\mathbf{x}_k^a \in \mathrm{R}^{n_a}$, and $n_a = n_x + n_w + n_v$. The initial augmented state estimate and square root covariances are assumed to be,

$$\hat{\mathbf{x}}_0^a = E[\mathbf{x}_0^a] = \begin{bmatrix} \hat{\mathbf{x}}_{\text{POI},0} \\ \bar{\mathbf{w}}_0 \\ \bar{\mathbf{v}}_0 \end{bmatrix} \tag{3.8}$$

$$\mathbf{S}^a = \begin{bmatrix} \mathbf{S}_{\text{POI},0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{w,0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{S}_{v,0} \end{bmatrix}, \tag{3.9}$$

where $\mathbf{0}$ is an appropriately sized matrix of zeros. Also, $\mathbf{S}_{(\cdot)}$ is an upper right triangular Cholesky factor such that $\mathbf{P}_{(\cdot)} = \mathbf{S}_{(\cdot)}^T \mathbf{S}_{(\cdot)}$. That is $\mathbf{S}_{\text{POI},0}$, $\mathbf{S}_{w,0}$, and $\mathbf{S}_{v,0}$ are the initial square root covariances for the state estimate, process noise, and measurement noise, respectively.

Next, sigma points are defined as,

$$\mathcal{X}_0 = \left[\begin{array}{ccc} \hat{\mathbf{x}}_{\text{POI},0} & \hat{\mathbf{x}}_{\text{POI},0} \cdot \mathbf{1}_{[1,n_x]} + \sigma_f \, \mathbf{S}_{\text{POI},0}^T & \hat{\mathbf{x}}_{\text{POI},0} \cdot \mathbf{1}_{[1,n_x]} - \sigma_f \, \mathbf{S}_{\text{POI},0}^T \end{array} \right] \tag{3.10}$$

$$\mathcal{W}_0 = \left[\begin{array}{ccc} \bar{\mathbf{w}}_0 & \bar{\mathbf{w}}_0 \cdot \mathbf{1}_{[1,n_w]} + \sigma_f \, \mathbf{S}_{w,0}^T & \bar{\mathbf{w}}_0 \cdot \mathbf{1}_{[1,n_w]} - \sigma_f \, \mathbf{S}_{w,0}^T \end{array} \right] \tag{3.11}$$

$$\mathcal{V}_0 = \left[\begin{array}{ccc} \bar{\mathbf{v}}_0 & \bar{\mathbf{v}}_0 \cdot \mathbf{1}_{[1,n_v]} + \sigma_f \, \mathbf{S}_{v,0}^T & \bar{\mathbf{v}}_0 \cdot \mathbf{1}_{[1,n_v]} - \sigma_f \, \mathbf{S}_{v,0}^T \end{array} \right] \tag{3.12}$$

where σ_f is a scaling for the distance of the sigma points from the mean, and $\mathbf{1}_{[1,(\cdot)]}$ is a $1 \times (\cdot)$ row vector of ones. A set of associated weights is then defined, which can be used to find the sample mean/covariance:

$$W_0^m = \frac{\sigma_f^2 - n_a}{\sigma_f^2}, \;\; W_0^c = \frac{\sigma_f^2 - n_a}{\sigma_f^2} + 3 - \frac{\sigma_f^2}{n_a}, \;\; W = \frac{1}{2\sigma_f^2}, \tag{3.13}$$

where m, c denote mean and covariance respectively.

The SR-SPIF algorithm requires three steps: State and Measurement Prediction, Effective Additive Measurement Noise Covariance Computation, and Measurement Fusion.

1. *State and Measurement Prediction*

The state prediction step propagates the state and disturbance sigma points through the nonlinear dynamics (Eqs. 3.14 and 3.15) and evaluates the sample mean (Eq. 3.16).

$$\mathcal{X}_{j,k+1}^- = f(\mathcal{X}_{j,k}, \mathcal{W}_{0,k}), \qquad j = 0, \cdots, 2n_x \tag{3.14}$$

$$\mathcal{X}_{j+2n_x,k+1}^- = f(\mathcal{X}_{0,k}, \mathcal{W}_{j,k}), \qquad j = 1, \cdots, 2n_w. \tag{3.15}$$

The predicted mean is then computed as

$$\overline{\mathbf{x}}_{\text{POI},k+1} = (W_0^m + 2n_v W)\mathcal{X}^-_{0,k+1} + \sum_{j=1}^{d} W\mathcal{X}^-_{j,k+1}, \tag{3.16}$$

where $d = 2(n_x + n_w)$. The centered sigma points are computed by subtracting the mean as

$$\mathcal{X}^{c0}_{k+1} = \mathcal{X}^-_{0,k+1} - \overline{\mathbf{x}}_{k+1} \tag{3.17}$$

$$\mathcal{X}^{c}_{k+1} = \left[\begin{array}{ccc} \mathcal{X}^-_{1,k+1} - \overline{\mathbf{x}}_{k+1} & \cdots & \mathcal{X}^-_{d,k+1} - \overline{\mathbf{x}}_{k+1} \end{array} \right] \tag{3.18}$$

The square root covariance for the POI states is formed through an orthogonalization and Cholesky update for the 0th sigma point, as its weight W_c^0 could be negative.

$$\overline{\mathbf{S}}_{\text{POI},k+1} = up\left\{\sqrt{W}\cdot orth\left\{(\mathcal{X}^c_{k+1})^T\right\}, (\mathcal{X}^{c0}_{k+1})\sqrt{|W^c_0|}, sgn(W^0_c)\right\} \tag{3.19}$$

For eventual fusion in the information domain, the square root information matrix, $\overline{\mathbf{R}}_{\text{POI},k+1}$, is computed. Since $\overline{\mathbf{S}}_{\text{POI},k+1}$ is upper triangular, the square root information matrix can be calculated with modest computation as

$$\overline{\mathbf{R}}_{\text{POI},k+1} = orth\left\{\left(\overline{\mathbf{S}}_{\text{POI},k+1}\right)^{-T}\right\} \tag{3.20}$$

where $\overline{\mathbf{R}}_{\text{POI},k+1}$ is the upper triangular square root factor of the information matrix (inverse covariance), $\overline{\mathbf{Y}}_{\text{POI},k+1} = \overline{\mathbf{R}}^T_{\text{POI},k+1}\overline{\mathbf{R}}_{\text{POI},k+1} = \overline{\mathbf{P}}^{-1}_{\text{POI},k+1}$.

For measurement prediction, the predicted state sigma points are propagated through the nonlinear measurement equation as

$$\begin{aligned} \mathcal{Z}_{j,k+1} &= h(\mathcal{X}^-_{j,k+1}, \mathcal{V}_{0,k+1}), \qquad j = 0, \cdots, d \\ \mathcal{Z}_{j+d,k+1} &= h(\mathcal{X}^-_{0,k+1}, \mathcal{V}_{j,k+1}), \qquad j = 1, \cdots, 2n_v \end{aligned} \tag{3.21}$$

The measurement mean is computed by

$$\overline{\mathbf{z}}_{k+1} = W_0^m \mathcal{Z}_{0,k+1} + \sum_{j=1}^{2n_a} W_j^m \mathcal{Z}_{j,k+1}. \tag{3.22}$$

The centered measurement sigma points are computed as

$$\begin{aligned} \mathcal{Z}_{k+1}^{c0} &= \mathcal{Z}_{0,k+1} - \overline{\mathbf{z}}_{k+1} \\ \mathcal{Z}_{k+1}^{c} &= \left[\begin{array}{ccc} \mathcal{Z}_{1,k+1} - \overline{\mathbf{z}}_{k+1} \ , & \cdots & , \mathcal{Z}_{2n_a,k+1} - \overline{\mathbf{z}}_{k+1} \end{array} \right] \end{aligned}$$

Next, the stochastic linearization of the nonlinear measurement is found using the state-measurement cross covariance,

$$P_{xz} = W \mathcal{X}_{k+1}^{c} \left(\mathcal{Z}_{k+1}^{c}\right)^T + W_0^c \mathcal{X}_{k+1}^{c0} \left(\mathcal{Z}_{k+1}^{c0}\right)^T \tag{3.23}$$

$$C_{\text{POI},k+1} = P_{xz}^T \overline{\mathbf{R}}_{\text{POI},k+1}^T \overline{\mathbf{R}}_{\text{POI},k+1} \tag{3.24}$$

The square root information state and matrix updates are then calculated,

$$\mathbf{i}_{k+1} = (\widetilde{S}_{v,k+1})^{-T} \left(\mathbf{z}_{k+1} - \overline{\mathbf{z}}_{k+1} + C_{\text{POI},k+1} \overline{\mathbf{x}}_{\text{POI},k+1}\right) \tag{3.25}$$

$$\mathbf{I}_{k+1} = (\widetilde{S}_{v,k+1})^{-T} C_{\text{POI},k+1}, \tag{3.26}$$

where $\widetilde{S}_{v,k+1}$ is the effective additive square root measurement error covariance which is developed below. The square root information state and gain are subsequently used for two purposes: 1) on-board the current UAV in its SR-SPIF geolocation estimator, and 2) communicating to the other UAVs for fusion into their SR-SPIF geolocation estimators.

2. *Effective Additive Measurement Noise Covariance Computation*

A set of associated weights is defined, which can be used to find the sample mean/covariance:

$$\widetilde{W}_0^m = \frac{\sigma_f^2 - 2n_v}{\sigma_f^2}, \ \ \widetilde{W}_0^c = \frac{\sigma_f^2 - 2n_v}{\sigma_f^2} + 3 - \frac{\sigma_f^2}{2n_v}, \ \ \widetilde{W} = \frac{1}{2\sigma_f^2}, \tag{3.27}$$

The mean of the measurement sigma points coming from the sensor noise is taken by

$$\widetilde{\mathbf{z}}_{k+1} = \widetilde{W}_0^m \mathcal{Z}_{0,k+1} + \sum_{j=1}^{2n_v} \widetilde{W} \mathcal{Z}_{d+j,k+1}, \tag{3.28}$$

and then subtracted from the measurement sigma points to get new centered measurement sigma points

$$\begin{aligned} \widetilde{\mathcal{Z}}_{k+1}^{c0} &= \mathcal{Z}_{0,k+1} - \widetilde{\mathbf{z}}_{k+1} \\ \widetilde{\mathcal{Z}}_{k+1}^{c} &= \left[\begin{array}{ccc} \mathcal{Z}_{d+1,k+1} - \widetilde{\mathbf{z}}_{k+1} \ , & \cdots & , \mathcal{Z}_{d+2n_v,k+1} - \widetilde{\mathbf{z}}_{k+1} \end{array} \right] \end{aligned} \tag{3.29}$$

The square root covariance of the equivalent additive sensor noise can by found by

$$\widetilde{S}_{v,k+1} = up\left\{ \sqrt{\widetilde{W}} \cdot orth\left\{ (\widetilde{\mathcal{Z}}_{k+1}^{c})^T \right\}, (\widetilde{\mathcal{Z}}_{k+1}^{c0}) \sqrt{|\widetilde{W}_0^c|}, sgn(\widetilde{W}_0^c) \right\} \tag{3.30}$$

Measurement Fusion

The square root information state and matrix updates are used for measurement fusion. First, consider the case of only local measurement fusion. An orthogonalization is performed to find the updated square root information matrix, $\mathbf{R}_{\mathrm{POI},k+1}$, for the POI states as

$$T_{k+1} \mathbf{R}_{\mathrm{POI},k+1} = orth\left\{ \left[\begin{array}{c} \overline{\mathbf{R}}_{\mathrm{POI},k+1} \\ \mathbf{I}_{k+1} \end{array} \right] \right\} \tag{3.31}$$

Finally, the POI state estimate ($\hat{\mathbf{x}}_{\mathrm{POI},k+1}$) and square root covariance ($S_{\mathrm{POI},k+1}$) are calculated, to be used both for the next sigma points and for the geolocation tracking outputs.

$$\hat{\mathbf{x}}_{\mathrm{POI},k+1} = (\mathbf{R}_{\mathrm{POI},k+1})^{-1} T_{k+1}^T \left[\begin{array}{c} \overline{\mathbf{R}}_{k+1} \overline{\mathbf{x}}_{\mathrm{POI},k+1} \\ \mathbf{i}_{k+1} \end{array} \right] \tag{3.32}$$

$$\mathbf{S}_{\mathrm{POI},k+1} = orth\left\{ \mathbf{R}_{\mathrm{POI},k+1}^{-T} \right\} \tag{3.33}$$

The above calculations are computationally moderate because the inversions are of upper right triangular matrix factors. The sigma points are then re-calculated for the prediction step (Eqs. 3.14 and 3.15), and the process repeats.

In the case of fusion using multiple cooperating UAVs, the UAVs transmit the information state and matrix updates. Then, fusion of the information from multiple UAVs is now a straightforward integration of the information state and matrix updates into the update equations (Eqs. 3.31-3.33). Equations 3.31-3.33 are rewritten in a form for cooperative fusion over N UAVs as

$$T_{k+1}\mathbf{R}_{\text{POI},k+1} = orth\left\{\begin{bmatrix} \overline{\mathbf{R}}_{\text{POI},k+1} \\ \mathbf{I}^1_{k+1} \\ \mathbf{I}^2_{k+1} \\ \vdots \\ \mathbf{I}^N_{k+1} \end{bmatrix}\right\} \tag{3.34}$$

$$\hat{\mathbf{x}}_{\text{POI},k+1} = (\mathbf{R}_{\text{POI},k+1})^{-1}T^T_{k+1}\begin{bmatrix} \overline{\mathbf{R}}_{\text{POI},k+1}\overline{\mathbf{x}}_{\text{POI},k+1} \\ \mathbf{i}^1_{k+1} \\ \mathbf{i}^2_{k+1} \\ \vdots \\ \mathbf{i}^N_{k+1} \end{bmatrix} \tag{3.35}$$

$$\mathbf{S}_{\text{POI},k+1} = orth\left\{\mathbf{R}^{-T}_{\text{POI},k+1}\right\} \tag{3.36}$$

Note that only the information matrix and state updates, $\mathbf{I}^j$ and $\mathbf{i}^j$, need to be exchanged between the UAVs. This is simpler than the form presented in [11] where the UAV navigation system statistics must also be known by all cooperating UAVs. Further, the information state and matrix updates are smaller in this formulation than in the original formulation and thus require less computation for fusion and less communication.

3.3 Filter Comparison

The SR-SPIF presented here can be seen as a refinement or improvement of the SR-SPIF presented in Chapter 2. The primary difference is the perspective with respect to the uncertain UAV states. In Chapter 2, the UAV states were considered as uncertain parameters and the formulation was a combined state and parameter estimation problem. However, for computational reasons, the update of the parameter states was not included. This lack of an update step is equivalent to considering the UAV state estimate errors as white.

The SR-SPIF presented here takes the perspective of considering the output of the onboard navigation system to be the mean and covariance of a white noise sequence. By considering the UAV states as noise, instead of parameters to be estimated, both communication and computation can be saved. Further, as will be shown here, the two approaches are statistically equivalent.

3.3.1 Presentation as Extended Information Filters

For clarity of comparison, both the new and original SR-SPIFs will be compared as extended information filters (EIFs). Since the prediction steps are exactly the same, only the update steps will be compared. Also, since only a single update step is compared the time step is omitted. Note that if equivalence is shown for one time step, then due to the recursive nature of the filter, equivalence is shown for all time steps.

Original Square Root Sigma Point Filter

The update portion of the original SR-SPIF can be detailed by defining the information state and matrix updates, $\mathbf{i}_{orig}$, and $\mathbf{I}_{orig}$, in non-square root form, along with the fusion step. First consider the creation of the information state and matrix updates, which are given by

$$\mathbf{i}_{\text{orig}} = \begin{bmatrix} \mathbf{C}_{\text{POI}}^T \mathbf{P}_{\text{SCR}}^{-1} \left(\mathbf{z} - \overline{\mathbf{z}} + \mathbf{C}_{\text{POI}} \overline{\mathbf{x}}_{\text{POI}} + \mathbf{C}_{\text{UAV}} \overline{\mathbf{x}}_{\text{UAV}}\right) \\ \mathbf{C}_{\text{UAV}}^T \mathbf{P}_{\text{SCR}}^{-1} \left(\mathbf{z} - \overline{\mathbf{z}} + \mathbf{C}_{\text{POI}} \overline{\mathbf{x}}_{\text{POI}} + \mathbf{C}_{\text{UAV}} \overline{\mathbf{x}}_{\text{UAV}}\right) \end{bmatrix} \tag{3.37}$$

$$\mathbf{I}_{\text{orig}} = \begin{bmatrix} \mathbf{C}_{\text{POI}} & \mathbf{C}_{\text{UAV}} \end{bmatrix}^T \mathbf{P}_{\text{SCR}}^{-1} \begin{bmatrix} \mathbf{C}_{\text{POI}} & \mathbf{C}_{\text{UAV}} \end{bmatrix}, \tag{3.38}$$

where $\mathbf{C}_{\text{UAV}}$ is the linearization of the measurement equation with respect to the UAV state and all other terms were defined in Section 3.2. The fusion and conversion to state space steps of the original SR-SPIF can be written as

$$\begin{bmatrix} \mathbf{P}_{\text{POI}} & \mathbf{P}_{\text{POI,UAV}} \\ \mathbf{P}_{\text{UAV,POI}} & \mathbf{P}_{\text{UAV}} \end{bmatrix}^{-1} = \begin{bmatrix} \overline{\mathbf{P}}_{\text{POI}}^{-1} & \mathbf{0} \\ \mathbf{0} & \overline{\mathbf{P}}_{\text{UAV}}^{-1} \end{bmatrix} + \mathbf{I}_{\text{orig}} \tag{3.39}$$

The block matrix inversion lemma can be used to determine the $\mathbf{P}_{\text{POI}}$ term in the above equation, which is after simplification

$$\mathbf{P}_{\text{POI}} = \left(\overline{\mathbf{P}}_{\text{POI}}^{-1} + \mathbf{C}_{\text{POI}}^T \left(\mathbf{P}_{\text{SCR}}^{-1} - \mathbf{P}_{\text{SCR}}^{-1} \mathbf{C}_{\text{UAV}} (\overline{\mathbf{P}}_{\text{UAV}}^{-1} + \mathbf{C}_{\text{UAV}}^T \mathbf{P}_{\text{SCR}}^{-1} \mathbf{C}_{\text{UAV}})^{-1} \mathbf{C}_{\text{UAV}}^T \mathbf{P}_{\text{SCR}}^{-1}\right) \mathbf{C}_{\text{POI}}\right)^{-1} \tag{3.40}$$

The state portion can be written as

$$\begin{bmatrix} \widehat{\mathbf{x}}_{\text{POI}} \\ \widehat{\mathbf{x}}_{\text{UAV}} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{\text{POI}} & \mathbf{P}_{\text{POI,UAV}} \\ \mathbf{P}_{\text{UAV,POI}} & \mathbf{P}_{\text{UAV}} \end{bmatrix} \left(\begin{bmatrix} \overline{\mathbf{P}}_{\text{POI}}^{-1} \overline{\mathbf{x}}_{\text{POI}} \\ \overline{\mathbf{P}}_{\text{UAV}}^{-1} \overline{\mathbf{x}}_{\text{UAV}} \end{bmatrix} + \mathbf{i}_{\text{orig}} \right). \tag{3.41}$$

Note that in the formulation in Chapter 2 the $\overline{\mathbf{P}}_{\text{UAV}}^{-1} \overline{\mathbf{x}}_{UAV}$ term in the above equation was actually included with the information state update, $\mathbf{i}_{\text{orig}}$, but has been moved here for clarity of the presentation. The updated state estimate can then be written

as

$$\widehat{\mathbf{x}}_{\text{POI}} = \mathbf{P}_{\text{POI}}\overline{\mathbf{P}}_{\text{POI}}^{-1}\overline{\mathbf{x}}_{\text{POI}} + \left(\mathbf{P}_{\text{POI}}\mathbf{C}_{\text{POI}}^{T}\mathbf{P}_{\text{SCR}}^{-1} + \mathbf{P}_{\text{POI,UAV}}\mathbf{C}_{\text{UAV}}^{T}\mathbf{P}_{\text{SCR}}^{-1}\right)\left(\mathbf{z} - \overline{\mathbf{z}} + \mathbf{C}_{\text{POI}}\overline{\mathbf{x}}_{\text{POI}}\right)$$
$$+\mathbf{P}_{\text{POI,UAV}}\mathbf{P}_{\text{UAV}}^{-1}\overline{\mathbf{x}}_{\text{UAV}} + \left(\mathbf{P}_{\text{POI}}\mathbf{C}_{\text{POI}}^{T}\mathbf{P}_{\text{SCR}}^{-1} + \mathbf{P}_{\text{POI,UAV}}\mathbf{C}_{\text{UAV}}^{T}\mathbf{P}_{\text{SCR}}^{-1}\right)\mathbf{C}_{\text{UAV}}\overline{\mathbf{x}}_{\text{UAV}}, \quad (3.42)$$

where, by block matrix inversion of Eqn. 3.39,

$$\mathbf{P}_{\text{POI,UAV}} = -\mathbf{P}_{\text{POI}}\mathbf{C}_{\text{POI}}^{T}\mathbf{P}_{\text{SCR}}^{-1}\mathbf{C}_{\text{UAV}}\left(\overline{\mathbf{P}}_{\text{UAV}}^{-1} + \mathbf{C}_{\text{UAV}}^{T}\mathbf{P}_{\text{SCR}}^{-1}\mathbf{C}_{\text{UAV}}\right)^{-1} \quad (3.43)$$

New Square Root Sigma Point Filter

Now, the update portion of the new SR-SPIF is written in extended information form. First, consider the form of the information state and matrix updates, which are given by

$$\mathbf{i}_{\text{new}} = \mathbf{C}_{\text{POI}}^{T}\left(\mathbf{P}_{\text{SCR}} + \mathbf{C}_{\text{UAV}}\overline{\mathbf{P}}_{\text{UAV}}\mathbf{C}_{\text{UAV}}^{T}\right)^{-1}\left(\mathbf{z} - \overline{\mathbf{z}} + \mathbf{C}_{\text{POI}}\overline{\mathbf{x}}_{\text{POI}}\right) \quad (3.44)$$

$$\mathbf{I}_{\text{new}} = \mathbf{C}_{\text{POI}}^{T}\left(\mathbf{P}_{\text{SCR}} + \mathbf{C}_{\text{UAV}}\overline{\mathbf{P}}_{\text{UAV}}\mathbf{C}_{\text{UAV}}^{T}\right)^{-1}\mathbf{C}_{\text{POI}} \quad (3.45)$$

Since only the state of the POI is considered the updated estimate and covariance can be written simply as

$$\mathbf{P}_{\text{POI}}^{-1} = \overline{\mathbf{P}}_{\text{POI}}^{-1} + \mathbf{I}_{\text{new}} \quad (3.46)$$

$$\widehat{\mathbf{x}}_{\text{POI}} = \mathbf{P}_{\text{POI}}\left(\overline{\mathbf{P}}_{\text{POI}}^{-1}\overline{\mathbf{x}}_{\text{POI}} + \mathbf{i}_{\text{new}}\right) \quad (3.47)$$

and substituting for the information state and matrix updates gives

$$\mathbf{P}_{\text{POI}} = \left(\overline{\mathbf{P}}_{\text{POI}}^{-1} + \mathbf{C}_{\text{POI}}^{T}\left(\mathbf{P}_{\text{SCR}} + \mathbf{C}_{\text{UAV}}\overline{\mathbf{P}}_{\text{UAV}}\mathbf{C}_{\text{UAV}}^{T}\right)^{-1}\mathbf{C}_{\text{POI}}\right)^{-1} \quad (3.48)$$

$$\widehat{\mathbf{x}}_{\text{POI}} = \mathbf{P}_{\text{POI}}\overline{\mathbf{P}}_{\text{POI}}^{-1}\overline{\mathbf{x}}_{\text{POI}} + \mathbf{P}_{\text{POI}}\mathbf{C}_{\text{POI}}^{T}\left(\mathbf{P}_{\text{SCR}} + \mathbf{C}_{\text{UAV}}\overline{\mathbf{P}}_{\text{UAV}}\mathbf{C}_{\text{UAV}}^{T}\right)^{-1}\left(\mathbf{z} - \overline{\mathbf{z}} + \mathbf{C}_{\text{POI}}\overline{\mathbf{x}}_{\text{POI}}\right) \quad (3.49)$$

3.3.2 Comparison of the SRSPIF Versions

Equivalence of the two filters can now be shown by comparing both the covariance and state estimate portions of each filter. First, consider the covariance part. For equivalence, it must be shown that the expression for $\mathbf{P}_{\text{POI}}$ in the original formulation, Eqn. 3.40, is the same as the expression in the new formulation, Eqn. 3.48. This can be done by using the matrix inversion lemma, Ref. [5], on the $\mathbf{P}_{\text{SCR}}^{-1} - \mathbf{P}_{\text{SCR}}^{-1}\mathbf{C}_{\text{UAV}}(\overline{\mathbf{P}}_{\text{UAV}}^{-1} + \mathbf{C}_{\text{UAV}}^{T}\mathbf{P}_{\text{SCR}}^{-1}\mathbf{C}_{\text{UAV}})^{-1}\mathbf{C}_{\text{UAV}}^{T}\mathbf{P}_{\text{SCR}}^{-1}$ term giving

$$\mathbf{P}_{\text{SCR}}^{-1}-\mathbf{P}_{\text{SCR}}^{-1}\mathbf{C}_{\text{UAV}}(\overline{\mathbf{P}}_{\text{UAV}}^{-1}+\mathbf{C}_{\text{UAV}}^{T}\mathbf{P}_{\text{SCR}}^{-1}\mathbf{C}_{\text{UAV}})^{-1}\mathbf{C}_{\text{UAV}}^{T}\mathbf{P}_{\text{SCR}}^{-1} = \left(\mathbf{P}_{\text{SCR}} + \mathbf{C}_{\text{UAV}}\overline{\mathbf{P}}_{\text{UAV}}\mathbf{C}_{\text{UAV}}^{T}\right)^{-1}. \tag{3.50}$$

and substituting into the expression for the covariance in the old formulation, Eqn. 3.40, which gives

$$\mathbf{P}_{\text{POI}}=\left(\overline{\mathbf{P}}_{\text{POI}}^{-1}+\mathbf{C}_{\text{POI}}^{T}\mathbf{P}_{\text{SCR}}^{-1}\mathbf{C}_{\text{POI}}-\mathbf{C}_{\text{POI}}^{T}\mathbf{P}_{\text{SCR}}^{-1}\mathbf{C}_{\text{UAV}}(\overline{\mathbf{P}}_{\text{UAV}}^{-1}+\mathbf{C}_{\text{UAV}}^{T}\mathbf{P}_{\text{SCR}}^{-1}\mathbf{C}_{\text{UAV}})^{-1}\mathbf{C}_{\text{UAV}}^{T}\mathbf{P}_{\text{SCR}}^{-1}\mathbf{C}_{\text{POI}}\right)^{-1}, \tag{3.51}$$

which is the exact same expression as given in the original formulation, Eqn. 3.40. The comparison of the state estimate portion is a little more involved. First, consider the expressions for each formulation, which are written below for comparison.

$$\begin{aligned} orig: \quad \widehat{\mathbf{x}}_{\text{POI}} &= \mathbf{P}_{\text{POI}}\overline{\mathbf{P}}_{\text{POI}}^{-1}\overline{\mathbf{x}}_{\text{POI}} + \left(\mathbf{P}_{\text{POI}}\mathbf{C}_{\text{POI}}^{T}\mathbf{P}_{\text{SCR}}^{-1} + \mathbf{P}_{\text{POI,UAV}}\mathbf{C}_{\text{UAV}}^{T}\mathbf{P}_{\text{SCR}}^{-1}\right)\cdot \\ &\quad (\mathbf{z} - \overline{\mathbf{z}} + \mathbf{C}_{\text{POI}}\overline{\mathbf{x}}_{\text{POI}}) + \mathbf{P}_{\text{POI,UAV}}\mathbf{P}_{\text{UAV}}^{-1}\overline{\mathbf{x}}_{\text{UAV}} + \\ &\quad \left(\mathbf{P}_{\text{POI}}\mathbf{C}_{\text{POI}}^{T}\mathbf{P}_{\text{SCR}}^{-1} + \mathbf{P}_{\text{POI,UAV}}\mathbf{C}_{\text{UAV}}^{T}\mathbf{P}_{\text{SCR}}^{-1}\right)\mathbf{C}_{\text{UAV}}\overline{\mathbf{x}}_{\text{UAV}} \end{aligned} \tag{3.52}$$

$$\begin{aligned} new: \quad \widehat{\mathbf{x}}_{\text{POI}} &= \mathbf{P}_{\text{POI}}\overline{\mathbf{P}}_{\text{POI}}^{-1}\overline{\mathbf{x}}_{\text{POI}} + \mathbf{P}_{\text{POI}}\mathbf{C}_{\text{POI}}^{T}\left(\mathbf{P}_{\text{SCR}} + \mathbf{C}_{\text{UAV}}\overline{\mathbf{P}}_{\text{UAV}}\mathbf{C}_{\text{UAV}}^{T}\right)^{-1}\cdot \\ &\quad (\mathbf{z} - \overline{\mathbf{z}} + \mathbf{C}_{\text{POI}}\overline{\mathbf{x}}_{\text{POI}}) \end{aligned} \tag{3.53}$$

By inspection the equivalence of the two expressions can be shown by demonstrating that the following two equations hold:

$$\mathbf{P}_{\text{POI}}\mathbf{C}_{\text{POI}}^{T}\mathbf{P}_{\text{SCR}}^{-1} + \mathbf{P}_{\text{POI,UAV}}\mathbf{C}_{\text{UAV}}^{T}\mathbf{P}_{\text{SCR}}^{-1} = \mathbf{P}_{\text{POI}}\mathbf{C}_{\text{POI}}^{T}\left(\mathbf{P}_{\text{SCR}} + \mathbf{C}_{\text{UAV}}\overline{\mathbf{P}}_{\text{UAV}}\mathbf{C}_{\text{UAV}}^{T}\right)^{-1} \tag{3.54}$$

$$\mathbf{P}_{\text{POI,UAV}}\overline{\mathbf{P}}_{\text{UAV}}^{-1} + \left(\mathbf{P}_{\text{POI}}\mathbf{C}_{\text{POI}}^{T}\mathbf{P}_{\text{SCR}}^{-1} + \mathbf{P}_{\text{POI,UAV}}\mathbf{C}_{\text{UAV}}^{T}\mathbf{P}_{\text{SCR}}^{-1}\right)\mathbf{C}_{\text{UAV}} = 0 \tag{3.55}$$

First, consider the validity of Eqn. 3.54. Substituting for $\mathbf{P}_{\text{POI,UAV}}$ (given in Eqn. 3.43) on the left hand side of Eqn. 3.54 and also applying the matrix inversion lemma to the right side and rearranging gives the same expression on each side of the equation,

$$\mathbf{P}_{\text{POI}}\mathbf{C}_{\text{POI}}^{T}\left(\mathbf{P}_{\text{SCR}}^{-1} - \mathbf{P}_{\text{SCR}}^{-1}\mathbf{C}_{\text{UAV}}(\mathbf{C}_{\text{UAV}}^{T}\mathbf{P}_{\text{SCR}}^{-1}\mathbf{C}_{\text{UAV}} + \overline{\mathbf{P}}_{\text{UAV}}^{-1})^{-1}\mathbf{C}_{\text{UAV}}^{T}\mathbf{P}_{\text{SCR}}^{-1}\right). \tag{3.56}$$

Now consider the validity of Eqn. 3.55. Rearranging Eqn. 3.55 gives

$$\mathbf{P}_{\text{POI,UAV}}\left(\overline{\mathbf{P}}_{\text{UAV}}^{-1} + \mathbf{C}_{\text{UAV}}^{T}\mathbf{P}_{\text{SCR}}^{-1}\mathbf{C}_{\text{UAV}}\right) + \mathbf{P}_{\text{POI}}\mathbf{C}_{\text{POI}}^{T}\mathbf{P}_{\text{SCR}}^{-1}\mathbf{C}_{\text{UAV}} = 0 \tag{3.57}$$

Substituting for $\mathbf{P}_{\text{POI,UAV}}$ (from Eqn. 3.43) gives

$$\begin{aligned} 0 \;=\; & -\mathbf{P}_{\text{POI}}\mathbf{C}_{\text{POI}}^{T}\mathbf{P}_{\text{SCR}}^{-1}\mathbf{C}_{\text{UAV}}\left(\overline{\mathbf{P}}_{\text{UAV}}^{-1} + \mathbf{C}_{\text{UAV}}^{T}\mathbf{P}_{\text{SCR}}^{-1}\mathbf{C}_{\text{UAV}}\right)^{-1}\left(\overline{\mathbf{P}}_{\text{UAV}}^{-1} + \mathbf{C}_{\text{UAV}}^{T}\mathbf{P}_{\text{SCR}}^{-1}\mathbf{C}_{\text{UAV}}\right) + \\ & \mathbf{P}_{\text{POI}}\mathbf{C}_{\text{POI}}^{T}\mathbf{P}_{\text{SCR}}^{-1}\mathbf{C}_{\text{UAV}}, \end{aligned} \tag{3.58}$$

which by inspection can be seen to hold true as all terms cancel. Therefore, the two filtering approaches are equivalent.

