UGV-TO-UAV COOPERATIVE RANGING FOR ROBUST NAVIGATION IN GNSS-CHALLENGED ENVIRONMENTS

Victor O Sivaneri

Follow this and additional works at: https://researchrepository.wvu.edu/etd

Recommended Citation

This Dissertation is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Dissertation in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/or on the work itself. This Dissertation has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.
UGV-TO-UAV COOPERATIVE RANGING FOR ROBUST NAVIGATION IN GNSS-CHALLENGED ENVIRONMENTS

Victor O. Sivaneri

Dissertation
Benjamin M. Statler College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in
Aerospace Engineering

Jason N. Gross, Ph.D., Chair
Patrick Browning, Ph.D.
Yu Gu, Ph.D.
Powsiri Klinkachorn, Ph.D.
Mario Perhinschi, Ph.D.

Department of Mechanical and Aerospace Engineering

Morgantown, West Virginia
2018

Keywords: Cooperative Navigation, Global Navigation Satellite Systems Challenged, Unmanned Aerial Vehicles, Sensor Fusion

Copyright © 2018 Victor O. Sivaneri
Abstract

UGV-to-UAV Cooperative Ranging for Robust Navigation in GNSS-Challenged Environments

Victor O. Sivaneri

Quadcopters have been used increasingly throughout the world for a wide range of applications, such as: delivering packages, cave or tunnel reconnaissance, building assessment, forest inventory, and many more. This increased use of Unmanned Aerial Vehicles (UAVs), especially in areas of Global Navigation Satellite System (GNSS)-challenged environments, shows a need to develop algorithms that are robust to these environments. To address this, this dissertation considers the cooperative ranging between a UAV, operating in a GNSS-challenged environment, with an Unmanned Ground Vehicle (UGV), and focuses on the design of the optimal motion of the UGV to best aid the UAV’s navigation solution. The overall goal of this research is to reduce the positioning error of the UAV. To achieve this goal, the project was divided into three phases. First, three novel cooperative strategies and two different estimation strategies for the UGV to assist the UAV were developed and compared. Each of these approaches reduces the uncertainty of a UAV’s navigation solution through the use of ranging measurement updates from a cooperative UGV, whose location is designed to improve positioning geometry of the UAV. During the first phase of the project, a Monte-Carlo simulation was employed to show that: UGV-to-UAV cooperative navigation can reduce the positioning error of a UAV operating in a GNSS-challenged environment by more than tenfold (i.e., 1-meter-level to 10-cm-level 3D positioning error) and the best approach to design the UGVs trajectory is a constrained nonlinear optimization that takes into account the GNSS satellites available to the UAV and the location of the UAV. During the second phase of the
project, the algorithm is extended to consider the benefits of employing a UGV path planning approach that leverages prior knowledge of the UAVs planned mission and the ephemeris of the GNSS satellites, and to consider employing multiple UGVs to help a UAVs navigation solution. Finally, during the last phase of the project, a series of experimental flight tests were conducted using an instrumented UAV and an instrumented UGV in order to demonstrate the applicability of this research, specifically the cooperative strategies.
Acknowledgments

I would like to thank my graduate research advisor, Dr. Jason Gross. Without your support and guidance, this project would not have been completed. I am glad to have been able to work on interesting projects throughout my time in your research group. I appreciate the time and effort you spent working with and teaching me over the course of this dissertation. Thank you.

I would also like to thank the other members of my committee, Dr. Patrick Browning, Dr. Yu Gu, Dr. Powsiri Klinkachorn, and Dr. Mario Perhinschi for their assistance and feedback that they provided on this project.

Thank you to my family who have supported me through everything I have done. Thanks to my parents, Nithi and Meena, for your continued love and support and for getting me to this point in my life and career. You have shown me that through hard work and dedication, I can achieve anything. My brother, Kevin, for your love and support. My sister, Mona, thanks for your love and support and keeping me sane throughout this process and always giving me a laugh when needed.

I would like to thank my fellow graduate students, past and present, for their help and support throughout my years: Shannen Daly, Stephane D’Urso, Jeremy Hardy, Jacob Hikes, Cagri Kilic, Sean Lantto, Nick Ohi, Scott Harper, Andrew Rhodes, Nate Tehrani, and Ryan Watson.

Finally, I would like to thank the U.S. National Geospatial-Intelligence Agency Academic Research Program (NARP), grant # HM0476-15-1-0004, for their support in this research. Approved for public release, 18-559.
Contents

Acknowledgments iv

List of Figures viii

List of Tables xii

1 Introduction 1
   1.1 Problem Overview ............................................. 1
   1.2 Concept Overview ................................................ 3
   1.3 Dissertation Outline .......................................... 5

2 Literature Review 6
   2.1 Global Navigation Satellite Systems ............................. 7
       2.1.1 Multipath ................................................... 10
   2.2 Cooperative Navigation .......................................... 12
       2.2.1 Ultra-Wideband Radios ..................................... 16
   2.3 GPS-Denied Navigation Techniques ............................... 18
   2.4 Path Planning ...................................................... 19
   2.5 Nonlinear Optimization for UAVs ................................ 21

3 Research Objectives .................................. 24
4 Background

4.1 Global Navigation Satellite System ..................................... 26
  4.1.1 Differential Global Navigation Satellite System .............. 30

4.2 Dilution of Precision ...................................................... 30

4.3 Extended Kalman Filter .................................................. 33

4.4 Kalman Filter .............................................................. 33
  4.4.1 Prediction ............................................................. 35
  4.4.2 Measurement Update ................................................ 35
  4.4.3 Process and Measurement Noise .................................... 36

4.5 Nonlinear Optimization .................................................. 37
  4.5.1 Unconstrained Optimization ....................................... 37
  4.5.2 Constrained Optimization .......................................... 40

5 Estimation Filter Design .................................................. 43

5.1 Reducing PDOP with a Single Ranging Source ......................... 43

5.2 Undifferenced GNSS Extended Kalman Filter ........................ 45
  5.2.1 Undifferenced GNSS EKF Design ................................ 45
  5.2.2 Undifferenced GNSS EKF Augmented with Cooperative Ranging 47

5.3 Differential GNSS Extended Kalman Filter ........................... 49
  5.3.1 Differential GNSS EKF Augmented with Cooperative Ranging 52
  5.3.2 Integer Ambiguity Fixing .......................................... 54

6 UGV Strategies ............................................................. 56

6.1 Locally Greedy Strategy .................................................. 56

6.2 Regionally Optimal Strategy ............................................. 59

6.3 Nonlinear Optimization Strategy ....................................... 61
  6.3.1 Path Planning ....................................................... 65
  6.3.2 Multiple Unmanned Ground Vehicles ............................. 68
7 Simulation Setup

8 Experimental Setup

8.1 Unmanned Aerial Vehicle 74
8.2 Unmanned Ground Vehicle 80
8.3 Experimental Procedure 83
8.4 Data Acquisition 86
8.5 Reference Solution Post-Processing 88

9 Results

9.1 One UAV and One UGV 92
  9.1.1 Nonlinear Optimization 100
  9.1.2 Incorporating Other Satellite Systems 107
9.2 Multiple UGVs 110
9.3 Experimental Results 112

10 Conclusions and Future Work

10.1 Conclusion 116
10.2 Future Work 117
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Concept diagram for cooperative navigation with ultrawide band (UWB) ranging between a UAV and UGV</td>
<td>3</td>
</tr>
<tr>
<td>2.1</td>
<td>Multipath example in an <em>Urban Canyon</em></td>
<td>12</td>
</tr>
<tr>
<td>2.2</td>
<td>Classes and methods for numerical optimization</td>
<td>22</td>
</tr>
<tr>
<td>4.1</td>
<td>Comparison of ‘poor’ PDOP and ‘good’ PDOP</td>
<td>31</td>
</tr>
<tr>
<td>5.1</td>
<td>Monte Carlo simulation result with a 55-degree elevation mask that illustrates the potential improvement of including a single additional ranging source that is optimally placed.</td>
<td>44</td>
</tr>
<tr>
<td>6.1</td>
<td>Regional optimal strategy using a grid to calculate minimum augmented-PDOP</td>
<td>59</td>
</tr>
<tr>
<td>6.2</td>
<td>Diagram of heading calculation between two UGV locations.</td>
<td>60</td>
</tr>
<tr>
<td>6.3</td>
<td>Nonlinear optimization strategy to calculate percent reduction of augmented-PDOP</td>
<td>65</td>
</tr>
<tr>
<td>7.1</td>
<td>Left panel: Elevation and Azimuth before and after high Elevation mask was applied Right panel: PDOP before and after high Elevation mask was applied</td>
<td>71</td>
</tr>
<tr>
<td>8.1</td>
<td>Initial UAV setup</td>
<td>75</td>
</tr>
</tbody>
</table>
8.2 Navigation solution using an Ublox GPS receiver and antenna, with an initialization time of 10 minutes ................. 76
8.3 Navigation solution using an Ublox GPS receiver and antenna .... 76
8.4 2nd UAV setup ........................................... 77
8.5 Novatel receiver and breakout board ........................ 78
8.6 Navigation solution using a Novatel GPS receiver and antenna .. 78
8.7 Xbee modem on the UAV ..................................... 79
8.8 Netburner on the UAV ....................................... 79
8.9 Final UAV setup ............................................. 79
8.10 Block diagram of UAV systems ............................ 80
8.11 UGV ............................................................ 81
8.12 Xbee modem on the UGV .................................. 82
8.13 Block diagram of UAV systems ............................ 83
8.14 Base station setup with laptop ............................. 84
8.15 UAV acquiring lock on satellites before flight ............... 84
8.16 UGV computer and program being initialized before starting movement ..................................................... 85
8.17 UAV flying and UGV moving according to algorithm ......... 86
8.18 UGV real-time software architecture ....................... 88
8.19 Left panel: Elevation and Azimuth before and after high Elevation mask was applied Right panel: PDOP before and after high Elevation mask was applied ........................................ 91
9.1 Simulation overview of the UAV with the cooperative UGV .... 93
9.2 Flight simulation example of the UAV with the cooperative UGV using a Locally Greedy (Left Panel) and Regionally Optimal (Right Panel) path planning strategy. ........................................ 94
9.3 Example flight trial of the UAV’s PDOP without UGV, with a *Locally Greedy* UGV, and with a *Regionally Optimal* UGV.

9.4 Example flight trial of the UAV’s positioning errors, for both filter designs, without a UGV, with a *Locally Greedy* UGV, and with a *Regionally Optimal* UGV.

9.5 3D position error of median of data sets with the undifferenced GNSS EKF

9.6 3D position error of median of data sets with the double–differenced GNSS EKF

9.7 25 flights of the 3 strategies and the undifferenced and the double–differenced Float and Fixed 3D Error

9.8 PDOP reduction of nonlinear optimization vs *Regionally Optimal* approach

9.9 PDOP reduction of nonlinear optimization between 10-step path planning and one-step path planning

9.10 Example of UGV path for the different methods: one-step path planning vs. 10-step path planning

9.11 PDOP reduction of nonlinear optimization between UAV’s position with no error and UAV’s position with error

9.12 PDOP reduction of nonlinear optimization between UAV’s position with error but weighting and UAV’s position with error but no weighting

9.13 PDOP reduction of nonlinear optimization between UAV with no positioning error and UAV’s position with error and with weighting

9.14 PDOP reduction of nonlinear optimization of UAV with positioning error and with weighting vs *Regionally Optimal* approach

9.15 Number of satellites, GPS + Galileo vs. GPS only, visible over one flight

9.16 PDOP comparison for GPS + Galileo vs. GPS + UGV, for one flight
9.17 % PDOP reduction of two UGVs from one UGV  

9.18 % PDOP reduction of two UGVs: decentralized from centralized  

9.19 Path of UAV (blue) and UGV (green)  

9.20 Comparison of PDOP  

9.21 East, North, Up position error (m) for the UAV with UGV cooperative ranging and without UGV cooperative ranging
List of Tables

7.1 Simulation parameters ............................................. 70
7.2 Parameters for nonlinear optimization algorithm ............... 72
8.1 Parameters for RTKPOST ........................................... 89
9.1 Summary of positioning performance with and without UGV cooperative navigation, for both control strategies and both filter designs. ................................................................. 98
9.2 Mean PDOP comparison for 50 data sets: nonlinear optimization cases versus Regionally Optimal .................................................. 101
9.3 Mean PDOP comparison for 50 data sets: nonlinear optimization one-step versus 10-step path planning ............................. 102
9.4 Mean sum PDOP comparison for 50 data sets: 10-step path planning nonlinear optimization cases ........................................ 106
9.5 Results for comparison of multi-satellite systems vs. multi-vehicles .. 110
9.6 Mean PDOP comparison for 50 data sets: one UGV versus two UGVs 111
9.7 Results for all experimental data ........................................ 114

xii
### Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>CODE</td>
<td>Center for Orbit Determination</td>
</tr>
<tr>
<td>DOP</td>
<td>Dilution of Precision</td>
</tr>
<tr>
<td>GDOP</td>
<td>Geometry Dilution of Precision</td>
</tr>
<tr>
<td>GNSS</td>
<td>Global Navigation Satellite System</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>DGNSS</td>
<td>Differential Global Navigation Satellite System</td>
</tr>
<tr>
<td>EKF</td>
<td>Extended Kalman Filter</td>
</tr>
<tr>
<td>IF</td>
<td>Ionosphere-Free</td>
</tr>
<tr>
<td>IGS</td>
<td>International GNSS Service</td>
</tr>
<tr>
<td>IMU</td>
<td>Inertial Measurement Unit</td>
</tr>
<tr>
<td>ITS</td>
<td>Intelligent Transportation Systems</td>
</tr>
<tr>
<td>LAMBDA</td>
<td>Least-squares AMBiguity Decorelation Adjustment</td>
</tr>
<tr>
<td>LLS</td>
<td>Linear Least Squares</td>
</tr>
<tr>
<td>MAV</td>
<td>Miniature Air Vehicles</td>
</tr>
<tr>
<td>PPP</td>
<td>Precise Point Positioning</td>
</tr>
<tr>
<td>RMS</td>
<td>Root Mean Square</td>
</tr>
<tr>
<td>STM</td>
<td>State Transition Matrix</td>
</tr>
<tr>
<td>TOA</td>
<td>Time of Arrival</td>
</tr>
<tr>
<td>UAV</td>
<td>Unmanned Air Vehicle</td>
</tr>
<tr>
<td>UGV</td>
<td>Unmanned Ground Vehicle</td>
</tr>
<tr>
<td>URE</td>
<td>User Range Error</td>
</tr>
<tr>
<td>UWB</td>
<td>UltraWide Band</td>
</tr>
<tr>
<td>WVU</td>
<td>West Virginia University</td>
</tr>
</tbody>
</table>
## Nomenclature