3.4 Conclusions

A new square root, sigma point information filter (SR-SPIF) was developed to solve the cooperative geolocation problem. In this new filter formulation, the uncertain UAV state was treated as a non-additive noise in the measurement equation and the mean was assumed to be given by the onboard navigation system. By treating the uncertain UAV state as noise instead of as a set of parameters to be estimated,

the measurement fusion step is considerably simplified. This simpler fusion step requires less computation and also less communication between the UAVs

The new SR-SPIF is compared to the original SR-SPIF and shown to be statistically equivalent. For clarity, the comparison is made by considering each as an extended information filter (EIF). Although, the numerical properties of an EIF and SR-SPIF are different, the statistical implication of the perspective of each of the two version of the SR-SPIF can be explored effectively with an EIF.

CHAPTER 4

DECENTRALIZED GEOLOCATION AND SENSOR BIAS ESTIMATION

4.1 Abstract

The cooperative geolocation of a point of interest (POI) using multiple UAVs with articulating camera sensors is addressed, where there are non zero mean errors (biases) in the estimate of the UAV state. The proposed approach is to use the on-board navigation solution in the estimator and further to consider biases across all UAVs, and jointly estimate both the biases and the unknown POI location. Furthermore, a decentralized solution is presented which uses marginalization of the biases, thus allowing the UAVs to share only information about the POI and model only their local biases. This decentralized approach saves significant computation and scales well with the number of UAVs. Real flight test data and hardware in the loop simulations are used to demonstrate the improvement in geolocation with bias estimation, as well as the effectiveness of the new decentralized POI and bias estimation algorithm, for both stationary and moving POIs.

4.2 Introduction

Uninhabited aerial vehicles (UAVs) are currently being developed and used for a wide variety of missions such as defense, search-and-rescue [22], and commercial applications [55]. A key technology in UAV systems being explored is tracking of a stationary or moving Point of Interest (POI) using visual cameras for payloads – termed "geolocation". Off the shelf digital cameras, which have been developed by

the electronics industry for the past decade for consumer usage, are now enabling very low cost UAV systems. Several groups have implemented gimballing camera systems on UAVs [25, 26], a few with target tracking results [49, 63]. The authors have developed and experimentally implemented a geolocation tracking algorithm for gimballing vision payloads on UAVs [10] using sigma point filtering concepts; extensions to information filtering and sensor fusion across multiple UAVs have also been developed [11].

The geolocation system for a UAV requires the complex integration of several hardware components (camera, UAV, GPS, attitude sensors) and software components (camera image processing, inner loop and path planning control, and estimation software) to develop accurate estimates of the object being tracked. In Refs. [10] and [11], the authors reduce the computation of geolocation by using the onboard navigation system, which integrates the UAV, GPS, and attitude sensors to give an estimate of the position and attitude of the UAV and also measurements of the camera gimbal angles. Further, there is an implicit assumption that the navigation system output and camera gimbal measurements have consistent statistics with zero mean errors. However, this assumption is not valid, as shown in Ref. [67], because both the output of the navigation system and the camera gimbal measurements have nonzero mean errors (biases) that significantly degrade geolocation performance. Ref. [10] compensates for these biases with an additional algorithm that fuses a conservative uniform density model for the bias.

Biases have also been shown to be a problem for estimation in other works. In one of the first treatments [21], Friedland showed that biases could be estimated efficiently in a linear system by partitioning the state. More recently in Refs. [32] and [54], bias estimation in a radar tracking context with multiple targets was

addressed. Bias estimation was further considered in Refs. [37] and [45], where multiple sensors were used to track multiple targets in a centralized formulation. Bias estimation has also been considered in a least squares estimation context by Dogancay in Ref. [16].

This paper proposes to jointly estimate the sensor biases and the unknown POI state in a decentralized manner, while using the solution from the onboard navigation system to save significant computation. Uniquely, the joint estimation problem is solved for multiple UAVs cooperating in a decentralized fashion such that the UAVs share information on the POI state, and model only their local biases. This decentralized formulation saves computation as well as communication, while giving geolocation accuracy comparable to the centralized case. Further, this decentralized approach fits nicely into the decentralized data fusion paradigm [17, 59] and allows for effective cooperation not only among UAVs with potentially different biases but different sensors altogether. A numerical observability analysis procedure is also developed and applied, which gives a meaningful measure of the degree of observability and also gives insight into the effects of UAV flight path on observability. The new decentralized approach is validated using both experimental flight data and high fidelity hardware in the loop simulations. Although multi-target tracking is not considered in this work, extensions based on Ref. [50] can be considered.

The paper is organized as follows. In Section 4.3, the cooperative geolocation problem is solved with explicit white, zero mean, and Gaussian assumptions on the estimate error of the UAV state. Then, in Section 4.4, a more realistic model including biases in the estimate of the UAV state is proposed and the joint POI and bias estimation problem is formulated and solved both in a centralized and

decentralized fashion. Section 4.5 develops a procedure for observability analysis of both biases and the POI state. Then, in Section 4.6 both experimental and simulated flight data for ScanEagle UAVs are used to evaluate the performance of the new decentralized approach compared to the centralized approach (baseline) for both stationary and moving POIs.

4.3 Cooperative Geolocation

Geolocation is the process of using sensor data to develop statistical estimates of a Point of Interest (POI) on the ground. For the application of a vision sensor on a UAV, the UAV, based on its position and orientation, points the camera (through a gimballing payload mount inside the UAV) at the POI on the ground. While the aircraft is moving (navigation and attitude), and the POI is potentially moving, the camera gimbals must adjust their angles to point at the POI. This application requires the camera to remain directed at the POI such that the POI always remains within the field of view of the camera. The objective of geolocation is then to estimate the position (2D or 3D) of the POI from the aircraft, gimbal, and camera measurements. Complicating this problem are uncertainties in the aircraft position and orientation, gimbal angles, camera specifications and measurements, and disturbances such as turbulence and engine vibrations.

The most accurate estimator tightly couples the UAV navigation (NAV), attitude (ATT), camera gimbal (GIM) and POI states in a single estimator which requires full UAV and gimbal models, and a model for the POI. However, this estimator requires very high computation, memory, and communication in the case of multiple UAVs. Fortunately, most UAVs use a navigation system with estimators

which provide statistics (estimates and covariances) for both the ATT and NAV states. In addition, the GIM states can be directly measured. Therefore, a geolocation estimator can be developed which estimates the POI state only, thus saving computation and memory.

A square root sigma point filter for geolocation, which utilizes the output statistics of the onboard navigation system of the UAVs to reduce computation is described in Ref. [10]. A Square Root, Sigma Point Information Filter (SR-SPIF) is developed in Ref. [11] to facilitate cooperative tracking of POIs utilizing multiple UAVs. In both cases, to develop the POI estimate, there is an implied assumption that the estimates of the NAV and ATT states as well as the measurements of the GIM states have zero mean errors. An Extended Information Filter [42] (EIF) is developed here, which utilizes the navigation system to solve the cooperative geolocation problem and makes explicit the assumptions about the estimates of the UAV state. Note that an EIF is used here and presented in general terms to make the development of the decentralized bias estimation approaches in Section 4.4 clear. However, other filtering techniques could also be used.

For this section, define the state to be estimated, $\mathbf{x}_k$, to be the state of the POI, $\mathbf{x}_{k,\mathrm{POI}}$, with discrete time dynamics governed by

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{w}_k) = \mathbf{f}_{\mathrm{POI}}(\mathbf{x}_{k,\mathrm{POI}}, \mathbf{w}_{k,\mathrm{POI}}) \tag{4.1}$$

where the disturbance, $\mathbf{w}_k = \mathbf{w}_{k,\mathrm{POI}}$, is zero mean, white, Gaussian noise with covariance $\mathbf{Q}_k = \mathbf{Q}_{k,\mathrm{POI}}$, and the subscript k denotes time step t_k. Note that a general state to be estimated, $\mathbf{x}_k$, is used so that the bias estimation approaches in Section 4.4 can be developed easily as extensions of the EIF presented here. Assume there are N UAVs with states, $\boldsymbol{\psi}^j_{k+1}$, for $j = 1, \ldots, N$, composed of UAV position, $\boldsymbol{\psi}^j_{\mathrm{k+1,NAV}}$, UAV attitude, $\boldsymbol{\psi}^j_{\mathrm{k+1,ATT}}$, and camera attitude, $\boldsymbol{\psi}^j_{\mathrm{k+1,GIM}}$ written

in vector form as

$$\boldsymbol{\psi}_{k+1}^{j} = \begin{bmatrix} \boldsymbol{\psi}_{\mathrm{k+1,NAV}}^{j} \\ \boldsymbol{\psi}_{\mathrm{k+1,ATT}}^{j} \\ \boldsymbol{\psi}_{\mathrm{k+1,GIM}}^{j} \end{bmatrix} \tag{4.2}$$

The UAVs are further assumed to have an onboard navigation system and measurements of the camera gimbal angles, which give an estimate of the UAV state, $\widehat{\boldsymbol{\psi}}_{k+1}^{j}$. For clarity in the development of the EIF, a simple model is used here and is given by

$$\boldsymbol{\psi}_{k+1}^{j} = \widehat{\boldsymbol{\psi}}_{k+1}^{j} + \boldsymbol{\eta}_{k+1}^{j}, \tag{4.3}$$

where the UAV state estimate error, $\boldsymbol{\eta}_{k+1}^{j}$, is zero mean, Gaussian, and white with covariance, ${}^{\eta}\mathbf{R}_{k+1}^{j}$. This model is known to not be theoretically correct because the statistics are not white, but correlated through the navigation filter. Many times the errors due to autocorrelation are small [39]. Biases may also exist, however, which can have a significant effect on accuracy [67]. The incorporation of biases in the UAV state estimate, the focus of this paper, is presented in Section 4.4 as an extension of the EIF algorithm presented here.

Measurements of the POI are made on each UAV using

$$\mathbf{z}_{k+1}^{j} = \mathbf{h}^{j}(\mathbf{x}_{k+1}, \boldsymbol{\eta}_{k+1}^{j}, \mathbf{v}_{k+1}^{j}) = \mathbf{h}_{\mathrm{SCR}}(\mathbf{x}_{k+1,\mathrm{POI}}, \widehat{\boldsymbol{\psi}}_{k+1}^{j} + \boldsymbol{\eta}_{k+1}^{j}, \mathbf{v}_{k+1,\mathrm{SCR}}^{j}) \tag{4.4}$$

where the sensor noise, $\mathbf{v}_{k+1}^{j}$, is zero mean, white, Gaussian noise with covariance ${}^{v}\mathbf{R}_{k+1}^{j}$. The process noise, sensor noises, and navigation system noises, $\mathbf{w}_k$, $\mathbf{v}_{k+1}^{j}$, and $\boldsymbol{\eta}_{k+1}^{j}$, respectively, are assumed to be uncorrelated with each other. Note that the measurement function in Equation 4.4 is a complicated nonlinear function of the POI state and the UAV state; a detailed development is given in Ref. [10] with a summary provided in Appendix C.

The cooperative geolocation problem can now be solved with an EIF as follows.

The information matrix, $\mathbf{Y}_k$, and information state, $\mathbf{y}_k$, are defined based on the state estimate error covariance, $\mathbf{P}_k$, and state estimate, $\widehat{\mathbf{x}}_k$, as

$$\mathbf{Y}_k = \mathbf{P}_k^{-1} \tag{4.5}$$

$$\mathbf{y}_k = \mathbf{Y}_k \cdot \widehat{\mathbf{x}}_k \tag{4.6}$$

The EIF algorithm is written for N UAVs, as computed on a local UAV l, as a recursion of the following five steps:

1. Time Propagation

$$\mathbf{Y}_{k+1}^{-} = (\mathbf{F}_k \mathbf{Y}_k^{-1} \mathbf{F}_k^T + \mathbf{\Gamma}_k \mathbf{Q}_k \mathbf{\Gamma}_k^T)^{-1} \tag{4.7}$$

$$\mathbf{y}_{k+1}^{-} = \mathbf{Y}_{k+1}^{-} \cdot \mathbf{f}[(\mathbf{Y}_k)^{-1}\mathbf{y}_k, \mathbf{0}], \tag{4.8}$$

where

$$\mathbf{F}_k = \nabla_{\mathbf{x}} \mathbf{f}[\mathbf{x}, \mathbf{w}]|_{\mathbf{x}=(\mathbf{Y}_k)^{-1}\mathbf{y}_k, \mathbf{w}=\mathbf{0}} \tag{4.9}$$

$$\mathbf{\Gamma}_k = \nabla_{\mathbf{w}} \mathbf{f}[\mathbf{x}, \mathbf{w}]|_{\mathbf{x}=(\mathbf{Y}_k)^{-1}\mathbf{y}_k, \mathbf{w}=\mathbf{0}} \tag{4.10}$$

2. Compute Local Information State and Matrix Updates

$$\begin{aligned} \mathbf{i}_{k+1}^l = {} & ({}^{\mathbf{x}}\mathbf{H}_{k+1}^l)^T (\mathbf{R}_{k+1}^l)^{-1} \cdot \\ & (\mathbf{z}_{k+1}^l - \mathbf{h}^l[(\mathbf{Y}_{k+1}^-)^{-1}\mathbf{y}_{k+1}^-, \mathbf{0}, \mathbf{0}] + {}^{\mathbf{x}}\mathbf{H}_{k+1}^l (\mathbf{Y}_{k+1}^-)^{-1}\mathbf{y}_{k+1}^-) \end{aligned} \tag{4.11}$$

$$\mathbf{I}_{k+1}^l = \left({}^{\mathbf{x}}\mathbf{H}_{k+1}^l\right)^T (\mathbf{R}_{k+1}^l)^{-1} ({}^{\mathbf{x}}\mathbf{H}_{k+1}^l), \tag{4.12}$$

where

$$\mathbf{R}_{k+1}^l = {}^{\mathbf{v}}\mathbf{H}_{k+1}^l {}^{\mathbf{v}}\mathbf{R}_{k+1}^l ({}^{\mathbf{v}}\mathbf{H}_{k+1}^l)^T + {}^{\boldsymbol{\eta}}\mathbf{H}_{k+1}^l {}^{\boldsymbol{\eta}}\mathbf{R}_{k+1}^l ({}^{\boldsymbol{\eta}}\mathbf{H}_{k+1}^l)^T \tag{4.13}$$

and

$$
\begin{aligned}
{}^{\mathbf{x}}\mathbf{H}^{l}_{k+1} &= \nabla_{\mathbf{x}}\mathbf{h}^{l}[\mathbf{x},\boldsymbol{\eta},\mathbf{v}]\big|_{\mathbf{x}=(\mathbf{Y}^{-}_{k+1})^{-1}\mathbf{y}^{-}_{k+1},\boldsymbol{\eta}=\mathbf{0},\mathbf{v}=\mathbf{0}} && (4.14)\\
{}^{\mathbf{v}}\mathbf{H}^{l}_{k+1} &= \nabla_{\mathbf{v}}\mathbf{h}^{l}[\mathbf{x},\boldsymbol{\eta},\mathbf{v}]\big|_{\mathbf{x}=(\mathbf{Y}^{-}_{k+1})^{-1}\mathbf{y}^{-}_{k+1},\boldsymbol{\eta}=\mathbf{0},\mathbf{v}=\mathbf{0}} && (4.15)\\
{}^{\boldsymbol{\eta}}\mathbf{H}^{l}_{k+1} &= \nabla_{\boldsymbol{\eta}}\mathbf{h}^{l}[\mathbf{x},\boldsymbol{\eta},\mathbf{v}]\big|_{\mathbf{x}=(\mathbf{Y}^{-}_{k+1})^{-1}\mathbf{y}^{-}_{k+1},\boldsymbol{\eta}=\mathbf{0},\mathbf{v}=\mathbf{0}} && (4.16)
\end{aligned}
$$

3. Transmit Local Information Updates, $\mathbf{i}^{l}_{k+1}$ and $\mathbf{I}^{l}_{k+1}$, to all UAVs

4. Receive Information Updates, $\mathbf{i}^{j}_{k+1}$ and $\mathbf{I}^{j}_{k+1}$, from all UAVs, $j = 1,\ldots,N, j \neq l$

5. Fuse Local and Received Information Updates

$$
\begin{aligned}
\mathbf{Y}_{k+1} &= \mathbf{Y}^{-}_{k+1} + \mathbf{I}^{l}_{k+1} + \sum_{j=1,j\neq l}^{N} \mathbf{I}^{j}_{k+1} && (4.17)\\
\mathbf{y}_{k+1} &= \mathbf{y}^{-}_{k+1} + \mathbf{i}^{l}_{k+1} + \sum_{j=1,j\neq l}^{N} \mathbf{i}^{j}_{k+1} && (4.18)
\end{aligned}
$$

The assumptions on the errors of the navigation system output (white, zero mean, Gaussian) are necessary for the computation of the information state and matrix updates (Step 2) and allow the uncertainty in both the vision system (${}^{\mathbf{v}}\mathbf{R}^{l}_{k+1}$) and the navigation system (${}^{\boldsymbol{\eta}}\mathbf{R}^{l}_{k+1}$) to be combined in Equation 4.13 to compute the effective measurement noise covariance, $\mathbf{R}^{l}_{k+1}$. Note that the superscript l is not included for the information matrix and state since they are the same on all UAVs. If communication losses are present, causing differences among the UAVs, then the above algorithm can be extended using the predicted information approach developed in Ref. [65].

4.4 Geolocation with Bias Estimation

The assumption in Section 4.3 that the errors in the estimate of the UAV state are zero mean, white, and Gaussian is not accurate in the practical case, for two reasons: correlated outputs of the navigation filter and biases in the outputs. Consider Figure 4.1, which shows a series of Sensed Points of Interest (SPOIs), which is defined as the line of sight intersection of the camera with the ground as computed based on the estimates of the UAV state (NAV, ATT, and GIM). A total of 2000 SPOIs are plotted for two orbits around a stationary POI from a flight test of the ScanEagle UAV on March 18, 2006 [67]. Figure 4.1 shows that the SPOI moves in a roughly circular pattern around the true POI location. The period of this oscillation corresponds directly to the UAVs orbit about the POI and is due to nonzero mean errors (biases) in the UAV state estimate.

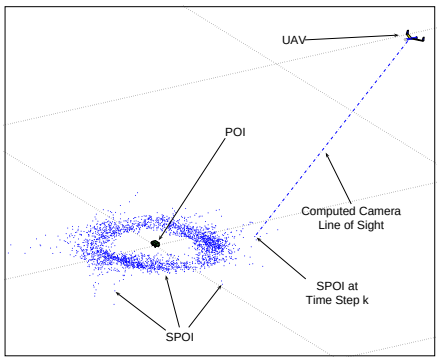


Figure 4.1: Single UAV tracking a stationary POI. The blue dots are computed camera line of sight intersections with the ground from two orbits of tracking.

Sensor biases were shown in Refs. [10] and [67] to be a significant source of error for geolocation and were compensated for by augmenting the output of the estimator with additional uncertainty, based on empirical data. However, this did nothing to improve the estimate itself, but simply improved estimator consistency. The approach taken here is to explicitly model the sensor biases,

and jointly estimate both the sensor biases and the unknown POI location in a decentralized manner. The biases, $\mathbf{b}_k^j$, are now modeled explicitly as part of the UAV navigation system output and camera gimbal measurement as

$$\boldsymbol{\psi}_k^j = \widehat{\boldsymbol{\psi}}_k^j + \mathbf{b}_k^j + \boldsymbol{\eta}_k^j, \tag{4.19}$$

where the model of the bias state, $\mathbf{b}_k^j$, used here is

$$\mathbf{b}_{k+1}^j = \mathbf{b}_k^j + \mathbf{w}_{k,\mathbf{b}^j}. \tag{4.20}$$

Autocorrelation of the UAV state estimate error could be taken into account in the same way as the biases [7]; by adding autocorrelation states, $\boldsymbol{\mu}_k^{j,m}$, in Eqn. 4.19 as

$$\boldsymbol{\psi}_k^j = \widehat{\boldsymbol{\psi}}_k^j + \mathbf{b}_k^j + \boldsymbol{\eta}_k^j + \boldsymbol{\mu}_k^{j,1} + \ldots + \boldsymbol{\mu}_k^{j,n_{\mu^j}}, \tag{4.21}$$

where each of the autocorrelation terms correspond to a different frequency of autocorrelation. The autocorrelation terms can be modeled as

$$\boldsymbol{\mu}_{k+1}^{j,m} = a^{j,m}\boldsymbol{\mu}_k^{j,m} + \mathbf{w}_{k,\boldsymbol{\mu}_{j,m}}, \tag{4.22}$$

where the parameter, $a^{j,m}$, is chosen to capture the appropriate autocorrelation frequency. Although autocorrelations are not considered in this work, the approach presented here is suitable for autocorrelation errors in addition to bias errors.

A key point of the approach here is that the navigation system and camera gimbal measurements are used directly, while only the biases, $\mathbf{b}^j$, are estimated, recursively, with the POI state, $\mathbf{x}_{k,\mathrm{POI}}$. This utilizes the navigation system to save significant computation, while effectively improving the estimate of the UAV state and thus improving geolocation.

4.4.1 Centralized Approach

With the goal of estimating the POI state, $\mathbf{x}_{\text{POI}}$, and the biases on each UAV, $\mathbf{b}^j$, the obvious approach is to create a single estimator; this is termed the centralized solution. In the centralized solution, the state to be estimated, $\mathbf{x}_k$, is modified to include the state of the POI, $\mathbf{x}_{k,\text{POI}}$, and the biases of each of the UAVs, $\mathbf{b}_k^j$, for $j = 1, \ldots, N$. The full state estimate and error covariance is now written as

$$\widehat{\mathbf{x}}_k = \begin{bmatrix} \widehat{\mathbf{x}}_{k,\text{POI}} \\ \widehat{\mathbf{b}}_k^1 \\ \vdots \\ \widehat{\mathbf{b}}_k^N \end{bmatrix} \qquad \mathbf{P}_k = \begin{bmatrix} \mathbf{P}_{k,\text{POI,POI}} & \mathbf{P}_{k,\text{POI},\mathbf{b}^1} & \ldots & \mathbf{P}_{k,\text{POI},\mathbf{b}^N} \\ \mathbf{P}_{k,\mathbf{b}^1,\text{POI}} & \mathbf{P}_{k,\mathbf{b}^1,\mathbf{b}^1} & \ldots & \mathbf{P}_{k,\mathbf{b}^1,\mathbf{b}^N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{k,\mathbf{b}^N,\text{POI}} & \mathbf{P}_{k,\mathbf{b}^N,\mathbf{b}^1} & \ldots & \mathbf{P}_{k,\mathbf{b}^N,\mathbf{b}^N} \end{bmatrix} \tag{4.23}$$

The EIF algorithm presented in Section 4.3 can now be applied by making changes to the dynamics model (Equation 4.1) and the measurement model (Equation 4.4). First, the dynamics model is augmented to include models of the bias states as

$$\mathbf{x}_{k+1} = \mathbf{f}\left(\mathbf{x}_k, \mathbf{w}_k\right) = \begin{bmatrix} \mathbf{f}_{\text{POI}}(\mathbf{x}_{k,\text{POI}}, \mathbf{w}_{k,\text{POI}}) \\ \mathbf{b}_k^1 + \mathbf{w}_{k,\mathbf{b}^1} \\ \vdots \\ \mathbf{b}_k^N + \mathbf{w}_{k,\mathbf{b}^N} \end{bmatrix} \tag{4.24}$$

and the process noise covariance, $\mathbf{Q}_k$, includes the covariance for the process noise of the POI as well as the bias and is given by

$$\mathbf{Q}_k = \begin{bmatrix} \mathbf{Q}_{k,\text{POI}} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{k,\mathbf{b}^1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \ldots & \mathbf{0} & \mathbf{Q}_{k,\mathbf{b}^N} \end{bmatrix}. \tag{4.25}$$

Second, the measurement model must be modified to include the biases, and is written for the j^{th} UAV as

$$\mathbf{z}_{k+1}^j = \mathbf{h}^j(\mathbf{x}_{k+1}, \boldsymbol{\eta}_{k+1}^j, \mathbf{v}_{k+1}^j) = \mathbf{h}_{\text{SCR}}(\mathbf{x}_{k+1,\text{POI}}, \widehat{\boldsymbol{\psi}}_{k+1}^j + \mathbf{b}_{k+1}^j + \boldsymbol{\eta}_{k+1}^j, \mathbf{v}_{k+1,\text{SCR}}^j) \tag{4.26}$$

The EIF algorithm from Section 4.3 can now be applied with the new dynamics and measurement equations.

Although the augmentation of the POI state with the biases of each UAV in the centralized approach is statistically correct, there are two significant problems with such a formulation. First, each UAV must model the dynamics and maintain estimates of the biases of all other UAVs (Equation 4.24). This violates one of the basic concepts of decentralized data fusion [59], and hence is termed here as the centralized approach. Second, the size of the state to be estimated scales with the number of UAVs (N) and therefore, the computation scales poorly with the number of UAVs. In fact, the information filter requires a matrix inverse to recover the state estimate and covariance, and this operation scales cubically with the size of the state. A decentralized approach is proposed in the next section, while the centralized solution is used in Section 4.6 to provide a benchmark for comparison.

4.4.2 Decentralized Approach

A decentralized approach, developed here, requires the UAVs to model only the POI and their own biases, and only share information related to the POI state. Thus, the decentralized approach scales well with the number of UAVs and requires less computation and communication than the centralized approach. In the decentralized approach, the state to be estimated on each UAV, $\mathbf{x}_k^j$, includes only the state of the POI, $\mathbf{x}_{k,\mathrm{POI}}$, and the local biases, $\mathbf{b}_k^j$, and is written as

$$\mathbf{x}_k^j = \begin{bmatrix} \mathbf{x}_{\mathrm{k,POI}} \\ \mathbf{b}_k^j \end{bmatrix}, \quad \text{for} \quad j = 1, \ldots, N. \tag{4.27}$$

Note that the superscript j is included as the state to be estimated is different on each UAV and hence the estimates and covariances (information states and matrices) are different on each UAV. The state dynamics are modified to include only the POI and local biases as

$$\mathbf{x}_{k+1}^{j} = \mathbf{f}\left(\mathbf{x}_{k}^{j}, \mathbf{w}_{k}^{j}\right) = \begin{bmatrix} \mathbf{f}_{\mathrm{POI}}(\mathbf{x}_{k,\mathrm{POI}}, \mathbf{w}_{k,\mathrm{POI}}) \\ \mathbf{b}_{k}^{j} + \mathbf{w}_{k,\mathbf{b}^{j}} \end{bmatrix}, \mathbf{Q}_{k}^{j} = \begin{bmatrix} \mathbf{Q}_{k,\mathrm{POI}} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{k,\mathbf{b}^{j}} \end{bmatrix} \tag{4.28}$$

Note that the measurement (Equation 4.26) for each UAV remains the same since it already contains only local biases.

In the decentralized formulation, the state estimate and error covariance are written as

$$\widehat{\mathbf{x}}_{k}^{j} = \begin{bmatrix} \widehat{\mathbf{x}}_{\mathrm{k,POI}}^{j} \\ \widehat{\mathbf{b}}_{k}^{j} \end{bmatrix}, \quad \mathbf{P}_{k}^{j} = \begin{bmatrix} \mathbf{P}_{k,\mathrm{POI,POI}}^{j} & \mathbf{P}_{k,\mathrm{POI},\mathbf{b}^{j}}^{j} \\ \mathbf{P}_{k,\mathbf{b}^{j},\mathrm{POI}}^{j} & \mathbf{P}_{k,\mathbf{b}^{j},\mathbf{b}^{j}}^{j} \end{bmatrix}, \tag{4.29}$$

and the usual conversion between state and information space applies:

$$\mathbf{Y}_{k}^{j} = (\mathbf{P}_{k}^{j})^{-1} \quad \mathbf{y}_{k}^{j} = \mathbf{Y}_{k}^{j}\widehat{\mathbf{x}}_{k}^{j} \tag{4.30}$$

The information state and matrix are propagated forward in time, using Equations 4.7 and 4.8 to get $\mathbf{Y}_{k+1}^{j-}$ and $\mathbf{y}_{k+1}^{j-}$. The local information state and matrix updates are computed using Equations 4.11 and 4.12 to get

$$\mathbf{I}_{k+1}^{j} = \begin{bmatrix} \mathbf{I}_{k,\mathrm{POI,POI}} & \mathbf{I}_{k,\mathrm{POI},\mathbf{b}^{j}} \\ \mathbf{I}_{k,\mathbf{b}^{j},\mathrm{POI}} & \mathbf{I}_{k,\mathbf{b}^{j},\mathbf{b}^{j}} \end{bmatrix}, \quad \mathbf{i}_{k+1}^{j} = \begin{bmatrix} \mathbf{i}_{k,\mathrm{POI}} \\ \mathbf{i}_{k,\mathbf{b}^{j}} \end{bmatrix} \tag{4.31}$$

With N decentralized estimators of the form described in Eqn. 4.29, it is noted that only the POI state is common across each of the estimators and thus it is desired to share only information related to the POI states. However, communication and fusion of only the POI state portion of the information updates, $\mathbf{I}_{k,\mathrm{POI,POI}}$ and $\mathbf{i}_{k,\mathrm{POI}}$, is unfortunately not statistically correct. The off diagonal portion of

the information matrix updates, $\mathbf{I}_{k,\mathrm{POI},\mathbf{b}^j}$, actually serves to reduce the information about the target because of the coupling with the uncertainty in the bias states. It is proposed to account for this coupling using two marginalization steps before communication and fusion.

The process of computing and fusing the marginalized information updates is written for a local UAV l as follows. The first step is to fuse the local information updates as

$$\mathbf{Y}^l_{k+1} = \mathbf{Y}^{l-}_{k+1} + \mathbf{I}^l_{k+1} \tag{4.32}$$

$$\mathbf{y}^l_{k+1} = \mathbf{y}^{l-}_{k+1} + \mathbf{i}^l_{k+1}, \tag{4.33}$$

which includes information about the local POI and bias estimates to be incorporated (Eqn. 4.31).

The second step is to compute two marginalizations in order to prepare the information to be communicated to the other UAVs. Marginalization of the *predicted* bias states begins by converting the predicted information state and matrix back to state space as

$$\mathbf{P}^{l-}_{k+1} = (\mathbf{Y}^{l-}_{k+1})^{-1} = \begin{bmatrix} \mathbf{P}^{l-}_{k,\mathrm{POI},\mathrm{POI}} & \mathbf{P}^{l-}_{k,\mathrm{POI},\mathbf{b}^l} \\ \mathbf{P}^{l-}_{k,\mathbf{b}^l,\mathrm{POI}} & \mathbf{P}^{l-}_{k,\mathbf{b}^l,\mathbf{b}^l} \end{bmatrix}, \quad \widehat{\mathbf{x}}^{l-}_{k+1} = \mathbf{P}^{l-}_{k+1}\mathbf{y}^{l-}_{k+1} = \begin{bmatrix} \widehat{\mathbf{x}}^{l-}_{\mathrm{k,POI}} \\ \widehat{\mathbf{b}}^{l-}_{k} \end{bmatrix}. \tag{4.34}$$

Then, in the state space, the bias states are marginalized out by simply removing the corresponding rows and columns of the predicted covariance and state estimate, since the joint POI and bias state is multi-variate Gaussian. The marginalized (M) covariance and state estimate are given as

$$\mathbf{P}^{l-}_{k+1,M} = \mathbf{P}^{l-}_{k+1,\mathrm{POI},\mathrm{POI}}, \quad \widehat{\mathbf{x}}^{l-}_{k+1,M} = \widehat{\mathbf{x}}^{l-}_{k+1,\mathrm{POI}} \tag{4.35}$$

The marginalized estimate and covariance are then converted back to information

space as

$$\mathbf{Y}^{l-}_{k+1,M} = (\mathbf{P}^{l-}_{k+1,M})^{-1} \quad \mathbf{y}^{l-}_{k+1,M} = \mathbf{Y}^{l-}_{k+1,M}\widehat{\mathbf{x}}^{l-}_{k+1,M}. \tag{4.36}$$

This marginalization process is then repeated for the *updated* information matrix, $\mathbf{Y}^{l}_{k+1}$, and state, $\mathbf{y}^{l}_{k+1}$, to get $\mathbf{Y}^{l}_{k+1,M}$ and $\mathbf{y}^{l}_{k+1,M}$.

The third step is to compute the marginalized information state and matrix updates as

$$\mathbf{I}^{l}_{k+1,M} = \mathbf{Y}^{l}_{k+1,M} - \mathbf{Y}^{l-}_{k+1,M} \tag{4.37}$$

$$\mathbf{i}^{l}_{k+1,M} = \mathbf{y}^{l}_{k+1,M} - \mathbf{y}^{l-}_{k+1,M} \tag{4.38}$$

The final step is to communicate the local marginalized information matrix and state updates to the other UAVs. Locally, all received updates are then fused as

$$\mathbf{Y}^{l}_{k+1} \Leftarrow \mathbf{Y}^{l}_{k+1} + \sum_{j=1,j\neq l}^{N} \begin{bmatrix} \mathbf{I}^{j}_{k+1,M} & \mathbf{0}_{n_{\text{POI}},n_{\mathbf{b}^l}} \\ \mathbf{0}_{n_{\mathbf{b}^l},n_{\text{POI}}} & \mathbf{0}_{n_{\mathbf{b}^l},n_{\mathbf{b}^l}} \end{bmatrix} \tag{4.39}$$

$$\mathbf{y}^{l}_{k+1} \Leftarrow \mathbf{y}^{l}_{k+1} + \sum_{j=1,j\neq l}^{N} \begin{bmatrix} \mathbf{i}^{j}_{k+1,M} \\ \mathbf{0}_{n_{\mathbf{b}^l},1} \end{bmatrix}, \tag{4.40}$$

where $\mathbf{0}_{m,n}$ is an m by n matrix of zeros, n_{POI} is the dimension of the POI state and $n_{\mathbf{b}^l}$ is the dimension of the biases being estimated on UAV l.

The marginalization steps accomplish two objectives:

1. The marginalized information updates are for the POI state only. This means that UAVs with different types of biases or even different types of sensors can effectively cooperate in geolocation. Further, no knowledge of the biases or even of the measurement models of the other UAVs is required.
2. The uncertainty in the bias states is used to reduce the information update on the POI, to account for uncertainty in the bias estimates. This improves

estimator consistency over no bias estimation, while still improving geolocation accuracy with local bias estimates.

However, marginalization causes some information to be lost. Specifically, the cross correlation among the biases is no longer maintained, i.e. $\mathbf{P}_{\mathbf{b}^j,\mathbf{b}^l} \equiv \mathbf{0}$ in the decentralized formulation. As long as the cross correlation is small, there is only a small reduction in geolocation accuracy over the centralized solution. This small loss of accuracy in geolocation is traded for significant gains in scalability, computation, and communication. Further, this decentralized solution fits nicely into the decentralized data fusion paradigm.

4.5 Observability Procedure

In Section 4.4, the POI and bias estimation problem is formulated with the potential for biases in the UAV position, $\mathbf{b}_{k,\text{NAV}}$, UAV attitude, $\mathbf{b}_{k,\text{ATT}}$, and camera attitude, $\mathbf{b}_{k,\text{GIM}}$. This section develops a procedure for evaluating the observability of both the POI and bias states. Due to the nonlinear measurement function for a camera sensor, shown in Appendix C, this is a nonlinear observability problem. Therefore, a numerical procedure which follows Refs. [48] and [47] is developed that utilizes the inverse of the observability grammian.

The observability grammian at time step, K, is given by

$$\mathcal{Q}_K = \mathcal{O}_K^T \mathcal{O}_K, \tag{4.41}$$

where $\mathcal{O}_K$ is the observability matrix at time step K and is given by

$$\mathcal{O}_K = \begin{bmatrix} {}^{\mathbf{x}}\mathbf{H}_1 \\ {}^{\mathbf{x}}\mathbf{H}_2\mathbf{F}_1 \\ \vdots \\ {}^{\mathbf{x}}\mathbf{H}_K\mathbf{F}_{K-1}\dots\mathbf{F}_1 \end{bmatrix}, \tag{4.42}$$

with ${}^{\mathbf{x}}\mathbf{H}$ defined in Eqn. 4.14 and $\mathbf{F}$ defined in Eqn. 4.9. Note that only one UAV is considered here and therefore the superscripts denoting the UAV are not used in this Section. Extensions of the observability analysis to multiple UAVs is straight forward.

Observability requires $\mathcal{Q}$ or equivalently $\mathcal{O}$ be full rank. A scaled version of the observability grammian, which gives physical insight into the degree of observability, is used here and defined as

$$\overline{\mathcal{Q}}_K = \mathcal{O}_K^T \mathcal{R}_K^{-1} \mathcal{O}_K, \tag{4.43}$$

where $\mathcal{R}_K$ is used to scale the observability grammian based on the effective sensor noise covariance,

$$\mathcal{R}_K = \begin{bmatrix} \mathbf{R}_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{R}_K \end{bmatrix}, \tag{4.44}$$

where it is noted that $\mathcal{R}_K$ is full rank.

Three important observations are made regarding this scaled observability grammian.

1. Since observability is determined by the rank of $\mathcal{Q}_K$, observability can also be inferred from $\overline{\mathcal{Q}}_K$ since

$$\text{Rank}(\mathcal{Q}_K) = \text{Rank}(\overline{\mathcal{Q}}_K) \tag{4.45}$$

2. $\overline{\mathcal{Q}}_K$ has a physically meaningful interpretation. In the case of no process noise; i.e., $\mathbf{Q} = \mathbf{0}$, $\overline{\mathcal{Q}}_K^{-1}$ is the error covariance at time step K if one starts with a diffuse prior, $\mathbf{Y}_0 = \mathbf{0}$.

3. If the POI is stationary, $\mathbf{F}_k = \mathcal{I}$, then $\overline{\mathcal{Q}}_K$ can be computed recursively by summing the information matrix updates, $\mathbf{I}_k$ in Equation 4.12, as

$$\overline{\mathcal{Q}}_K = \sum_{k=1}^{K} \mathbf{I}_k \tag{4.46}$$

The process for determining observability is summarized as follows. First, define the state to be considered, $\mathbf{x}^*$. For example, the most general joint POI and bias state is $\mathbf{x}^* = \left[\mathbf{x}_{\text{POI}}^T,\ \mathbf{b}_{\text{NAV}}^T,\ \mathbf{b}_{\text{ATT}}^T,\ \mathbf{b}_{\text{GIM}}^T\right]^T$. Define a time step of interest, such as $K = k_{\text{orbit}}$ where k_{orbit} is the number of time steps for the UAV to make a single orbit about the POI. Then, compute $\overline{\mathcal{Q}}_K$ using Eqn. 4.43. Finally, check the rank of $\overline{\mathcal{Q}}_K$; the state $\mathbf{x}^*$ is then observable if and only if $\overline{\mathcal{Q}}_K$ is full rank. If $\overline{\mathcal{Q}}_K$ is full rank, then the degree of observability can be assessed by taking the inverse of $\overline{\mathcal{Q}}_K$ to get $\overline{\mathcal{P}}_K$. The matrix $\overline{\mathcal{P}}_K$ denotes the error covariance when starting with a diffuse prior and no process noise. Large values of the diagonal of $\overline{\mathcal{P}}_K$ indicate poor observability. Further, if some of the elements of the state $\mathbf{x}^*$ have comparable units, e.g., $\mathbf{b}_{\text{ATT}}$ and $\mathbf{b}_{\text{GIM}}$, then the modes and degree of observability can be determined by analyzing the eigenvalues, Λ, and eigenvectors, V, of the submatrix of $\overline{\mathcal{P}}_K$ corresponding to those states. Large elements of Λ indicate poor observability and the corresponding eigenvector yields the related combination of the elements of $\mathbf{x}^*$.

4.6 Validation with Experimental and Simulated Flight Data

The decentralized bias estimation approach presented in Section 4.4 is evaluated and compared with the centralized solution using both experimental flight test data and hardware in the loop (HiL) simulations. Additionally, the procedure for observability analysis developed in Section 4.5 is used to evaluate the observability of the POI and bias states. Note that due to International Traffic in Arms Regulations (ITAR) restrictions, all results are scaled to avoid showing absolute performance. However, since a consistent scaling is used for all results, all relative comparisons should be clear.

4.6.1 Experimental and Simulated Flight Test Setup

Both experimental flight data and hardware in the loop (HiL) simulations are used. Figure 4.2 shows the orbit configuration used for both the experimental and HiL tests, including relative phasing between the UAVs, β, and orbit offsets from the POI, Δ. The UAVs are shown as triangles and the POI is shown as a star.

Experimental flight data was collected during a cooperative UAV flight test on March 16, 2007. In this test, two UAVs orbited a stationary POI, with a POI centered orbit, $\Delta = 0$, and an orbit radius of 500 m. For this test, a van was used as the POI and the true location was measured using a local GPS receiver. The experimental flight test is described in more detail in Ref. [57].

In addition to the experimental flight data, a series of high fidelity HiL simulations were performed. The HiL simulations include realistic environmental effects

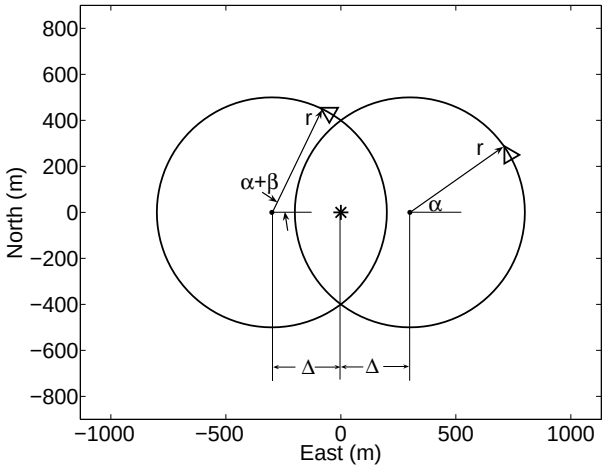


Figure 4.2: Diagram illustrating the configuration (in projected 2D) of two UAVs (triangle) orbiting above a POI ($*$) performing geolocation. The orbits are parameterized by the relative phase, β, and orbit offset, Δ.

such as wind, as well as the actual onboard guidance software that coordinates the motion of the UAVs developed by Wise and Rysdyk [58, 68]. HiL simulations were performed to generate flight data for both stationary and moving POIs. Tests were performed for a range of relative phase angles between the UAVs and offsets to their orbits from the POI. The HiL test configurations used are summarized in Table 4.1

Table 4.1: Test configurations for the HiL simulations.

HiL Test Number	POI Motion	Phase, β (deg)	Orbit Offset, Δ (m)
1	Stationary	$[0, 30, \ldots, 180]$	0
2	Stationary	150	$[0, 50, \ldots, 500]$
3	Moving	$[0, 30, \ldots, 180]$	0

For each HiL test, bias errors were added to the UAV attitude and camera attitude. The values of the biases were chosen to be representative, and are not shown due to ITAR restrictions. The trajectory of the POI in the moving POI case is based on a city driving model, which is characterized by frequent stops and 90°

turns. The FlockGuidance software developed by Wise and Rysdyk [58, 68] is used to maintain the configuration, shown in Figure 4.2, relative to the moving POI. For each test of a moving POI, a different POI trajectory is used and constant bias errors are added to both the camera and UAV attitudes estimates. Note that the bias errors are the same for each test but different for each UAV.

4.6.2 Stationary POI Results

Geolocation and bias estimation for a stationary POI are evaluated both with experimental flight test data and HiL simulations. A stationary POI is modeled as

$$\mathbf{x}_{k+1,\text{POI}} = \mathbf{x}_{k,\text{POI}} + \mathbf{w}_{k,\text{POI}}, \tag{4.47}$$

where the state of the POI, $\mathbf{x}_{k,\text{POI}}$, is the position in a local North-East-Down (NED) frame. The process noise, $\mathbf{w}_{k,\text{POI}}$, is white, Gaussian noise with a relatively small covariance, $\mathbf{Q}_{k,\text{POI}} = 0.05 \cdot \mathcal{I}_{n_{\text{POI}}}$ m/s, which is included to prevent premature convergence. The stationary POI results are organized as follows. First, the case of a single UAV is considered, including an observability analysis as well as geolocation error evaluation. Second, cooperative geolocation is evaluated using the experimental flight test data, which includes two UAVs in circular orbits about a stationary POI.

Single UAV

First, consider the observability of the POI and bias states using a single UAV, a stationary POI, and a POI centered orbit for the UAV. Observability is evaluated using the procedure developed in Section 4.5. The state $\mathbf{x}^*$ is defined to be the

state of the POI and all possible biases,

$$\mathbf{x}^* = \begin{bmatrix} \mathbf{x}_{\text{POI}} \\ \mathbf{b}_{\text{NAV}} \\ \mathbf{b}_{\text{ATT}} \\ \mathbf{b}_{\text{GIM}} \end{bmatrix}, \tag{4.48}$$

where the dimension of $\mathbf{x}^*$ is 12. Further, define $K = k_{\text{orbit}}$, where k_{orbit} is the number of time steps for the UAV to complete one orbit about the POI. Note that one orbit is used here because the UAV continually repeats orbits about the POI. The observability of $\mathbf{x}^*$ is evaluated by computing $\overline{\mathcal{Q}}_K$, as in Eqn. 4.43 and then computing the rank giving, in this case,

$$\text{Rank}(\overline{\mathcal{Q}}_K) = 9, \tag{4.49}$$

which, is a rank deficiency of 3. This indicates that not all states of $\mathbf{x}^*$ are observable.

Consider the effect of removing the UAV navigation bias from $\mathbf{x}^*$, i.e., redefining $\mathbf{x}^*$ to be

$$\mathbf{x}^* = \begin{bmatrix} \mathbf{x}_{\text{POI}} \\ \mathbf{b}_{\text{ATT}} \\ \mathbf{b}_{\text{GIM}} \end{bmatrix}, \tag{4.50}$$

where the dimension of $\mathbf{x}^*$ is now 9. Recomputing $\overline{\mathcal{Q}}_K$ and taking the rank gives

$$\text{Rank}(\overline{\mathcal{Q}}_K) = 9, \tag{4.51}$$

which is full rank. This indicates that $\mathbf{x}_{\text{POI}}$, $\mathbf{b}_{\text{ATT}}$, and $\mathbf{b}_{\text{GIM}}$ are all observable, but $\mathbf{b}_{\text{NAV}}$ is not. Intuitively, this implies that navigation biases are translated directly into POI biases. It is noted that the observability of UAV navigation biases was explored in Ref. [2], where the navigation biases were modeled as

$$\mathbf{b}_{k+1,\text{NAV}} = a\ \mathbf{b}_{k,\text{NAV}} + \mathbf{w}_{k,\text{NAV}}. \tag{4.52}$$

It was then shown that position biases are unobservable with $a = 1$, which is the bias model used here. However, if the biases are taken as autocorrelated errors, with $a \neq 1$, then the position biases are observable. However, the time scale for GPS based navigation errors is too large to be useful here as geolocation is desired in less than an orbit and this would require multiple orbits.

Since the ATT and GIM states are observable, one can check the degree of observability as described in Section 4.5. First, invert $\overline{\mathcal{Q}}_K$ to get $\overline{\mathcal{P}}_K$ and then analyze the eigenvectors and eigenvalues of the submatrix of $\overline{\mathcal{P}}_K$ corresponding to $\mathbf{b}_{\text{ATT}}$ and $\mathbf{b}_{\text{GIM}}$. In this case, the eigenvectors and eigenvalues are

$$V = \begin{bmatrix} 0.18 & 0.68 & 0.01 & 0.04 & -0.71 & -0.01 \\ 0.01 & 0.04 & -0.47 & -0.83 & -0.01 & -0.30 \\ 0.02 & 0.00 & -0.81 & 0.28 & 0.00 & 0.51 \\ 0.70 & -0.18 & 0.24 & -0.34 & -0.02 & 0.55 \\ -0.19 & -0.68 & -0.02 & -0.01 & -0.71 & -0.01 \\ -0.66 & 0.19 & 0.24 & -0.35 & -0.01 & 0.59 \end{bmatrix} \begin{matrix} \leftarrow roll \\ \leftarrow pitch \\ \leftarrow yaw \\ \leftarrow pan \\ \leftarrow tilt \\ \leftarrow scan \end{matrix} \tag{4.53}$$

$$\Lambda = \begin{bmatrix} 26.45 & 7.56 & 1.55 & 0.64 & 0.00 & 0.00 \end{bmatrix}, \tag{4.54}$$

where the columns of V are the eigenvectors and the elements of Λ are the corresponding eigenvalues.

Since $\mathbf{b}_{\text{ATT}}$ and $\mathbf{b}_{\text{GIM}}$ have the same units, the scaled eigenvalues in Eqn. 4.54 provide a meaningful measure of the observability and the eigenvectors in Eqn. 4.53 give the modes of observability. Note that large eigenvalues correspond to poor observability. The least observable mode is the combination of camera pan and camera scan with an eigenvalue of 26.4. This indicates that a pan bias is difficult to distinguish from a scan bias. The other poorly observable mode is the combination of UAV roll and camera tilt with an eigenvalue of 7.5. So, even though, as indicated

by the rank condition, all of the attitude biases are observable in a POI centered orbit, some of them are poorly observable.

The poor observability of the combination of both ATT and GIM biases in a POI centered orbit indicates that estimating only a subset of the biases may be effective in improving geolocation accuracy. To test this, four choices of biases to estimate are evaluated here using the 2007 flight test data with a single UAV in a POI centered orbit ($\Delta = 0$) about a stationary POI: 1) No biases (No Bias) 2) Camera gimbal bias (GIM) 3) UAV attitude (ATT) 4) UAV attitude and camera gimbal biases (ATT-GIM). The geolocation error for each of the four bias choices is shown in Figure 4.3 for the first five minutes of the 2007 flight test data.

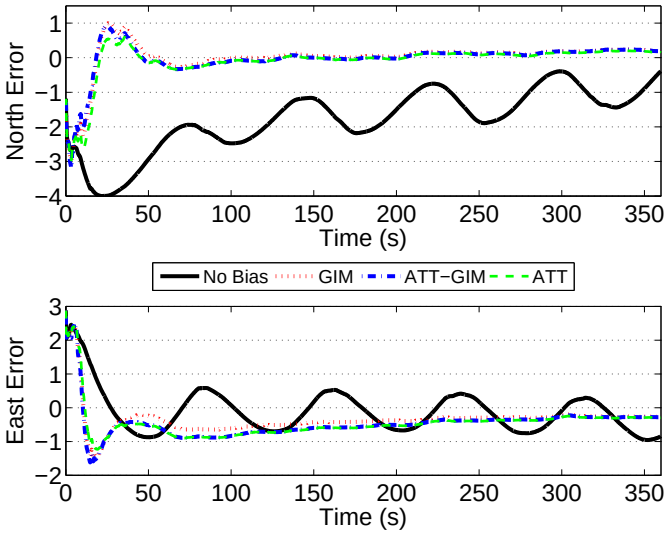
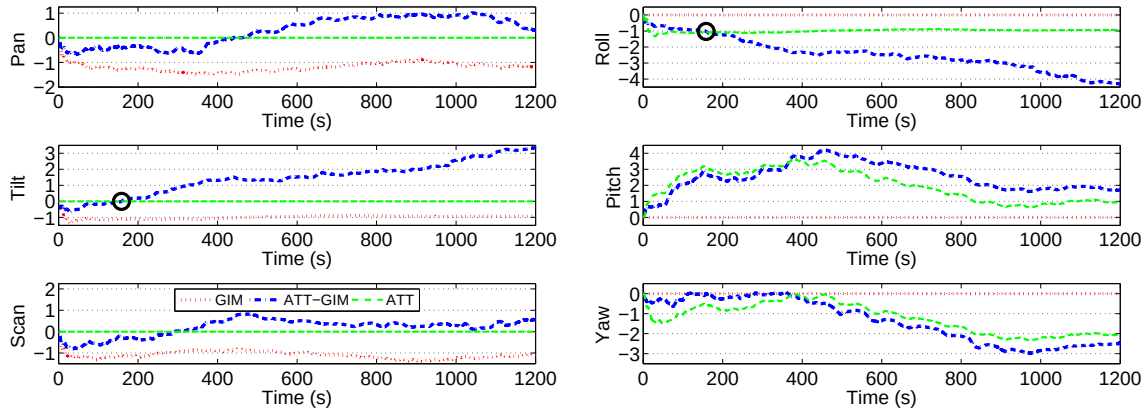


Figure 4.3: Geolocation error during the first six minutes of the 2007 flight test data with the UAV in a POI centered orbit about a stationary POI and using four different choices of biases to estimate: 1) No biases (No Bias) 2) Camera gimbal bias (GIM) 3) UAV attitude (ATT) 4) UAV attitude and camera gimbal biases (ATT-GIM).

Two significant observations can be made from Figure 4.3. First, all forms of sensor bias estimation improve geolocation accuracy, both in terms of steady state estimate error and convergence time. When the sensor biases are not modeled,

the geolocation estimate oscillates with a period that is the same as the orbit of the UAV. When estimating the sensor bias, the geolocation estimate converges to within 1 of the POI location in 30 seconds, compared to more than 6 minutes with no bias estimation. The second important observation is that the choice of sensor biases to model has little impact on geolocation accuracy in this case, i.e., a centered orbit about a stationary POI. This is because the UAV attitude and camera attitude biases are poorly observable when flying in a POI centered orbit.

As further evidence of the lack of observability in POI centered orbits, Figure 4.4 shows the estimates of the sensor biases for three bias estimation choices: 1) GIM 2) ATT 3) ATT-GIM. Note that the estimates for biases that are not part of that estimator are plotted as zero, e.g., the roll bias estimate for GIM bias estimation in Figure 4.4(b).



(a) Camera Gimbal Bias Estimates

(b) UAV Attitude Bias Estimates

Figure 4.4: Bias estimates for the 2007 flight test data using a single UAV and a centered orbit around a stationary POI.

Consider first the camera tilt and UAV roll bias estimates, shown in Figure 4.4. For a POI centered orbit, the camera is perpendicular to the fuselage of the UAV, creating a situation in which a bias in camera tilt has the same effect as a bias in UAV roll. Consequently, the tilt bias estimate from GIM bias estimation is the

same as the roll estimate from ATT bias estimation, i.e., a value of ≈ -1. Further, in the case of ATT-GIM bias estimation, the tilt bias estimate drifts positive while the roll bias estimate drifts negative. This drift is due to the poor observability between the camera tilt and UAV roll, as described earlier and shown in Eqn. 4.53. Interestingly, when the tilt bias estimate and the roll bias estimate are added for the ATT-GIM case, a value of ≈ -1 results, which is the same as the GIM and ATT cases. This is easiest to see near ≈ 160 seconds, where the circles show the convergence of all three bias estimation cases. Similar observations can be made by comparing the estimates of the other bias variables, albeit with more complicated analysis.

The lack of observability for POI centered orbits is not necessarily a problem; in fact, it can be exploited as a benefit. If the biases are being estimated only to improve geolocation, i.e., the true biases are not required, then, the reduced parameterization of the ATT *or* GIM cases will suffice to improve geolocation, as shown in Figure 4.3. This also serves to reduce computation, which scales with the cube of the dimension of the state to estimated.

It is insightful, then, to study the observability and performance of geolocation for non-centered orbits about a stationary POI. The effect of orbit offset on observability can be evaluated by considering the worst observability mode and its change with orbit offset, Δ, where the worst mode is defined to be $\lambda_{\max}$ with

$$\lambda_{\max} = \max_i \lambda_i, \tag{4.55}$$

where λ_i are the eigenvalues of $\overline{\mathcal{P}}_K = \overline{\mathcal{Q}}_K^{-1}$. HiL simulation tests were run for orbit offsets up to $\Delta = 500$m (equal to the orbit radius), and Figure 4.5 plots $\lambda_{\max}$ as a function of the orbit offset when estimating both ATT and GIM biases. Figure 4.5 shows that observability significantly improves with orbit offset up to an offset of

150m, with limited improvement afterwards. These results indicate that the biases in both the UAV attitude and camera attitude can be effectively estimated with a circular orbit which has a center offset from the POI, i.e., $\Delta \neq 0$.

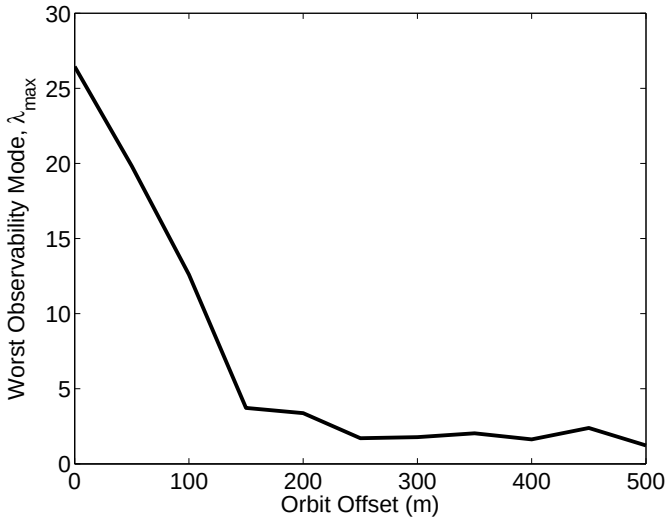


Figure 4.5: Observability of least observable mode over a range of orbit offsets.

The effect of orbit offset on geolocation accuracy is also evaluated with HiL simulation. Thirty minutes of simulated data from the HiL simulation is used for each of the 11 orbit offsets tested, $\Delta \in [0\text{ m}, 0\text{ m}, \ldots, 500\text{ m}]$. Figure 4.6 shows the steady state geolocation error for each of the orbit offset tests. The steady state geolocation error is defined here to be the Euclidean distance between the POI estimate and the true POI location, averaged over the last two minutes of the thirty minutes of data.

Figure 4.6 shows that geolocation accuracy degrades with orbit offset not only for the No Bias case, but also for the GIM only and ATT only cases as well. This is because, as the orbit offset increases, all of the biases are observable, and geolocation is now dependent on all of the biases. For the ATT and GIM cases, the estimator attempts to fit an under-parameterized set of biases to the true set of six. Since ATT-GIM bias estimation includes all of the biases, it is invariant to

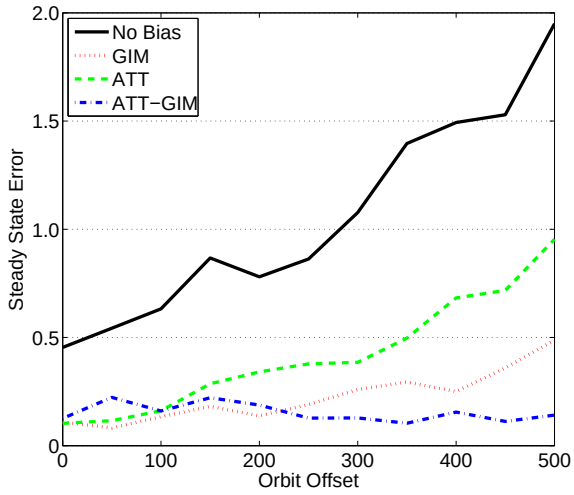


Figure 4.6: Average steady state geolocation error from HiL simulation with orbit offsets ranging from 0 m to 500 m (the orbit radius was 500 m).

orbit offset.

Cooperative UAV Flight Test Results

Cooperative geolocation is experimentally evaluated using the 2007 flight test data, which included two UAVs in a circular orbit centered over a stationary POI. The experimental flight data is broken into four subsets of six minutes each. Figure 4.7 shows geolocation errors averaged over those four subsets. Six different cases are included in Figure 4.7 based on a 3×2 selection of three cooperation choices: 1) a single UAV 2) centralized with two UAVs and 3) decentralized with two UAVs and two bias choices: 1) No Bias and 2) ATT-GIM Bias. The error is defined to be the Euclidean distance between the true and estimated POI locations.

Four important conclusions can be drawn from Figure 4.7. First, comparing a single UAV, with and without bias estimation shows that geolocation accuracy improves by approximately 75% with bias estimation. Second, comparing the de-

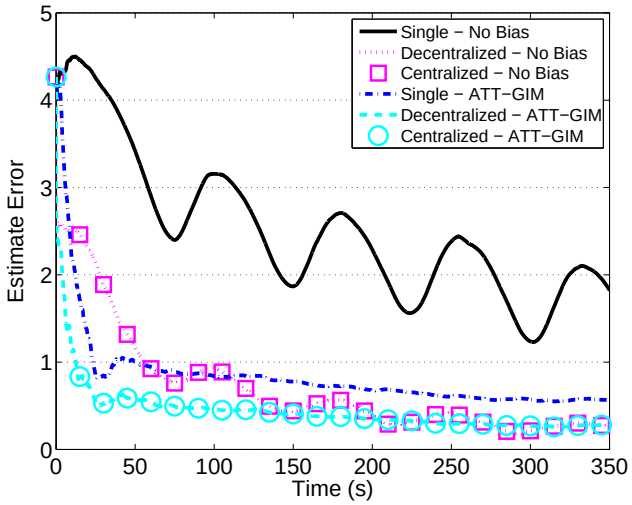


Figure 4.7: Geolocation error averaged over four subsets of the 2007 flight test with three choices of cooperation: Single UAV, Decentralized, and Centralized and also with two choices of biases to estimate: No Bias, and ATT-GIM bias.

centralized No Bias vs. ATT-GIM cases, bias estimation improves the convergence time ($\approx$ 1 minute with bias estimation compared to $\approx$ 4 minutes without bias estimation), but gives approximately the same steady state error. The centralized (No Bias vs. ATT-GIM) comparison gives similar results. Third, comparing a single UAV with No Bias estimation to using two UAVs (decentralized or centralized) with No Bias estimation shows that using two UAVs improves geolocation accuracy significantly, both in terms of the steady state error (≈ 1.5 for one UAV compared to ≈ 0.3 for two UAVs) and convergence time ($\approx$ 6 minutes for one UAV and $\approx$ 4 minutes for two UAVs). Fourth, with ATT-GIM bias estimation, comparing the decentralized bias estimation approach and the centralized bias estimation approach shows that the decentralized approach remains within 0.03% of the centralized approach at all times, when locating a stationary POI. This shows that the algorithmic assumption in the decentralized approach, that the cross correlation between the biases on each of the UAVs is small and can be removed through marginalization, is reasonable in this case.