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c$</td>
<td>m</td>
<td>Speed of light</td>
</tr>
<tr>
<td>$\delta t_u$</td>
<td>s</td>
<td>Receiver clock bias</td>
</tr>
<tr>
<td>$\delta t^k$</td>
<td>s</td>
<td>Satellite clock bias</td>
</tr>
<tr>
<td>$e_l$</td>
<td>deg</td>
<td>Elevation angle</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>m</td>
<td>Un-modeled error sources</td>
</tr>
<tr>
<td>$G$</td>
<td></td>
<td>Geometry matrix</td>
</tr>
<tr>
<td>$H$</td>
<td></td>
<td>Observation matrix</td>
</tr>
<tr>
<td>$I$</td>
<td></td>
<td>Identity matrix</td>
</tr>
<tr>
<td>$K$</td>
<td></td>
<td>Kalman Gain</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>MHz</td>
<td>Wavelength</td>
</tr>
<tr>
<td>$N$</td>
<td></td>
<td>Carrier-phase bias</td>
</tr>
<tr>
<td>$n$</td>
<td></td>
<td>Satellites in view</td>
</tr>
<tr>
<td>$P$</td>
<td></td>
<td>Error Covariance matrix</td>
</tr>
<tr>
<td>$Q$</td>
<td></td>
<td>Process noise covariance matrix</td>
</tr>
<tr>
<td>$R$</td>
<td></td>
<td>Measurement noise covariance matrix</td>
</tr>
<tr>
<td>$\rho$</td>
<td>m</td>
<td>Pseudorange vector</td>
</tr>
<tr>
<td>$\sigma_{URE}$</td>
<td></td>
<td>Standard deviation of the User Range Error</td>
</tr>
<tr>
<td>$t^k$</td>
<td>s</td>
<td>Time signal was transmitted by satellite</td>
</tr>
<tr>
<td>$t_u$</td>
<td>s</td>
<td>Time signal was received by the user</td>
</tr>
<tr>
<td>$T$</td>
<td></td>
<td>Troposphere Delay</td>
</tr>
<tr>
<td>$\dot{v}$</td>
<td>m</td>
<td>UWB predicted range</td>
</tr>
<tr>
<td>$\Phi$</td>
<td></td>
<td>State transition matrix</td>
</tr>
<tr>
<td>$\phi$</td>
<td></td>
<td>Carrier-phase</td>
</tr>
<tr>
<td>$x$</td>
<td></td>
<td>State vector</td>
</tr>
<tr>
<td>$\psi$</td>
<td>deg</td>
<td>UGV heading angle</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Portions of this chapter were described in a 2016 ION GNSS+ Conference Proceedings article (Sivaneri and Gross, 2016) [1] and a 2017 AST Journal Paper (Sivaneri and Gross, 2017) [2]

1.1 Problem Overview

Global Positioning System (GPS) has been around for a few decades. Since then the applications for using GPS has skyrocketed from hand-held devices, to cars, planes, robots, drones, and many more applications [3]. One of the many fields that has benefited from this technology is Unmanned Aerial Vehicles (UAVs). UAVs have had a great increase in capability in recent years due to instruments such as GPS, inertial measurement unit (IMU), cameras, and a plethora of other sensors that have become more accurate and lighter, smaller, and cheaper allowing them to be placed on these vehicles. This has enabled many advancements in the field of robotics and allowed for a wide range of new research, such as robots being autonomous, using drones for surveillance, search and rescue missions in dangerous environments, and various indoor and outdoor applications.

While there are many applications that use GPS on UAVs, there are many situa-
tions where GPS may not be readily available due to jamming or because of the environment. While adding more sensors may be helpful, there may be a weight constraint for the UAVs. One way to avoid using additional sensors is the use of cooperative or collaborative vehicles to assist with positioning in these scenarios. Collaborative or cooperative navigation research has been increasingly active in recent years, especially in support of military operations [4] [5] and Intelligent Transportation Systems [6] (ITS). This is due to the fact that these applications are oftentimes confronted with GNSS-challenged environments. For example, vehicles may go from situations where GNSS is readily available to completely denied (e.g., being jammed). In the context of autonomous driving, urban canyons often lead to extreme multipath, complete GNSS blockages and/or the reception of non-Line of Sight (LOS) signals, which can lead to position errors as large as hundreds of meters [7]. These scenarios are a few of the better-known cases in which collaborative or cooperative navigation would be beneficial. Other potential applications include the use of a small quadrotor UAV for bridge inspection or structural health monitoring of a building [8], for surveillance applications in urban environments [9], or applications that require quadrotor UAVs to transition from indoor-to-outdoor operations or vice versa [10].

The problems listed convey a need to be able to use UAVs within a GNSS-challenged scenario, specifically an urban canyon. In a GNSS-challenged environment, such as an urban canyon, the positioning of the vehicle can be severely degraded. In an urban canyon environment, the number of satellites is tremendously reduced due to buildings blocking the satellite signal and signals bouncing off buildings to create multipath, an effect on some signals. In this environment, there is a need to be able to navigate safely through an urban canyon. This dissertation aims to solve this problem through cooperative ranging for robust navigation between UAVs and UGVs.
1.2 Concept Overview

Figure 1.1 [1] shows the assumed set-up for the cooperative navigation design within the context of an urban canyon setting. In this case, the UAV is considered to be in a GNSS-challenged environment (urban canyon). It is further assumed that the cooperative vehicle is not GNSS-challenged. For example, the UGV is assumed to reside in a city intersection with an open-sky access, just outside the forest canopy, not directly under the bridge being inspected, etc.

In this context, (1) a single UGV acting as a ranging source is able to yield a wide range of geometry with respect to the location of a UAV and (2) a UGV is naturally positioned to improve a UAV’s navigation solution geometry as it emanates its ranging signal from the surface of the Earth, a location that a GNSS transmitter cannot be located. By leveraging these characteristics, this cooperative navigation algorithm is shown to yield significant increases in the accuracy of the positioning of the UAV faced with GNSS-challenged conditions.

This study employs the use of a cooperative ranging signal for cooperative navigation in order to develop a system architecture that increases the UAV’s navigation
accuracy in an *urban canyon*. To accomplish this main objective, this research was composed of three phases:

**Phase 1:** Three cooperative strategies and two estimation strategies are developed. During this phase of the research study, three cooperative strategies, a *Locally Greedy* approach, a *Regionally Optimal* approach, and a nonlinear optimization approach is developed to control the movement of the secondary vehicle, the UGV. To compare these strategies, an Undifferenced GNSS Extended Kalman Filter and a Differential GNSS Extended Kalman Filter were used to compare the positioning accuracy of the UAV using simulation data.

**Phase 2:** Expanding the study for path planning and multiple cooperative vehicles. Using the knowledge gained from the previous phase, the goal of this phase is to incorporate a path planning approach for the UGV, utilizing the UAV’s path and the satellite ephemeris data. In addition, comparing the use of multiple satellite systems versus using a cooperative vehicle was studied.

**Phase 3:** Designing an experimental setup. The final phase of this project is to build a UAV and a UGV to test a cooperative strategy in the field. Also, the experimental data is processed using a Differential GNSS Extended Kalman Filter to demonstrate the navigational accuracy of the UAV that could be attained with the use of a cooperative vehicle.

While this study focuses on the aim of solving the problem of navigating through an *urban canyon*, the theory and algorithms developed can be used as a basis for other *GNSS-challenged* scenarios. This technology can be used for cooperatively navigating cars through a city, or having swarms of UAVs flying though a city. There is a significant need for this type of technology, as cars and other vehicles are being driven autonomously and there are many situations where GNSS can be readily available to GNSS-denied situations. This study shows the benefit of having a cooperative vehicle help the positioning accuracy of another vehicle, both in simulation and in
experimental applications. In addition to the theoretical contributions, this research also includes hardware and software designs needed to apply this research in an experimental situation. The goal of the study is to use a cooperative ranging source for robust navigation in a GNSS-challenged situation. The cooperative strategies and estimation strategies designed are validated with simulation and experimental data, which shows the feasibility and reliability of this research.

1.3 Dissertation Outline

The remaining dissertation is organized as follows. Chapter 2 is an overview of the past research in the fields of Global Navigation Satellite Systems, Cooperative navigation, path planning, Kalman filter, and Nonlinear optimization. Chapter 3 describes the objectives of this dissertation. Chapter 4 provides an overview of the Global Navigation Satellite System, Extended Kalman Filter, the differential GNSS Extended Kalman Filter, cooperative navigation, nonlinear optimization, and path planning approach. Chapter 5 describes the formulation for the Dilution of Precision, the Undifferenced GNSS Extended Kalman, and the Differential GNSS Extended Kalman Filter. Next, Chapter 6 discusses the three different UGV trajectory control strategies. Chapter 7 presents the simulation environment used for this work. Chapter 8 discusses the UAV and UGV components and system software, and the experimental setup. Chapter 9 discusses the results of a series of flight simulations of one UAV and one UGV and also one UAV and multiple UGVs, as well as results for the experimental demonstration. The final chapter, Chapter 10 summarizes this dissertation and discusses the future work.
Chapter 2

Literature Review

Portions of this chapter were described in a 2016 ION GNSS+ Conference Proceedings article (Sivaneri and Gross, 2016) [1] and a 2017 AST Journal Paper (Sivaneri and Gross, 2017) [2]

In this chapter, the literature review will provide an overview of the Global Navigation Satellite System (GNSS), cooperative navigation, GPS-denied navigation techniques, path planning, the Kalman Filter, and the Nonlinear Optimization for UAVs. The GNSS section entails the different satellite systems used throughout the world how they are similar, how they differ from each other, and the effects of multipath in an urban canyon. A description of cooperative navigation and the difference between cooperative and collaborative navigation is discussed further in the next section, as well as the use of Dilution of Precision and Ultra-Wideband radios in cooperative navigation. The different types of cooperative navigation and the relevant types are summarized in the cooperative navigation section. Next, GPS-denied navigation techniques will be discussed. Then, an overview of path planning in the field of UAVs and ground rovers are discussed. There are a wide variety of path planning techniques, and the ones pertaining to cooperative navigation will be reviewed in this section. An introduction to the Kalman Filter will be discussed. Finally, the background of
nonlinear optimization is reviewed and the use of how it is incorporated in cooperative navigation.

2.1 Global Navigation Satellite Systems

Global Navigation Satellite Systems (GNSS) is a generic term for all the satellite navigation systems in the world. Currently there are six operational systems, the United States Global Positioning System (GPS), the Russian GLONASS, the European Galileo system, the Chinese BeiDou Navigation Satellite system known as BeiDou, the Indian Navigation Indian Constellation (NavIC), and the Japan Quasi-Zenith Satellite System. These systems provide the user with a three-dimensional positioning solution by using signals transmitted by satellites orbiting the earth. But each system has some similarities and some differences, which will be discussed throughout this section.

GPS is a global 3D passive, trilateration, time-of-arrival (TOA) navigation system [11]. The satellites broadcast ranging signals and navigation data allowing users to measure their pseudoranges in order to measure their position, velocity, and time [12]. The ranging signals indicate the time when the signals were received, and the navigation data message includes information about the satellite’s orbit and timing parameters.

The architecture for GNSS consist of three segments: 1) space segment, 2) control or ground segment, and 3) user segment [13]. The space segment are the satellites. For GPS, a minimum of 24 satellites are operational, but up to 32 can be in service at a given time, for GLONASS there are 27 total satellites with 24 in full service, for Galileo 24 satellites are in full service and 6 spares, and BeiDou-2 will have 27 satellites in operation. For a user with an open-sky, a minimum of four GPS satellites are visible, but typically 6-8 GPS satellites are within view. The GPS satellites are
in a circular orbit with approximately a 12-hour period. There are 6 orbital planes each inclined at 55 degrees from the equator. Within these orbital planes, there lie 4 satellites and at least one spare satellite per plane. For GLONASS, Galileo, and Beidou-2, there are 3 orbital planes inclined at 64.8, 56, and 55 degrees, respectively [11].

The control or ground segment monitors, maintains, and reports satellite health, maintains the GPS time scale, predicts satellite orbits and clocks, and performs maneuvers, if needed. For GPS, there are two master control stations (a primary one and a backup), four data uploading stations, and 16 monitoring stations, positioned throughout the world [14]. The master control station adjusts the satellites’ orbit parameters and on-board high-precision clocks, when necessary, to maintain accuracy. The monitor stations monitor the satellites’ status and signals and sends this information to the master control station. There are 17 monitor systems and are run by the Department of Defense in Colorado Springs, 11 by the National Geospatial Intelligence Agency, and 6 by the United States Air Force. The user segment comprises the equipment that processes the received signals from the satellites and uses them to derive location and time information. This equipment consists of antennas, receivers, processors, and power supplies [11]. In this study, mainly GPS is used for implementation in the filters, but there is a comparison study between using multiple constellations, such as GPS and Galileo, to having multiple cooperative ranging vehicles.

A GNSS receiver calculates its position based on information from the satellites, but there are many error sources pertaining to this process. If left uncorrected, these errors can cause position errors to be quite large [11]. GPS error sources can be grouped into 3 categories: 1) propagation medium model uncertainty, 2) control segment errors, 3) measurement noise and RF interference.

The propagation medium model uncertainty deals with the errors associated with
the atmosphere, specifically ionosphere and troposphere [11]. The ionosphere is the layer of atmosphere between 80 km and 600 km above the earth and is a layer of ionized gases, free electrons, and ions. The ionospheric delay can cause significant satellite positioning error, typically ± 5 meters [15]. The ionization is the work of the sun’s radiation, meaning the state of the ionosphere varies with the intensity of solar activity, season, time of day, and location [11]. This variation in delay makes it very difficult to predict how much ionospheric delay is impacting the calculated position. A benefit of the delay due to ionosphere is that it also varies based on the radio frequency of the signal source. As mentioned, GPS signals operate on the L1 and L2 band, for a receiver having dual frequency capability, it can use this to its advantage. Galileo operates on a similar frequency as GPS, 1575.42 MHz for L1 and 1176.45 MHz for L2. Where GLONASS uses a range for the frequencies, L1 has a range of 1598.0625 - 1609.3215 MHz and a L2 band of 1242.9375 - 1251.6875 MHz [16]. This technique is known as frequency division multiple access (FDMA). Whereas Beidou has signal characteristics of 1561.098 MHz for B1 and 1207.140 MHz for B2. By comparing the measurements from L1 and from L2, the receiver can determine the amount of ionospheric delay [17]. The troposphere, is the layer of atmosphere right above the surface of the Earth and is composed of dry gases (mainly $N_2$ and $O_2$) and water vapor [13]. Variations in troposphere delay are caused by the change in humidity, temperature, and atmospheric pressure in the troposphere. Unlike the ionosphere, the troposphere is non-dispersive for GPS frequencies thus the delay cannot be estimated from GPS measurements and must be estimated from models [15].