4.6.3 Moving Point of Interest

Bias estimation for a moving POI is evaluated in this section using the HiL simulation data for two ScanEagle UAVs locating a moving POI, described in Section 4.64.6.1. The path of the moving POI is based on a city driving model characterized by frequent stops and 90° turns. A representative moving POI trajectory is shown in Figure 4.8. The SPOIs, computed camera line of sight intersections with the ground, are also included in Figure 4.8 as red dots for reference as an approximate evaluation of the camera measurement.

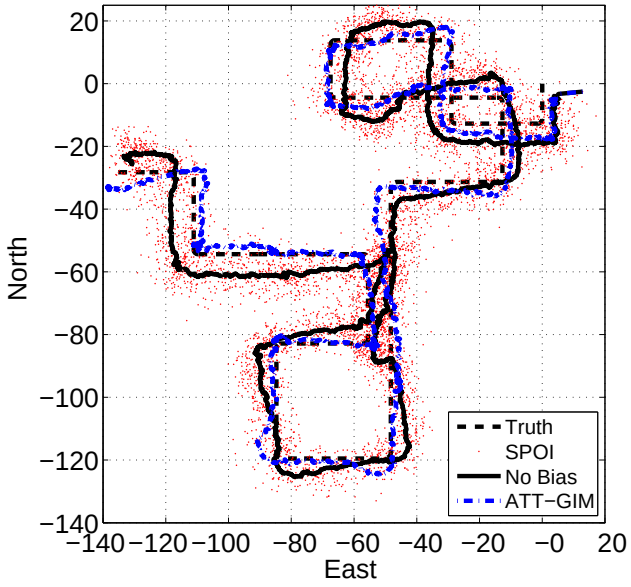


Figure 4.8: Representative trajectory from the HiL city driving simulator and geolocation estimates using a single UAV and two bias estimation choices: No Bias, and ATT-GIM bias.

A simple ballistic model is used as the POI dynamics model for the geolocation estimator and is written as

$$\mathbf{x}_{k+1,\mathrm{POI}} = \begin{bmatrix} \mathcal{I}_{3,3} & \delta t \mathcal{I}_{3,3} \\ \mathbf{0}_{3,3} & \mathcal{I}_{3,3} \end{bmatrix} \mathbf{x}_{k,\mathrm{POI}} + \begin{bmatrix} \mathbf{0}_{3,3} \\ \mathcal{I}_{3,3} \end{bmatrix} \mathbf{w}_{k,\mathrm{POI}}, \tag{4.56}$$

where $\mathcal{I}_{3,3}$ is a 3×3 identity matrix. The POI state stacks three position states

and three velocity states, both in the local North-East-Down (NED) frame.

Figure 4.8 also shows the estimates of the POI using one UAV for two cases of bias estimation: 1) no bias estimation and 2) ATT-GIM bias estimation. When no biases are estimated (black line in Figure 4.8) the estimate approximately follows the SPOIs. This is due to the freedom in a ballistic POI model used here, as the estimator considers the SPOI to be a statistically reasonable estimate for the POI state. When biases are assumed in both the ATT-GIM states (blue dash-dot line), the estimate follows the true POI path more closely.

The geolocation performance (average error) for the moving POI case is presented in Figure 4.9 for a 3×2 selection of three cooperation cases: 1) a single UAV 2) centralized with two UAVs and 3) decentralized with two UAVs; and two bias cases: 1) No Bias and 2) ATT-GIM Bias. The data from the seven 30 min HiL tests of a moving POI are each broken into four non-overlapping subsets of seven minutes each; giving 28 data sets (7 tests varying relative phase $\beta \in [0°, 30°, \ldots, 180°]$, and 4 subsets of each). The geolocation error in Figure 4.9 is the average over these 28 data sets. The average geolocation error is computed by finding the Euclidean distance from the estimated POI location and the true POI location at each time step, for each data set, and then, taking the average over the 28 data sets.

Four conclusions can be drawn from Figure 4.9. First, using only one UAV and not estimating biases is the least accurate approach, with a steady state error of ≈ 6.5 and is used here as a baseline for comparison. Second, adding a second cooperating UAV with No Bias estimation (centralized and decentralized are the same in this case) provides some reduction in steady state error over the single UAV with No Bias estimation case with a steady state error of ≈ 4.2. Third, including bias estimation with a single UAV provides even more improvement in steady state

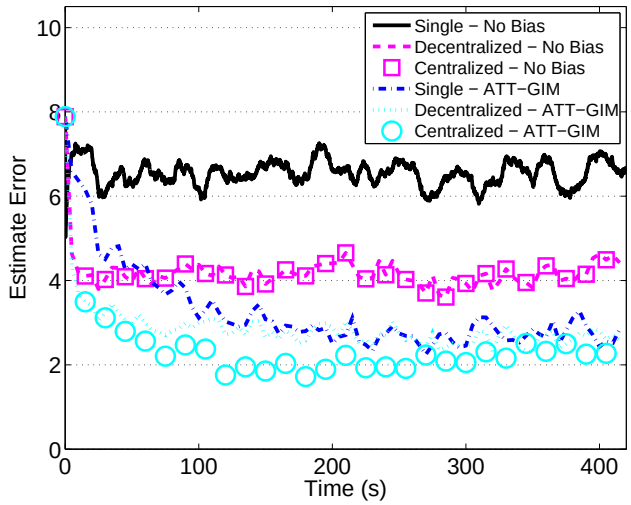


Figure 4.9: Geolocation error averaged over four subsets of the HiL moving POI test set with three choices of cooperation: Single UAV, Decentralized, and Centralized and also with two choices of biases to estimate: No Bias, and ATT-GIM bias.

error over the single UAV No Bias case and has a steady state error of ≈ 3. Fourth, the centralized and decentralized approaches for two UAVs with ATT-GIM bias estimation are the most accurate, particularly early (first four minutes) and only moderately so afterwards. Finally, the decentralized approach has approximately 5% larger steady state error than the centralized approach; this error was 0.3% in the stationary POI case. This indicates that the moving POI case results in a larger coupling of the biases of each UAV than the stationary POI case, and a small amount of information is lost through the marginalization process. This small loss of accuracy is traded with the scalability of the decentralized implementation in terms of reduced computation and communication.

4.7 Conclusions

The cooperative geolocation of a point of interest (POI) using multiple UAVs with articulating camera sensors is addressed, where there are non zero mean errors (biases) in the estimate of the UAV state. A decentralized, joint POI and bias estimation approach is proposed which utilizes a marginalization step allowing each UAV to model only the POI state and the local biases. Marginalization is used to share information only related to the POI state by assuming that the cross correlation between the biases across each of the UAVs is small and can be ignored. The proposed decentralized solution is scalable in that only information about the POI is shared among the UAVs.

An observability analysis of the joint geolocation and bias estimation problem showed that biases in the UAV navigation solution are not observable, and pass directly to the POI estimate. Also, biases in the UAV attitude and camera attitude are observable, albeit poorly for the case of a single UAV in a POI centered orbit. While potentially problematic, this poor observability of the biases allows for computational savings for the POI centered orbit case, by estimating only a subset of the biases without a degradation of geolocation accuracy. As the orbit center is offset from the POI, the biases are well observable. In this case, all biases must be estimated in order to yield accurate geolocation estimates; estimating only a subset of the biases degrades geolocation accuracy.

The proposed decentralized joint POI and bias estimation approach is validated using both experimental flight test data and hardware in the loop simulation data for two ScanEagle UAVs locating stationary and moving POIs. Including bias estimation coupled with POI estimation improves geolocation accuracy, in terms of convergence time and steady state estimate error for all cases. For the single UAV

case, the steady state estimate error is reduced by more than 50%. When locating a stationary POI, the decentralized approach recovers the centralized solution to within 0.03%. This validates the assumption that the cross correlation of the biases is small in this case. When locating a moving POI, however, the decentralized approach deviates from the centralized approach by approximately 5%. However, the small reduction in accuracy is traded for scalability in reduced computation and communication.

CHAPTER 5

COOPERATIVE GEOLOCATION IN THE PRESENCE OF COMMUNICATION LOSS

5.1 Abstract

Cooperative estimation using multiple mobile sensor nodes communicating over a lossy network is considered. A new method, termed Predicted Information, is developed from a separable formulation of the Extended Information Filter. Two variations of the predicted information method are presented which trade between accuracy and computational complexity. The first variation estimates the information matrix updates directly from estimates of the cooperating sensor node states. The second variation uses a piecewise constant approximation to predict the information matrix updates. The predicted information method is shown to give the exact solution for linear systems when the measurement dynamics are constant or known by all sensor nodes. The predicted information method is evaluated with a cooperative geolocation problem with two uninhabited aerial vehicles using gimballing camera sensors. Flight test data and high fidelity hardware in the loop simulations are used to compare the predicted information method with three benchmark methods from the literature for tracking both stationary and maneuvering targets, and for single extended losses and random dropouts.

5.2 Introduction

Cooperative estimation is the process where multiple possibly mobile sensor nodes take measurements of the state of interest and work together to find a joint esti-

mate. Fusion into a single estimate on each node can be formulated by exchanging information from each of the measurements (information fusion [41]) or state estimates based on local measurements (state fusion [29]). Cooperative estimation applications span from space [19], to air, to undersea [18]. Of particular interest here are target tracking applications utilizing multiple Uninhabited Aerial Vehicles (UAVs). Currently UAVs are being used to track ground targets in situations ranging from military missions to search and rescue operations [12] and even to animal migration [53]. Several groups have implemented target tracking systems on UAVs [49, 63, 15]. It is well known that data fusion techniques can significantly increase the information (decrease the uncertainty) of the target being tracked based on multi-sensor measurements [36, 41]. Recognizing this, Refs. [51, 11, 23] have implemented UAV systems using decentralized fusion (information filtering) concepts.

Communication is an important component of the cooperative estimation process, and recent research has focused on the effects of digital communication in both control and estimation. Delchamps [14] presented a seminal work describing the effect of quantization on controller performance. More recently, significant work has been done on the effect of network communication between sensors, controllers, and system plants [60, 69, 52, 70]. In addition to quantization effects, the cooperative estimation system must be robust to communication losses and delays. For example, in autonomous underwater vehicle applications, communication is particularly limited in reliability as well as bandwidth. In [[1]], Akyildiz *et al* characterize the underwater communication channel noting that communication is lost or delayed frequently due multi-path and obstructions.

The problem of cooperative estimation in the presence of communication loss

is considered in this paper. Communication loss refers to a situation in which the nodes are unable to communicate with each other for a period of time that is unknown a priori. It is assumed here that the communication losses are symmetric and known by each sensor node. Communication delay is a related problem and refers to the situation in which sensor nodes must communicate over a medium which induces potentially unknown and varying delays in the data transmission; for example, communication over an ad hoc wireless network. It is assumed here that the delays are not symmetric between nodes and there is no confirmation of receipt provided from the receiving node. In this case, any time a sensor node sends data, it will have no knowledge of how long before the data is received at the other end. Also, there is no guarantee that the data will be received at a sensor node in the order it was generated. Under this scenario, it is the responsibility of the receiver to make effective use of the data received, even if the data is both late and out of order.

The delayed data problem was explored in the Kalman filtering domain by Larsen *et al.* in Ref. [35], where the focus was on estimation with a combination of fast and slow sensors. In Ref. [30], Julier and Uhlmann developed an approach algebraically equivalent to the results of Larsen. In a centralized estimation framework, Bar-Shalom developed an exact solution for out-of-sequence measurements[3]. Nettleton and Durrant-Whyte [44] examined the delayed data problem in the information filtering domain, which is more amenable to decentralized sensor fusion. This approach is theoretically equivalent to centralized estimation in linear systems, but requires large memory and bursty communication. None of the existing methods address the communication loss (or delay) problem in the nonlinear decentralized estimation context. These cases arise when the POI dynamics or measurements are nonlinear, which is common in the UAV tracking

problem [11].

This paper develops a solution to the decentralized estimation problem using multiple mobile sensor nodes with communication over a lossy network. The new method, termed the Predicted Information (PI) method, is developed as an approximation to the Separable Extended Information Filter (SEIF), which is an alternate form of the Extended Information Filter (EIF) derived in this work. Our approach is to predict the information matrix updates during the communication loss to more easily update the estimator when communication is re-established. Two variations are presented which trade accuracy with computation, memory, and communication load. The first variation estimates the information matrix updates over time, while the second variation uses a piecewise constant approximation to the information matrix updates. Although the delayed data problem is not specifically addressed here, the PI method can be used with delayed data but requires minor modifications.

The PI method is validated and compared to benchmark methods from the literature [29, 44, 50] using ScanEagle UAV flight data for the cooperative geolocation of both stationary and moving targets. The ScanEagle is a long endurance UAV ($\sim$ 24 hours) developed by The Insitu Group and is equipped with a digital video camera integrated into an inertially-stabilized pan / tilt nose turret along with vision tracking software that allows the camera to remain pointed at both stationary and moving targets as the UAV maneuvers. Flight tests of two ScanEagle UAVs were performed on March 16, 2007 [57] and this data along with hardware in the loop simulations are used in the examples presented.

The paper is outlined as follows. First, the general decentralized cooperative estimation problem is presented with a description of the Extended Information

Filter (EIF). Second, a formulation of the EIF is presented which explicitly separates the contributions of the a priori information and the information coming from each sensor node. Next, the PI method is developed from the Separable EIF (SEIF) for the communication loss problem. Then, three methods from the literature are summarized as benchmark approaches for communication loss. Finally, a challenging and practically motivated nonlinear cooperative geolocation example is presented as validation. For the geolocation example, experimental flight data and hardware in the loop simulations are used to track both stationary and moving targets for two communication loss scenarios.

5.3 Decentralized Cooperative Estimation

Consider a general state of interest, $\mathbf{x}$, whose discrete time dynamics are governed by

$$\mathbf{x}_{k+1} = \mathbf{f}[\mathbf{x}_k, \mathbf{w}_k], \tag{5.1}$$

where the disturbance, $\mathbf{w}_k$, is zero mean, white, Gaussian noise with covariance $\mathbf{Q}_k$, and the subscript k denotes time step t_k. Assume there are n_s mobile sensor nodes with states, $\boldsymbol{\psi}^s_{k+1}$, taking measurements of the state of interest as

$$\mathbf{z}^s_{k+1} = \mathbf{h}^s[\mathbf{x}_{k+1}, \boldsymbol{\psi}^s_{k+1}, \mathbf{v}^s_{k+1}], \qquad s \in \mathbb{S} = \{1, \ldots, n_s\}, \tag{5.2}$$

where the sensor noise, $\mathbf{v}^s_{k+1}$, is zero mean, white, Gaussian noise with covariance ${}^v\mathbf{R}^s_{k+1}$. The process and sensor noises, $\mathbf{w}_k$, and $\mathbf{v}^s_{k+1}$, are assumed to be uncorrelated with each other. The sensor node dynamics are given by

$$\boldsymbol{\psi}^s_{k+1} = \mathbf{g}^s[\boldsymbol{\psi}^s_k, \boldsymbol{u}^s_k, \mathbf{w}^s_k], \tag{5.3}$$

where $\boldsymbol{u}^s_k$ is a control input and $\mathbf{w}^s_k$ is a zero mean white Gaussian noise disturbance. The sensor nodes are further assumed to have an onboard navigation system which

gives an estimate of the sensor node state, $\widehat{\boldsymbol{\psi}}^s_{k+1}$ such that

$$\boldsymbol{\psi}^s_{k+1} = \widehat{\boldsymbol{\psi}}^s_{k+1} + \boldsymbol{\eta}^s_{k+1}, \tag{5.4}$$

where the sensor node state estimate error, $\boldsymbol{\eta}^s_{k+1}$, is assumed zero mean, Gaussian, and white with covariance, ${}^{\boldsymbol{\eta}}\mathbf{R}^s_{k+1}$. In fact, the sensor node state estimate does not have zero mean, white, and Gaussian errors. Because it is the output of an on-board estimator, it will have autocorrelation errors. However, these have been shown to be small for similar applications [39]. Further, assuming that $\boldsymbol{\eta}^s_{k+1}$ is white is equivalent to assuming white noise in the measurement equation (Eqn. 5.2), which has been used in many instances with success [62, 46]. Additionally, it may not be zero mean due to alignment and turret mounting errors (biases), which enter the problem in a similar manner. It is well known that estimation accuracy can be improved via bias estimation [37, 45, 66]. However, the inclusion of bias models is beyond the scope of this work as communication issues are the main focus here.

Let $\mathbb{Z}^{\mathbb{S}}_{k+1}$ denote the set of all measurements from all sensor nodes up to time step $k+1$ such that

$$\mathbb{Z}^{\mathbb{S}}_{k+1} = \{\mathbf{z}^1_1, \ldots, \mathbf{z}^1_{k+1}, \ldots, \mathbf{z}^s_1, \ldots, \mathbf{z}^s_{k+1}\} \tag{5.5}$$

The goal of decentralized cooperative estimation is to utilize all measurements, $\mathbb{Z}^{\mathbb{S}}_{k+1}$, to develop an accurate representation of the probability density function of the state of interest, $p[\mathbf{x}_{k+1}|\mathbb{Z}^{\mathbb{S}}_{k+1}]$, on each sensor node. Note that frequently, only the first and second moments of $p[\mathbf{x}_{k+1}|\mathbb{Z}^{\mathbb{S}}_{k+1}]$ are maintained, denoted here as estimate mean, $\widehat{\mathbf{x}}_{k+1}$, and estimate error covariance, $\mathbf{P}_{k+1}$, respectively. Furthermore, for decentralization, the development of $p[\mathbf{x}_{k+1}|\mathbb{Z}^{\mathbb{S}}_{k+1}]$ must utilize the computational capabilities of each sensor node rather than a central processing node.

The Information Filter (IF), used as a basis here, was presented by Rao, Durrant-Whyte, and Sheen in Ref. [50] as the optimal solution to the linear decentralized estimation problem. Mutambara presented the IF and Extended Information Filter (EIF) for nonlinear systems in the widely used form in Ref. [43], and further showed that the EIF is algebraically equivalent to the Extended Kalman Filter (EKF). More recently, information filtering concepts were used to develop a more accurate and robust filter for nonlinear systems, termed the Square-Root, Sigma Point Information Filter (SR-SPIF) in Ref. [11].

The information matrix, $\mathbf{Y}_k$, and information state, $\mathbf{y}_k$, are defined based on the state estimate error covariance, $\mathbf{P}_k$, and state estimate, $\widehat{\mathbf{x}}_k$, as

$$\mathbf{Y}_k = \mathbf{P}_k^{-1} \tag{5.6}$$

$$\mathbf{y}_k = \mathbf{Y}_k \cdot \widehat{\mathbf{x}}_k \tag{5.7}$$

The EIF algorithm can then be written for n_s sensor nodes, as computed on a local node l, with a recursion of the following five steps:

1. Time Propagation

$$\mathbf{Y}_{k+1}^- = (\mathbf{F}_k)^{-T}\mathbf{Y}_k(\mathbf{F}_k)^{-1}(\mathcal{I} + \mathbf{\Gamma}_k\mathbf{Q}_k(\mathbf{\Gamma}_k)^T(\mathbf{F}_k)^{-T}\mathbf{Y}_k(\mathbf{F}_k)^{-1})^{-1} \tag{5.8}$$

$$\mathbf{y}_{k+1}^- = \mathbf{Y}_{k+1}^- \cdot \mathbf{f}[(\mathbf{Y}_k)^{-1}\mathbf{y}_k, \mathbf{0}], \tag{5.9}$$

where $\mathcal{I}$ denotes the identity matrix and

$$\mathbf{F}_k = \nabla_{\mathbf{x}}\mathbf{f}[\mathbf{x}, \mathbf{w}]|_{\mathbf{x}=(\mathbf{Y}_k)^{-1}\mathbf{y}_k, \mathbf{w}=\mathbf{0}} \tag{5.10}$$

$$\mathbf{\Gamma}_k = \nabla_{\mathbf{w}}\mathbf{f}[\mathbf{x}, \mathbf{w}]|_{\mathbf{x}=(\mathbf{Y}_k)^{-1}\mathbf{y}_k, \mathbf{w}=\mathbf{0}} \tag{5.11}$$

2. Compute Local Information State and Matrix Updates

$$\begin{aligned}
\mathbf{i}^l_{k+1} &= ({}^{\mathbf{x}}\mathbf{H}^l_{k+1})^T({}^{\mathbf{v}}\mathbf{H}^l_{k+1}{}^{\mathbf{v}}\mathbf{R}^l_{k+1}({}^{\mathbf{v}}\mathbf{H}^l_{k+1})^T + {}^{\boldsymbol{\psi}}\mathbf{H}^l_{k+1}{}^{\boldsymbol{\psi}}\mathbf{R}^l_{k+1}({}^{\boldsymbol{\psi}}\mathbf{H}^l_{k+1})^T)^{-1} \cdot \\
&\quad (\mathbf{z}^l_{k+1} - \mathbf{h}^l[(\mathbf{Y}^-_{k+1})^{-1}\mathbf{y}^-_{k+1}, \widehat{\boldsymbol{\psi}}^l_{k+1}, \mathbf{0}] + {}^{\mathbf{x}}\mathbf{H}^l_{k+1}(\mathbf{Y}^-_{k+1})^{-1}\mathbf{y}^-_{k+1}) \qquad (5.12) \\
\mathbf{I}^l_{k+1} &= \left({}^{\mathbf{x}}\mathbf{H}^l_{k+1}\right)^T ({}^{\mathbf{v}}\mathbf{H}^l_{k+1}{}^{\mathbf{v}}\mathbf{R}^l_{k+1}({}^{\mathbf{v}}\mathbf{H}^l_{k+1})^T + {}^{\boldsymbol{\eta}}\mathbf{H}^l_{k+1}{}^{\boldsymbol{\eta}}\mathbf{R}^l_{k+1}({}^{\boldsymbol{\eta}}\mathbf{H}^l_{k+1})^T)^{-1}{}^{\mathbf{x}}\mathbf{H}^l_{k+1} \qquad (5.13)
\end{aligned}$$

where

$$\begin{aligned}
{}^{\mathbf{x}}\mathbf{H}^l_{k+1} &= \nabla_{\mathbf{x}}\mathbf{h}^l[\mathbf{x}, \widehat{\boldsymbol{\psi}} + \boldsymbol{\eta}, \mathbf{v}]|_{\mathbf{x}=(\mathbf{Y}^-_{k+1})^{-1}\mathbf{y}^-_{k+1}, \boldsymbol{\eta}=\mathbf{0}, \mathbf{v}=\mathbf{0}} \qquad (5.14) \\
{}^{\mathbf{v}}\mathbf{H}^l_{k+1} &= \nabla_{\mathbf{v}}\mathbf{h}^l[\mathbf{x}, \widehat{\boldsymbol{\psi}} + \boldsymbol{\eta}, \mathbf{v}]|_{\mathbf{x}=(\mathbf{Y}^-_{k+1})^{-1}\mathbf{y}^-_{k+1}, \boldsymbol{\eta}=\mathbf{0}, \mathbf{v}=\mathbf{0}} \qquad (5.15) \\
{}^{\boldsymbol{\eta}}\mathbf{H}^l_{k+1} &= \nabla_{\boldsymbol{\eta}}\mathbf{h}^l[\mathbf{x}, \widehat{\boldsymbol{\psi}} + \boldsymbol{\eta}, \mathbf{v}]|_{\mathbf{x}=(\mathbf{Y}^-_{k+1})^{-1}\mathbf{y}^-_{k+1}, \boldsymbol{\eta}=\mathbf{0}, \mathbf{v}=\mathbf{0}} \qquad (5.16)
\end{aligned}$$

3. Transmit Local Information Updates, $\mathbf{i}^l_{k+1}$ and $\mathbf{I}^l_{k+1}$, to all Sensor Nodes, $s \in \mathbb{S} \setminus \{l\}$

4. Receive Information Updates, $\mathbf{i}^s_{k+1}$ and $\mathbf{I}^s_{k+1}$, from all Sensor Nodes, $s \in \mathbb{S}\setminus\{l\}$

5. Fuse Local and Received Information Updates

$$\begin{aligned}
\mathbf{Y}_{k+1} &= \mathbf{Y}^-_{k+1} + \mathbf{I}^l_{k+1} + \sum_{s\in\mathbb{S}\setminus\{l\}} \mathbf{I}^s_{k+1} \qquad (5.17) \\
\mathbf{y}_{k+1} &= \mathbf{y}^-_{k+1} + \mathbf{i}^l_{k+1} + \sum_{s\in\mathbb{S}\setminus\{l\}} \mathbf{i}^s_{k+1} \qquad (5.18)
\end{aligned}$$

Note that when the system is linear, Equations 5.1 and 5.2 are linear, and the system matrices: $\mathbf{F}_k$, $\boldsymbol{\Gamma}_k$, ${}^{\mathbf{x}}\mathbf{H}^l_{k+1}$, ${}^{\mathbf{v}}\mathbf{H}^l_{k+1}$, ${}^{\boldsymbol{\eta}}\mathbf{H}^l_{k+1}$ are known and independent of the state. For a linear system, the EIF reduces to the IF, which simplifies the information state prediction step, Equation 5.9, to be

$$\begin{aligned}
\mathbf{L}_{k+1} &= \mathbf{Y}^-_{k+1}\mathbf{F}_k\mathbf{Y}^{-1}_k \qquad (5.19) \\
\mathbf{y}^-_{k+1} &= \mathbf{L}_{k+1}\mathbf{y}_k, \qquad (5.20)
\end{aligned}$$

where $\mathbf{L}_{k+1}$ is termed the information state propagation matrix. In addition, computation of the information state update, $\boldsymbol{i}^l_{k+1}$, in Equation 5.12, simplifies to be

$$\mathbf{i}^l_{k+1} = ({}^{\mathbf{x}}\mathbf{H}^l_{k+1})^T({}^{\mathbf{v}}\mathbf{H}^l_{k+1}{}^{\mathbf{v}}\mathbf{R}^l_{k+1}({}^{\mathbf{v}}\mathbf{H}^l_{k+1})^T + {}^{\boldsymbol{\eta}}\mathbf{H}^l_{k+1}{}^{\boldsymbol{\eta}}\mathbf{R}^l_{k+1}({}^{\boldsymbol{\eta}}\mathbf{H}^l_{k+1})^T)^{-1}\mathbf{z}^l_{k+1} \quad (5.21)$$

5.4 Separable Extended Information Filter

An alternative form of the EIF is now presented, termed the Separable EIF (SEIF), which separates the information matrix and state into components corresponding to the a priori information and information contributed by each of the sensor nodes. The full derivation of the SEIF is given in Appendix A and a summary is presented here. The SEIF is summarized in the following six steps as implemented on node l. The cumulative information variables of the SEIF are initialized as $\widetilde{\mathbf{y}}_k = \mathbf{y}_k$, $\widetilde{\mathbf{i}}^l_k = \mathbf{0}$, $\widetilde{\mathbf{Y}}_k = \mathbf{Y}_k$, and $\widetilde{\mathbf{I}}^l_k = \mathbf{0}$.

1. Compute discount factor $\mathbf{D}_k$

$$\mathbf{D}_k = (\mathcal{I} + \boldsymbol{\Gamma}_k\mathbf{Q}_k(\boldsymbol{\Gamma}_k)^T(\mathbf{F}_k)^{-T}\mathbf{Y}_k(\mathbf{F}_k)^{-1})^{-1} \quad (5.22)$$

2. Time Propagation

$$\begin{aligned}
\mathbf{Y}^-_{k+1} &= (\mathbf{F}_k)^{-T}\mathbf{Y}_k(\mathbf{F}_k)^{-1}\mathbf{D}_k & (5.23)\\
\mathbf{L}_{k+1} &= \mathbf{Y}^-_{k+1}\boldsymbol{\Phi}_k(\mathbf{Y}_k)^{-1} & (5.24)\\
\widetilde{\mathbf{Y}}_{k+1} &= (\mathbf{F}_k)^{-T}\widetilde{\mathbf{Y}}_k(\mathbf{F}_k)^{-1}\mathbf{D}_k & (5.25)\\
\widetilde{\mathbf{y}}_{k+1} &= \mathbf{L}_{k+1}\widetilde{\mathbf{y}}_k & (5.26)
\end{aligned}$$

3. Compute local information updates and package into cumulative information

updates

$$\mathbf{I}_{k+1}^{l} = \left({}^{\mathbf{x}}\mathbf{H}_{k+1}^{l}\right)^{T}\left({}^{\mathbf{v}}\mathbf{H}_{k+1}^{l}\,{}^{\mathbf{v}}\mathbf{R}_{k+1}^{l}({}^{\mathbf{v}}\mathbf{H}_{k+1}^{l})^{T} + {}^{\boldsymbol{\eta}}\mathbf{H}_{k+1}^{l}\,{}^{\boldsymbol{\eta}}\mathbf{R}_{k+1}^{l}({}^{\boldsymbol{\eta}}\mathbf{H}_{k+1}^{l})^{T}\right)^{-1}\,{}^{\mathbf{x}}\mathbf{H}_{k+1}^{l} \quad (5.27)$$

$$\begin{aligned}\mathbf{i}_{k+1}^{l} = {} & ({}^{\mathbf{x}}\mathbf{H}_{k+1}^{l})^{T}\left({}^{\mathbf{v}}\mathbf{H}_{k+1}^{l}\,{}^{\mathbf{v}}\mathbf{R}_{k+1}^{l}({}^{\mathbf{v}}\mathbf{H}_{k+1}^{l})^{T} + {}^{\boldsymbol{\eta}}\mathbf{H}_{k+1}^{l}\,{}^{\boldsymbol{\eta}}\mathbf{R}_{k+1}^{l}({}^{\boldsymbol{\eta}}\mathbf{H}_{k+1}^{l})^{T}\right)^{-1}\cdot \\ & (\mathbf{z}_{k+1}^{l} - \mathbf{h}^{l}[(\mathbf{Y}_{k+1}^{-})^{-1}\mathbf{y}_{k+1}^{-}, \widehat{\boldsymbol{\psi}}_{k+1}^{l}, \mathbf{0}] + {}^{\mathbf{x}}\mathbf{H}_{k+1}^{l}(\mathbf{Y}_{k+1}^{-})^{-1}\mathbf{y}_{k+1}^{-}) \end{aligned} \quad (5.28)$$

$$\widetilde{\mathbf{I}}_{k+1}^{l} = (\mathbf{F}_{k})^{-T}\widetilde{\mathbf{I}}_{k}^{l}(\mathbf{F}_{k})^{-1}\mathbf{D}_{k} + \mathbf{I}_{k+1}^{l} \quad (5.29)$$

$$\widetilde{\mathbf{i}}_{k+1}^{l} = \mathbf{L}_{k+1}\widetilde{\mathbf{i}}_{k}^{l} + \mathbf{i}_{k+1}^{l} \quad (5.30)$$

4. Transmit Local Cumulative Information Updates, $\widetilde{\mathbf{i}}_{k+1}^{l}$ and $\widetilde{\mathbf{I}}_{k+1}^{l}$, to all Sensor Nodes, $s \in \mathbb{S} \setminus \{l\}$

5. Receive Cumulative Information Updates, $\widetilde{\mathbf{i}}_{k+1}^{s}$ and $\widetilde{\mathbf{I}}_{k+1}^{s}$, from all Sensor Nodes, $s \in \mathbb{S} \setminus \{l\}$

6. Fuse Local and Received Cumulative Information Updates

$$\mathbf{y}_{k+1} = \widetilde{\mathbf{y}}_{k+1} + \widetilde{\mathbf{i}}_{k+1}^{l} + \sum_{s \in \mathbb{S}\setminus\{l\}} \widetilde{\mathbf{i}}_{k+1}^{s} \quad (5.31)$$

$$\mathbf{Y}_{k+1} = \widetilde{\mathbf{Y}}_{k+1} + \widetilde{\mathbf{I}}_{k+1}^{l} + \sum_{s \in \mathbb{S}\setminus\{l\}} \widetilde{\mathbf{I}}_{k+1}^{s} \quad (5.32)$$

There are two key parts of the SEIF. First, is the creation of the terms, $\widetilde{\mathbf{Y}}$, $\widetilde{\mathbf{I}}$, $\widetilde{\mathbf{y}}$, and $\widetilde{\mathbf{i}}$, which account for the information accumulated by each sensor node ($\widetilde{\mathbf{I}}$, $\widetilde{\mathbf{i}}$) and the a priori information ($\widetilde{\mathbf{Y}}$, $\widetilde{\mathbf{y}}$). Second, is the transition matrix, $\boldsymbol{\Phi}_{k}$, shown in Equation 5.24, which is used to propagate the cumulative information state and update forward in time. If the state dynamics are linear, then the transition matrix, $\boldsymbol{\Phi}_{k} = \mathbf{F}_{k}$. Appendix B shows how to compute the transition matrix for nonlinear state dynamics.

Note that no approximations are made in the development of the SEIF from the EIF, and therefore, the SEIF and EIF are algebraically equivalent. However, the

computation for the SEIF is a bit higher due to the prediction of the cumulative information variables. The utility of the SEIF will be made more clear in Section 5.5 when implemented in the context of a communication loss.

5.5 Predicted Information Method

The Predicted Information (PI) method presented here is derived based on the separable formulation of the EIF presented in Section 5.4. The explicit separation of the information into contributions from prior information and from each sensor node makes it possible to combine information that was not communicated, due to a communication loss, into a single, approximate cumulative update. In this formulation, during a communication loss, two filters are used:

1. an EIF, which fuses only local information and acts as the current local estimator during the communication loss
2. a PI filter, which is used to quickly update the local estimator with information from other nodes when communication is restored.

Consider trying to implement the SEIF in the case of an $m-1$ step complete communication loss for all sensors between time steps k and $k+m$. A complete communication loss is one in which no sensor node is able to communicate with any other sensor node, which is the worst case example. During the communication loss, the fusion step of the SEIF (step 6 in Section 5.4) cannot be performed because the transmission and reception steps (steps 4 and 5 in Section 5.4) are not completed. Without the fusion step, the first three steps also fail at the next time iteration because they require the information matrix, $\mathbf{Y}_{k+1}$. It is proposed

here that the information matrix $\mathbf{Y}_{k+1}$ is estimated as $\widehat{\mathbf{Y}}^l_{k+1}$ by predicting the information *matrix* updates that should have been received from each sensor node, $[\widehat{\mathbf{I}}^s_{k+1}]^l \approx \mathbf{I}^s_{k+1}$ (explored in Section 5.5-5.5.1), and using the standard EIF fusion step. This gives

$$\mathbf{Y}_{k+1} \approx \widehat{\mathbf{Y}}^l_{k+1} = \widehat{\mathbf{Y}}^{l-}_{k+1} + \mathbf{I}^l_{k+1} + \sum_{s \in \mathbb{S} \setminus \{l\}} [\widehat{\mathbf{I}}^s_{k+1}]^l, \tag{5.33}$$

where the use of a hat over a variable here denotes an estimated or approximate value. The SEIF can then proceed during the communication loss with the same recursion but using the predicted information. Note that during the communication loss, the other variables, $\mathbf{D}_k$, $\mathbf{L}_{k+1}$, $\widetilde{\mathbf{Y}}_{k+1}$, $\widetilde{\mathbf{y}}_{k+1}$, $\widetilde{\mathbf{I}}_{k+1}$, and $\widetilde{\mathbf{i}}_{k+1}$ are approximate due to their dependence on the information matrix, $\mathbf{Y}_{k+1}$, and are denoted with a hat as $\widehat{\mathbf{D}}_k$, $\widehat{\mathbf{L}}_{k+1}$, $\widehat{\widetilde{\mathbf{Y}}}_{k+1}$, $\widehat{\widetilde{\mathbf{y}}}_{k+1}$, $\widehat{\widetilde{\mathbf{I}}}_{k+1}$, and $\widehat{\widetilde{\mathbf{i}}}_{k+1}$, respectively. It is important to note that Predicted Information refers to the prediction of the uncertainty about a measurement, given by $\mathbf{I}$, not prediction of the measurements themselves. This approximation allows for the use of the SEIF recursion which efficiently accumulates local information into cumulative information updates which can be communicated and fused easily.

During a communication loss, the PI method consists of three parts: 1. Initialization, 2. Propagation During Loss, 3. Communication and Fusion. The three parts are shown below, as applied on local node l, for an $m-1$ step communication loss between time steps k and $k+m$.

1. Initialization

   When communication is lost at time step $k+1$ (last completed communication at time step k), the PI filter is initialized from the current EIF state as $\widehat{\widetilde{\mathbf{Y}}}^l_k = \mathbf{Y}_k$, $\widehat{\widetilde{\mathbf{I}}}^l_k = \mathbf{0}$, $\widehat{\widetilde{\mathbf{y}}}^l_k = \mathbf{y}_k$, $\widehat{\widetilde{\mathbf{i}}}^l_k = \mathbf{0}$, and $\widehat{\mathbf{Y}}^l_k = \mathbf{Y}_k$.

2. Propagation During Loss

   While communication is unavailable, two sets of recursions are used. The first is the already running EIF, which fuses only local updates and is used to develop: $\mathbf{F}_j^l$, $\mathbf{\Gamma}_j^l$, ${}^{\mathbf{x}}\mathbf{H}_j^l$, ${}^{\mathbf{v}}\mathbf{H}_j^l$, ${}^{\boldsymbol{\eta}}\mathbf{H}_j^l$, $\mathbf{\Phi}_j^l$, $\mathbf{i}_j^l$, and $\mathbf{I}_j^l$. Second, is the PI filter, shown in Figure 5.1.

3. Communication and Fusion

   Once communication is available, the cumulative information updates, $\widehat{\widetilde{\mathbf{I}}}_{k+m}^{\,l}$, and $\widehat{\widetilde{\mathbf{i}}}_{k+m}^{\,l}$, are transmitted to all other nodes, $s \in \mathbb{S}$, $\widehat{\widetilde{\mathbf{I}}}_{k+m}^{\,s}$, and $\widehat{\widetilde{\mathbf{i}}}_{k+m}^{\,s}$ are received from all other nodes, $s \in \mathbb{S}$. At local node l, all received cumulative updates at time step $k+m$ are then fused according to

$$\mathbf{Y}_{k+m}^l = \widehat{\widetilde{\mathbf{Y}}}_{k+m}^{\,l} + \sum_{s \in \mathbb{S}\backslash\{l\}} \widehat{\widetilde{\mathbf{I}}}_{k+m}^{\,s} \tag{5.34}$$

$$\mathbf{y}_{k+m}^l = \widehat{\widetilde{\mathbf{y}}}_{k+m}^{\,l} + \sum_{s \in \mathbb{S}\backslash\{l\}} \widehat{\widetilde{\mathbf{i}}}_{k+m}^{\,s} \tag{5.35}$$

5.5.1 Predicting Information Updates

The development of the predicted information matrix updates, $[\widehat{\mathbf{I}}_{j+1}^s]^l$, is dependent on the sensor node state dynamics in Equation 5.3. Two different approaches are used in this work. The first approach, termed Estimate Information (EI), estimates the information matrix updates directly from estimates of the states of the cooperating sensor nodes at each time step as

$$[\widehat{\mathbf{I}}_{j+1}^s]_{\mathrm{EI}}^l = \left([{}^{\mathbf{x}}\widehat{\mathbf{H}}_{j+1}^s]^l\right)^T \left([{}^{\mathbf{v}}\widehat{\mathbf{H}}_{j+1}^s]^l {}^{\mathbf{v}}\mathbf{R}_{j+1}^s ([{}^{\mathbf{v}}\widehat{\mathbf{H}}_{j+1}^s]^l)^T + [{}^{\boldsymbol{\eta}}\widehat{\mathbf{H}}_{j+1}^s]^l {}^{\boldsymbol{\eta}}\mathbf{R}_{j+1}^s ([{}^{\boldsymbol{\eta}}\widehat{\mathbf{H}}_{j+1}^s]^l)^T\right)^{-1} [{}^{\mathbf{x}}\widehat{\mathbf{H}}_{j+1}^s]^l, \tag{5.36}$$

where the measurement function linearizations, ${}^{\mathbf{x}}\widehat{\mathbf{H}}_{j+1}^s$, ${}^{\mathbf{v}}\widehat{\mathbf{H}}_{j+1}^s$, and ${}^{\boldsymbol{\eta}}\widehat{\mathbf{H}}_{j+1}^s$ are estimated on local node l based on local estimates as $[{}^{(\cdot)}\mathbf{H}_{j+1}^s]^l = \nabla_{(\cdot)}\mathbf{h}^s[\mathbf{x}, [\widehat{\boldsymbol{\psi}}_{j+1}^s]^l + \boldsymbol{\eta}, \mathbf{v}]\big|_{\mathbf{x}=(\mathbf{Y}_j^{l-})^{-1}\mathbf{y}_j^{l-}, \boldsymbol{\eta}=\mathbf{0}, \mathbf{v}=\mathbf{0}}$ and sensor node l's estimate of the state of sensor node

Recursion for $j = k, \ldots, k+m$

(a) Compute predicted discount factor $\widehat{\mathbf{D}}_j^l$

$$\widehat{\mathbf{D}}_j^l = (\mathcal{I} + \mathbf{\Gamma}_j^l \mathbf{Q}_j (\mathbf{\Gamma}_j^l)^T (\mathbf{F}_j^l)^{-T} \widehat{\mathbf{Y}}_j^l (\mathbf{F}_j^l)^{-1})^{-1}$$

(b) Propagate cumulative information matrix variables, $\widehat{\widetilde{\mathbf{Y}}}_{j+1}^{\,l}$ and $\widehat{\widetilde{\mathbf{I}}}_{j+1}^{\,l}$

$$\begin{aligned} \widehat{\widetilde{\mathbf{Y}}}_{j+1}^{\,l} &= (\mathbf{F}_j^l)^{-T} \widehat{\widetilde{\mathbf{Y}}}_j^{\,l} (\mathbf{F}_j^l)^{-1} \widehat{\mathbf{D}}_j^l + \mathbf{I}_{j+1}^l \\ \widehat{\widetilde{\mathbf{I}}}_{j+1}^{\,l} &= (\mathbf{F}_j^l)^{-T} \widehat{\widetilde{\mathbf{I}}}_j^{\,l} (\mathbf{F}_j^l)^{-1} \widehat{\mathbf{D}}_j^l + \mathbf{I}_{j+1}^l \end{aligned}$$

(c) Compute time propagation of the predicted information matrix

$$\widehat{\mathbf{Y}}_{j+1}^{l-} = (\mathbf{F}_j^l)^{-T} \widehat{\mathbf{Y}}_j^l (\mathbf{F}_j^l)^{-1} \widehat{\mathbf{D}}_j^l$$

(d) Compute the predicted information state propagation matrix, $\widehat{\mathbf{L}}_{j+1}^l$ and propagate the cumulative information state variables, $\widetilde{\mathbf{y}}_{j+1}^l$ and $\widetilde{\mathbf{i}}_{j+1}^l$

$$\begin{aligned} \widehat{\mathbf{L}}_{j+1}^l &= \widehat{\mathbf{Y}}_{j+1}^{l-} \widehat{\mathbf{\Phi}}_j^l (\widehat{\mathbf{Y}}_j^l)^{-1} \\ \widehat{\widetilde{\mathbf{y}}}_{j+1}^{\,l} &= \widehat{\mathbf{L}}_{j+1}^l \widehat{\widetilde{\mathbf{y}}}_j^{\,l} + \mathbf{i}_{j+1}^l \\ \widehat{\widetilde{\mathbf{i}}}_{j+1}^{\,l} &= \widehat{\mathbf{L}}_{j+1}^l \widehat{\widetilde{\mathbf{i}}}_j^{\,l} + \mathbf{i}_{j+1}^l \end{aligned}$$

(e) Predict $[\widehat{\mathbf{I}}_{j+1}^s]_{(\cdot)}^l$ for all $s \in \mathbb{S} \setminus \{l\}$, where $(\cdot)$ is the prediction method described in Section 5.5.1

(f) Complete the predicted measurement update of the predicted information matrix

$$\widehat{\mathbf{Y}}_{j+1}^l = \widehat{\mathbf{Y}}_{j+1}^{l-} + \mathbf{I}_{j+1}^l + \sum_{s \in \mathbb{S} \setminus \{l\}} [\widehat{\mathbf{I}}_{j+1}^s]_{(\cdot)}^l$$

Figure 5.1: Summary of the PI filter during a complete communication loss of $m-1$ time steps.

s at time step $j+1$ is denoted as $[\widehat{\boldsymbol{\psi}}_{j+1}^s]^l$. The process of directly predicting information matrix updates requires three conditions:

1. Each sensor node must have access to the measurement function for every

other sensor node including sensor noise statistics (i.e. Equation 5.2 must be known by all sensor nodes)

2. Each sensor node must have a model of the dynamics of the sensor node states for every sensor node and navigation system error statistics (Equation 5.3)
3. When communication is available, in addition to transmitting information updates, the sensor nodes must also transmit their sensor node state estimates, $\widehat{\boldsymbol{\psi}}_k^s$

Condition number one is easily met by simply sharing the sensor models among the cooperating sensor nodes. Condition number two is the most uncertain and essentially requires the 'plan' of each sensor node to be known or the state not to vary much. For the UAV example presented here this is reasonable. Condition number three is straight forward but does require a small increase in communication (10 floating point numbers for each UAV in the geolocation example).

An alternative, simpler, approach, termed Constant Information (CI), is to assume that the information matrix updates change slowly over the time of the communication loss, and can be well approximated as constant. This would give the following predicted information matrix update during the communication loss

$$[\widehat{\mathbf{I}}_{k+j}^s]_{\mathrm{CI}}^l = \mathbf{I}_k^s \qquad \forall \quad j \ \in \ \{1, \ldots, m-1\} \tag{5.37}$$

The applicability of the CI approach is problem dependent and the accuracy depends on how quickly the true information matrix updates change. recall that only the information matrix updates are being approximated, which contain the measurement statistics and not the measurements themselves. Both the EI and CI approaches to predicting information matrix updates are compared in Section 6.3 for a cooperative geolocation example with UAVs.

5.5.2 Stability and Performance

The PI method was developed from the separable formulation of the EIF for nonlinear estimation. Since analytic stability proofs for the EIF/EKF are only available in special cases [33], a general analytic stability proof of the PI method is not expected. Therefore, the stability and performance of the PI method is evaluated empirically in Section 5.7. However, in the case of a linear system the performance of the PI method can be shown analytically.

When applying the PI method to the general nonlinear system presented in Section 5.3, there are two sources of error:

1. error in the predicted information matrix updates, i.e. $[\widehat{\mathbf{I}}^s_{j+1}]^l_{(\cdot)} \neq \mathbf{I}^s_{j+1}$
2. incorrect linearization for the system matrices during the communication loss (Equations 5.10, 5.11, 5.14, 5.15, and 5.16)

During a communication loss there is no feedback and in general the errors will continue to accumulate. However, for a linear system, where all system matrices are not sensor state dependent, both sources of error are eliminated. The linearization error is removed because the system is already linear. Furthermore, since the measurement functions are known by all sensor nodes and are not dependent on the states of either the POI or mobile sensor node dynamics, the information matrix updates can be predicted exactly during a communication loss,

$$[\widehat{\mathbf{I}}^s_{k+j}]^l_{\mathrm{EI}} = \mathbf{I}^s_{k+j} \quad \forall \quad j \in \{1, \ldots, m-1\}. \tag{5.38}$$

Thus, for linear systems, the PI method with the EI prediction, is guaranteed to achieve the optimal solution after a communication loss, as if the loss had not occurred for any finite duration communication loss. Further, if the system

matrices are linear and constant, then the PI method with CI prediction is also exact.

5.5.3 Maintaining Equivalent Estimates Across Sensor Nodes

One disadvantage of the PI method is that, after a communication loss, estimates on each sensor node can be different because a different predicted a priori information term, $\widehat{\widetilde{\mathbf{Y}}}^{l}$, is computed and used on each local node. A slight modification can be made to obtain equivalent estimates across sensor nodes by also communicating $\widetilde{\mathbf{y}}$ and $\widetilde{\mathbf{Y}}$ to all other nodes, and performing a covariance intersection step [29] in place of Equations 5.34 and 5.35. Formally, this is written as

$$\mathbf{Y}_{k+1}^{l} = \sum_{s\in\mathbb{S}} \left(\omega^{s} \widehat{\widetilde{\mathbf{Y}}}_{k+1}^{s} + \widehat{\widetilde{\mathbf{I}}}_{k+1}^{s} \right) \tag{5.39}$$

$$\mathbf{y}_{k+1}^{l} = \sum_{s\in\mathbb{S}} \left(\omega^{s} \widehat{\widetilde{\mathbf{y}}}_{k+1}^{s} + \widehat{\widetilde{\mathbf{i}}}_{k+1}^{s} \right), \tag{5.40}$$

where the ω^{s} parameters are chosen to maximize $\mathrm{Det}[\mathbf{Y}_{k+1}^{l}]$ subject to the constraints:

$$\omega^{s} \in [0,1] \quad \forall \ s \in \mathbb{S} \tag{5.41}$$

$$\sum_{s\in\mathbb{S}} \omega^{s} = 1. \tag{5.42}$$

However, significant computation and communication can be saved by forgoing the optimization step and instead using an equal weighting, $\omega^{s} = \frac{1}{n_s}$. This is reasonable because the differences in the predicted a priori information terms are not of statistical significance. The sensor nodes can then transmit a weighted sum of their a priori information and cumulative information terms as a single update.

The fusion step then takes the form,

$$\mathbf{Y}_{k+1}^{l} = \left(\frac{1}{n_s}\widehat{\widetilde{\mathbf{Y}}}_{k+1}^{l} + \widehat{\widetilde{\mathbf{I}}}_{k+1}^{l}\right) + \sum_{s\in\mathbb{S}\setminus\{l\}}\left(\frac{1}{n_s}\widehat{\widetilde{\mathbf{Y}}}_{k+1}^{s} + \widehat{\widetilde{\mathbf{I}}}_{k+1}^{s}\right) \tag{5.43}$$

$$\mathbf{y}_{k+1}^{l} = \left(\frac{1}{n_s}\widehat{\widetilde{\mathbf{y}}}_{k+1}^{l} + \widehat{\widetilde{\mathbf{i}}}_{k+1}^{l}\right) + \sum_{s\in\mathbb{S}\setminus\{l\}}\left(\frac{1}{n_s}\widehat{\widetilde{\mathbf{y}}}_{k+1}^{s} + \widehat{\widetilde{\mathbf{i}}}_{k+1}^{s}\right). \tag{5.44}$$

5.5.4 General Communication Loss Scenarios

Frequently in practice, communication losses affect only a subset of the nodes and further, are not always symmetric. Define $\mathbb{R}_{k+1}^{l}$ to be the set of all remote nodes from which node l has received updates at time step $k+1$. Also, Define $\mathbb{T}_{k+1}^{l}$ to be the set of all remote nodes to which node l has transmitted updates at time step $k+1$. If either $\mathbb{R}_{k+1}^{l}$ or $\mathbb{T}_{k+1}^{l}$ or both are proper subsets of $\mathbb{S}\setminus\{l\}$, then a communication loss has occurred. The previously described PI method can still be used, but in a modified form. In these cases, cumulative information updates must be accumulated for *each* of the communication links that are lost. The changes to the PI method are summarized as follows, from the perspective of node l.

1. Initialization now includes cumulative information updates for each of the other nodes
   $\widehat{\widetilde{\mathbf{I}}}_{k}^{l} = \mathbf{0} \Rightarrow \widehat{\widetilde{\mathbf{I}}}_{k}^{l,s} = \mathbf{0}$ and $\widehat{\widetilde{\mathbf{i}}}_{k}^{l} = \mathbf{0} \Rightarrow \widehat{\widetilde{\mathbf{i}}}_{k}^{l,s} = \mathbf{0}$ for $s \in \mathbb{S}\setminus\{l\}$
2. Transmit cumulative information updates to all available nodes $t \in \mathbb{T}_{j+1}^{l}$ and reset cumulative information updates to zero after successful transmission
   $\widehat{\widetilde{\mathbf{I}}}_{k}^{l,t} = \mathbf{0}$ and $\widehat{\widetilde{\mathbf{i}}}_{k}^{l,t} = \mathbf{0}$ for $t \in \mathbb{T}_{j+1}^{l}$
3. Receive cumulative information updates from all available nodes $r \in \mathbb{R}_{j+1}^{l}$

and fuse as

$$\begin{aligned}\widehat{\widehat{\mathbf{Y}}}_{j+1}^{l} &= \widehat{\widehat{\mathbf{Y}}}_{j+1}^{l} + \sum_{r\in\mathbb{R}_{j+1}^{l}} \widehat{\widehat{\mathbf{I}}}_{j+1}^{r,l} \\ \widehat{\widehat{\mathbf{y}}}_{j+1}^{l} &= \widehat{\widehat{\mathbf{y}}}_{j+1}^{l} + \sum_{r\in\mathbb{R}_{j+1}^{l}} \widehat{\widehat{\mathbf{i}}}_{j+1}^{r,l}\end{aligned}$$

Note that step 2 includes a resetting of the cumulative information updates to $\mathbf{0}$ after a successful transmission, to prevent double counting of information. This is equivalent to keeping cumulative information updates only for the nodes that lost communication and using the standard EIF updates, $\mathbf{I}_k$ and $\mathbf{i}_k$, for the nodes still in communication.

5.6 Benchmark Methods for Communication Loss in Decentralized Estimation

The PI method developed in Section 5.5 is compared to three benchmark methods for decentralized nonlinear estimation from the literature. The benchmark methods vary in their trade between estimator accuracy, communication requirements, memory storage, and computational complexity.

5.6.1 Drop Information

Perhaps the simplest approach, referred to as Drop Information (DI), is to disregard and not use any information updates that do not arrive in time. This approach is implied in one of the original developments of the IF [50] for decentralized data fusion, where sensor nodes are specifically allowed to be added/subtracted from

the sensor network. Specifically, the DI method differs from the EIF/IF algorithm described in Section 5.3 only in the fusion step, Equations 5.17 and 5.18. Recall, $\mathbb{R}^l_{k+1}$ is the set of all remote nodes from which node l has received updates at time step $k+1$; during a communication loss, $\mathbb{R}^l_{k+1}$ is a proper subset of $\mathbb{S} \setminus \{l\}$. The fusion step of the DI method is then written as

$$\mathbf{y}^l_{k+1} = \mathbf{y}^{l-}_{k+1} + \mathbf{i}^l_{k+1} + \sum_{r \in \mathbb{R}^l_{k+1}} \mathbf{i}^r_{k+1} \tag{5.45}$$

$$\mathbf{Y}^l_{k+1} = \mathbf{Y}^{l-}_{k+1} + \mathbf{I}^l_{k+1} + \sum_{r \in \mathbb{R}^l_{k+1}} \mathbf{I}^r_{k+1} \tag{5.46}$$

The DI method has a number of advantages. First, the DI method is very modular since no sensor node must know anything about the communication topology. Second, communication losses have no effect on the required memory storage or the size of the information transmissions. Finally, the computation required in the DI method at each sensor node actually *decreases* when communication is lost because fewer updates are fused. The biggest disadvantage of the DI method is estimation accuracy. Dropping information, as the name implies, leads to suboptimal estimates which may be significant if the network is lossy. Furthermore, the sensor nodes do not maintain the same estimate.

5.6.2 Store and Burst

The Store and Burst (SB) method presented here is a general version of the method developed in Ref. [44], where computational simplifications were made specifically for linear systems. The SB method requires each sensor node to save their local information state, $\mathbf{y}$, and information matrix, $\mathbf{Y}$, from the last time step where all information updates were available, i.e. before the communication loss. Then, when new updates are made available through the communication network, the

filter iterates from this step to the most recent time.

To understand how the SB method works, consider an $m-1$ time step complete communication loss, on sensor node l, where communication is available at time step k and not again until time step $k+m$. The SB method during a communication loss can be summarized in three steps: 1. Initialization - beginning of communication loss, 2. Storage - during communication loss, 3. Burst and Fusion - after communication is restored.

1. Initialization

   When communication is lost at time step $k+1$, the information state and matrix, $\mathbf{y}_k^l$, and $\mathbf{Y}_k^l$, from time step k are stored in memory.

2. Storage

   While communication is lost, two steps occur. First, the SB method uses the prediction step of the EIF, Equations 5.8 and 5.9 and fuses only locally available information updates, $\mathbf{i}_{k+j}^l$ and $\mathbf{I}_{k+j}^l$ for $j \in \{1, \ldots, m-1\}$, as computed in Equations 5.12 and 5.13. Second, during the communication loss, local information updates, $\mathbf{I}_j^l$ and $\mathbf{i}_j^l$ for $j \in \{1, \ldots, m-1\}$, are stored for later transmission when communication is restored.

3. Burst and Fusion

   After communication is restored at time step $k+m$, node l sends *all* saved information updates,

   $\{\boldsymbol{i}_{k+1}^l, \ldots, \boldsymbol{i}_{k+m}^l, \boldsymbol{I}_{k+1}^l, \ldots, \boldsymbol{I}_{k+m}^l\}$, to all other nodes, and receives information updates from all cooperating sensor nodes, $\{\boldsymbol{i}_{k+1}^s, \ldots, \boldsymbol{i}_{k+m}^s, \boldsymbol{I}_{k+1}^s, \ldots, \boldsymbol{I}_{k+m}^s\}$ for all $s \in \mathbb{S} \setminus \{l\}$, in a single burst. Each node then runs a *full* EIF recursion from time step k to $k+m$ (Equations 5.8 to 5.18 using all saved and received information updates.

The biggest advantage of the SB method is accuracy. For a linear system, the SB method recovers the optimal solution, as if communication was not lost, once all updates have been received at each node. For nonlinear systems, the accuracy is typically still quite good, although not optimal because of linearization errors. Note that optimal is used here to mean that the estimate and uncertainty, after communication is restored, return immediately to the estimate and uncertainty that would have come with no communication loss at all. The biggest disadvantage of the SB method is the memory, communication, and computation required. Specifically, the SB method requires large memory during the communication loss to store the information updates, as well as sudden bursts of both communication and computation when communication is restored.

5.6.3 Hybrid State Fusion

State fusion is a cooperative estimation approach where first, only local measurements are used to develop a local state estimate; then, the *state* estimates are then shared amongst all nodes and fused to create a new state estimate. State fusion can even be used to combine estimates from more complicated estimation algorithms such as Gaussian Sum Filters [28]. In Ref. [29], Julier and Uhlmann present a method, known as Covariance Intersection (CovI), as a statistically consistent approach for fusing estimates at a given time with unknown correlation. Covariance Intersection uses a convex combination of the state estimates and covariances to provide an updated state estimate and covariance. Covariance Intersection can naturally be expressed as a convex combination of the IF variables. For example,

simultaneously fusing state estimates from n_s nodes yields

$$\mathbf{Y}_k^{\text{CovI}} = \sum_{s=1}^{n_s} \omega^s \mathbf{Y}_k^s \tag{5.47}$$

$$\mathbf{y}_k^{\text{CovI}} = \sum_{s=1}^{n_s} \omega^s \mathbf{y}_k^s, \tag{5.48}$$

where the parameters, $\{\omega^1, \ldots, \omega^{n_s}\}$, are typically optimized over some metric, such as

$$\{\omega^1, \ldots, \omega^{n_s}\} = \underset{\{\omega^1,\ldots,\omega^{n_s}\}}{\text{argmax}} \text{Det}\left[\sum_{s}^{n_s} \omega^s \mathbf{Y}_k^s\right], \tag{5.49}$$

subject to the constraints

$$\sum_{s=1}^{n_s} \omega^s = 1 \tag{5.50}$$

$$\omega^s \in [0,1] \quad \text{for} \quad s \in \{1, \ldots, n_s\}. \tag{5.51}$$

Note that using the determinant, in Equation 5.49, is equivalent to minimizing the volume of the updated uncertainty ellipsoid.

State estimates in the recursive filtering domain, by design, accumulate information from a time sequence of measurements. Therefore, it is expected that performing a state fusion after a long communication loss would combine information from the measurements taken during the communication loss. The benefit of state fusion is that each node fuses local information in real time and only shares the current information matrix and state. So, communication and memory requirements are low. However, the downside is that the accuracy is not as good as centralized estimation.

The Hybrid State Fusion (HSF) approach presented here is a slight modification of the Covariance Intersection method. The HSF method combines Covariance Intersection with the measurement fusion of the EIF at the time of communication, and is summarized for local node l in Figure 5.2. During a communication loss, the

HSF functions as the DI approach. When communication resumes, the Covariance Intersection method is used to fuse information accumulated on the other nodes during the loss. When there is no communication loss, the Covariance Intersection step has no effect since each of the nodes has the same information. In this case, the HSF method functions as the original EIF.