The control segment errors are consisted of the errors in the broadcast orbits and clocks presented in the navigation message [11]. The atomic clocks in the GNSS satellites are very accurate but do drift a small amount. Unfortunately, even a small amount has a profound effect on position error, a 10-nanosecond clock error would result in a 3-meter positioning error. Similarly, the broadcast orbit errors do vary
slightly from their known orbits resulting in a significant error in the position. These errors are minimized by the ground control system, who is monitoring the uncertainty and sending corrections up to the satellite [17].

The measurement noise and RF interference pertain to receiver noise and multipath. Receiver noise refers to the positioning errors due to the GPS hardware, the antenna, amplifiers, cables, receiver, interference from other GPS signals, and signal quantization noise [11]. Multipath refers to the occurrence of a signal reaching an antenna via two or more paths. This occurs when a GPS signal is reflected off an object, such as a building, to the GPS antenna [18]. In an urban canyon, this phenomenon occurs frequently and is hard to model due to the difference in building materials. This delay can cause the receiver to calculate an incorrect position. In this study, these individual noise sources are modeled and discussed later on.

2.1.1 Multipath

As previously mentioned, GNSS provides an accurate and reliable positioning system, as long as each satellite’s signal travels along a direct path directly to the receiver’s antenna. But in an urban canyon this reliability degrades, due to signal blockage and reflections caused by tall buildings, tremendously. These reflections off buildings, as displayed in Figure 2.1, and other surfaces are known as multipath. The GPS receiver could detect the same signal twice but at different ranges [19]. Since the multipath signal takes a longer path, therefore a longer propagation time than the direct signal, it results in an error in the pseudorange measurements which then affects the positioning accuracy [20]. For a static receiver, the propagation geometry varies slightly, and the multipath parameters are almost constant for a short duration. But for moving receivers, it can experience a wide range of values in fractions of a second [21]. Thus, multipath must be taken into account for this dissertation, as the UAV will be moving.
There are several multipath-reduction techniques that try to take advantage of signal propagation geometry such as using a special antenna, having multiple antennas, or having a static receiver and measuring the multipath parameters [20]. Using a special antenna, adding a metallic disk on the bottom of the antenna’s base, acts as an impedance at the GPS signal frequency. The antenna gets protection from multipath from the ground and the horizontal directions, by use of this disk [20]. While this added disk provides some protection to multipath, this does not help much in an urban canyon, where most of the reflected signals are from above [20]. This ring would also add weight to the antenna which is not practical for a UAV flying through a city block. Employing the multiple antennas tactic, would not be applicable for this research study due to the fact that buildings might not be made of the same material and would require another UAV. For the last tactic stated, having a stationary antenna, for this dissertation the UAV is moving through an urban canyon and will not be staying in one place for a significant amount of time.

While these multipath mitigation techniques are more on the hardware side, there are software related solutions to mitigate the multipath. Many methods have been implemented to mitigate multipath errors at the receiver baseband signal processing level, narrow correlator [22], strobe and edge correlator [23], gated correlator, and high-resolution correlator (HRC) [24]. HRC reduces the code phase multipath by using special designed reference signals to correlate with the incoming signal corrupted by multipath. A disadvantage of this algorithm is the poor performance for short-delayed multipath signals [25]. Multipath mitigation techniques (MMT) can estimate the code and the carrier–phases of both LOS and multipath signals, but in order to perform well, this method needs a higher sampling frequency which leads to a higher computational cost [26]. In the absence of multipath, the delay lock loop (DLL) implements an approximation of a maximum likelihood (ML) time-delay estimator, known as multipath estimation delay lock loop (MEDLL) [27]. The advantages and
disadvantages of MEDLL are the same as MMT [24].

2.2 Cooperative Navigation

Collaborative or cooperative navigation research has been increasingly active in recent years, especially in support of military operations and Intelligent Transportation Systems (ITS). This is due to the fact that these applications are oftentimes confronted with *GNSS-challenged* environments. These scenarios are a few of the better-known cases in which collaborative or cooperative navigation would be beneficial.

The difference between collaborative navigation and cooperative navigation is the ability for the user to independently estimate their accurate position [28]. That
is, in cooperative navigation, vehicles help each other to determine their locations, but it is often assumed that each vehicle cannot independently determine its own position to a sufficient level of accuracy [29]. Collaborative positioning techniques, on the other hand, typically use locally available or opportunistic measurements, such as measurements from neighboring vehicles or civil infrastructure, to help reduce navigation uncertainty. Many collaborative navigation techniques focus on the use of Vehicle Ad-Hoc Networks (VANETs), where multi-sensor fusion is used on individual nodes, and the collaboration between nodes is opportunistic in nature. Most of the collaborative navigation communication is with a ranging signal for vehicle-to-vehicle and vehicle-to-infrastructure communication [30] and does not focus on the optimal geometry of the network, but instead use what is available. The work herein focuses on cooperative navigation between a UAV and UGV, in which ranging measurement updates are provided by a UGV, and the motion of the UGV is designed to provide the most favorable geometry.

An additional cooperative navigation system has been developed in [31], where multiple miniature air vehicles (MAVs) work cooperatively to be able to navigate in a GNSS-degraded or denied area. In this case, the MAVs are assumed to be instrumented with Inertial Measurement Units (IMUs) which is shared amongst other MAVs and coupled with range and/or bearing measurement in order to estimate the relative position and velocity of itself and the other MAVs. The authors put the restraint that the MAVs are moving in level flight and the velocity of the MAV is aligned with the wind frame. The work assumes that the MAVs are unable to stop, which would not be helpful in an urban canyon setting with a UAV doing a building inspection. Similarly, cooperative localization of a multiple fixed-wing UAVs in a GPS-denied scenario was also recently investigated in [32], where dynamic model information, IMU, airspeed, and altimeter data were shared across UAVs to better determine position with updates from a ranging source between each UAV. Similar
to the current study, an EKF-based centralized cooperative localization algorithm was implemented. Centralized and multi-centralized schemes are better than decentralized for optimal solutions [33] but have practical implementation issues and are not scalable. Similar methods have also been studied on a group of robots [34], in which relative position was estimated by using wheel encoders and multiple cameras to implement bearing-only cooperative localization. The limitation in this work is the system must be observable for the localization errors to be bounded. In these three works, the use of cooperative navigation was implemented to assist in positioning and the secondary vehicles are moving too. However, the optimization of the secondary vehicles position, to improve the positioning of the main vehicle is not implemented as the focus of the dissertation.

While the previous works did not emphasize the control or design of the location of the moving cooperative nodes, these next papers [35] [6] [36] [37] discuss the design process of a localization system for a team of vehicles/robots. [35] used stochastic mapping and navigation to have an algorithm to control the robot and was shown to be superior to straight-line motion and random motion. The previous study used an EKF-based Simultaneous localization and mapping (SLAM) algorithm to determine the next place for their rover to move toward, by minimizing the uncertainty, the sum of the areas of the individual error ellipses of the rover and the mapped features. The adaptive sensing technique used was a local method, only the next move of the robot is considered, unlike this dissertation where the UGV’s location is optimized for the whole flight. While this paper shows that there is a better path than a straight line, it would not help in an urban canyon, due to a lack of a cooperative vehicle. In [6] the design of a localization system for a team of robots that collaborate to map and explore unknown environments was explored. The paper’s goal was to determine the optimal action given the current knowledge of the environment, sensors, and robot dynamics in the framework to maximize the robot’s information about its location.
and all the features from the map. The approach taken in the paper, termed ‘leapfrogging’, uses three rovers as fixed beacons, while the others are free to move. The positioning of the beacon is selected by using the maximum likelihood technique. Any of the rovers can be selected as a beacon and if the information is worse at the current time–step than at the previous time–step, the rovers do not move. When the beacons are at the vertices of an equilateral triangle, the localization algorithm is the most accurate. The concept of the cooperative positioning system used in [36] is similar to what was found in [6]. The robots were divided into two separate groups, one group moved while the other group stayed stationary, then the stationary group moved while the other group stayed put. The paper looked at multiple configurations of the robots: 1) having the two child robots move next to the parent 2) have one child in front and one behind the parent and 3) a fixed triangular shape. The next step is calculated by minimizing a weighted least squares problem for the covariance of the rovers. The paper found the different configurations are best suited for different applications, 1) and 2) are suited for environments with obstacles and 3) is suited for large open areas. In [37] the group of robots perform cooperative localization using a maximum likelihood estimator in order to build an occupancy grid map. The robots use collaboration to obtain relative position and orientation of the robots with respect to each other. The position between the robots is obtained by using ultrasound pulses and then using trilateration. During the initialization stage of the robots, the algorithm may get stuck in a local minimum. The previous papers have some limitations, the papers show different methods in localizing a group of vehicles, but for only a local optimum, stationary beacons are used, vehicles have to be in close proximity to each other. In this study, using the Locally Greedy, Regionally Optimal, and nonlinear optimization approach are being studied, both cooperative vehicles are able to move, and there is no restriction to how close the vehicles must be to each other.
The differential GNSS EKF used in this study incorporates the use of double-differenced satellite observations. Several authors have used Differential GNSS (DGNSS) to increase navigation accuracy. For example, in [38], the authors used DGPS, inertial, and vision sensors for multiple UAV cooperative fault detection. Fewer have used DGPS/UWB in a cooperative navigation setting. For example, within [39] augmented carrier-phase DGPS is coupled with UWB for relative vehicle positioning. In this application, infrastructure points were used to transmit the information of the UWBs, but these points are unable to move. More recently, in [40], the combination of carrier-phase DGNSS with INS and UWB was considered between UAVs and focused on the need to enhance navigation on the small UAVs that demonstrated fast dynamics and large bank angles which caused degraded GNSS. However, again, none of these prior works focus on the potential of cooperatively moving one vehicle with respect to another in order to increase navigation observability.

2.2.1 Ultra-Wideband Radios

When a vehicle tries to obtain its position, GPS is often the first sensor that is used to due to its global coverage and ease of use. But when driving/flying in an **urban canyon** there tends to be multipath, signal interference, and reduced satellites visible due to blockage of buildings. To combat this problem, signals from other sensors are desired and one solution is the use of UWB radios. UWB devices operate on a frequency band of 3.1-10.6 Hz within the measurement systems domain [41]. An advantage to having this large frequency band to work with is the UWB is immune to multipath and has the ability to penetrate walls. With the up and coming of autonomous vehicles, the U.S. Federal Communication Commission (FCC) has allocated a Dedicated Short Range Communication (DSRC) spectrum, between 5.85 GHz and 5.925 GHz, to be used only for Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) [41].

Several authors have considered the use of peer-to-peer ranging to aide a GNSS-
degraded navigation solution, such as using stationary nodes [42] [30] or moving nodes [43] [44] [45]. Likewise, using UWB radios on a vehicle-to-vehicle platform, where the vehicles exchanged their position with each other, was considered in [43]. Similar to the previous work, UWBs are being utilized to communicate the distance between the cooperating vehicles. [43] showed that the UWBs can readily achieve better than 70-centimeter accuracy. [46] researched using multiple UWBs with GPS Real-time Kinematic (RTK) in hostile environments. RTK positioning using GPS provides centimeter-level accuracy but is limited to areas of good satellite visibility. The results show when employing an elevation mask, the GPS only solution has very poor dilution of precision (DOP) values, while the GPS+UWB solution has an improved DOP. While the study by [46] is very similar to the current study, the secondary vehicles, which are stationary, are not optimized to help in the positioning of the primary vehicle.

In [45], the GPS pseudorange, Doppler, carrier–phase, UWB range, and bearing measurements were combined to show that these additional measurements can significantly improve the across-track position. This work uses three of more vehicle to simulate a group of vehicles traveling together, and using collaborative navigation to improve performance with GPS alone. In this previous study, there was a situation where an urban canyon existed. Some satellites were dropped, and the GPS-only solution was biased. But by adding in the UWB, a large amount of the biases was removed and the errors were smaller. In [44] a similar concept from the previous papers was implemented. The paper was solving the vehicle position estimation problem, while using radio-ranging technology, road maps, and vehicle kinematics to aid in the process. But it is different in the case that the paper considers a $n$ number of vehicles. A two-step process is executed to gain information about the inter-vehicle distances and velocities of all the vehicles. First, the vehicles read their own speed and ranging measurements are gathered. Second, the information collected is shared
to all other vehicles. Further, the design and evaluation of an estimation strategy for determining the relative pose of the aircraft, between UAVs in a GPS-Challenged [40] or GPS-denied environment [47, 48] has been considered. While these are comparable to this dissertation, in terms of their exploitation of peer-to-peer ranging, on the contrary, there is not an emphasis on the control or design for the location of moving cooperative navigation nodes, which is a focus of this study.

2.3 GPS-Denied Navigation Techniques

When GPS is not available, there are other techniques and sensors that are used to help navigate the UAV through the GPS-denied situations. The key challenge for enabling GPS-denied flight is the system must be able to estimate its position and velocity by sensing unknown structures and obstacles with accuracy and low latency to navigate the vehicle. SLAM is using sensor data to build a map of the environment around the vehicle, while using SLAM to estimate the vehicle’s position [49]. In [50], a form of cooperative SLAM is used, where one vehicle is responsible for maintaining the estimates of the map and poses for each robot. In [51], the authors used a 3D laser range finder for gauging and digitization of 3D indoor environments. Visual odometry is another technique for GPS-denied areas, where a single camera is sufficient for estimating relative motion of the vehicle by using features in consecutive image frames [52]. [53] used ultrasound sensors for controlling a flying vehicle in a structured testing environment. [54] used a dead-reckoning approach which used a 6-DOF inertial measurement unit (IMU) attached to a user’s boot. The IMU provided rate-of-rotation and acceleration measurements that are used to estimate the location of the user relative to a known starting point.
2.4 Path Planning

Path planning attempts to solve the problem of finding a collision-free path and using different approaches to achieve a goal position or orientation of the vehicle [55]. Path planning has been a widely studied subject, especially in recent years with the boom of robots and UAVs, and has been used for many years in cases of vehicles trying to save fuel, vehicles having certain targets [56], and planes finding the quickest, most economical, and safest route to get from point A to point B. Cooperative control and path planning for a team of vehicles has been an interest for many fields and is researched intensely, such as [57] [58].