1. Prediction

$$\begin{aligned}\mathbf{Y}^{l-}_{k+1} &= (\mathbf{F}^l_k)^{-T}\mathbf{Y}^l_k(\mathbf{F}^l_k)^{-1}(\mathcal{I}+\mathbf{\Gamma}^l_{k+1}\mathbf{Q}_k(\mathbf{\Gamma}^l_{k+1})^T(\mathbf{F}^l_k)^{-T}\mathbf{Y}^l_k(\mathbf{F}^l_k)^{-1})^{-1}\\ \mathbf{y}^{l-}_{k+1} &= \mathbf{Y}^{l-}_{k+1}\mathbf{f}[(\mathbf{Y}^l_k)^{-1}\mathbf{y}^l_k,\mathbf{0}]\end{aligned}$$

2. Transmit $\mathbf{I}^l_{k+1}$, $\mathbf{i}^l_{k+1}$, $\mathbf{Y}^{l-}_{k+1}$, and $\mathbf{y}^{l-}_{k+1}$ to other nodes for $t \in \mathbb{T}^l_{k+1}$

3. Receive $\mathbf{I}^r_{k+1}$, $\mathbf{i}^r_{k+1}$, $\mathbf{Y}^{r-}_{k+1}$, and $\mathbf{y}^{r-}_{k+1}$ from other nodes for $r \in \mathbb{R}^l_{k+1}$

4. State Fusion using Covariance Intersection for all received state estimates

$$\begin{aligned}\mathbf{Y}^{l-}_{k+1} &\leftarrow \omega^l\mathbf{Y}^{l-}_{k+1}+\sum_{r\in\mathbb{R}^l_{k+1}}\omega^r\mathbf{Y}^{r-}_{k+1}\\ \mathbf{y}^{l-}_{k+1} &\leftarrow \omega^l\mathbf{y}^{l-}_{k+1}+\sum_{r\in\mathbb{R}^l_{k+1}}\omega^r\mathbf{y}^{r-}_{k+1}\end{aligned}$$

5. Fuse local and received information updates

$$\begin{aligned}\mathbf{Y}^l_{k+1} &= \mathbf{Y}^{l-}_{k+1}+\mathbf{I}^l_{k+1}+\sum_{r\in\mathbb{R}^l_{k+1}}\mathbf{I}^r_{k+1}\\ \mathbf{y}^l_{k+1} &= \mathbf{y}^{l-}_{k+1}+\mathbf{i}^l_{k+1}+\sum_{r\in\mathbb{R}^l_{k+1}}\mathbf{i}^r_{k+1}\end{aligned}$$

Figure 5.2: Summary of the Hybrid State Fusion method.

5.7 Cooperative Geolocation Using Flight Test Results

A challenging and practically motivated cooperative geolocation example with experimental flight data is used to understand and compare the performance of the PI method, developed in Section 5.5, relative to the three benchmark methods, shown in Section 5.6. Figure 5.3 shows a conceptual scenario for the geolocation problem. Each UAV, based on its position and orientation, points a camera (using a gimballing payload mount inside the UAV) at the Point of Interest (POI) on the ground. While the aircraft is moving (navigation and orientation), and the POI is potentially moving, the camera gimbal must make adjustments to point at the POI. The objective of geolocation is then to estimate the position (2D or 3D) of the POI from the aircraft, gimbal, and camera measurements/estimates. It is assumed here that the POI always remains within the camera's field of view, even while the UAV and POI are moving [10]. Complicating the geolocation problem are uncertainties in the aircraft position and orientation, gimbal angles, camera specifications and measurements, and disturbances such as turbulence and engine vibrations. Each UAV communicates appropriate information to every other UAV to fuse together a cooperative estimate of the POI.

Both experimental flight tests and hardware in the loop (HiL) simulations of the ScanEagle UAVs are used in this example. The ScanEagle UAVs are produced by The Insitu Group with the centerpiece of ScanEagle being a digital video camera integrated into an inertially-stabilized pan / tilt nose turret. The operator can command the camera to pan back-and-forth for wide-area search, or to remain locked onto a POI while the aircraft maneuvers using a local, computer vision based feedback loop [57]; the latter mode is used here. When the user selects a POI for geolocation, the gimballing turret and on-board vision software attempt

Figure 5.3: Overview of the cooperative geolocation problem, showing the tracking Point of Interest (POI) on the ground, with multiple UAVs pointing cameras at the POI.

to maintain the POI in the center of the frame, from frame to frame, with an associated error covariance. Therefore, the 'measurement' of the POI is assumed to be at the center of the image frame. The measurement equations, Equation 5.2, are nonlinear functions of the UAV states and the POI states, which yield the location of the POI in the camera screen in terms of pixels. The full derivation of the measurement equation is developed in Ref. [10], and a summary is presented in Appendix C.

The experimental flight data used for this evaluation was collected during cooperative UAV flight tests on March 16, 2007 [57]. In these tests, two UAVs orbited a stationary POI with an orbit radius of 500m. Furthermore, the UAVs were separated in their orbits by $\sim 90^\circ$ of relative phasing. Since no moving POI data was available for this study, HiL simulations are used to test for a moving POI. The HiL simulator provides a very accurate representation of the true system, including the actual hardware and realistic noises. In fact, the HiL simulator is used by Insitu to validate tests before flight [57, 58]. The motion of the POI is based on city driving model, which is characterized by frequent starts, stops, and 90° turns.

A sample trajectory is shown in Figure 5.4. Note that the units are not included due to ITAR restrictions.

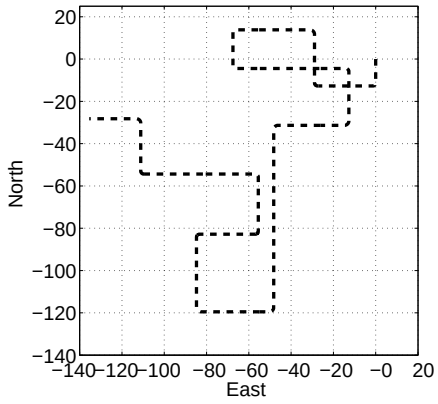


Figure 5.4: Sample moving POI trajectory from the hardware in the loop simulations based on a city driving model.

A linear POI model of the form

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{\Gamma}_k \mathbf{w}_k \tag{5.52}$$

is used for both the stationary and moving POI. However, the state and system matrices are different for the stationary/moving POI cases. For the stationary POI case, the state, $\mathbf{x}_k \in \mathcal{R}^{3\times 1}$, is the POI location in a local North-East-Down (NED) frame and the system matrices are assumed to be

$$\mathbf{F}_k = \mathcal{I}_3, \quad \mathbf{\Gamma}_k = \Delta t \mathcal{I}_3, \tag{5.53}$$

where Δt is the time step, which in this example is set to 0.1 sec. Recall, $\mathcal{I}_3$ denotes the 3×3 identity matrix. The process noise, $\mathbf{w}_k$, is assumed to be zero mean, white, Gaussian distributed with covariance $\mathbf{Q}_k = \alpha^2 \mathcal{I}_3$, where the parameter $\alpha = 0.001$ m/s.

For the maneuvering POI case, the state, $\mathbf{x}_k \in \mathcal{R}^{6\times 1}$, is composed of the position and velocity, both expressed in the local NED frame. The system matrices are given

by

$$\mathbf{F}_k = \begin{bmatrix} \mathcal{I}_3 & \Delta t \mathcal{I}_3 \\ \mathbf{0}_3 & \mathcal{I}_3 \end{bmatrix}, \quad \mathbf{\Gamma}_k = \begin{bmatrix} \mathbf{0}_3 \\ \Delta t \mathcal{I}_3 \end{bmatrix}, \tag{5.54}$$

where $\mathbf{0}_3$ denotes a 3×3 matrix of zeros. The process noise covariance is given by

$$\mathbf{Q}_k = \begin{bmatrix} \alpha^2 & 0 & 0 \\ 0 & \alpha^2 & 0 \\ 0 & 0 & \beta^2 \end{bmatrix}, \tag{5.55}$$

where the parameters are $\alpha = 0.5 \quad \mathrm{m/s^2}$ and $\beta = 0.05 \quad \mathrm{m/s^2}$. Note that β corresponds to vertical accelerations and is smaller than α because ground POIs typically accelerate more horizontally than vertically. For both the stationary and moving POI cases the parameters α and β were chosen based on estimator tuning.

Two realistic communication loss scenarios are considered here.

1. Extended Loss - Full all to all communication is available until the occurrence of an m step complete communication loss. Full communication is available for the remainder of time after the communication loss. The extended loss scenario is representative of cases such as equipment failure, environmental effects, or enemy jamming.
2. Random Drop - At each time step the communication has probability, p, of being lost. The random drop scenario is a good model of communication over a wireless network.

Although, both extended losses and random drops could occur on the same tracking mission, each is considered separately here for clarity of the analysis.

During a communication loss, the EI variation of the PI method requires each UAV to estimate the state (position, attitude, camera attitude, and camera field

of view), of the other UAV at each time step. The ScanEagle UAV has a guidance loop which keeps the UAV flying in an orbit around the POI. This fact is used to propagate the UAV position and heading angle estimate forward in time, by assuming a constant UAV velocity and using the local POI estimate as the orbit center. The remaining UAV states (roll angle, pitch angle, camera attitude, and camera field of view), are estimated as constant.

5.7.1 Single, Extended Communication Loss

Consider first the case of an extended communication loss when locating a stationary POI. For Figure 5.5, the flight data was broken into 250 sets and an $m = 50$ step (5 second) communication loss was imposed from time step 300 to 350 in each set. Figure 5.5 shows the average estimate error over the 250 sets. The estimate error is defined to be the Euclidian distance from the estimate to the true POI location. Due to International Traffic in Arms Regulations (ITAR), the labels from the vertical axis of Figure 5.5 have been removed. Recall that the PI method can be implemented using two approaches for predicting information matrix updates: Constant Information (CI) and Estimate Information (EI). Also, the Full Communication (FC) error is included to provide a point of comparison as the 'best' one could do, but would not be an option during a communication loss.

Even with the short communication loss shown in Figure 5.5, three observations are clear. First, as expected, the estimation error for all methods increases compared to the FC case, due to communication loss. Second, during the communication loss, all methods are equally accurate. Third, the accuracy of each of the methods immediately after communication is restored at time step 351 varies, where CI, EI, and SB all converge immediately to approximately the Full Com-

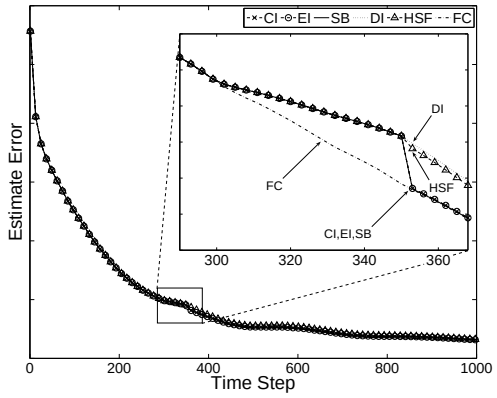


Figure 5.5: Estimate error (distance) averaged of 250 tests with experimental flight data for several cases: Full Communication (FC), two variations of Predicted Information method (CI, EI), and the three benchmark methods (DI, HSF, and SB).

munication estimate.

Immediately after communication is restored is an obvious time to evaluate the estimator accuracy and is the time taken here. However, instead of using the positioning error as a metric, the Kullback-Leibler Divergence (KLD) from the Full Communication estimate is used. The KLD is used in this work for two reasons. First, the KLD takes into account both differences in the estimate mean and the covariance. Second, it captures the accuracy in the case of a moving POI without having to create an arbitrary weighting of the position and velocity accuracy. Further, the KLD is an established measure of the difference between two probability distributions [13]. The KLD between the communication loss methods and Full Communication is the KLD between two Gaussian distributions and is given by

$$KLD_{(\cdot)} = \frac{1}{2}\left(\ln(\frac{|\mathbf{P}_{(\mathrm{FC})}|}{|\mathbf{P}_{(\cdot)}|}) + \mathrm{tr}(\mathbf{P}_{(\mathrm{FC})}^{-1}\mathbf{P}_{(\cdot)}) + (\hat{\mathbf{x}}_{(\mathrm{FC})} - \hat{\mathbf{x}}_{(\cdot)})^T\mathbf{P}_{(\mathrm{FC})}^{-1}(\hat{\mathbf{x}}_{(\mathrm{FC})} - \hat{\mathbf{x}}_{(\cdot)}) - n_x\right), \tag{5.56}$$

where the subscript $(\cdot)$ is used to denote the method being compared.

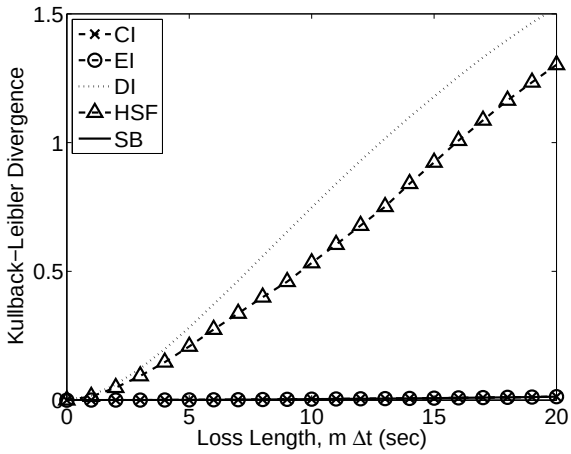


Figure 5.6: Kullback-Leibler Divergence from Full Communication for the two variations of Predicted Information method (CI, EI), and the three benchmark methods (DI, HSF, and SB) when locating a stationary POI, averaged over 5000 tests with experimental flight data.

For the case of a stationary POI, Figure 5.6 shows the KLD for each method compared to the Full Communication solution as a function of the length of a single communication loss of m time steps; here, m is varied from 0 to 200 (0 sec. to 20 sec.). Note that the results in Figure 5.6 are averaged over 5000 tests; the flight data was broken into 100 sets of length 1000 time steps (100 sec.) and communication losses started at 50 different times. Three conclusions can be drawn from Figure 5.6. First, as expected, the KLD metric increases with the length of the communication loss for all methods. Second, the CI, EI, and SB methods are all approximately equivalent and are by far the most accurate; after a 200 step (20 second) communication loss, the maximum KLD for CI, EI and SB is only 0.015. Thus, after an extended loss of 20 sec., while tracking a stationary POI, the CI, EI, and SB methods recover to nearly the Full Communication solution, almost as if the loss had never occurred. Third, the Hybrid State Fusion (HSF) approach is slightly more accurate than simply dropping information (DI).

Figure 5.7 shows the KLD averaged over 5000 tests of HiL simulation data, for

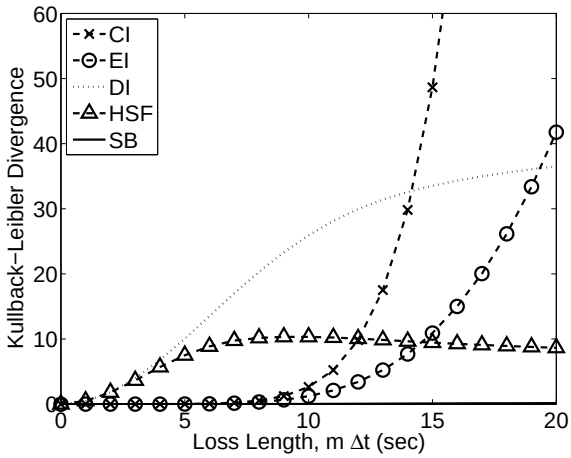


Figure 5.7: Kullback-Leibler Divergence from Full Communication for the two variations of Predicted Information method (CI, EI), and the three benchmark methods (DI, HSF, and SB) when locating a maneuvering POI, averaged over 5000 tests with HiL simulation data.

each of the methods as a function of the length of the communication loss for the case of locating a moving POI. As expected, the KLD metric increases for all of the methods as the communication loss gets longer. However, the relative accuracy of each of the methods depends on the length of the communication loss. For communication losses of less than seven seconds, the CI, EI, and SB methods are nearly equivalent and perform the best. However, as the communication losses get longer, each of these methods begin to perform worse. The CI method becomes worse than the HSF method at about 12 seconds and worse than the DI method at $\sim$ 14 seconds. The EI method performs better than the CI method but also degrades with the length of the communication loss and becomes worse than the HSF method at $\sim$ 15 seconds and owrse than the DI method at $\sim$ 19 seconds. The degradation of the PI methods is due to the accumulation of error during the communication loss. The EI method is able to more accurately predict the information matrix updates and hence can be used for longer communication losses than the CI method before the accuracy degrades.

Although estimation accuracy is important, other factors such as computation, communication, and memory requirements must also be considered. Most importantly, consider the computation loads of each of the communication loss methods. During a communication loss, the DI, HSF and SB methods require the same computation. They implement the EIF recursion, Equations 5.8 to 5.18, fusing only locally available information. The CI and EI methods require approximately twice as much; the EIF recursion plus the PI filter shown in Figure 5.1. However, after communication is restored, the CI, EI, and DI methods require the least computation; an addition of n_s matrices and n_s vectors. The HSF method requires the covariance intersection step in addition to the matrix and vector additions of the CI, EI, and DI methods. The SB method requires computation which scales linearly with the length of the communication loss, m EIF recursions to iterate from the beginning of the communication loss to the end. Note that this is essentially the computation performed by the CI and EI methods during the loss. However the SB method must perform this burst of computation in a short period of time, which may not be possible if the communication loss is long enough.

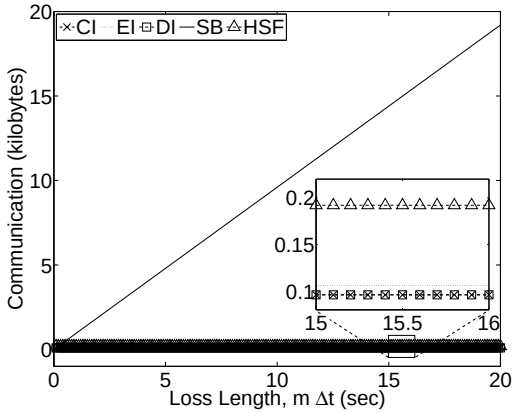


Figure 5.8: Communication required by each of the communication loss methods after an m step communication loss when locating a stationary POI.

During a communication loss there are no communication requirements. How-

ever, after communication is restored, the amount of data that must be sent varies between each of the methods. Figure 5.8 shows the communication requirements of each of the methods versus the length of the communication loss. Two observations can be made about Figure 5.8. First, the SB method requires linearly increasing communication with the length of the communication loss. The other methods (CI, EI, DI, and HSF) require constant levels of communication with the CI and DI methods requiring the least, followed by the EI method with slightly more (10 floating point numbers for the geolocation example), and the HSF method requiring the most. Second, the communication requirements only become a significant problem, even for the SB method, when the communication loss gets long (20 kilobytes for a 20 second loss). Note that the communication requirements shown in Figure 5.8 are for locating a stationary POI and are approximately doubled for locating a moving POI. However the relative ordering between the methods remains the same.

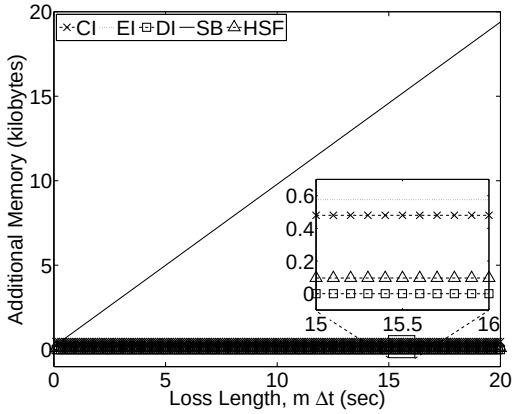


Figure 5.9: Additional memory required by each of the communication loss methods for an m step communication loss when locating a stationary POI.

The memory requirements are related to the communication requirements for each method and are shown in Figure 5.9. The observations about communication requirements essentially hold for the memory requirements. The SB method re-

quires linearly increasing memory with the length of the communication loss. The other methods, (CI, EI, DI, and HSF), have constant memory requirements with the DI method requiring the least, followed by the HSF method, the CI method, and finally the EI method requires the most. As with the communication requirements, the memory requirements are not significant even for the SB method until the communication loss gets very long.

5.7.2 Random Losses

In addition to single extended losses, the case of random losses is also considered. In the random loss scenario, the probability of any single information exchange being lost is denoted by p and is assumed to be independent of previous information exchanges. The KLD from FC converges to a steady state value in time, which is dependent on the probability of loss, p. This steady state value of the KLD from FC is used here as the accuracy metric and is defined formally as the average over the last 500 time steps (50 sec.) for 3000 time steps (300 sec.) of tracking. Note that the flight data is broken into 100 sets that are 3000 time steps (300 sec.) long and the results presented in this section are averages over these 100 sets.

For the case of a stationary POI, Figure 5.10 shows the steady state KLD metric averaged over 100 tests of the experimental flight data for each of the methods and drop probabilities varying from $p = 0$ to $p = 0.9$. Three observations can be made about Figure 5.10. First, the CI, EI, and SB methods are the most accurate and further, are indistinguishable from FC, even as the drop probability tends to large values such as $p = 0.9$. Second, the DI and HSF methods diverge significantly as the probability of loss, p, increases. Third, the HSF method provides essentially no improvement over DI.

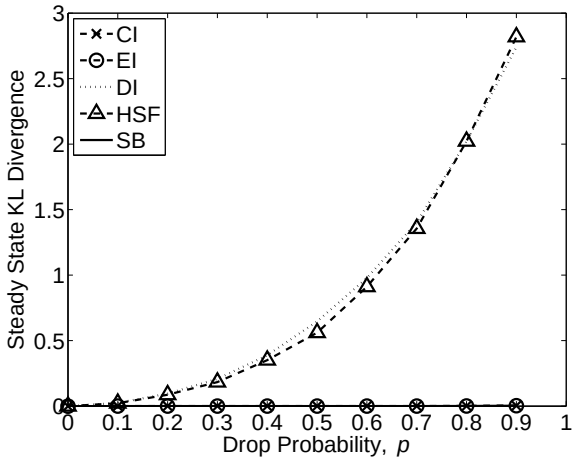


Figure 5.10: Steady State Kullback-Leibler Divergence as compared to the Full Communication case for the two variations of Predicted Information method (CI, EI), and the three benchmark methods (DI, HSF, and SB) when locating a stationary POI, averaged over 100 tests with HiL simulation data.

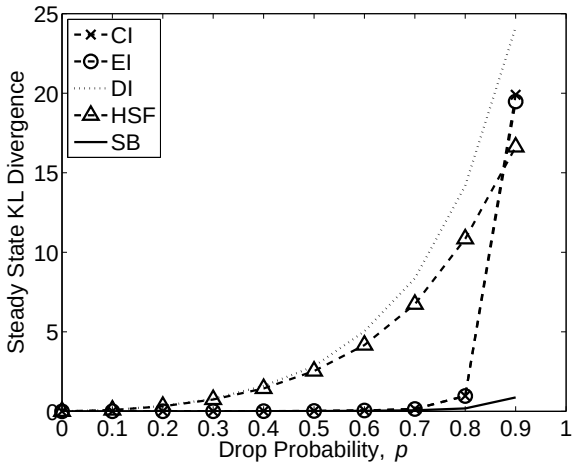


Figure 5.11: Steady State Kullback-Leibler Divergence as compared to the Full Communication case for the two variations of Predicted Information method (CI, EI), and the three benchmark methods (DI, HSF, and SB) when locating a maneuvering POI, averaged over 100 tests with HiL simulation data..

The steady state KLD metric for each of the methods when tracking a maneuvering POI using HiL simulation data is shown in Figure 5.11, with drop probabilities varying from $p = 0$ to $p = 0.9$. As expected, the accuracy of each of the methods degrades as the probability of loss, p, increases. The degradation in accuracy for the PI methods (CI,EI) is not noticeable until $p = 0.7$. However, the accuracy of the PI methods then degrades sharply at $p = 0.9$. The accuracy of the SB method also degrades but not noticeably until $p = 0.8$. The accuracy of the DI and HSF methods degrades smoothly and significantly as the probability of loss, p, increases; with the HSF methods being little better than simply dropping information.

The random loss scenario can be thought of as a repeated extended loss where the length of each loss is determined probabilistically based on p. Under the assumptions of the random loss scenario, it can be shown that the expected length of each communication loss is given by $m_{RL} = \frac{p}{1-p}$. So, the expected communication loss length depends nonlinearly on p and further diverges to infinity as p approaches one. Consider now the similarities between Figures 5.6 and 5.10 and between Figures 5.7 and 5.11. The dependence of the KL divergence from Full Communication on the single extended loss length for each of the methods (Figures 5.6 and 5.7) is similar to the dependence of the steady state KL divergence on expected loss length for the random drop scenario (Figures 5.10 and 5.11). However, the steady state KL divergence is much higher for corresponding loss lengths. This is due to the accumulation of error from the repetition of the communication losses.

5.8 Conclusions

Cooperative estimation using multiple mobile sensor nodes communicating over a lossy network is considered in this paper. A new method, termed Predicted Information (PI), is developed from a separable formulation of the Extended Information Filter. Although derived for nonlinear systems, it is shown to give the exact solution for linear systems when the measurement dynamics are constant or known by all sensor nodes. Two variations of the predicted information method are presented, which trade between accuracy and computational complexity. The first variation, Estimate Information (EI), estimates the information matrix updates directly from estimates of the cooperating sensor node states. The second variation, Constant Information (CI), uses a piecewise constant approximation to predict the information matrix updates. Note that although the differences in accuracy between the PI methods is not significant in the geolocation example, there are reasonable examples with more predictable sensor node dynamics where the accuracy would be significantly different. The predicted information method is evaluated with a cooperative geolocation problem with two Uninhabited Aerial Vehicles (UAVs) using gimballing camera sensors. Flight test data is used to compare the PI method with three benchmark methods from the literature, Drop Information (DI), Store and Burst (SB), and Hybrid State Fusion (HSF), for tracking both stationary and maneuvering Points of Interest (POIs). Furthermore, two types of communication losses are considered: single extended losses and random dropouts.

When locating a stationary POI, in the presence of either extended losses or random drop outs, the new PI methods (CI,EI) are shown to be as accurate as the most accurate method (SB) and more accurate than the other two benchmark methods (HSF, DI). Further, the PI methods are able to match the tracking ac-

curacy of the SB method, but with significantly lower memory, communication, and computation requirements. Note that the computational load of the SB methods scales linearly with the length of the communication loss and can become too great if the communication loss is long enough. However, the computational load of the proposed PI method remains constant independent of the length of the communication loss.

When tracking a maneuvering POI in the presence of an extended communication loss, it is noted that there exists a crossover point where the PI methods (CI, EI) are no longer more accurate than the simple DI and HSF methods. This crossover point varies depending on the application, but could be evaluated in simulation. If the given application has a lossy network, and is known to have communication losses that persist beyond the crossover point, then a switch from one of the PI methods to one of the other methods from the literature is suggested. Combinations of these methods could also be used. Similarly, with random dropouts there exists a crossover probability ($p \approx 0.9$ for the geolocation example) where the PI methods begin to lose accuracy as compared to the other methods. This drop probability is quite high compared to most practical applications.

CHAPTER 6

FLIGHT RESULTS FROM TRACKING GROUND TARGETS USING SEASCAN UAVS WITH GIMBALLING CAMERAS

6.1 Abstract

Flight test results using a SeaScan UAV with a gimballing camera to track both stationary and moving ground targets are presented. We experimentally studied the effect of UAV altitude above the target, camera field of view, and orbit center offsets within the geolocation tracking performance for both stationary and moving targets. In addition, all of the tests were performed using two different aircraft navigation systems, showing important sensitivities to avionics system accuracies. Sensor biases are shown to directly cause slowly varying errors in the geolocation estimates which can dominate tracking performance. These errors, which typically oscillate with the UAV orbit, are adequately bounded with a geolocation estimator which captures both the target tracking uncertainty, as well as unobservable sensor biases.

6.2 Introduction

Currently, Uninhabited Aerial Vehicles (UAVs) are being used to track ground targets in applications ranging from military missions to search and rescue operations [12] and even to animal migration [53]. Ground targets can be tracked using digital cameras, radar, infrared cameras or any combination. There are many reasons to use digital cameras for tracking ground targets when using UAVs. Digital cameras use significantly less power and are much cheaper than radar. The sensor

output is a digital image of the target and surroundings, and therefore is intuitive to a human operator. Digital cameras are also now small, lightweight and inexpensive, which is a major benefit for UAVs because the weight, size and cost of the payload are critical to the end missions and applications where UAVs will be used.

Several groups have implemented gimballing camera systems on UAVs [25, 26, 64], and a subset of these now include target tracking results [49, 63, 15, 10]. Refs. [51], and [23] have implemented UAV systems with cameras using decentralized fusion (information filtering) concepts. Stepanyan and Hovakimyan [56] demonstrate visual tracking of one aircraft with another aircraft using only a monocular camera.

In related research, several groups are using vision systems for UAV navigation, such as for road following [20] or obstacle avoidance [34]. The latter implements a bearings-only Simultaneous Localization and Mapping (SLAM) algorithm to localize both the vehicle and obstacles and navigate using only a low cost inertial measurement unit and a monocular camera. Kaaniche [31] *et al* present a traffic surveillance approach with UAVs using a graph cut formulation and a verification step.

The authors have developed and implemented experimentally a geolocation tracking algorithm for gimballing vision payloads on UAVs [11]. The implementation occurred on the SeaScan UAV, a long endurance (24hr) UAV developed by the Insitu Group. The algorithm includes attributes such as the ability to fuse information across multiple UAVs, and the inclusion of uniform bias uncertainties. As part of the verification of this work, sets of geolocation flight tests were performed. Flight tests were nominally a single UAV in a circular orbit above the

target, tracking the target with the two axis gimballing camera. Servo control of the camera axes was separate from the geolocation tracking estimator as an "inner" control loop. Important parameters varied during this flight test include UAV altitude, orbit center offsets, camera field of view, and type of target (stationary, moving). In addition, the UAV was equipped with two different UAV navigation systems, demonstrating the sensitivity of the results to the avionics accuracy.

This paper is organized as follows. First, the geolocation estimator used for the tracking is introduced. This is followed by a description of the SeaScan UAV hardware. Finally the effect of each of the flight test parameters on geolocation performance is presented.

6.3 Geolocation with the SeaScan UAV

Geolocation is the process of using sensory data to develop statistical estimates of a Point of Interest (POI) on the ground. For the application of a vision sensor on a UAV, the UAV, based on its position and orientation, points the camera (through a gimballing payload mount inside the UAV) at the POI on the ground. While the aircraft is moving (navigation and orientation), and the POI is potentially moving, the camera gimbals must adjust their angles to point at the POI. This application requires the camera to remain directed at the POI such that the POI always remains within the field of view of the camera. The objective of geolocation is then to estimate the position (2D or 3D) of the POI from the aircraft, gimbal, and camera measurements. Complicating this problem are uncertainties in the aircraft position and orientation, gimbal angles, camera specifications and measurements, and disturbances such as turbulence and engine vibrations.

Building a centralized estimator to estimate UAV navigation (NAV), attitude (ATT), gimbal (GIM) and POI states would require full UAV and gimbal models, and a model for the POI. Therefore, the implemented estimator would require a very high load of computation, memory, and communication in the case of multiple UAVs. Fortunately, most UAVs use an inertial measurement system which includes estimators which provide statistics (estimates and covariances) for both the UAV ATT and NAV states. In addition, the GIM states can be directly measured. Therefore, an estimator can be developed which develops estimates of the POI state statistics only, thus saving computation and memory.

The dynamic tracking model of the POI, and the nonlinear camera measurements (dependent on POI, NAV, ATT, and GIM states) in this case are written as:

$$\mathbf{x}_{k+1,POI} = f_{POI}(\mathbf{x}_{k,POI}, \mathbf{w}_{k,POI}) \tag{6.1}$$

$$\mathbf{z}_{k+1,SCR} = g_{SCR}(\mathbf{x}_{k+1,POI}, \begin{bmatrix} \mathbf{x}_{k+1,NAV} \\ \mathbf{x}_{k+1,ATT} \\ \mathbf{x}_{k+1,GIM} \end{bmatrix}, \mathbf{v}_{k+1,SCR}) \tag{6.2}$$

Ref. [10] describes a square root sigma point filter for this modified geolocation model, as well as a compensation technique for bias uncertainties. A Square Root, Sigma Point Information Filter (SR-SPIF) is developed in Ref. [11] in order to facilitate cooperative tracking of POIs.

6.3.1 The SeaScan UAV

The SeaScan UAV, produced by The Insitu Group and shown in Figure 6.1, is a long endurace UAV (24hr) used for defense and civilian applications. The SeaScan

Figure 6.1: The SeaScan UAV.