With a similar motivation to this dissertation, the use of radiometric signals of opportunity have be shown to improve the estimation navigation and timing information and found that adopting an information-based optimal motion planning performed better than having a pre-described path [59]. [59] studied five different trajectories, a random trajectory based on velocity, a pre-defined circular trajectory around the known anchor signal of opportunity, and optimal trajectories based on D, A, and E-optimality criteria. The D-optimality is based on the minimization of the volume of the uncertainty ellipsoid, A-optimality is the minimization of the average variance of the estimates, and E-optimality is the minimization of the length of the largest axis of the uncertainty ellipsoid. The study showed that using one of the optimal trajectories performed better than the random and pre-defined trajectories, and basing the path on the minimization of the volume of the uncertainty ellipsoid proved to be the best solution. The optimal motion planning evaluated different actions that the receiver platform could take, then would move to a location that maximizes the information about the environment. Likewise, the use of a cooperative navigation algorithm to navigate vehicles through a field with obstacles has been demonstrated in [60]. In this case, a UAV provides a low-resolution map to a UGV, so it can plan its movements based on the objects ahead. The UAV is flying a pre-determined path
and is used to map out the obstacles for the UGV to avoid. A dynamic programming approach is used, where the primary observation is the UGV sensor measurement. If no UGV measurement is available, the UAV sensor measurement is used, and if no previous measurement exists, no measurement data is stored. Due to the fact that the UAV's path does not depend on the UGV, there are scenarios where the UGV is unable to complete its task of visiting all the target points. The UGV gets stuck in a local minima along the obstacle's edge. Different from previous approaches, in this research study, signals of opportunity or cooperation through improved situational awareness are not assumed, but instead assume active ranging measurement updates support from a cooperative UGV whose location can be strategically placed. The UGV measurement will always be used to improve the positioning of the UAV.

[61] studies the cooperative vehicle problem for surveillance and reconnaissance missions with communication constraints between the vehicles. The cooperation between a ground vehicle and an aerial vehicle are used to visit a set of targets and every target is visited either by the UAV or ground vehicle. The UAV communicates with the ground vehicle and then the ground vehicle communicates with a ground station. The travel cost used between the pair of targets is the Euclidean distance between the pair of targets or the terrain. In [62], the researchers address the problem of UAV path planning for a team of cooperating UGVs. As with the previous study, the UAV needs to periodically fly over each UGV to collect the information and transmit it to the other UGVs. Proposed in [62] is a path planning approach to determine the sequence in which the UAV should visit the UGVs and to optimize the access locations of the UAV in the area of each UGV. The current study uses a different travel cost function, while also optimizing the position of the UAV. In the previous studies, their goal is to only hit a certain number of targets, while minimizing the cost function.

Another paper that studied path planning for optimal cooperative navigation is [63], where the work presented a path planning method that attempts to minimize
the navigation uncertainty of the vehicles. Each vehicle has an odometer to measure relative changes in position and heading and the ability to measure the distance between each vehicle. The vehicle kinematics are modeled using a simple unicycle model with no sideslip. The paper tried to solve the problem of constructing a set of trajectories that minimize position estimation uncertainty while satisfying vehicle motion constraints. Waypoints were used along the path to update the vehicle’s position. Using a simple stochastic search algorithm to design trajectories that minimize the objective function, the authors were able to reduce the position uncertainty by a factor of 15, when comparing the case of the vehicle driving in a straight path. The downside of using waypoints are you must know the vehicle’s path beforehand or the vehicle does not take the best and most efficient route.

2.5 Nonlinear Optimization for UAVs

Optimization is solving the problem of minimizing or maximizing a function of variables that is possibly subject to inequality and/or equality constraints on these variables. Many of these problems arise throughout engineering, mathematics, economics, and business. Nonlinear optimization deals with optimization problems where the objective function or constraint equations are nonlinear. This differs from linear optimization where the solution optimizes linear objective functions and constraint equations [64]. And from quadratic optimization which has quadratic objective functions and constraint equations, but there are quadratic techniques used in nonlinear optimization [64], Figure 2.2.

The study of optimization of cooperation of multiple vehicles is a topic that has been and is being studied for uses such as two vehicles working together to accomplish a goal. For these operations, teams of small UAVs can be used to cover a greater distance and provide a cost-effective alternative to larger, more expensive UAVs. In
these types of situations of cooperation between vehicles there are two main types centralized and decentralized to capture the complex interactions between two or more vehicles. Centralized cooperation uses the approach to solve the problem globally, but the biggest downside is the computational power used. The computational need is scaled based on the number of vehicles employed and the problem type (quadratic and mixed-integer linear programming are more difficult)[65]. All control processing occurs in a single vehicle, meaning one vehicle, robot or UAV, coordinates where the remaining ‘n’ vehicles should move. [45] used a centralized approach, with V2V relative positioning, where all the observations from all the vehicles are gathered and processed which is not as robust and requires communication to all involved vehicles.

Decentralized optimization is the idea where control processing is applied to solve problems for large scale systems, where the control is distributed among the vehicles. The vehicles have common goals, objectives, and constraints and interact with each other. The primary objective is to improve the performance of the system by solving local optimization problems for each vehicle without needing to run a global optimization for each vehicle [65]. Decentralized optimization can be different from each other

Figure 2.2: Classes and methods for numerical optimization
by the type of interactions between the different vehicles, the model of information
between the systems, and the control design technique used on the system [66]. One
issue with decentralized algorithms is the non-guarantee of having the solution be a
global solution.

Decentralized control has been successfully applied to control multiple vehicles
[67] and [68], and formation flight control of multiple UAVs [69]. Where most of the
decentralized control designs are based off the decentralized receding horizon control
(RHC) scheme. RHC is a control scheme that repeatedly solves a constrained optimi-
mization problem, using cost functions and constraints over a moving time horizon
to dictate what the vehicle should do [70]. [71] researched using a decentralized ap-
proach for cooperative navigation for vehicle-to-vehicle using GPS integrated with
UWB range. The researchers used a differential GPS filter, including UWB mea-
surements to estimate the relative positioning. The previous study shows that using
the decentralized approach improved the RMS compared to the centralized approach.
But unlike this current study, [71] does not optimize the secondary vehicles to help
the positioning of the primary vehicle.
Chapter 3

Research Objectives

The main objective of this research is to implement cooperative navigation between a UAV and a UGV to solve the problem of degraded position of a UAV in a GNSS-challenged scenario, such as an urban canyon. In this work, a novel cooperative navigation architecture is investigated between a UGV and a UAV, in which a ranging update is used to provide range measurements between the two vehicles, and the UGV is strategically moved in order to reduce the navigation estimation error of the UAV. In this work, multiple new estimation filter architectures are implemented, dynamic UAV trajectories are incorporated, and the modeling of heightened multipath errors is considered. Further, in this dissertation, three different UGV cooperative trajectory control strategies are also considered, namely, traversing to a location that is Regionally Optimal, simply basing the UGV trajectory on its most favorable nearby location at each time-step, and using a nonlinear optimization. To offer important insight, a side-by-side comparison of the different estimation architectures and UGV trajectory design approaches is conducted in order to reveal the benefits or downsides of each approach. Another objective for this dissertation is to expand the simulation, built in the first phase of this dissertation, to include more vehicles, add more cooperative UGVs, and discovering a method to incorporate multiple vehicles. The final
objective for this work is to implement a complementary experimental platform to test out the different strategies of the two-vehicle system in the real world. The novel cooperative navigation architecture developed was implemented on the experimental platforms to test the feasibility of the system. In summary, the contributions of this work are as follows:

1. Develop three cooperative strategies: Locally Greedy, Regionally Optimal, and nonlinear optimization

2. Develop two different estimation strategies: undifferenced GNSS EKF and differential GNSS EKF augmented with cooperative ranging

3. Show the feasibility of the cooperative strategy on a complementary experimental platform

The cooperative strategies that are novel strategies are the Locally Greedy and Regionally Optimal. The implementation of the nonlinear optimization was novel in this dissertation. The elements that are novel are the constraint functions and the function to be optimized. The estimation strategies were variations of the EKF. The novel aspect is including the UGV in the formulation of the equations. As stated in the literature review, there are plenty of works out there that use cooperative navigation between vehicles, use Ultra-WideBand radios as a means of communicating between the vehicles, and use DOP for placement of the secondary vehicle. But there are few studies that optimize the secondary vehicle’s position based on position dilution of precision (PDOP) to help the positioning accuracy of the primary vehicle. This dissertation implements the use of a UGV to help improve the positioning accuracy of the UAV. The UGV’s movements are optimized based on the improvement of the geometry of the UAV with respect to the satellites, the reduction of PDOP.
Chapter 4

Background

*Portions of this chapter were described in a 2016 ION GNSS+ Conference Proceedings article (Sivaneri and Gross, 2016) [1] and a 2017 AST Journal Paper (Sivaneri and Gross, 2017) [2]*

This chapter discusses the background of Global Navigation Satellite Systems, how an Extended Kalman Filter is setup, how the differential GNSS EKF is different from the EKF, and how nonlinear optimization is used. This chapter is an introduction to the topics; a more in-depth and relevant discussion will occur later in the dissertation.

4.1 Global Navigation Satellite System

The GNSS observables that are discussed are the pseudorange and carrier–phase. The pseudorange, range from satellite to user, is obtained by differencing the time-of-arrival and transmission time, Eq. 4.1. Each satellite generates its signal based on the clock on-board and each receiver generates a replica of the signal with its own on-board clock. This measurement is biased due to the satellite and receiver clocks not being synchronized.

\[ \rho_k = c(t_u - t^k) \]  

(4.1)
c is the speed of light, \( k \) is the satellite index, \( t_u \) is the time the signal was received by the user receiver, \( t^k \) is the time the signal was transmitted by the satellite and error sources have been neglected. Ideally, the true range measurement between the user and the satellite would be \( r \), but instead it is a biased and noisy measurement of \( r \). They are called pseudoranges since it does not truly represent a geometric range, due to the GNSS receiver clock bias and other error sources. The pseudorange measurement, including error sources and different delays, is shown in Eq. 4.2.

\[
\rho^k = r^k + c(\delta t_u - \delta t^k) + I^k + T^k + \epsilon_p
\] (4.2)

\( r^k \) is the range measurement between the satellite and user, \( \delta t_u \) is the receiver clock bias, \( \delta t^k \) is the satellite clock bias, \( I_p \) is the Ionospheric delays, \( T_p \) is the Tropospheric delays, and \( \epsilon_p \) is the error due to unmodeled effects, modeling errors, and measurement errors. The receiver clock bias, \( \delta t_u \), and satellite clock bias, \( \delta t^k \), are the amount the receiver and satellite clocks are advanced in relation to GPS time (GPST).

The carrier–phase measurements are more precise than the pseudorange measurements but are ambiguous and indirect measurements. The measurement is the difference between the phases of the receiver-generated signal and the carrier received from the satellite at the time the measurement was sent [11]. Carrier–phase measurements are defined by the number of cycles generated or received since the starting point, Eq. 4.3.

\[
\phi(t) = \phi(t_0) + f(t - t_0)
\] (4.3)

\( f \) is the instantaneous frequency, \( \phi(t_0) \) is the initial phase offset, \( t - t_0 \) is the time at each epoch. In this model, the carrier–phase measurements would be a fraction of a cycle, but there is no information regarding the total number of phase cycles between the user and satellite. This unknown number of phase cycles is known as the
integer ambiguity. Before the carrier–phase measurements can be incorporated, the integer ambiguity must be calculated, which is addressed in Eq. 4.4.

\[ \phi(t) = \phi_u(t_0) + \phi^k(t - \tau) + N \] (4.4)

\( \phi_u(t) \) is the phase of the receiver-generated signal, \( \phi^k(t - \tau) \) is the phase of the signal received from the satellite at time \( t \), \( \tau \) is the transit time of the signal, and \( N \) is the integer ambiguity. The estimation of \( N \) is the integer ambiguity resolution. The carrier–phase measurement, accounting for clock bias, initial phase offset, ionosphere and troposphere delays, and measurement errors, is shown in Eq. 4.5.

\[ \phi^k = \frac{1}{\lambda} [r^k + I^k + T^k] + \frac{c}{\lambda} (\delta t_u - \delta t^k) + N + \epsilon \phi \] (4.5)

c is the speed of light, \( \lambda \) is carrier wavelength, \( I^k \) is the ionosphere delay, and \( T^k \) is the troposphere delay.