UAV can accommodate a variety of payloads, but the centerpiece is a digital video camera integrated into an inertially-stabilized pan / tilt nose turret. The daylight camera has an acuity $\approx 50\%$ better than that of the unaided eye at the telescopic end. It can resolve POI's such as small boats and logs from five miles away. The operator can command the camera to pan back-and-forth for wide-area search, or to remain locked onto a POI while the aircraft maneuvers; the latter mode is used here.

Software processes the images from the camera. When the user selects a POI for geolocation, the gimballing turret and ground software attempt to maintain the POI in the center of the frame, from frame to frame. Therefore, the "measurement" of the POI is assumed to be at the center of the image frame. The measurement equation, (6.2), is a complicated function of the UAV states and the POI states, which yield the screen coordinates in terms of pixels. The full derivation of the measurement equation is developed in Ref. [10].

6.3.2 Sensor Bias

A key element of this estimator is the implementation of an augmentation for sensor bias uncertainty. Sensor bias is used in this context to mean slow varying errors in the estimators of the UAV position (NAV), attitude (ATT), or camera gimbal (GIM) states. Biases can arise from a variety of conditions, including mounting conditions, timing delays, sensor characteristics, thermal changes, and GPS inaccuracies.

Figure 6.2 shows a stationary POI example with a known POI altitude. In this case, the measurement equation (6.2) can be reconfigured and solved directly for x_{POI}, given a measurement z_{SCR}. This is called the sensed point of interest (SPOI). Physically, the SPOI is the intersection of the camera line of sight with the ground as computed from the UAV and camera states.

Figure 6.2 shows that the SPOI moves in a roughly circular path around the true POI location; the period of this oscillation corresponds directly to the UAVs orbit about the POI. These periodic SPOI errors can be attributed to a bias in the UAV altitude, UAV attitude, or camera attitude states.

What is more challenging is that these biases are typically unobservable to the estimator. The POI state estimates and covariances can typically be used to bound the geolocation tracking error. A common metric is to calculate and plot a 2σ bound on the estimate based on the UAV states, POI estimate and POI estimator covariance. But, any biases in the sensors will directly create instantaneous errors in the POI state estimates, and these errors are not captured in the 2σ bounds. Therefore, these errors must be addressed.

For stationary POI's, the biases can be reduced by assuming a stationary POI

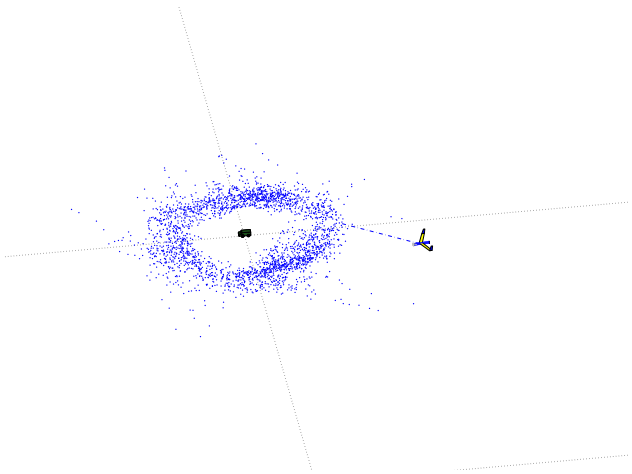


Figure 6.2: Single UAV tracking a stationary POI. The blue dots are computed camera line of sight intersections with the ground from two orbits of tracking.

model, and averaging over time. But this does not work, in general, for moving POI's. Ref. [10] developed an approach to augment the 2σ uncertainty bounds of the estimator with a probabilistic model of the bias uncertainties. A uniform distribution of each of the biases was used, with the intuition that one bias was not more likely than the next. Augmenting the Gaussian tracking error distribution with a series of uniform distributions from the sensor biases created a non-Gaussian distribution that could be used to model the true tracking error distribution; bounds with 95% confidence could be generated in real time.

Figure 6.3 shows the POI tracking errors and uncertainty bounds from the same flight data set as used in Figure 6.2. The tracking errors were found by subtracting off a "truth" GPS measurement of the POI location. The SPOI is also plotted, and the oscillations can easily be seen in the data. The geolocation state estimates oscillate similar to the SPOI, but at a smaller amplitude that decreases over time. This is a result of the using a stationary POI model, and a relatively small white noise intensity on the disturbance (w_{POI}). The dash-dot lines denote the 2σ uncertainty bounds directly from the geolocation estimator. Notice

that these bounds are quite small, and do not bound the true uncertainty of the estimator. In addition, the bounds are relatively insensitive to the oscillations of the orbit. This indicates that the system is dominated by sensor biases. The outer dashed lines are the augmented 95% probability, uncertainty bounds, which include components for the 2σ estimator bounds and the bias distribution. The bias bounds capture the true uncertainty in that the zero error always falls within the bounds. In addition, note that the bounds also oscillate with the orbital uncertainty; this correlates well with the true bias sensitivities, such as those shown in Figure 6.2.

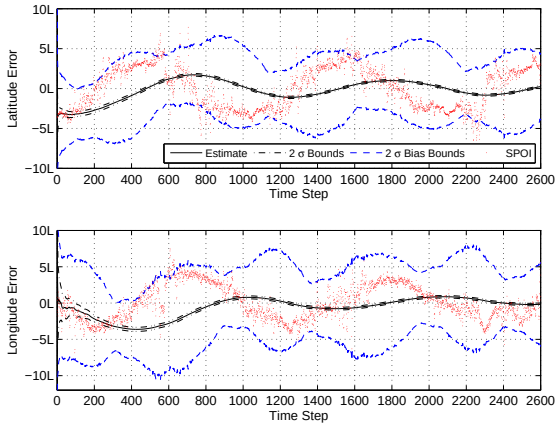


Figure 6.3: Geolocation errors and bounds using the Athena GuideStar navigation system with a camera field of view or 3.84 deg, an altitude above target of 320 m, and no orbit center offset.

6.4 Flight Results

Flight tests were performed on 18 March 2006. A "truth" was set up using a GPS antenna and receiver in a car near the flight test range. Over five hours of flight tests were recorded over the two days using a variety of POI's. The flight tests evaluated stationary and moving POIs, different altitudes of the UAV, and different

orbit offsets. In all tests, the UAVs onboard vision system kept the camera pointed at the POI. All telemetry was saved in order to evaluate estimation and tracking accuracy as a function of the different parameters. For all cases the orbit radius was fixed at 500 m and the altitude, camera field of view, and orbit offset were varied. In addition, two different UAV navigation systems were used: Athena GuideStar III and Helmsman. The Athena GuideStar III navigation system is an off the shelf system [**?**] with accuracies given as $\sigma = 0.3$ deg for attitude estimates (roll, pitch, yaw), and $\sigma = 2$ m navigation solution. Helmsman was developed by The Insitu Group as an in house guidance solution. The POI tracking results and their dependencies on each of the parameters varied during the test are summarized in the following sections. Note that due to International Traffic in Arms Regulations (ITAR), the specific performance data has been scaled. The unit of length used here will be denoted as, L and consequently we will use a volume unit, V, with $V = L^3$. The relative effect of each of the parameters within the flight tests, however, are still evident.

For the evaluation of each the test parameters, three measures of performance will be used. First the output of the bias estimator, denoted as Bias Bound Volume, which is a 95% confidence bound which includes the tracking estimators 2σ bound. Next the tracking estimators 2σ bound, denoted as SPF bound. Finally an experimental measure of the error which is denoted as the experimental error covariance or error volume. The experimental error covariance is defined by taking the expectation of the outer product of the position estimate errors. Mathematically this can be written as

$$P_{err} = \frac{1}{N}\sum_{i=1}^{N}(\hat{x}_{\mathrm{i,POI}} - x_{\mathrm{i,POI}})(\hat{x}_{\mathrm{i,POI}} - x_{\mathrm{i,POI}})^T \tag{6.3}$$

Note that given a covariance matrix, P, the 2σ volume, $V_{2\sigma}$, is defined as

$$V_{2\sigma} = \frac{4}{3}\pi det(\sqrt{P})8, \tag{6.4}$$

where $\sqrt{(\cdot)}$ is the principal matrix square root, the factor of $\frac{4}{3}$ is for the volume of an ellipse and the factor of 8 scales the volume to the 2σ value.

6.4.1 Stationary POI

For the stationary POI tests, a vehicle was parked on the test range at an altitude $\approx$ 180 m above sea level. The UAV orbited the POI with a radius of 500 meters and used the onboard vision system to keep the car image in the camera screen. The UAV altitude, camera field of view, and orbit center were varied for each of the tests and are summarized in Table 6.1.

Table 6.1: Flight tests for a stationary POI.

Test Number	Altitude (m)	Field of View (deg)	Orbit Offset (m)
1	500	7.51	0
2	500	3.84	0
3	500	1.78	0
4	750	7.51	0
5	750	3.84	0
6	750	1.78	0
7	750	3.84	300
8	750	3.84	500
9	750	3.84	700

Orbit Offsets

The orbit offset is defined as the 2D off-set of the orbit center from the POI position, projecting the UAV to the ground. When there is zero off-set, the slant range, or

range to the POI, is a constant. When the off-set is equal to the orbital radius, the UAV varies from being directly overhead of the POI (smallest slant range), to quite far from the POI (largest slant range).

Test numbers 5, 7, 8, and 9 were used to evaluate the effect of orbit offsets on geolocation performance because the only difference in these tests was the orbit offset. Figure 6.4 shows the tracking performance from test number 8. Because the orbit offset for test number 8 was 300 m, the slant range to the POI varied significantly throughout the test. As the slant range varies from approximately 550 m to 1150 m (factor of two), the corresponding bias bound ranges from approximately 3L to 10L (factor of three).

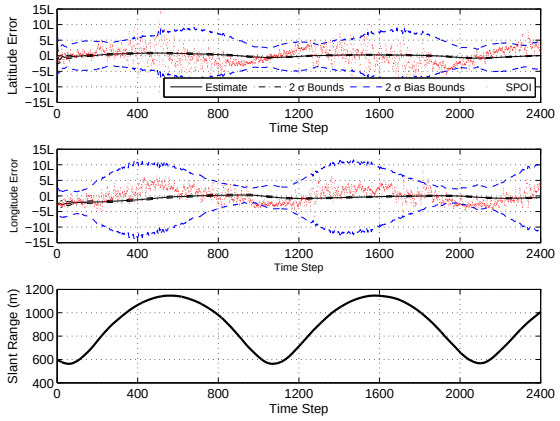


Figure 6.4: Geolocation errors and slant range to POI from flight test eight using GuideStar.

Figures 6.5 and 6.6 show the min, max, and mean volume of the 95% bias bounds, 2σ SPF covariance bounds, and the experimental estimate error covariance volume for each of the offset values for the Helmsman and GuideStar avionics systems. Several trends are evident. First, the variation in the uncertainty volume increases with increasing orbit off-set. While this can hurt tracking performance, it is noted that the minimum volume coincides with the minimum slant range, which

occurs when the off-set is equal to the orbital radius (500 m). The second trend to note is that the volume from the bias bounds is roughly 50 times larger than the volume from the estimator bounds. Obviously, this is significant. Finally, the bias uncertainty volume using the Helmsman avionics system is roughly twice as large as that using the GuideStar avionics. These results indicate that the orbit off-set is one of the most significant factors in tracking performance.

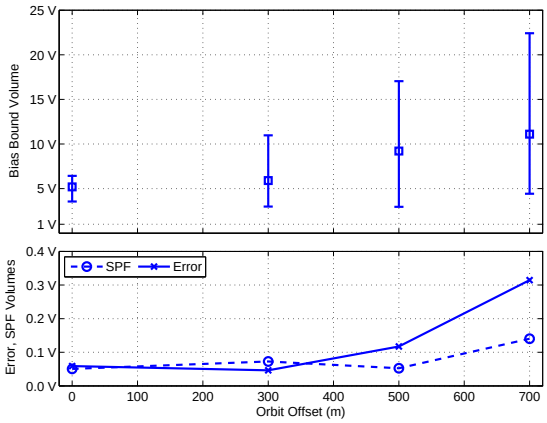


Figure 6.5: Uncertainty volumes over a range of orbit offsets using the Helmsman avionics system.

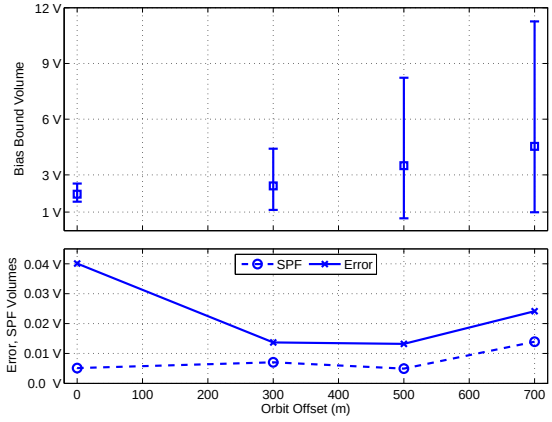


Figure 6.6: Uncertainty volumes over a range of orbit offsets using the GuideStar avionics system.

Variations in Relative Altitude and Camera Field of View

Tests 1 to 6 were used to judge the effect of changes in altitude on tracking performance. In tests 1-3, the UAV altitude was 500 m, giving a relative altitude above the POI of 320 m. In tests 4-6, the UAV altitude was 750 m, giving a relative altitude above the POI of 570 m.

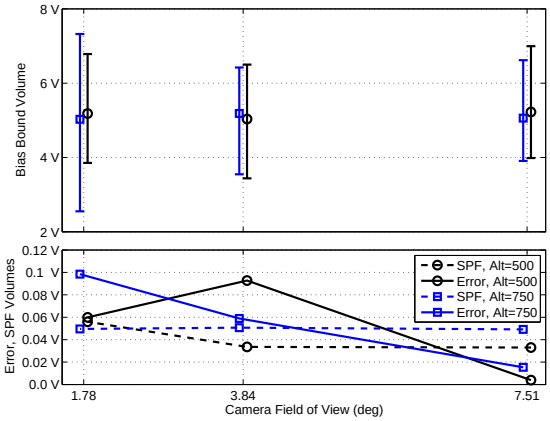


Figure 6.7: Uncertainty volumes over a range of camera field of view levels using the Helmsman guidance system.

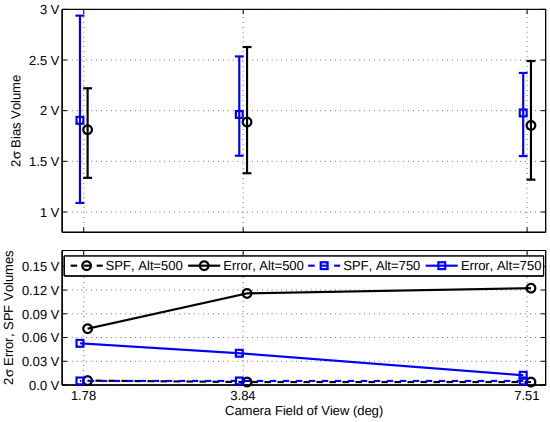


Figure 6.8: Uncertainty volumes over a range of camera field of view levels using the GuideStar navigation system.

Figures 6.7 and 6.8 show the changes in bias uncertainty volume as a function of the UAV altitude and camera field of view for these six tests, and for both the

Helmsman and GuideStar avionics. The primary effect of increasing the altitude is a narrowing of the variation of the bias uncertainty volume as the altitude increases. This is due to the nature of the camera measurement which is essentially a bearing sensor. The only case where this did not occur is at high zoom (field of view of 1.78 deg), where the onboard vision system lost POI lock and less than one orbit of data was recorded. For GuideStar, the UAV altitude of 750 m provided better tracking performance across camera field of view levels.

The effect of camera field of view on tracking performance is evaluated using test numbers 1-6. Three different field of view settings were tested: 7.51, 3.84, and 1.78 deg. Figures 6.7 and 6.8 show the three uncertainty volume metrics over the range of field of view levels. Several trends are noted. First, the SPF covariance is insensitive to camera field of view for both avionics systems. But, the true error is quite sensitive to camera field of view. For the Helmsman navigation system example, as the field of view ranges from 7.51 deg to 1.78 deg, the uncertainty volume increases from 0.02V to 0.1V. This is because of the performance of the image tracking loop at high zoom levels. With the camera field of view set to 1.78 deg the vision system had significant trouble remaining locked onto the target and during test number three only 850 time steps of the test were performed due to vision system difficulties. In the GuideStar case, the true error fluctuates more, but is smaller than the Helmsman case. A second trend is that the bias bound volume does not change as a function of field of view, except in the case when the field of view was at 1.78 and altitude at 750 m, which was very sensitive.

6.4.2 Moving POI

For the moving POI tests, a GPS unit was placed in a car which was driven at low speeds down a road near the test range. The UAV camera field of view and orbit center location were varied for each of the tests and are summarized in Table 6.2.

Table 6.2: Flight test parameters for the moving POI tests.

Test Number	Altitude (m)	Field of View (deg)	Orbit Offset (m)
10	750	7.51	0
11	750	3.84	0
12	750	1.78	0
13	750	3.84	300
14	750	3.84	500
15	750	3.84	700

The Geolocation behavior with a moving POI is different than with a stationary POI. Figure 6.9 shows the Geolocation errors and bound from test number 11. The uncertainty volume is significantly larger for the moving POI than for the stationary POI. Perhaps more importantly, the slowly varying errors due to the sensor biases in the measurements are present in the estimator output. Because the process noise intensity must be increased to track a moving POI, the geolocation estimator cannot distinguish between the measurement biases and POI dynamics.

Orbit Offsets

Test numbers 11, 13, 14, and 15 were used to evaluate the effect of orbit offsets on geolocation performance for the moving target case. Figures 6.10 and 6.11 show the three volume metrics (Bias, SPF, and Error) for each of the offset values, using Helmsman and using GuideStar. The same trends from the stationary case

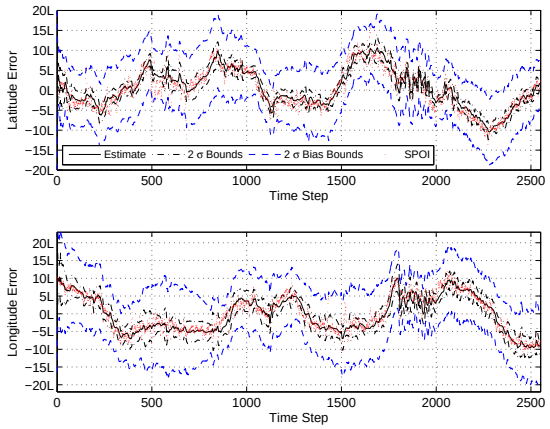


Figure 6.9: Geolocation errors and slant range to POI from flight test eleven using the Helmsman navigation system.

persist with the moving target case. Also, as with the stationary case, the Athena GuideStar navigation system improves the tracking performance. In the moving target case, this improvement is roughly a factor of two in the uncertainty volume, which is the same as in the stationary case. One notable difference is that the SPF bounds are greater than the experimental error bounds for the moving target case but were not for the stationary target case. This is because the tracking filter was tuned for more highly maneuverable targets than were demonstrated in this flight test.

Camera Field of View

The effect of camera field of view on moving target tracking performance is evaluated based on tests 10-12 with three different field of view levels: 1.78, 3.84, and 7.51 deg. Figure 6.7 shows the 95% uncertainty volumes over the range of field of view levels. As with the stationary case, the effect of field of view level is limited in terms of the geolocation performance. Again the SPF bound is higher than the error the process noise was significantly increased for tracking highly maneuverable

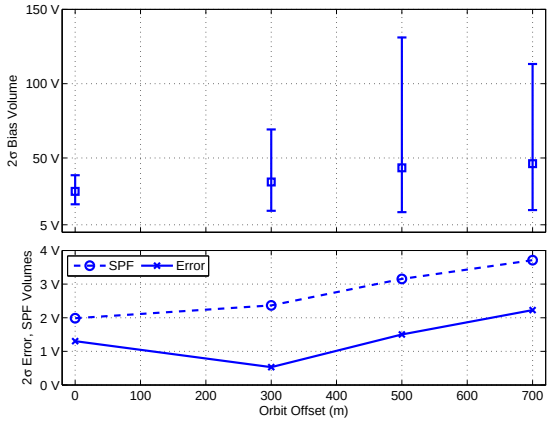


Figure 6.10: Uncertainty volumes over a range of orbit offsets using the Helmsman navigation system.

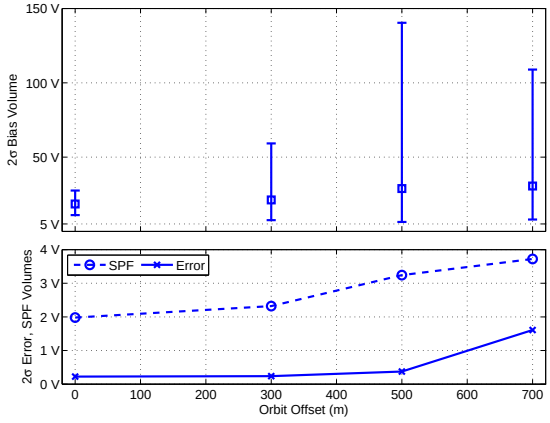


Figure 6.11: Uncertainty volumes over a range of orbit offsets using the GuideStar navigation system.

targets.

6.5 Conclusions

Flight test results using a SeaScan UAV with a gimballing camera to track both stationary and moving ground targets has been presented. Important geolocation system parameters were experimentally studied, including the effect of UAV alti-

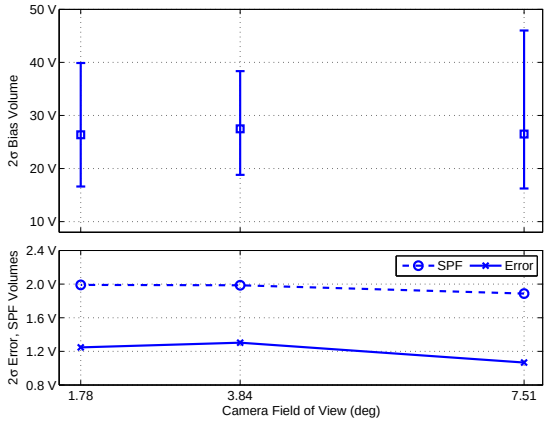


Figure 6.12: Uncertainty volumes over a range of camera field of view levels using Helmsman.

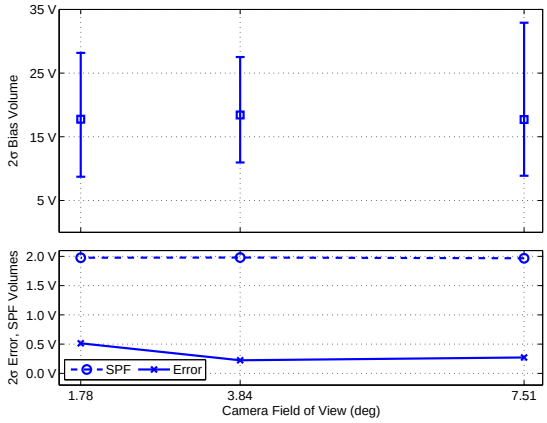


Figure 6.13: Uncertainty volumes over a range of camera field of view levels using GuideStar.

tude, camera field of view, and orbit center off-sets on the tracking performance. All tests were performed with two different aircraft navigation systems.

Sensor biases were shown to cause a slowly varying, typically oscillatory set of errors. In this work, a sensor bias model with a uniform distribution was used to augment the uncertainty developed from the tracking estimator. With a stationary target, the slow varying errors in the SPOI caused large geolocation errors at the initialization, but they decreased over time. The estimator is able to converge

to the correct answer after several orbits because the stationary tracking model enables a simple averaging over time. However, in the moving target case, the estimator is not able to clearly distinguish between the effects of the bias errors, and the actual target motion.

The parameter that caused the most sensitivity in geolocation performance, of those tested, was orbit off-set. Because the camera is a bearing only sensor, the bias errors are influenced directly by the slant range to the target. Therefore, the uncertainty volume is a maximum when the slant range is a maximum, and a minimum when the slant range is a minimum. This variation can be used to improve geolocation, especially in the multiple vehicle case, by noting that the uncertainty volume is a minimum when the UAV is directly overhead.

In addition to slant range, the relative altitude and consequently the elevation angle are contributing factors to geolocation performance. Due to the nature of the camera sensor, a low elevation angle leads to greater SPOI errors in the direction from the UAV toward the target. This leads to a trade off between increasing the altitude above the target to get a higher elevation angle and trying to decrease the slant range.

CHAPTER 7

CONCLUSIONS

The central contribution of this dissertation is the solution to the problem of cooperative geolocation of a point of interest (POI) using multiple uninhabited aerial vehicles (UAVs) with gimballing camera sensors and two associated complications: sensor bias and communication loss. Several other groups have implemented gimballing camera systems on UAVs [23, 51, 25, 26], a few with initial target tracking results [49, 63, 15]. In a more closely related work Vercauteren and Wang [62] develop a Sigma Point Information Filter. Most of these works typically only address a subset of the important requirements for vision tracking systems for production UAVs, namely 1) scalability to a modest number of UAVs, 2) robustness to communication loss and bandwidth limitations, and 3) numerical stability and efficiency in real time implementation for nonlinear, decentralized tracking.

This dissertation details the theoretical development of a distributed, cooperative estimation methodology for multiple UAVs which track stationary and moving POIs with on board cameras which meets all of the above requirements. A square root, sigma point information filter (SR-SPIF) is developed in [11] and presented in Chapter 2 to solve the cooperative geolocation problem. The square root formulation is used to maintain numerical integrity in real time. The sigma point formulation is used for its accuracy with nonlinear dynamics and nonlinear measurement equations. An information form is used for ease in fusing measurements from other UAVs. The SR-SPIF presented in Chapter 2 incorporates uncertainty in the UAV states with a combined state (POI state) and parameter (UAV state) formulation with a non-standard update, which utilizes the onboard navigation system to save computation. Chapter 3 presents an improved SR-SPIF which in-

corporates the UAV state uncertainty as a non-additive, nonzero mean noise. This formulation saves both computation and communication. Both versions of the SR-SPIF have the following important properties:

Decentralized: Each vehicle has its own geolocation estimator, and then communicates only necessary information to the other vehicles. This minimizes memory and communication, and enables network robustness.

Information Form: An information form is used to 1) minimize the amount of information shared between vehicles, 2) simplify the multiple vehicle fusion problem, and 3) simplify the problem of delayed data (from communication drop-outs). For single vehicle tracking the use of the information form leads to a slight increase in the computational burden. However, when performing tracking with cooperative vehicles, the information form can significantly reduce the amount of computation.

Sigma Points: Sigma points are used to develop statistical linearizations of the dynamics, which have been shown to be more accurate than the traditional Extended Kalman Filter (or the Extended Information Filter) [27]. The use of sigma points requires a small increase in computation compared to the Extended Kalman or Information Filters. However, it is proposed here that the increase in tracking performance with the nonlinear measurement is justification for this increase in computational cost.

Square Root: A square root version of the estimator is used for its numerical accuracy in real time implementation. The square root implementation requires very little additional complexity when used in conjunction with sigma points. The square root version is equivalent in computation to the Square Root Sigma Point Filter [6], which was shown to work in real time at 20Hz

for an UAV aerodynamic model estimator [9].

In addition to the development of the SR-SPIF, where the errors in the estimate of the UAV state are assumed to be zero mean, white, and Gaussian, nonzero mean errors (biases) are also treated in this dissertation. Sensor biases were shown in Refs. [10] and [67] (included in Chapter 6)to be a significant source of error for geolocation using the ScanEagle UAV and were compensated for by augmenting the output of the estimator with additional uncertainty, based on empirical data. However, this did nothing to improve the estimate itself, but simply improved estimator consistency. Biases have also been shown to be a problem for estimation in other works. In one of the first treatments [21], Friedland showed that biases could be estimated efficiently in a linear system by partitioning the state. More recently in Refs. [32] and [54], bias estimation in a radar tracking context with multiple targets was addressed. Bias estimation was further considered in Refs. [37] and [45], where multiple sensors were used to track multiple targets in a centralized formulation. Bias estimation has also been considered in a least squares estimation context by Dogancay in Ref. [16].

In Chapter 4, an approach is developed to jointly estimate the sensor biases and the unknown POI state in a decentralized manner, while using the solution from the onboard navigation system to save significant computation. The decentralized formulation allows the UAVs to share information on only the POI state, and model only their local biases, saving computation as well as communication, and moreover, giving geolocation accuracy comparable to the centralized case. Further, this decentralized approach fits nicely into the decentralized data fusion paradigm [17, 59] and allows for effective cooperation not only among UAVs with potentially different biases but different sensors altogether. A numerical observ-

ability analysis procedure is also developed in Chapter 4 and applied, which gives a meaningful measure of the degree of observability and also gives insight into the effects of UAV flight path on observability. The new decentralized approach is validated using both experimental flight data and high fidelity hardware in the loop simulations.

The second problem related to geolocation which is solved in this dissertation is cooperative estimation in the presence of communication loss. Communication is an important component of the cooperative estimation process, and recent research has focused on the effects of digital communication in both control and estimation. Delchamps [14] presented a seminal work describing the effect of quantization on controller performance. More recently, significant work has been done on the effect of network communication between sensors, controllers, and system plants [60, 69, 52, 70]. In addition to quantization effects, the cooperative estimation system must be robust to communication losses and delays. For example, in autonomous underwater vehicle applications, communication is particularly limited in reliability as well as bandwidth. In [1], Akyildiz *et al* characterize the underwater communication channel noting that communication is lost or delayed frequently due multi-path and obstructions.

The problem of cooperative estimation in the presence of communication loss is considered in Chapter 5. Communication loss refers to a situation in which the sensor nodes are unable to communicate with each other for a period of time that is unknown a priori. It is assumed here that the communication losses are symmetric and known by each sensor node. Communication delay is a related problem and refers to the situation in which sensor nodes must communicate over a medium which induces potentially unknown and varying delays in the data transmission;

for example, communication over an ad hoc wireless network. It is assumed here that the delays are not symmetric between nodes and there is no confirmation of receipt provided from the receiving node. In this case, any time a sensor node sends data, it will have no knowledge of how long before the data is received at the other end. Also, there is no guarantee that the data will be received at a sensor node in the order it was generated. Under this scenario, it is the responsibility of the receiver to make effective use of the data received, even if the data is both late and out of order.

The delayed data problem was explored in the Kalman filtering domain by Larsen *et al.* in Ref. [35], where the focus was on estimation with a combination of fast and slow sensors. In Ref. [30], Julier and Uhlmann developed an approach algebraically equivalent to the results of Larsen. In a centralized estimation framework, Bar-Shalom developed an exact solution for out-of-sequence measurements[3]. Nettleton and Durrant-Whyte [44] examined the delayed data problem in the information filtering domain, which is more amenable to decentralized sensor fusion. This approach is theoretically equivalent to centralized estimation in linear systems, but requires large memory and bursty communication. None of the existing methods address the communication loss (or delay) problem in the nonlinear decentralized estimation context. These cases arise when the POI dynamics or measurements are nonlinear, which is common in the UAV tracking problem [11].

In Chapter 5, a new method is developed for decentralized estimation using multiple UAVs communicating over a lossy network. The new method, termed the Predicted Information (PI) method, is developed as an approximation to the Separable Extended Information Filter (SEIF), which is an alternate form of the

Extended Information Filter (EIF) derived in Appendix A. The basic concept of the PI method is to predict the information matrix updates during the communication loss to more easily update the estimator when communication is re-established. Two variations are presented that trade accuracy with computation, memory, and communication load. The first variation estimates the information matrix updates over time, while the second variation uses a piecewise constant approximation to the information matrix updates. Although the delayed data problem is not specifically addressed here, the PI method can be used with delayed data but requires minor modifications.