This dissertation employs the use of dual-frequency undifferenced GNSS observables. Due to the fact that it is undifferenced data, the error sources include the ionospheric delay, tropospheric delay, and receiver clock bias and must be included in the measurement model. To eliminate the ionospheric delay, the ionospheric-free (IF) pseudorange and carrier–phase combinations are employed, which exploits the dispersive nature of the ionosphere to eliminate its impact to first order. The GPS IF combination for pseudorange and carrier–phase are shown in Eq. 4.6 and Eq. 4.7, respectively.

\[ \rho_{IF}^k = \rho_{L1}^k \left[ \frac{f_1^2}{f_1^2 - f_2^2} \right] - \rho_{L2}^k \left[ \frac{f_2^2}{f_1^2 - f_2^2} \right] = 2.546 \rho_{L1}^k - 1.546 \rho_{L2}^k \] (4.6)

\[ \phi_{IF}^k = \phi_{L1}^k \left[ \frac{f_1^2}{f_1^2 - f_2^2} \right] - \phi_{L2}^k \left[ \frac{f_2^2}{f_1^2 - f_2^2} \right] = 2.546 \phi_{L1}^k - 1.546 \phi_{L2}^k \] (4.7)
In Eq. 4.6 and Eq. 4.7, the $f_1$ and $f_2$ are the $L_1$ and $L_2$ frequencies, $\rho_{L1}$ and $\rho_{L2}$ are the pseudorange measurements on the $L_1$ and $L_2$ frequencies, $\phi_{L1}$ and $\phi_{L2}$ are the carrier–phase measurements on the $L_1$ and $L_2$ frequencies. The superscript $k$ in Eq. 4.6 and Eq. 4.7 is used to denote the measurement between satellite $k$ and the user. The remaining error sources must be modeled and estimated for each pseudorange and carrier–phase measurement as shown in Eq. 4.8 and 4.9, respectively.

$$\rho_{IF}^k = r^k + c\delta t_u + T_z m(\epsilon^k) + \epsilon_{\rho}$$ (4.8)

$$\phi_{IF}^k = r^k + c\delta t_u + T_z m(\epsilon^k) + \lambda_{IF} N_{IF}^k + \epsilon_{\phi}$$ (4.9)

In Eqs. 4.8 and 4.9 $c$ is the speed of light, $\delta t_u$ is the receiver clock bias, $T_z$ is the tropospheric delay in the zenith direction, $m(\epsilon^k)$ is a mapping function dependent on elevation angle, $\lambda_{IF}$ is the wavelength corresponding to the IF combination, $N_{IF}$ is phase ambiguity (for the carrier–phase measurement model), $r^k$ is geometric range between the user and the satellite, as shown in Eq. 4.10, and $\epsilon$ are the remaining un-modeled error sources for pseudorange ($\rho$) and carrier–phase ($\phi$), respectively.

$$r^k = \sqrt{(x^k_x - x_u)^2 + (y^k_y - y_u)^2 + (z^k - z_u)^2}$$ (4.10)

In the equation, $k$ denotes the satellite and $u$ denotes the user. The troposphere mapping function allows for the reduction of the troposphere delay model to a single unknown delay in the zenith direction with respect to the user and is scaled according to the user to satellite elevation angles. The troposphere zenith delay, $T_z$, as shown in Eqs. 4.8 and 4.9 is composed of both a dry and wet component. The wet delay (i.e., approximately 10% of the total delay [72]) is typically estimated as it is more difficult to model whereas the dry delay is modeled. In this simulation, the Hopfield model [72] is used to model the wet and dry delay. The mapping function implemented was
taken from [11], and is shown in Eq. 4.11.

\[
m(e^k) = \frac{1.001}{\sqrt{0.00201 + \sin(e^k)^2}}
\]  

(4.11)

4.1.1 Differential Global Navigation Satellite System

Double–differenced measurements are constructed by first forming single–differenced measurements of the pseudorange and carrier–phase for each satellite in view, as in Eq. 4.12, followed by calculating the difference between each single–differenced and a single–differenced reference satellite Eq. 4.13. This process is repeated for pseudorange and carrier–phase data for all frequencies (i.e., L1 and L2) [11].

\[
\Delta \rho^f_k = \rho^A^k_f - \rho^B^k_f
\]  

(4.12)

In Eq. 4.12 \( \rho^A^k_f \) is the pseudorange for receiver A, \( f \) denotes L1 or L2, of the UAV of each satellite in view, \( \rho^B^k_f \) is the pseudorange for receiver B, \( f \) denotes L1 or L2, of the UGV of each satellite in view.

\[
\nabla \Delta \rho_f = \Delta \rho^k_f - \Delta \rho^\text{refSat}_f
\]  

(4.13)

The single–differenced pseudorange, L1 or L2, is denoted as \( \Delta \rho^\text{refSat,k}_f \) of the reference satellite.

4.2 Dilution of Precision

The dilution of precision provides a simple characterization of the user-satellite geometry. In general, the more favorable the geometry, the lower the DOP. The lower the DOP, the better the quality of position estimate. During the early years, GPS receivers were limited to tracking only four satellites at a time, thus PDOP was a
metric used to select the ‘best’ four satellites in view. Nowadays, receivers do not have that limitation and all satellites in view can be tracked. But PDOP is still a metric used to indicate the quality of the position estimate. With four satellites, precision dilution of precision is the lowest if three satellites are evenly distributed near the horizon and the fourth is directly overhead. If the satellites are grouped close together in the sky, the PDOP value is high, resulting in poor positioning performance, this is shown in Figure 4.1. In this study, PDOP is used as the main metric in the cooperative algorithms.

In [73], the authors examined whether the use of DOP was an adequate metric for collaborative positioning. A simulation model and an experimental demonstration was performed to show that the DOP model can be used as a reference to analyze GNSS collaborative positioning accuracy. The results provided show that if the additional vehicle has poor DOP, the overall DOP is still reduced and provides navigational help. There are papers that use DOP as a metric for placement of the secondary vehicle(s) but differ in some aspects from this study. For example, in [42] the best position for placing base stations, such that an area is covered by four base stations at all times, was considered. The previous study is similar to this current research work, in that the placement of the secondary vehicle is based on the DOP. Specifically, the previous work was using the Geometric Dilution of Preci-
sion (GDOP). The drawback to the previous work is the limitation of being inside a square area due to the fact that these base-stations are unable to move. Due to the restriction for needing four base-stations in order to perform localization, the user is severely restricted to the interior area of the four base-stations.

The Dilution of Precision metric provides a simple characterization of how user-satellite geometry impacts the positioning errors. In short, it represents how much the random errors of ranging sources scale into the position domain when using trilateration. The more favorable the user-satellite geometry, the lower the DOP metric, and the better the position estimate will be (i.e., assuming that all other error sources are constant). Starting from the Linear Least Squares (LLS) GNSS-solution,

\[
\Delta x = (G^T G)^{-1} G^T \Delta \rho
\]  
(4.14)

\(G\) is the \textit{Geometry matrix}, in Eq. 4.15 that is constructed by creating a set of unit-vectors that point along the direction cosines from the user to the satellite locations. The pseudorange, \(\rho\), measurement model is given by the equation, Eq. f4.2.

\[
G = \begin{bmatrix}
-1^1 & 1 \\
\vdots & \vdots \\
-1^n & 1
\end{bmatrix}
\]  
(4.15)

In Eq. 4.15 the \(\mathbf{1}\) is the unit vector constructed by direction cosines from the user to the satellite locations.

In this LLS solution, it is assumed that the measurement residuals are zero-mean \(E[\tilde{\epsilon}_\rho]\) and the variance of the error is given by

\[
E[\tilde{\epsilon}_\rho \tilde{\epsilon}_\rho^T] = P_\epsilon = \sigma_{URE}^2 \mathbf{I}
\]  
(4.16)

\(\sigma_{URE}\) is the standard deviation of the \textit{User Range Error} and is provided by the GNSS
control segments [74] and $I$ is the identity. Using the LLS solution for position and clock bias estimation and the zero-mean assumption, the estimation covariance matrix can be formed

$$\text{cov}[\Delta x] = \sigma_{URE}^2 (G^TG)^{-1} = \sigma_{URE}^2 H$$

(4.17)

Using the previous equation, the Horizontal Dilution of Precision, $H^{DOP}$, is formed and shown in Eq. 4.18.

$$H^{DOP} = (G^TG)^{-1} = \text{diag} \begin{bmatrix} H_{11}, & H_{22}, & H_{33}, & H_{44} \end{bmatrix}$$

(4.18)

The elements of the $H^{DOP}$ are known as the Position Dilution of Precision (PDOP),

$$PDOP = \sqrt{H_{11}^{DOP} + H_{22}^{DOP} + H_{33}^{DOP}}$$

(4.19)

Within this study, in addition to GNSS ranging signals, the impact on the Geometry matrix by augmenting the measurement set with a ranging measurement from a cooperative UGV is considered. In this sense, the UGV effectively acts as another satellite observation in which there is an ability to control its location to support favorable geometry.

4.3 Extended Kalman Filter

4.4 Kalman Filter

The Kalman Filter was developed in 1960 as an optimal solution to the discrete-data linear filtering problem [75]. Since then, the Kalman Filter has been the subject of extensive research and application, such as navigating the Apollo spacecraft, predicting stock market fluctuations, and particularly in the field of autonomous and coopera-
The Kalman Filter is a well-known data processing technique that attempts to minimize the mean of the squared error between the observation and the estimations [76]. It is recursive, so that new measurements can be processed as they arrive. The Kalman Filter consists of two series of steps, the prediction step and the measurement-update step. These equations are found in numerous papers and textbooks such as [76] [77] [78] [79] [80] [81]. The Kalman Filter combines the information from the two steps, predicted and measured values, and improves both by suppressing the noise in the measurement and improves the accuracy of the state and parameters [80]. The drawbacks to the Kalman Filter are it is assumed that the system is linear, and the noise is Gaussian [82]. There are many scenarios when the system is not linear, thus filters that process nonlinear systems are discussed, namely the Extended Kalman Filter.

The Extended Kalman Filter (EKF) is the approach most utilized for dealing with nonlinear systems by using a linearized version of the nonlinear system modeled over the last estimate. Using an EKF for the application of GPS is not novel, as it has been used in many studies. Specifically for cooperative navigation, it has been used by many researchers [9] [10] [44] [45] [63] [5], to incorporate a measurement from the cooperative vehicle(s). The EKF has shown to be a useful method of obtaining good estimates of the system state. In this study, the EKF is the algorithm that is used to evaluate the UAV’s position.

A Kalman Filter is different from batch processing where all the data must be present before running through the filter. Undifferenced GNSS EKF is a method that has been used for UAV navigation [83]. The advantages of undifferenced GNSS EKF is it does not need additional equipment. The use of undifferenced GNSS EKF is investigated as a part of this dissertation as it would reduce the communication required between the UAV and UGV in comparison to differential approaches.
4.4.1 Prediction

The prediction step is used to propagate the state estimate and an estimate of the error covariance from the previous time to the current time. The propagation of the state estimate is displayed below Eq. 4.20.

\[
\hat{x}_k^- = \Phi(t_k, t_{k-1}) \hat{x}_{k-1}^+ + w_k
\]  

(4.20)

Where \( \Phi \) is the state transition matrix (STM), the \( \hat{\cdot} \) is to designate an estimated parameter, the subscript \( t_k \) and \( t_{k-1} \) represents the time–step, the \( - \) denotes that the estimate is before the measurement update, \( \hat{x}_{k-1}^+ \) is the state estimate from the previous time–step, and \( w_k \) is the process noise [82].

The Kalman filter uses three different matrices that are specific to the application because the Kalman Filter is known to be sensitive to these tuning parameters these matrices must be tuned correctly [12]. These matrices are as follows, state covariance matrix, \( P \), the process noise covariance matrix \( Q \), and the measurement covariance matrix, \( R \). If \( P \) is set to zero, the filter ignores and learns nothing from the measurements and if \( P \) is very large, the filter trusts the measurements more than the model. \( Q \) should be small enough to use the measurement, but large enough so the filter estimate is used [84]. The error covariance matrix is formed by Eq. 4.21.

\[
P_k^- = \Phi(t_k, t_{k-1}) P_{k-1}^+ \Phi(t_k, t_{k-1})^T + Q_k
\]  

(4.21)

The process noise covariance is denoted as \( Q_k \) [82].

4.4.2 Measurement Update

In this step, the state estimate and the covariance matrix is updated with the new measurements [82]. The equations used to propagate the state estimate and the
covariance matrix are displayed in Eqs 4.22 and 4.23.

\[
\hat{x}_k^+ = \hat{x}_k^- + K_k(z_k - H_kx_k^-)
\]  \hspace{1cm} (4.22)

\[
P_k^+ = (I - K_kH_k)P_k^- (I - K_kH_k)^T + K_k R_k K_k^T
\]  \hspace{1cm} (4.23)

The observation matrix is \(H_k\) and \(z\) the measurement vector. These two parameters’ derivation will be discussed later on. The Kalman Gain is denoted as \(K_k\) and is used to decide the weighting value for the state estimate and the measurements is shown in Eq. 4.24 [82].

\[
K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R)^{-1}
\]  \hspace{1cm} (4.24)

### 4.4.3 Process and Measurement Noise

The process noise covariance matrix has the relation shown below.

\[
w_x \sim N(0, Q)
\]  \hspace{1cm} (4.25)

Where \(w_x\) is the process noise [82].

The measurement covariance matrix, \(R\), has the relation shown below.

\[
v_x \sim N(0, R)
\]  \hspace{1cm} (4.26)

Where \(v_x\) is the measurement noise [82].
4.5 Nonlinear Optimization

For nonlinear optimization, there are many methods for solving nonlinear problems. Some of these methods include Gauss-Newton, Linear-Markov, Newton-Raphson, sequential quadratic programming, interior-point method, genetic algorithm, and convex programming to name a few [64]. These different methods are used based on whether the problem is an unconstrained or constrained optimization. If it is constrained, whether it is a single or multi-objective problem. This section will discuss a few of the optimization methods and the method chosen for this study.

4.5.1 Unconstrained Optimization

Unconstrained optimization considers the problem of minimizing or maximizing an objective function that depends on real variable with no restrictions on their values. When solving maximization problems, the problems are turned to minimization problems by multiplying the function by $-1$. The general mathematical function for this problem is shown in Eq.4.27.

$$\min_{x} f(x) \quad (4.27)$$

Where $x \in \mathbb{R}^n$ is a real vector with $n \geq 1$ and $f : \mathbb{R}^n \to \mathbb{R}$ is a smooth function. Finding a global minimum can be difficult because the knowledge of $f$ is usually only local [85]. Most algorithms are able to only find a local minimum, which is a point that achieves the smallest value of $f$ in its neighborhood. To know whether a point is a local minimum or global minimum, there are tests for optimality which serve as the basis for these algorithms. The optimality conditions in unconstrained optimization are necessary, these optimality conditions are known as necessary and sufficient optimal conditions [86]. Every local minimum of Eq. 4.27 satisfies the first-order necessary condition Eq. 4.28 which states that the gradient must be zero at the
local minimum,

$$\nabla f(y^*) = 0$$  \hspace{1cm} (4.28)

where \(y^*\) is the local minimum. And satisfies the second-order necessary condition Eq. 4.29,

$$\nabla^2 f(y^*) \geq 0$$  \hspace{1cm} (4.29)

which means that the Hessian, \(\nabla^2 f(y^*)\), must be positive semi-definite. The second-order sufficient condition for the local minimum states that the Hessian is positive definite, Eq. 4.30.

$$\nabla^2 f(y^*) > 0$$  \hspace{1cm} (4.30)

Every local minimum satisfies the two necessary conditions but not necessarily satisfies the sufficient condition. And every point that satisfies the necessary conditions and the sufficient condition is guaranteed to be a local minimum [86].

There are two main types of algorithms used in unconstrained nonlinear optimization: line search methods and trust region methods. In line search methods, one first chooses the search direction from the current point using information about the function, then chooses a step length. The steepest-descent direction is the choice for search direction for a line search method [85]. For the trust region methods, one chooses an approximation to the function around the current point. A commonly used model of the trust region method is the Newton-Raphson Method or the Newton’s method [87]. The idea is to construct an iterative procedure by using a linear Taylor approximation around the current point.

For nonlinear unconstrained optimization, such as Newton’s method, the calculation of the Hessian is needed for every iteration. While exact, this calculation is
numerically expensive and in general not beneficial except in convergence. Taking this into account, the Quasi-Newton method uses the approximation of the inverse of the Hessian and is beneficial because the linear system of equations does not need to be solved to calculate the search direction. Instead, a simple matrix-vector multiplication is used [88].

The nonlinear least-squares problem is a special case of unconstrained nonlinear optimization. This type of problem arises in many practical problems, especially in data-fitting applications, from chemistry, financial, to engineering. In the least-squares problem, the objective function $f$, Eq. 4.31 has the following special form:

$$f(x) = \frac{1}{2} \sum_{j=1}^{m} r_j^2(x) \quad (4.31)$$

where $r$ is the residual. Any application uses this form to measure the difference between the model and the output of the system at various observation points. Values are selected for the parameters that best match the model to the data. The biggest advantage of the least-squares problem is that by knowing the Jacobian matrix, the first part of the Hessian can be computed ‘for free’. The Gauss-Newton Method, simplest method, is one of the methods that minimizes nonlinear objective functions and exploits the structure in the gradient and the Hessian. It is a modified Newton’s Method with line search. Instead of generating the search direction by solving the standard Newton equations, the calculation of the Hessian is excluded from obtaining the search direction, and only the approximation of Hessian is calculated [85].

Another method used is the Levenberg-Marquardt Method, damped least-squares method, which uses the trust-region strategy rather than the line search strategy. It can be considered as an interpolation between the Gauss-Newton method and the gradient descent method. The Gauss-Newton method is vulnerable when the Jacobian is rank-deficient, but the Levenberg-Marquardt Method avoids this weakness. Both methods do not use the second-order Hessian component, so the local convergence for
both methods is similar. Even if the initial guess is far from the solution corresponding to the minimum, the iteration can still converge on the solution[89].

4.5.2 Constrained Optimization

Constrained optimization, similar to unconstrained optimization, is minimizing the objective function but subject to constraints. The constrained nonlinear problem with equality and inequality constraints is Eq. 4.32

$$\min_{y \in R^n} f(y)$$

subject to \( g(y) \leq 0 \)

$$h(y) = 0$$

(4.32)

where \( f \) and \( g(y) \) and \( h(y) \) are all smooth, real-valued functions on a subset of \( R^n \).

The main techniques that are used to solve constrained optimization problems are reduced-gradient methods, sequential linear methods, quadratic programming methods, and methods using augmented Lagrangian and exact penalty functions. As with unconstrained optimization, constrained has necessary and sufficient conditions. But there are different conditions depending on the type of constrained problem [90]. For the equality constrained problem there are necessary and sufficient conditions and for inequality constrained problems there are necessary and sufficient conditions that vary slightly from the equality constraints. And finally, there are conditions for an equality and inequality constrained problem. For this last combination, there are first-order and second-order necessary conditions and a second-order sufficient condition to see if a point is a strict local minimizer [91]. The Lagrangian is used to express the first and second-order conditions for a local minimizer. The first-order necessary conditions for the existence of a local minimizer requires the existence of
Lagrange multipliers. Lagrange multipliers only allows equality constraints, so for inequality constraints, the Karush-Kuhn-Tucker (KKT) conditions, Eq. 4.33 generalizes the method of Lagrange multipliers. Meaning the KKT conditions are analogous to the condition that the gradient must be zero at a minimum but modified to take constraints into account.

\[
\Delta_y L(y, \Lambda) = 0
\]
\[
\Lambda_{g, i} g_i(y) = 0
\]  (4.33)

Where \(\Lambda\) is the concatenation of \(\Lambda_g\) and \(\Lambda_h\) the Lagrange multiplier vector. The length is the total number of constraints. Necessary optimal conditions are used for finding a candidate solution for being optimal. In order to verify optimality, sufficient optimality conditions are needed. A local minimum that meets the KKT conditions is a local minimum, and if the second-order sufficient condition is met, the point is a strict local minimizer [92].

One algorithm for the inequality constrained nonlinear optimization problems, and used in this study, is the interior-point method, more accurately the interior-point barrier method. This is an iterative method which produces a sequence of points lying in the relative interior of the feasible set [64] or an approach to solve a sequence of approximate minimization problems. Eq. 4.32 is the original problem, where the approximate problem is shown in Eq. 4.34 for each \(\mu > 0\) [93].

\[
\min_{y, s} f_\mu(y, s) = \min_{y, s} f(y) - \mu \sum_i \ln(s_i)
\]
\[
\text{subject to } h(y) = 0 \text{ and } g(y) + s = 0
\]  (4.34)

The slack variables, \(s_i\), are restricted to be positive to keep the \(\ln(s_i)\) bounded.
This added logarithmic term is called a barrier function [92]. The approximate problem is a simpler one in which constraints are replaced by a penalty term. The purpose of this penalty term is to give large objective functions values to points near the relative boundary of the feasible set, which effectively becomes a barrier against leaving the feasible set [94]. For this dissertation, the MATLAB ‘fmincon’ function, nonlinear optimization, was used. An in-depth analysis of this algorithm is discussed in Chapter 7.
Chapter 5

Estimation Filter Design

Portions of this chapter were described in a 2016 ION GNSS+ Conference Proceedings article (Sivaneri and Gross, 2016) [1] and a 2017 AST Journal Paper (Sivaneri and Gross, 2017) [2]

This chapter discusses the details of how the Position Dilution of Precision is reduced when adding in a ranging source, the equations for the Undifferenced GNSS Extended Kalman Filter with and without the ranging source, and the equations for the Differential GNSS Extended Kalman Filter with and without the ranging source.

5.1 Reducing PDOP with a Single Ranging Source

To motivate the potential of the cooperative approaches taken in this dissertation, it is important to consider the PDOP reduction that a single additional ranging source could potentially realize. To show this, a simple Monte-Carlo simulation was conducted to determine the maximum amount of PDOP reduction realizable by the addition of a single ranging source across different GNSS satellite geometries. That is, to vary the GNSS satellite constellation geometry, the location of the simulated user’s location and the GNSS time of week were randomized many times for a given scenario. Next, to simulate a GNSS-challenged condition, a high elevation mask was applied
to simulate a user’s GNSS visibility being severely impacted. As an example, Figure 5.1 shows the percentage of PDOP reduction that could occur with the inclusion of an optimally placed ranging source located that is in surrounding proximity of the user’s location relative to their GNSS-only PDOP.

Figure 5.1: Monte Carlo simulation result with a 55-degree elevation mask that illustrates the potential improvement of including a single additional ranging source that is optimally placed.

In Figure 5.1 [1], a 55-degree elevation mask was used to simulate a UAV in the center of a city block. From this analysis, it is apparent that the poorer the satellite geometry, the more benefit the single ranging source can potentially offer. For example, the PDOP can be reduced by up to 75% when the GNSS-only PDOP is 6. However, to motivate the need for a cooperative strategy in lieu of simply relying on a UGV placed anywhere, the minimum potential PDOP reduction is also shown, which suggests that a poorly placed UGV could offer no geometric benefit.
5.2 Undifferenced GNSS Extended Kalman Filter

This section gives an overview of the estimation filter designs developed for this study. First, an overview of the undifferenced GNSS EKF design is discussed. Then, the inclusion of a cooperative ranging signal from a UGV is discussed. Next, the differential GNSS EKF design as well as the inclusion of the ranging radio measurement within its measurement update are detailed. Finally, three cooperative strategies for the UGV motion with respect to the UAV are discussed, which are dubbed the Locally Greedy strategy, the Regionally Optimal strategy, and the Nonlinear Optimization.

5.2.1 Undifferenced GNSS EKF Design

The filter parameters that have to be initialized in the EKF are the three matrices, $P$ the state covariance matrix, $Q$ the process noise covariance matrix, and $R$ the measurement noise covariance matrix. The initial state covariance matrix, $P$ is used to define the amount of uncertainty that is assumed in the initial states. The amount of uncertainty in the GPS states (the carrier–phase bias for each satellite in view), is a diagonal matrix with 25 meters in every diagonal term. The $P$ is shown in Eq. 5.1.

$$
P = \begin{bmatrix}
P_{pos}(3\times3) & 0_{(3\times1)} & 0_{(3\times1)} & 0_{(3\times n)} \\
0_{(1\times3)} & P_{clk}(1\times1) & 0 & 0_{(1\times n)} \\
0_{(1\times3)} & 0 & P_{trop}(1\times1) & 0_{(1\times n)} \\
0_{(n\times3)} & 0_{(n\times1)} & 0_{(n\times1)} & P_{\phi(n\times n)}
\end{bmatrix} \tag{5.1}
$$

The $P_{pos}$ is the uncertainty in position, $P_{clk}$ is the uncertainty in clock bias, $P_{trop}$ is the uncertainty in the troposphere, and $P_{\phi}$ is the uncertainty in the carrier–phase measurements of GPS.

The process noise covariance matrix is implemented to describe how much the state of the system changes over time because of the process noise. Similar to the state covariance matrix, the process noise covariance matrix is made up of the uncertainty
in the GPS. The $Q$ matrix is shown in Eq. 5.2.

$$Q = \begin{bmatrix} Q_{pos}(3\times3) & 0_{(3\times1)} & 0_{(3\times1)} & 0_{(3\times1)} \\ 0_{(1\times3)} & Q_{clk}(1\times1) & 0 & 0_{(1\times1)} \\ 0_{(1\times3)} & 0 & Q_{trop}(1\times1) & 0_{(1\times1)} \\ 0_{(n\times3)} & 0_{(n\times1)} & 0_{(n\times1)} & Q_{\phi}(n\times n) \end{bmatrix} \tag{5.2}$$

The $Q_{pos}$ is the uncertainty in position, $Q_{clk}$ is the uncertainty in clock bias, $Q_{trop}$ is the uncertainty in the troposphere, and $Q_{\phi}$ is the uncertainty in the carrier–phase measurements of GPS.

The measurement noise covariance matrix, $R$, is used to account for errors in the GPS that are not taken into account in the simulation, shown in Eq. 5.3.

$$R = \begin{bmatrix} R_{Phase(n\times n)} & 0_{(n\times n)} \\ 0_{(n\times n)} & R_{Range(n\times n)} \end{bmatrix} \tag{5.3}$$

The $R_{Phase}$ represents the carrier–phase measurements and the $R_{Range}$ is the pseudorange measurements.

As mentioned, the pseudorange is much less precise than the carrier–phase. This is taken into account in this matrix, with the values corresponding to the pseudorange being 100 times worse than the carrier–phase data [11].

The STM used in this simulation is shown in Eq. 5.4 is the identity matrix. The length of the matrix is dependent on the number of satellites, $n$, in view at that specific time–step.

$$\Phi = \begin{bmatrix} I_{6\times6} & 0 \\ 0 & I_{n\times n} \end{bmatrix} \tag{5.4}$$
The estimated state vector is shown in Eq. 5.5.

\[ \mathbf{x} = \begin{bmatrix} \delta r \\ \delta t_u \\ T_w \\ N_1 \\ \vdots \\ N_n \end{bmatrix} \] (5.5)

The position-error with respect to a nominal guess is denoted as \( \delta r \), \( \delta t_u \) is the estimated receiver clock bias with respect to a nominal guess, \( T_w \) is the estimated troposphere delay, and \( N_{1..n} \) are the estimated carrier–phase biases of the ionospheric-free carrier–phase data for each satellite in view.

### 5.2.2 Undifferenced GNSS EKF Augmented with Cooperative Ranging

In this study, it is assumed that a cooperative ranging measurement such as a pair of UWB radios, is used as a Time of Arrival measurement (ToA) [95] between a UGV and a UAV. An advantage of using UWB signals is they have been shown to work in non-LOS application, can penetrate walls, and are not significantly impacted by multipath [43]. Another benefit of using the UWB radios are their weight, relatively low-power consumption, and low-cost [96].

The GNSS observation matrix, \( \mathbf{H}_{UD}^{obs} \), represents the sensitivity of the measurement models to the state being estimated. The first 3 columns of \( \mathbf{H}_{UD}^{obs} \) are the partial derivatives of the measurement models with respect to the user’s Earth Centered Earth Fixed (ECEF) position. Column four is the partial derivative with respect to the GNSS receiver clock bias. The troposphere’s zenith delay partials are comprised of the elevation dependent mapping function and appear in column 6 of \( \mathbf{H}_{UD}^{obs} \). The
rest of the columns are populated with an identity matrix over the block of rows that correspond to the carrier–phase observations. This identity matrix represents the partial derivative of the carrier–phase observational model with respect to the carrier–phase biases. This is the $H_{UD}^{obs}$ matrix when the UGV is not being used.

The UGV ranging source can be considered similar to an additional GNSS satellite measurement, with the exception that its measurement model only has partial derivatives that are sensitive to the UAV’s position with no sensitivity to the GNSS receiver clock bias, troposphere delay, or GNSS carrier–phase ambiguity. Throughout the paper, the cooperative ranging measurement will be defined as, $\upsilon$.

$$
H_{UD}^{obs} = 
\begin{bmatrix}
  u_1^X & u_1^Y & u_1^Z & 1 & M_{el}^1 & : \\
  : & : & : & : & 0_{n \times n} & : \\
  u_n^X & u_n^Y & u_n^Z & 1 & M_{el}^n & : \\
  u_1^X & u_1^Y & u_1^Z & 1 & M_{el}^1 & : \\
  : & : & : & : & I_{n \times n} & : \\
  u_n^X & u_n^Y & u_n^Z & 1 & M_{el}^n & : \\
  \gamma_X & \gamma_Y & \gamma_Z & 0 & 0 & 0_{1 \times n}
\end{bmatrix}
$$

(5.6)

The symbols in the previous equation are denoted as: $n$ is the number of satellites in view, $u_n^X$, $u_n^Y$, and $u_n^Z$ are the respective satellite positions in X, Y, and Z, $\gamma_X$, $\gamma_Y$, and $\gamma_Z$ is the X, Y, and Z location of the UGV. To include the cooperative ranging measurement, it must be predicted for inclusion in the filter in order to form an innovation residual. The predicted UAV to UGV cooperative range is formed by calculating the distance between the estimated UAV and UGV positions, Eq. 5.7 and Eq. 5.8.

$$
\hat{\upsilon} = \| \hat{r}_{UAV} - r_{UGV} \|_2
$$

(5.7)
\[ z_{\text{uwb}} = \bar{\upsilon} - \hat{\upsilon} \]  

(5.8)

In Eq. 5.7, \( r_{UAV} \) is the position of the UAV, \( r_{UGV} \) is the position of the UGV, \( \hat{\upsilon} \) is the UWB measured range, and in Eq. 5.8 \( \bar{\upsilon} \) is the UWB predicted range.