Another important contribution of this dissertation is the experimental and simulated flight data. Each of the three theoretical developments in this dissertation are validated using experimental flight data using the ScanEagle UAV. Flight tests were performed on October 6, 2004 [11], March 18, 2006 [67], and March 16, 2007 [57]. In addition to using the flight tests for validation, the 2006 flight test was used for a parametric study of the effect of various flight parameters on geolocation accuracy [67]. The results of the study are included in Chapter 6.

A high fidelity hardware in the loop (HiL) simulation test bed for distributed tracking was also developed. The distributed tracking test bed was used for validation of both the bias estimation and communication loss algorithms. Further, the simulation test bed was used in the experimental demonstrations of a flight test of cooperative tracking. [58]

APPENDIX A

DERIVATION OF SEPARABLE EXTENDED INFORMATION FILTER

To develop the SEIF, consider the recursive equations for the information matrix of the EIF shown in Equations 5.8 and 5.17 at the first time step, $k = 0$. Substituting Equation 5.8 into Equation 5.17 yields

$$\begin{aligned} \mathbf{Y}_1^l &= [(\mathbf{F}_0^l)^{-T}(\mathbf{Y}_0^l)(\mathbf{F}_0^l)^{-1}\boldsymbol{D}_0^l] + \mathbf{I}_1^l + \sum_{s\in\mathbb{S}\setminus\{l\}} \mathbf{I}_1^s \\ &= \widetilde{\mathbf{Y}}_1^l + \widetilde{\mathbf{I}}_1^l + \sum_{s\in\mathbb{S}\setminus\{l\}} \widetilde{\mathbf{I}}_1^s, \end{aligned} \tag{A.1}$$

where the discount factor $\boldsymbol{D}_0^l$ is defined as

$$\boldsymbol{D}_0^l = (\mathcal{I} + \boldsymbol{\Gamma}_0^l \mathbf{Q}_0 (\boldsymbol{\Gamma}_0^l)^T (\mathbf{F}_0^l)^{-T} (\mathbf{Y}_0^l \mathbf{I}_0^s)(\mathbf{F}_0^l)^{-1})^{-1}. \tag{A.2}$$

The contribution of prior information and the sensor nodes in Equation A.1 is broken into the cumulative information matrix, $\widetilde{\mathbf{Y}}_1^l$, and updates, $\widetilde{\mathbf{I}}_1^l$ and $\widetilde{\mathbf{I}}_1^s$, which are

$$\widetilde{\mathbf{Y}}_1^l = (\mathbf{F}_0^l)^{-T}(\mathbf{Y}_0^l)(\mathbf{F}_0^l)^{-1}\boldsymbol{D}_0^l \tag{A.3}$$

$$\widetilde{\mathbf{I}}_1^l = \mathbf{I}_1^l \tag{A.4}$$

$$\widetilde{\mathbf{I}}_1^s = \mathbf{I}_1^s \quad \text{for} \quad s \in \mathbb{S} \setminus \{l\}. \tag{A.5}$$

Propagating another time step to $k = 1$, the cumulative information matrix variables by substituting Equation A.1 into Equations 5.8 and 5.17 and again grouping terms gives

$$\mathbf{Y}_2^l = \widetilde{\mathbf{Y}}_2^l + \widetilde{\mathbf{I}}_2^l + \sum_{s\in\mathbb{S}\setminus\{l\}} \widetilde{\mathbf{I}}_2^s, \tag{A.6}$$

$$\tag{A.7}$$

where

$$\widetilde{\mathbf{Y}}_2^l = (\mathbf{F}_1^l)^{-T}(\widetilde{\mathbf{Y}}_1^l)(\mathbf{F}_1^l)^{-1}\boldsymbol{D}_1^l \tag{A.8}$$

$$\widetilde{\mathbf{I}}_2^l = (\mathbf{F}_1^l)^{-T}(\widetilde{\mathbf{I}}_1^l)(\mathbf{F}_1^l)^{-1}\boldsymbol{D}_1^l + \mathbf{I}_2^l \tag{A.9}$$

$$\widetilde{\mathbf{I}}_2^s = (\mathbf{F}_1^l)^{-T}(\widetilde{\mathbf{I}}_1^s)(\mathbf{F}_1^l)^{-1}\boldsymbol{D}_1^l + \mathbf{I}_2^s \quad \text{for} \quad s \in \mathbb{S} \setminus \{l\}. \tag{A.10}$$

and therefore by induction the recursion for the cumulative information matrix variables is given by

$$\mathbf{D}_k^l = (\mathcal{I} + \boldsymbol{\Gamma}_k^l \mathbf{Q}_k (\boldsymbol{\Gamma}_k^l)^T (\mathbf{F}_k^l)^{-T} (\mathbf{Y}_k^l \mathbf{I}_k^s)(\mathbf{F}_k^l)^{-1})^{-1} \tag{A.11}$$

$$\widetilde{\mathbf{Y}}_{k+1}^l = (\mathbf{F}_k^l)^{-T}(\widetilde{\mathbf{Y}}_k^l)(\mathbf{F}_k^l)^{-1}\mathbf{D}_k^l \tag{A.12}$$

$$\widetilde{\mathbf{I}}_{k+1}^l = (\mathbf{F}_k^l)^{-T}(\widetilde{\mathbf{I}}_k^l)(\mathbf{F}_k^l)^{-1}\mathbf{D}_k^l + \mathbf{I}_{k+1}^l \tag{A.13}$$

$$\widetilde{\mathbf{I}}_{k+1}^s = (\mathbf{F}_k^l)^{-T}(\widetilde{\mathbf{I}}_k^s)(\mathbf{F}_k^l)^{-1}\mathbf{D}_k^l + \mathbf{I}_{k+1}^s \quad \text{for} \quad s \in \mathbb{S} \setminus \{l\}. \tag{A.14}$$

The information state, $\mathbf{y}$, can be written in a similar fashion. First, consider the time propagation of the information state for the EIF shown in Equation 5.9 with the substitution of the fusion step shown in Equation 5.18 which gives

$$\mathbf{y}_{k+1}^{l-} = \mathbf{Y}_{k+1}^{l-} \cdot \mathbf{f}\left[(\mathbf{Y}_k^l)^{-1}\left(\mathbf{y}_k^{l-} + \mathbf{i}_k^l + \sum_{s\in\mathbb{S}\setminus\{l\}} \mathbf{i}_k^s\right), \mathbf{0}\right] \tag{A.15}$$

Since the state transition function, $\mathbf{f}[\cdot,\cdot]$, may be nonlinear, Equation A.15 is not directly separable. However, if $\mathbf{x} = \mathbf{0}$ is a fixed point of $\mathbf{f}$; i.e., $\mathbf{f}[\mathbf{0},\mathbf{0}] = \mathbf{0}$, then an equivalent state transition matrix, $\boldsymbol{\Phi}$, may be found such that

$$\mathbf{Y}_{k+1}^{l-}\cdot\mathbf{f}\left[(\mathbf{Y}_k^l)^{-1}\left(\mathbf{y}_k^{l-} + \mathbf{i}_k^l + \sum_{s\in\mathbb{S}\setminus\{l\}} \mathbf{i}_k^s\right), \mathbf{0}\right] = \mathbf{Y}_{k+1}^{l-}\cdot\boldsymbol{\Phi}_k\cdot(\mathbf{Y}_k^l)^{-1}\left(\mathbf{y}_k^{l-} + \mathbf{i}_k^l + \sum_{s\in\mathbb{S}\setminus\{l\}} \mathbf{i}_k^s\right). \tag{A.16}$$

If $\mathbf{0}$ is not a fixed point of $\mathbf{f}$ and cannot be made a fixed point by a suitable change of coordinates then $\boldsymbol{\Phi}$ is only guaranteed to exist for $\boldsymbol{x} \neq \mathbf{0}$. Note that Appendix B shows one approach to finding $\boldsymbol{\Phi}$. The equivalent state transition matrix, $\boldsymbol{\Phi}$,

allows the information state propagation to be written in a separable formulation, very similar to the IF, as

$$\widetilde{\mathbf{y}}_{k+1}^{l} = \mathbf{L}_{k+1}^{l}\widetilde{\mathbf{y}}_{k}^{l} \tag{A.17}$$

$$\widetilde{\mathbf{i}}_{k+1}^{l} = \mathbf{L}_{k+1}^{l}\widetilde{\mathbf{i}}_{k}^{l} + \mathbf{i}_{k+1}^{l} \tag{A.18}$$

$$\widetilde{\mathbf{i}}_{k+1}^{s} = \mathbf{L}_{k+1}^{s}\widetilde{\mathbf{i}}_{k}^{s} + \mathbf{i}_{k+1}^{s} \qquad \text{for} \quad s \in \mathbb{S} \setminus \{l\} \tag{A.19}$$

where the separable information state transition matrix, $\mathbf{L}_{k+1}^{l}$, is given by

$$\mathbf{L}_{k+1}^{l} = \mathbf{Y}_{k+1}^{l-}\mathbf{\Phi}_{k}(\mathbf{Y}_{k}^{l})^{-1}. \tag{A.20}$$

The form of the fusion step is the same as for the information matrix, and is given as

$$\mathbf{y}_{k+1}^{l} = \widetilde{\mathbf{y}}_{k+1}^{l} + \widetilde{\mathbf{i}}_{k+1}^{l} + \sum_{s\in\mathbb{S}\setminus\{l\}} \widetilde{\mathbf{i}}_{k+1}^{s} \tag{A.21}$$

APPENDIX B

COMPUTING EQUIVALENT STATE TRANSITION MATRIX

The equivalent state transition matrix, $\mathbf{\Phi}$, required for the Separable EIF can be computed in a number of simple ways such as

$$\mathbf{\Phi} = \text{diag}[f[\mathbf{x}, \mathbf{0}]./\mathbf{x}], \tag{B.1}$$

where ./ denotes element wise division and all elements of $\mathbf{x}$ are assumed nonzero. However, this approach breaks down when used with the PI method because during a communication loss, the state estimates are different on each of the sensor nodes. So, instead of an exact equivalent state transition matrix, an approximate one is sought that is accurate in a neighborhood of the state estimate, which will be approximately the same on all sensor nodes. This approximate equivalent state transition matrix can be computed with a 'sigma point' approach as follows. Define a set of sigma points as

$$\mathcal{X}_k^l = \left[\widehat{\mathbf{x}}_k^l, \;\; \widehat{\mathbf{x}}_k^l + \sigma_f \sqrt{\mathbf{P}_k^l}, \;\; \widehat{\mathbf{x}}_k^l - \sigma_f \sqrt{\mathbf{P}_k^l}\right], \tag{B.2}$$

where σ_f is a scaling parameter for the spacing of the sigma points. The sigma points are then propagated through the nonlinear dynamics as

$$\mathcal{X}_{k+1}^{l-,i} = \mathbf{f}\left[\mathcal{X}_k^{l,i}, \mathbf{0}\right] \quad \text{for} \;\; i = 1, \ldots, 2n_x + 1, \tag{B.3}$$

where the superscript i denotes the i^{th} column of the matrix (i^{th} sigma point). A least squares solution for $\mathbf{\Phi}$ in the neighborhood of $\widehat{\mathbf{x}}_k^l$ is then found by pseudoinverse as

$$\mathbf{\Phi}^l = \mathcal{X}_{k+1}^{l-} \cdot (\mathcal{X}_k^l)^T \cdot \left(\mathcal{X}_k^l \cdot (\mathcal{X}_k^l)^T\right)^{-1}. \tag{B.4}$$

The primary benefit of computing a $\mathbf{\Phi}$ which is accurate in a neighborhood of the estimate is the insensitivity to small differences in estimates on each of the sensor nodes.

APPENDIX C

CAMERA MEASUREMENT

The sensor measurement is formally defined as the location of the POI in the camera screen as determined by the onboard vision system and is given by (dropping the superscript and subscript denoting which UAV at which time step, respectively)

$$\begin{aligned} \mathbf{z} &= \mathbf{h}_{\mathrm{SCR}}(\mathbf{x}_{\mathrm{POI}}, \boldsymbol{\psi}, \mathbf{v}) \\ &= \begin{bmatrix} \lambda_y & 0 \\ 0 & \lambda_z \end{bmatrix} \begin{bmatrix} R^{\mathrm{CAM}}_{\mathrm{NED}}(2) \\ R^{\mathrm{CAM}}_{\mathrm{NED}}(3) \end{bmatrix} \left[R^{\mathrm{CAM}}_{\mathrm{NED}}(1)\left(\mathbf{x}_{\mathrm{POI}} - \boldsymbol{\psi}_{\mathrm{NAV}}\right)\right]^{-1} \left(\mathbf{x}_{\mathrm{POI}} - \boldsymbol{\psi}_{\mathrm{NAV}}\right) + \mathbf{v} \end{aligned} \tag{C.1}$$

where $\mathbf{v}$ is the noise in the vision system and

$$\lambda_y = \frac{\tan\left(FOV/2\right)}{p_{max-y}}, \lambda_z = \frac{\tan\left(FOV/2\right)}{p_{max-z}}$$

are pixel length scale factors, which depend on the camera field of view, FOV, and $p_{max-(y,z)}$, which is the maximum camera pixels in the y, z directions. Note that the UAV position and POI location are assumed to be in a local North-East-Down (NED) frame. The rotation matrix $R^{\mathrm{CAM}}_{\mathrm{NED}}$ is a combination of the rotation matrices from the local NED frame to the aircraft body (ABC) frame, and then to the camera (CAM) frame, given as

$$R^{\mathrm{CAM}}_{\mathrm{NED}} = \begin{bmatrix} R^{\mathrm{CAM}}_{\mathrm{NED}}(1) \\ R^{\mathrm{CAM}}_{\mathrm{NED}}(2) \\ R^{\mathrm{CAM}}_{\mathrm{NED}}(3) \end{bmatrix} = R^{\mathrm{CAM}}_{\mathrm{ABC}} R^{\mathrm{ABC}}_{\mathrm{NED}} \tag{C.2}$$

where

$$R_{\text{NED}}^{\text{ABC}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C(\phi) & S(\phi) \\ 0 & -S(\phi) & C(\phi) \end{bmatrix} \begin{bmatrix} C(\theta) & 0 & -S(\theta) \\ 0 & 1 & 0 \\ S(\theta) & 0 & C(\theta) \end{bmatrix} \begin{bmatrix} C(\psi) & S(\psi) & 0 \\ -S(\psi) & C(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{(C.3)}$$

$$R_{\text{ABC}}^{\text{CAM}} = \begin{bmatrix} C(s) & S(s) & 0 \\ -S(s) & C(s) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C(t) & 0 & S(t) \\ 0 & 1 & 0 \\ -S(t) & 0 & C(t) \end{bmatrix} \begin{bmatrix} C(p) & S(p) & 0 \\ -S(p) & C(p) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{(C.4)}$$

and $\boldsymbol{\psi}_{\text{ATT}} = [\phi, \theta, \psi]^T$ are the roll, pitch, and yaw angles of the aircraft and $\boldsymbol{\psi}_{\text{GIM}} = [p, t, s]^T$ are the pan, tilt, and scan of the gimbal and $C(\cdot)$ and $S(\cdot)$ represent $\cos(\cdot)$ and $\sin(\cdot)$ respectively.

BIBLIOGRAPHY

[1] Ian F. Akyildiz, Dario Pompili, and Tommaso Melodia. Underwater acoustic sensor networks: Research challenges. *Ad Hoc Networks (Elsevier)*, 3(3):257–279, Feb. 2005. doi: 10.1016/j.adhoc.2005.01.004.

[2] Yaakov Bar-Shalom. Mobile radar bias estimation using unknown location targets. In *International Conference on Information Fusion*, volume 1, Paris, France, July 2000.

[3] Yaakov Bar-Shalom. Update with out-of-sequence measurements in tracking: Exact solution. *IEEE Transactions on Aerospace and Electronic Systems*, 38(3):769–777, July 2002. doi: 10.1109/TAES.2002.1039398.

[4] Yaakov Bar-Shalom, X Rong Li, and Thiagalingam Kirubarajan. *Estimation with Application to Tracking and Navigation.* John Wiley and Sons, 2001.

[5] Yaakov Bar-Shalom, X Rong Li, and Thiagalingam Kirubarajan. *Estimation with Application to Tracking and Navigation.* John Wiley and Sons, 2001.

[6] Shelby Brunke and Mark E. Campbell. Square root sigma point filtering for aerodynamic model estimation. *AIAA Journal of Guidance, Control, and Dynamics*, 27(2):314–317, 2004.

[7] A. E. Bryson and D. E. Johansen. Linear filtering for time-varying systems using measurements containing colored noise. *IEEE Transactions on Automatic Control*, 10(1):4–10, Jan. 1965.

[8] M. Campbell, R. D'Andrea, J.-W. Lee, and E. Scholte. Experimental demonstrations of semi-autonomous control. In *Proceedings of the American Control Conference*, Boston MA, 2004.

[9] M. E. Campbell, J. Han, J. Lee, E. Scholte, and J. Ousingsawat. Validation of active state model based control using the seascan uav. In *AIAA Unmanned Unlimited Systems, Technologies, and Operations Conference*, San Diego CA, 2003.

[10] Mark E Campbell and Matthew Wheeler. A vision based geolocation tracking system for uav's. Keystone, CO, Aug. 2006. AIAA Guidance Navigation and Control Conference. AIAA Paper 2006-6246, to appear in AIAA Journal of Guidance, Control, and Dynamics.

[11] Mark E. Campbell and William W. Whitacre. Cooperative tracking using vision measurments on seascan uavs. *IEEE Transactions on Control System Technology*, 15(4):613–626, July 2007. doi: 10.1109/TCST.2007.899177.

[12] Chang Chung-Kuo and J. Huang. Video surveillance for hazardous conditions using sensor networks. In *IEEE International Conference on Networking, Sensing, and Control*, volume 2, pages 1008–1013, Taipei, Taiwan, Mar. 2004. doi: 10.1109/ICNSC.2004.1297085.

[13] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*, pages 243–259. John Wiley and Sons, New Jersey, second edition, 1991.

[14] David F. Delchamps. Stabilizing a linear system with quantized state feedback. *IEEE Transactions on Automatic Control*, 35(8):916–924, Aug. 1990. doi: 10.1109/9.58500.

[15] Vladimir Dobrokhodov, Isaac Kaminer, and Kevin Jones. Vison-based tracking and motion estimation for moving targets using small uavs. Keystone, CO, Aug. 2006. AIAA Guidance Navigation and Control Conference. AIAA Paper 2006-6606.

[16] Kutluyil Dogancay. Bias compensation for bearings-only pseudolinear target track estimator. *AIAA Modeling and Simulation Technologies Conference*, 54(1):59–68, Jan. 2006. doi: 10.1109/TSP.2005.861088.

[17] H.F. Durrant-Whyte and M. Stevens. Data fusion in decentralized sensing networks. In *The 4th International Conference on Information Fusion*, Montreal, Canada, 2001.

[18] Donald P. Eickstedt and Michael R. Benjamin. Cooperative target tracking in a distributed autonomous sensor network. In *OCEANS 2006*, Boston, MA, Sept. 2006. doi: 10.1109/OCEANS.2006.306976.

[19] Philip Ferguson and Jonathan How. Decentralized estimation algorithms for formation flying spacecraft. Austin, TX, Aug. 2003. AIAA Guidance Navigation and Control Conference. AIAA Paper 2003-5442.

[20] Eric Frew, Tim McGee, ZuWhan Kim, Xiao Xiao, Stephen Jackson, Michael Morimoto, Sivakumar Rathinam, Jose Padial, and Raja Sengupta. Vision-based road-following using a small autonomous aircraft. In *IEEE Aerospace Conference*, pages 3006– 3015, 2004.

[21] Bernard Friedland. Treatment of bias in recursive filtering. *IEEE Transactions on Automatic Control*, 14(4):359–367, August 1969.

[22] Tomonari Furukawa, Frederic Bourgault, and Durrant-Whyte Hugh Lavis, Benjamin. Recursive bayesian search-and-tracking using coordinated uavs for lost targets. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Orlando, FL, May 2006. doi: 10.1109/ROBOT.2006.164208.

[23] Ben Grocholsky, Selcuk Bayraktar, Vijay Kumar, and George Pappas. Uav and ugv collaboration for active ground feature search and localization. In *AIAA 3rd "Unmanned Unlimited" Technical Conference*, Chicago, IL, Sept. 2004. AIAA Paper 2004-6565.

[24] G. Holland, T. McGeer, and H. Youngren. Autonomous aerosondes for economical atmospheric soundings anywhere on the globe. *Bulletin of the American Meteorological Society*, (73):19871998, 1992.

[25] Fei Bin Hsiao, Yu-Hsu Chieri, Tsung-Liang Liu, Mong Tse Lee, Wen-Ying Chang, Siang Yi Han, and Yi-Hsuan Wang. A novel unmanned aerial vehicle system with autonomous flight and auto-lockup capability. In *AIAA Aerospace Sciences Meeting and Exhibit*, Reno NV, 2005. American Institute of Aeronautics and Astronautics. AIAA Paper 2005-1050.

[26] Ole C. Jakobsen and Eric N. Johnson. Control architecture for a uav mounted pan/tilt/roll camera gimbal. In *Infotech@Aerospace*, Arlington VA, 2005. American Institute of Aeronautics and Astronautics. AIAA Paper 2005-7145.

[27] S. Julier, J. Uhlmann, and H.F. Durrant-Whyte. A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control*, 45(3):477–482, 2000.

[28] Simon J. Julier. An empirical study into the use of chernoff information for robust, distributed fusion of gaussian mixture models. In *International Conference on Information Fusion*, Florence, Italy, July 2006. doi: 10.1109/ICIF.2006.301755.

[29] Simon J. Julier and Jeffrey K. Uhlmann. A non-divergent estimation algorithm in the presence of unknown correlations. In *Proceedings of the American Control Conference*, pages 2369–2373, Albuquerque, NM, June 1997. doi: 10.1109/ACC.1997.609105.

[30] Simon J. Julier and Jeffrey K. Uhlmann. Fusion of time delayed measurements with uncertain time delays. In *Proceedings of the American*

Control Conference, pages 4028–4033, Portland, OR, June 2005. doi: 10.1109/ACC.2005.1470607.

[31] K. Kaaniche, B. Champion, C. Pegar, and P. Vasseur. A vision algorithm for dynamic detection of moving vehicles with a uav. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1878–1883, 2005.

[32] K. Kastella, B. Yeary, T. Zadra, R. Brouillard, and E. Frangione. Bias modeling and estimation for gmti applications. In *International Conference on Information Fusion*, volume 1, July 2000.

[33] Barbara La Scala, Robert Bitmead, and Matthew James. Conditions for stability of the extended kalman filter and their application to the frequency tracking problem. *Mathematics of Control, Signals, and Systems*, 8(1):1–26, Mar. 1995. doi: 10.1007/BF01212364.

[34] Jack Langelaan and Steve Rock. Towards autonomous uav flight in forests. In *AIAA Guidance, Navigation, and Control Conference*, San Francisco CA, 2005. American Institute of Aeronautics and Astronautics.

[35] Thomas D. Larsen, Nils A. Andersen, Ole Ravn, and Niels K. Poulsen. Incorporation of time delayed measurements in a discrete-time kalman filter. In *IEEE Conference on Decision and Control*, volume 4, pages 3972–3977, Dec. 1998. doi: 10.1109/CDC.1998.761918.

[36] Martin E. Liggins, Chee-Yee Chong, Ivan Kadar, Mark G. Alford, Vincent Vannicola, and Stelios Thomopoulos. Distributed fusion architectures and algorithms for target tracking. *Proceedings of the IEEE*, 85(1):95–107, Jan. 2006. doi: 10.1109/JPROC.1997.554211.

[37] X. Lin, Y. Bar-Shalom, and T. Kirubarajan. Multisensor multitarget bias estimation for general asynchronous sensors. *IEEE Transactions on Aerospace and Electronic Systems*, 41(3):899–921, July 2005. doi: 10.1109/TAES.2005.1541438.

[38] Tad McGeer and Juris Vagners. Historic crossing: an unmanned aircraft's atlantic flight. *GPS World*, 10(2):24–30, Feb 1999 1999.

[39] Isaac Miller and Mark Campbell. Particle filtering for map-aided localization in sparse gps environments. In *IEEE Conference on Robotics and Automation*, Pasadena, CA, May 2008. doi: 10.1109/ROBOT.2008.4543474.

[40] A. Mutambara. *Decentralized Estimation and Control for Multisensor Systems*. CRC Press, 1998.

[41] A. Mutambara. *Decentralized Estimation and Control for Multisensor Systems*, pages 55–79. CRC Press, Boca Raton, FL, 1998.

[42] Arthur G. O. Mutambara. Information based estimation for both linear and nonlinear systems. In *Proceedings of the American Control Conference*, San Diego, California, 1999. doi: 10.1109/ACC.1999.783583.

[43] Arthur G. O. Mutambara and M. S. Al-Haik. State and information estimation for linear and nonlinear systems. *ASME Journal of Dynamic Systems, Measurement, and Control*, 121(2):318–320, June 1999. doi: 10.1115/1.2802474.

[44] Eric Nettleton and Hugh Durrant-Whyte. Delayed and asequent data in decentralised sensing networks. In *Sensor Fusion and Decentralized Control in Robotic Systems IV*, Boston, MA, Oct. 2001. Society of Photo-Optical Instrumentation Engineers. doi: 10.1117/12.444148.

[45] N. Okello and S. Challa. Joint sensor registration and track to track fusion for distributed trackers. *IEEE Transactions on Aerospace and Electronic Systems*, 40(3):808–823, July 2004. doi: 10.1109/TAES.2004.1337456.

[46] Yaakov Oshman and Pavel Davidson. Optimization of observer trajectories for bearings-only target localization. *IEEE Transactions on Aerospace and Electronic Systems*, 35(3):892–902, July 1999. doi: 10.1109/7.784059.

[47] Mark L. Psiaki. Autonomous low-earth-orbit determination for two spacecraft from relative position measurements. *AIAA Journal of Guidance, Control, and Dynamics*, 22(2):305–312, March-April 1999. doi: 10.2514/2.4379.

[48] Mark L. Psiaki. Autonomous low-earth-orbit determination from magnetometer and sun sensor data. *AIAA Journal of Guidance, Control, and Dynamics*, 22(2):296–304, March-April 1999. doi: 10.2514/2.4378.

[49] M. Quigley, M.A. Goodrich, S. Griffiths, A. Eldredge, and R. W. Beard. Target acquisition, localization, and surveillance using a fixed-wing mini-uav and gimbaled camera. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2600–2605, Barcelona, Spain, Apr. 2005.

[50] B. S. Y. Rao, H. F. Durrant-Whyte, and J. A. Sheen. A fully decentralized

multi-sensor system for tracking and surveillance. *International Journal of Robotics Research*, 12(1):20–44, Feb. 1993. doi: 10.1177/027836499301200102.

[51] M. Ridley, E. Nettleton, S. Sukkarieh, and H. Durrant-Whyte. Tracking in decentralised air-ground sensing networks. In *Proceedings of the Fifth International Conference on Information Fusion*, volume 1, pages 616–623, Annapolis, MD, July 2002. doi: 10.1109/ICIF.2002.1021211.

[52] Luca Schenato. Optimal estimation in networked control systems subject to random delay and packet drop. *IEEE Transactions on Automatic Control*, 53(5):1311–1317, June 2008. doi: 10.1109/TAC.2008.921012.

[53] Philip J. Seddon and Richard F. Maloney. Tracking wildlife radio-tag signals by light fixed-wing aircraft, 2004. New Zealand Department of Conservation Technical Series 30.

[54] Peter Shea, Tim Zadra, Dale Klamer, Ellen Frangione, and Rebecca Brouillard. Precision tracking of ground targets. In *Proceedings of the IEEE Aerospace Conference*, volume 3, pages 473–482, March 2000. doi: 10.1109/AERO.2000.879873.

[55] Suman Srinivasan, Haniph Latchman, John Shea, Tan Wong, and Janice McNair. Airborne traffic surveillance systems: Video surveillance of highway traffic. In *The 2nd ACM Internation Workshop on Video Surveillance and Sensor Networks*, New York, NY, Oct. 2004. doi: 10.1145/1026799.1026821.

[56] V. Stepanyan and N Hovakimyan. Visual tracking of a maneuvering target. AIAA Guidance Navigation and Control Conference, 2006.

[57] Davis Stevenson, Matthew Wheeler, Mark Campbell, William Whitacre, Rolf Rysdyk, and Richard Wise. Cooperative tracking flight test. Hilton Head, SC, Aug. 2007. AIAA Guidance Navigation and Control Conference. AIAA Paper 2007-6756.

[58] Davis Stevenson, Matthew Wheeler, Charlie Matlack, Richard Wise, and William Whitacre. Developing a robust and flexible simulation environment to support cooperative tracking of uavs. Hilton Head, SC, Aug. 2007. AIAA Paper 2007-6565.

[59] Salah Sukkarieh, Eric Nettleton, Jong-Hyuk Kim, Matthew Ridley, Ali Goktogan, and Hugh Durrant-Whyte. The anser project: Data fusion across multiple uninhabited air vehicles. 22(7-8):505–539, July-August 2003. doi: 10.1177/02783649030227005.

[60] S. Tatikonda and S. Mittler. Control under communication constraints. *IEEE Transactions on Automatic Control*, 49(7):1056–1068, July 2004. doi: 10.1109/TAC.2004.831187.

[61] The Insitu Group. The seascan uav. www.insitu.com.

[62] Tom Vercauteren and Xiaodong Wang. Decentralized sigma-point information filters for target tracking in collaborative sensor networks. *IEEE Transactions on Signal Processing*, 53(8):2997–3009, 2005. doi: 10.1109/TSP.2005.851106.

[63] Ick H. Wang, Vladimir N. Dobrokhodov, Isaac I. Kaminer, and Kevin D. Jones. On vision-based target tracking and range estimation for small uavs. In *AIAA Guidance Navigation and Control Conference*, San Francisco CA, Aug. 2005. AIAA Paper 2005-6401.

[64] M. Wheeler, B. Schrick, W. Whitacre, M. Campbell, R. Rysdyk, and R. Wise. Cooperative tracking of moving targets by a team of autonomous uavs. Digital Avionics Systems Conference, 2006.

[65] William Whitacre and Mark Campbell. Cooperative geolocation with uavs under communication constraints. Hilton Head, SC, August 2007. AIAA Guidance Navigation and Control Conference. AIAA Paper 2007-6759.

[66] William Whitacre and Mark Campbell. Cooperative geolocation and sensor bias estimation for uavs with articulating cameras. Chicago, IL, Aug. 2009. AIAA Guidance Navigation and Control Conference. AIAA Paper 2009-6220.

[67] William Whitacre, Mark Campbell, Matthew Wheeler, and Davis Stevenson. Flight results from tracking ground targets using seascan uavs with gimballing cameras. New York, NY, July 2007. Proceedings of the American Control Conference. doi: 10.1109/ACC.2007.4282696.

[68] R. Wise and R. Rysdyk. Uav coordination for autonomous target tracking. Keystone, CO, Aug. 2006. AIAA Guidance Navigation and Control Conference. AIAA Paper 2006-6453.

[69] W.S. Wong and R.W. Brockett. Systems with finite communication bandwidth constraints. i. state estimation problems. *IEEE Transactions on Automatic Control*, 42(9):1294–1299, Sept. 1997. doi: 10.1109/9.623096.

[70] J. Wu and T. Chen. Design of networked control systems with packet

dropouts. *IEEE Transactions on Automatic Control*, 52(7):1314–1319, July 2008. doi: 10.1109/TAC.2007.900839.