The overall measurement vector, \( Z_{UD} \), of the undifferenced EKF is given as shown in Eq. 5.9.

\[
Z_{UD} = \begin{bmatrix}
\Delta \Phi_{L1} \\
\Delta \Phi_{L2} \\
\Delta \rho_{L1} \\
\Delta \rho_{L2} \\
\Delta \upsilon
\end{bmatrix}
\]  

(5.9)

where the \( \Delta \)s denotes the fact the the GNSS measurements are the observed-minus-computed innovation residuals that are formed by differencing the receiver’s observables with their modeled counterparts from Eqs. 4.6 and 4.7, which are evaluated about an assumed nominal user position.

### 5.3 Differential GNSS Extended Kalman Filter

The advantages of using DGNSS over undifferenced GNSS EKF are: 1) Single-differenced measurement eliminates satellite clock errors, orbit errors, and localized atmospheric errors 2) Double-differenced measurement eliminates the receiver clock errors. The disadvantage is the need for more equipment, namely a second receiver, to provide the double-differenced measurement and a modem to transmit this data. Further, the double-differenced carrier-phase leads to double-differenced carrier-phase ambiguities that must be estimated.

Double-differenced measurements are constructed by first forming single-differenced
measurements of the pseudorange and carrier-phase for each satellite in view, as in Eq. 5.10, and then differencing each single-differenced measurement with a single-differenced measurement of a chosen reference satellite, as shown in Eq. 5.11. This process is repeated for both pseudorange and carrier-phase data for both carrier frequencies (i.e., L1 and L2).

\[
\Delta \rho_f = \rho_{f, UAV}^k - \rho_{f, UGV}^k
\]  

(5.10)

In Eq. 5.10 \( \rho_{f, UAV}^k \) is the pseudorange of the UAV of each satellite in view, \( \rho_{f, UGV}^k \) is the pseudorange of the UGV of each satellite in view, and in both terms \( f \) is the frequency, L1 or L2.

\[
\nabla \Delta \rho_f = \Delta \rho_f^k - \Delta \rho_f^{refSat}
\]  

(5.11)

The single-differenced pseudorange is denoted as \( \Delta \rho_f^{refSat, k} \) of the reference satellite, where \( f \) is denoted as the frequency L1 or L2. In the differential GNSS EKF, the \( P, Q, \) and \( R \) matrices are similar to the undifferenced GNSS EKF, but with the DGNSS there is no clock bias, clock drift, or troposphere to include in the matrices. The state covariance matrix, \( P \), is shown in 5.12.

\[
P = \begin{bmatrix}
P_{pos(3x3)} & 0_{(3xl)} \\
0_{(l\times3)} & P_{DD(l\times l)}
\end{bmatrix}
\]  

(5.12)

The \( P_{pos} \) is the uncertainty in position and \( P_{DD} \) is the uncertainty in the double-differenced measurements of GPS, where the length of \( P_{DD} \) is twice the number of double-differenced measurements, one for each frequency, L1 and L2, and \( l \) is twice the number of satellites measurements used at the time step, \( l = 2(n - 1) \). For DD GNSS EKF, the number of satellites are \( n - 1 \) due to the reference satellite being used to make the double-differenced measurements.
The process noise covariance matrix is implemented to describe how much the state of the system changes over time because of the process noise. Similar to the state covariance matrix, the process noise covariance matrix is made up of the uncertainty in the GPS. The \( Q \) matrix is shown in Eq. 5.2.

\[
Q = \begin{bmatrix}
Q_{\text{pos}}_{(3 \times 3)} & 0_{(3 \times l)} \\
0_{(l \times 3)} & Q_{\text{\Phi}_D D(l \times l)}
\end{bmatrix}
\tag{5.13}
\]

The \( Q_{\text{pos}} \) is the uncertainty in position and \( Q_{DD} \) is the uncertainty in the double-differenced measurements of GPS, where the length of \( Q_{DD} \) is twice the number of double-differenced measurements, one less than the number of satellites in view. The measurement noise covariance matrix, \( R \), is used to account for errors in the GPS that are not taken into account in the simulation, shown in Eq. 5.3.

\[
R = \begin{bmatrix}
R_{\text{Phase}}_{(l \times l)} & 0_{(l \times l)} \\
0_{(l \times l)} & R_{\text{Range}}_{(l \times l)}
\end{bmatrix}
\tag{5.14}
\]

The \( R_{\text{Phase}} \) represents the carrier-phase measurements and the \( R_{\text{Range}} \) is for the pseudorange measurements. The STM used in this simulation is shown in Eq. 5.16 is the identity matrix. The length of the matrix is dependent on the number of satellites, \( n \), in view at that specific time-step.

\[
\Phi = \begin{bmatrix}
I_{3 \times 3} & 0 \\
0 & I_{l \times l}
\end{bmatrix}
\tag{5.15}
\]

In Eq. 5.15, the \( n \) is based on the number of double-differenced measurements at that time-step. The estimated state vector for doubled-difference filter is given as
shown in Eq. 5.16.

\[
\mathbf{x}_{DD} = \begin{bmatrix}
\Delta \rho_{UAV-UGV} \\
\nabla \Delta N_{L1}^{1,\text{refSat}} \\
\vdots \\
\nabla \Delta N_{L1}^{n-1,\text{refSat}} \\
\nabla \Delta N_{L2}^{1,\text{refSat}} \\
\vdots \\
\nabla \Delta N_{L2}^{n-1,\text{refSat}}
\end{bmatrix}
\]  
(5.16)

5.3.1 Differential GNSS EKF Augmented with Cooperative Ranging

The measurement vector, \( \mathbf{Z}_{DD} \), is formed by stacking the double-differenced carrier-phase and pseudorange L1 and L2 data and cooperative ranging measurement, respectively, as shown below in Eq. 5.17.

\[
\mathbf{Z}_{DD} = \begin{bmatrix}
\nabla \Delta \Phi_{L1} \\
\nabla \Delta \Phi_{L2} \\
\nabla \Delta \rho_{L1} \\
\nabla \Delta \rho_{L2} \\
\Delta \upsilon
\end{bmatrix}
\]  
(5.17)

To form the observation matrix, \( \mathbf{H}^{obs}_{DD} \), there are a few calculations that need to be done. First is to determine the unit-vector between the reference satellite and the UGV location, as shown in Eq. 5.18. Then the unit-vector between each satellite and the UGV, Eq. 5.19 is determined. Finally, the first three columns of \( \mathbf{H}^{obs}_{DD} \) are formed by differencing all satellites (i.e., including the UGV measurement) with the
reference satellite’s unit vector, as shown in Eq. 5.20.

\[ \mathbf{R}^{refSat}_{UGV} = \frac{\mathbf{refSat}_{xyz} - \mathbf{UGV}_{xyz}}{\| \mathbf{refSat}_{xyz} - \mathbf{UGV}_{xyz} \|_2} \quad (5.18) \]

\[ \mathbf{R}^{Sat,k}_{UGV} = \frac{\mathbf{Sat}^k_{xyz} - \mathbf{UGV}_{xyz}}{\| \mathbf{Sat}^k_{xyz} - \mathbf{UGV}_{xyz} \|_2} \quad (5.19) \]

\[ \mathbf{R}^{Sat,k}_{refSat} = \frac{\mathbf{R}^{Sat,k}_{UGV} - \mathbf{R}^{refSat}_{UGV}}{\| \mathbf{R}^{Sat,k}_{UGV} - \mathbf{R}^{refSat}_{UGV} \|_2} \quad (5.20) \]

The rest of the \( \mathbf{H}^{obs}_{DD} \) matrix is populated, and shown in 5.21 with \( \lambda \), carrier–phase wavelength, of the respective satellite systems, only for the carrier–phase L1 and L2 double–differenced observation in order to account for the unknown ambiguities. Unlike the undifferenced GNSS EKF observation matrix, calculating the single differenced data eliminates the satellite clock bias and ephemeris errors and the double–differenced measurements cancel out the receiver clock bias. When the cooperative ranging radio updates are employed, \( \mathbf{H}^{obs}_{DD} \) only slightly differs. The UGV is considered an additional GNSS satellite measurement and its position is added to the end of the observation matrix. \( n \) is the number of satellites observed at that
5.3.2 Integer Ambiguity Fixing

Using carrier–phase data requires solving the whole number of cycles. The carrier–phase measurements are from the receiver to each of the visible satellites. The integer part of the state vector is associated with the carrier–phase ambiguities. The parts of the state vector and the error-covariance matrix that are associated with the ambiguity states are input into the Least-squares AMBiguity Decorelation Adjustment (LAMBDA) method. LAMBDA was created by Peter Teunissen and the researchers at Delft University to compute the integer ambiguity set that is most probable [97].

The LAMBDA method uses the integer least squared (ILS) solution for integer estimation based on the float ambiguity vector and the associated variance-covariance matrix. The fixed ambiguities are tested against the ratio of the quadratic forms of residuals of the most-likely and the second most-likely integer candidates. This ratio
test, Eq. 5.22, is compared to a critical value.

\[
\text{Accept Fix iff } \frac{F(N^1)}{F(N^2)} < \frac{1}{C}
\]  \hspace{1cm} (5.22)

The best candidate, \( N^1 \), and the second-best candidate, \( N^2 \), are the two candidates that minimize the quadratic cost function \( F(N) \). \( C \) is the critical value and is set to a constant in this study. In this study, an aggressive ratio was used—2. The smaller values of \( C \) will lead to more candidates failing, but a less chance of incorrectly accepting a set of incorrect integer ambiguities [97].
Chapter 6

UGV Strategies

Portions of this chapter were described in a 2016 ION GNSS+ Conference Proceedings article (Sivaneri and Gross, 2016) [1] and a 2017 AST Journal Paper (Sivaneri and Gross, 2017) [2]

The three strategies that were evaluated included: (1) having the UGV choose the minimum PDOP of the UAV if it were to select from points immediately around the UGV, (2) having the UGV calculate the minimum PDOP of the UAV if it were to be located anywhere within a grid centered at the UAV, then moving in the direction of this Regionally Optimal location, and (3) having the UGV calculate the minimum PDOP of the UAV, using nonlinear optimization, located anywhere within a grid centered at the UAV, then moving in that direction. For the approaches, the maximum distance that the UGV is assumed to move over one GNSS measurement update interval is one meter.

6.1 Locally Greedy Strategy

In this approach, first the UAV receives the signals from all available GNSS satellites to calculate its position solution. After the UAV communicates the satellites it has in-view to the UGV, the UGV determines which location it should move, Eq. 6.2, in
order to reduce the UAV’s PDOP. This is accomplished by the UGV calculating what
the UAV’s PDOP would become when incorporating a UWB ranging update from
each of the UGV’s candidate locations. With the Locally Greedy approach, the list
of candidate position only includes positions immediately surrounding the UGV. For
this implementation, the number of positions surrounding the UGV were discretized
to be the ten points that encircle the UGV’s current location.

To implement this approach, ten candidate UGV heading angles, shown in Eq.
6.1, were selected and candidate positions were calculated using Eq. 6.2.

\[ \Psi_{\ell=1:10} = \left[ 0, \ldots, 2\pi \right] \]  

\[ r_{k+1}^{UGV,ENU} = \begin{bmatrix} r_{k}^{UGV,E} + d \times \cos(\Psi_{\ell}) \\ r_{k}^{UGV,N} + d \times \sin(\Psi_{\ell}) \\ 0 \end{bmatrix} \]  

where \( r_{k+1}^{UGV,ENU} \) is the UGV’s candidate location for heading angle \( \Psi_{\ell} \), \( r_{k}^{UGV,E} \) is the
UGV’s current East position, \( r_{k}^{UGV,N} \) is the UGV’s current North position, \( d \) is the
move distance of the UGV, \( \Psi_{\ell} \) is the candidate heading location around the UGV
that is being evaluated. With each candidate UGV location, the UAV’s GNSS-only
Geometry Matrix, \( G \), is augmented using unit vector to the candidate UGV position
and current best estimate of the UAV’s position.

\[ u_{uwb} = \frac{r_k^{UAV} - r_{k+1}^{UGV,ENU}}{||r_k^{UAV} - r_{k+1}^{UGV,ENU}||_2} \]  

where the \( u_{uwb} \) is the unit vector distance between the UAV and the candidate UGV
position. With the set of UAV Geometry matrices augmented with each UGV can-
didate location, the PDOP for each candidate 1 to \( \ell \) is evaluated, and the minimum
PDOP is selected as indicated in Eq. 6.4.

\[
PDOP_{\text{min}} = \arg\min(PDOP_1, \, ..., \, PDOP_N) \tag{6.4}
\]

Once the minimum PDOP of the UAV is identified, the UGV is moved to the location that corresponds to the minimum UAV PDOP.

Additional UGV path planning logic was also included to ensure that it does not get too close to the UAV. This is included to ensure that the UGV doesn’t also enter the GNSS-challenged environment, itself. For the time being, this is implemented as a simple perimeter of radius 70 meters around the UAV’s best-known location as a ‘no-UGV-zone’. As such, if the UGV’s next desired trajectory position falls inside the perimeter, the following steps are taken. First, the slope of the distance between the UAV and UGV is found using Eq. 6.5.

\[
m = \frac{\Delta r_N}{\Delta r_E} \tag{6.5}
\]

where \(m\) is the slope, \(\Delta r_N\) is the North component of the distance between the UAV and UGV, and \(\Delta r_E\) is the East component of the distance between the UAV and UGV. Next, the intersection of the perimeter and the UGV, is determined, based on the slope and the equation for a circle as shown in Eq. 6.6 and 6.7.

\[
r_{UGV,E}^{k+1} = \text{sign}(r_k^{UGV,E}) \sqrt{\frac{r_{\text{perim}}^2}{(m^2 + 1)}} \tag{6.6}
\]

\[
r_{UGV,N}^{k+1} = r_{k}^{UGV,N} \tag{6.7}
\]

where \(r_{k+1}^{UGV,E}\) is the UGV’s next East position, \(r_{\text{perim}}\) is the radius of the perimeter, and \(r_{k+1}^{UGV,N}\) is the UGV’s next North position. In Eq. 6.6, the sign operator is to ensure UGV is located in the proper quadrant of the circle.
6.2 Regionally Optimal Strategy

For the *Regionally Optimal* cooperative UGV path planning strategy, a square grid is setup with the UAV at the center. Then, cooperative-range-augmented-PDOP from including a ranging observation emanating from every point on the grid is computed. As an example of this approach, Figure 6.1 shows the percentage reduction possible for the square grid at one time-step, for one simulation scenario. The yellow represents a region in which a 70 % reduction is achievable. After evaluating this grid, the minimum overall PDOP is determined is displayed in Eq. 6.8.

\[
P D O P_{\text{min}} = \text{argmin}(PDOP_{1\text{grid}}, \ldots, PDOP_{n\text{grid}})
\]  

(6.8)

Once the East and North location of where the PDOP is minimum is found, the UGV is driven in that direction. This is accomplished by first determining the distance between the current UGV position and the location where the PDOP of the UAV is

Figure 6.1: Regional optimal strategy using a grid to calculate minimum augmented-PDOP
minimum over the grid as shown in Eq. 6.9 and 6.10.

\[
\Delta r_{UGV,E}^k = r_{UGV,E}^k - E_{PDOP_{min}}^{Grid} \tag{6.9}
\]

\[
\Delta r_{UGV,N}^k = r_{UGV,N}^k - N_{PDOP_{min}}^{Grid} \tag{6.10}
\]

where \( E_{PDOP_{min}}^{Grid} \) is the location where the PDOP would be minimized for the UGV location within the grid. From here, the heading angle, \( \Psi \), is found to determine which direction the UGV should move, as shown in Figure 6.2. The heading was calculated by Eq. 6.11.

\[
\Psi = \tan^{-1}(\Delta r_{UGV,N}^k, \Delta r_{UGV,E}^k) \tag{6.11}
\]

Since there is a constraint that the UGV can only move a maximum of one meter per time–step, the next UGV location is determined based on the move distance and the heading, Eq. 6.12 and 6.13.

\[
r_{k+1}^{UGV,E} = r_{k}^{UGV,E} + d * \cos(\Psi) \tag{6.12}
\]
\[ r_{k+1}^{UGV,N} = r_k^{UGV,N} + d \ast sin(\Psi) \]  

(6.13)

where \( d \) is the maximum move distance of the UGV and \( \Psi \) is the heading angles. As stated, there is a check in place to make sure the UGV does not come too close to the UAV. If it does, the same procedure is followed as described in Section 6.1.

### 6.3 Nonlinear Optimization Strategy

The nonlinear optimization used in this study was the \textit{fmincon} algorithm provided by MATLAB optimization toolbox. This algorithm has various inputs and outputs that can be specified and changed depending on the optimization problem. In the \textit{fmincon} function, several nonlinear algorithms can be selected, Trust Region Reflective algorithm, Active Set algorithm, Sequential Quadratic Programming algorithm, or the Interior Point Algorithm. Several trials were conducted on the different algorithms and it was decided that for this study the Interior Point Algorithm was the best algorithm. These tests were based on a few different criteria such as time elapsed for the algorithm and the amount of PDOP reduction. The Interior Point Algorithm used in this study is, more specifically, an Interior Point Barrier algorithm. The added logarithmic term, shown in Eq. 4.34, is called a barrier function.

To solve this approximation problem, this method uses one of two main types of steps at each iteration, a direct step or a conjugate gradient step. The direct step method attempts to solve the KKT equations, Eq. 4.33, for the approximate problem by a linear approximation, also called the Newton step. At each iteration the algorithm decreases a merit function, such as Eq. 6.14.

\[ f_\mu(y, s) + \nu|| (h(y), g(y) + s) || \]  

(6.14)
The \( \nu \) may increase with iteration number and force the solution towards feasibility. If the merit function is not decreased with the previous step, the algorithm will reject the previous step and try a new step. If a complex value is obtained either for the objective function or for one of the nonlinear constraints, the algorithm will reject the step and try a new step. Using a linearized Lagrangian to solve Eq. 4.33,

\[
\begin{bmatrix}
H & 0 & J_h^T & J_g^T \\
0 & S\Lambda & 0 & -S \\
J_h & 0 & I & 0 \\
J_g & -S & 0 & I \\
\end{bmatrix}
\begin{bmatrix}
\Delta y \\
\Delta s \\
-\Delta x \\
-\Delta \lambda \\
\end{bmatrix} =
\begin{bmatrix}
\nabla f - J_h^T x - J_g^T \lambda \\
S\lambda - \mu e \\
h \\
g + s \\
\end{bmatrix}
\]  

(6.15)

where \( J_g \) is the Jacobian of the constraint function \( g \), \( J_h \) is the Jacobian of the constraint function \( h \), \( S = \text{diag}(s) \), \( \lambda \) is the Lagrange multiplier vector associated with constraints \( g \), \( \Lambda = \text{diag}(\lambda) \), \( x \) is the Lagrange multiplier vector associated with constraints \( h \), and \( e \) is a vector of ones and the same size as \( g \). The Interior Point Barrier method tries to solve for \( \Delta y \) and \( \Delta s \) by making an LDL factorization of the matrix. This factorization is the most expensive step, computationally. This factorization allows for checking if the projected Hessian is positive definite, if not the conjugate gradient step is used.

The algorithm attempts to solve the problem by using the direct step, but if it fails, the next solution is to try the conjugate gradient step, using a trust region. The algorithm will go directly to the conjugate gradient step if the approximate problem is not locally convex near the current iterate. This method attempts to solve the approximate problem Eq. 4.34, by adjusting \( y \) and \( s \), keeping \( s \) positive. The goal is to minimize a quadratic approximation to the approximate problem in a trust region but within the bounds of the constraint equation(s). The Lagrange multipliers are calculated by approximately solving the KKT equations Eq. 6.16, with \( \lambda \) being
positive.

\[ \nabla_y L = \nabla_y f(y) + \sum_{i} \lambda_i \nabla g_i(y) + \sum_{j} x_j \nabla h_j(y) \quad (6.16) \]

A step \((\Delta y, \Delta s)\) is used to approximately solve, Eq. 6.17,

\[
\min_{\Delta y, \Delta s} \nabla f^T \Delta y + \frac{1}{2} \Delta y^T \nabla^2_{yy} L \Delta y + \mu \varepsilon^T S^{-1} \Delta s + \frac{1}{2} \Delta s^T S^{-1} \Lambda \Delta s \quad (6.17)
\]

subject to linearized constraints, Eq. 6.18

\[
g(y) + \mathbf{J}_g \Delta x + \Delta s = 0
\]

\[
h(y) + \mathbf{J}_h \Delta x = 0 \quad (6.18)
\]

The gradient step method minimizes the norm of the linearized constraints inside a region with a radius. Next, Eq. 6.17 is solved with the constraints and are matched to the residual from solving Eq. 6.18 while staying within the trust region of Radius \(R\) and keeping \(s\) positive.

One of the options that can be selected when using the Interior Point algorithm is the calculation of the Hessian. The choices are:

- **Broyden, Fletcher, Goldfarb, and Shanno (BFGS):** Calculates the Hessian by a dense quasi-Newton approximation

- **LBFGS:** Calculates the Hessian by a limited-memory, large-scale Quasi-Newton approximation

- **Finite Difference:** Calculates a Hessian-times-vector product by finite differences of the gradient(s)

- **Other:** User specifies which function to compute the Hessian
In this study, the BFGS for the calculation of the Hessian approximation was used. Other options include setting the tolerances on the objective function, tolerances on the constraint violation, finite difference step size, finite difference type, maximum function evaluations, max iterations, optimality tolerance, gradients for nonlinear constraint and objective functions, and step tolerances. The selected options for this study will be discussed in Chapter 7.

In this approach, the nonlinear optimization, it is similar to the Regionally Optimal strategy where there is a square grid setup with the UAV at the center, at each time–step. Except, instead of a brute force method of searching at every point in the grid, this method uses the algorithm discussed to find the minimum PDOP in the search area. Since this method does not need to calculate the PDOP at every grid point, the grid around the UAV can be expanded due to it being more computationally efficient compared to the Regionally Optimal approach. An example of this approach, Figure 6.3 shows the percentage reduction possible for the square grid at one time–step, for a random simulation scenario. The yellow region represents a region where a 30% reduction is possible. The white circle at the center of the plot shows the ‘no-UGV-zone’. The output for the nonlinear optimization directly solves for the lowest PDOP and best UGV position, so there is no use for Eq. 6.8.

Once the algorithm calculates where the optimal UGV location should be, it follows the same process as discussed in Chapter 6.2, and using Eqs. 6.9 - 6.13. Another benefit of using this method, is the ability to add in constraints into the algorithm with ease. With the two previous UGV strategies, Eqs. 6.5 - 6.7, were needed to be able to keep the UGV outside of the ‘no-UGV-zone’. But with this approach, an inequality constraint equation, Eq. 6.19, is used to exclude that area around the UAV from being used in the calculation of the PDOP, so the UGV will
Figure 6.3: Nonlinear optimization strategy to calculate percent reduction of augmented-PDOP

not be allowed in that area.

\[ c = -\sqrt{UGV_{East}^2 + UGV_{North}^2 + R} \]  

(6.19)

\( R \) is the ‘no-UGV-zone’ radius around the UAV. For the \textit{fmincon} function, the algorithm is optimizing the UGV location based on, as with the other algorithms in this study, PDOP. The location of the UGV is the starting point for the algorithm, which it optimizes until the minimum PDOP is found, while keeping the constraints in mind. Once the optimal UGV location is calculated, a similar method is used to find the heading, as in the previous strategies for the UGV. After the UGV moves, the previous process is repeated.

6.3.1 Path Planning

A path planning approach was examined to see if there is any benefit of having the UGV calculate the next ‘n’ epochs compared to moving at each epoch. The idea for path planning for this study is that the GNSS satellite ephemeris (i.e. orbital elements) are known, so their positions can be predicted in advance. For this study,
10 epochs was chosen to be the number of epochs for the path planning approach. The difference between single epoch and the path planning approach is taking in the matrices for UGV initial start, satellite geometry, and the UAV’s location, rather than information for one epoch. The inequality constraint is similar in the single epoch approach, except now there is an inequality constraint for each epoch, Eq. 6.20.

\[ c(n) = \sqrt{UGV_{n,East}^2 + UGV_{n,North}^2} \]  (6.20)

In the previous equation, \( n \) is the number of epochs. Another constraint equation, Eq. 6.21 was added for the distance the UGV was allowed to move at one epoch, with respect to the previous epoch. For example, at epoch 6 it can only be a maximum of the move distance away from epoch 5.

\[ ceq(j) = \sqrt{(UGV_{j,East} - UGV_{j+1,East})^2 + (UGV_{j,North} - UGV_{j+1,North})^2} + d \]  (6.21)

In Eq. 6.21, \( j \) is \( n - 1 \) epochs and \( d \) is the move distance allowed for the UGV at each time-step. With the path planning approach, the PDOP, Eq. 6.22 now is optimized over the entire ‘n’ epochs.

\[ \sum PDOP_{min} = (PDOP_{path}^1 + PDOP_{path}^2 + ... + PDOP_{path}^n) \]  (6.22)

Once the algorithm determines the minimum PDOP location for the ‘n’ epochs, and the corresponding UGV location. Then the heading for the first epoch is calculated, by using the same methods as in the Regionally Optimal approach. And for the subsequent UGV locations, the heading is calculated from the updated UGV position of the previous epoch to the optimal UGV location, found from the nonlinear
optimization, of the next epoch.

\[
\Delta r_i^{UGV,E} = r_i^{UGV,E} - E_{i+1}^{path} 
\] (6.23)

\[
\Delta r_i^{UGV,N} = r_i^{UGV,N} - N_{i+1}^{path} 
\] (6.24)

where \( E, N_{i+1}^{path} \) is the location where the PDOP would be minimized for the UGV location from the nonlinear optimization. From here, the heading angle, \( \Psi \), is found to determine which direction the UGV should move, as shown in Figure 6.2. The heading was calculated by Eq. 6.25.

\[
\Psi_i = \text{atan2}(\Delta r_i^{UGV,N}, \Delta r_i^{UGV,E}) 
\] (6.25)

Since there is a constraint that the UGV can only move a maximum of one meter per time-step, the next UGV location is determined based on the move distance and the heading, Eq. 6.26 and 6.27.

\[
r_{i+1}^{UGV,E} = r_i^{UGV,E} + d \cdot \cos(\Psi) 
\] (6.26)

\[
r_{i+1}^{UGV,N} = r_i^{UGV,N} + d \cdot \sin(\Psi) 
\] (6.27)

where \( d \) is the maximum move distance of the UGV and \( \Psi \) is the heading angles. The equations are for the assumption that at each future epoch the UAV’s position is known. Even if you know the exact path of the UAV, there is still some deviation with where the UAV actually is and where the UGV thinks the UAV is going to be located. This deviation can cause the calculation of the PDOP to differ. To combat this problem, the PDOP is still taken as a sum, but now there is a weight associated
with each additional PDOP term as shown in Eq. 6.28.

\[ \sum PDOP_{\text{min}} = (W_1 \cdot PDOP^1_{\text{path}} + W_2 \cdot PDOP^2_{\text{path}} + \ldots + W_{10} \cdot PDOP^{10}_{\text{path}}) \]

(6.28)

This weighting system tells the optimizer to give more weight to the first epoch then decrease the weight as the epoch number increases. This tells the constraint function to put a greater emphasis on the location of the first UGV movement.

### 6.3.2 Multiple Unmanned Ground Vehicles

Having multiple vehicles is becoming more and more common with technological advances and easier to implement it in a system. This dissertation explores the idea of having a second UGV in the field to help the UAV navigate through the urban canyon. With the addition of another vehicle, it introduces more variables and more dynamics that need to be included in the equations. The nonlinear optimization was used to explore the use of having multiple UGVs assisting the UAV. There were two approaches, the centralized and decentralized nonlinear optimization. For the decentralized approach, it has one UGV moving first then the second will move. The order of the UGV movement is decided on the PDOP of each UGV, and whichever has the highest PDOP will move first and plan out the next ‘n’ steps. After the first UGV finds the optimal path, based on the nonlinear optimization solution, then the second UGV will calculate the best path. The centralized approach is known for being more expensive and having one central vehicle tell the other vehicles where to move. The ‘main’ UGV, lower PDOP, decides the best location for both the UGVs to move. After this calculation, both UGVs move toward the same location.
Chapter 7

Simulation Setup

*Portions of this chapter were described in a 2016 ION GNSS+ Conference Proceedings article (Sivaneri and Gross, 2016) [1] and a 2017 AST Journal Paper (Sivaneri and Gross, 2017) [2]*

This chapter describes the simulation setup used in this study such as the different parameters and different error sources used for the generation of the simulated GNSS data as well as the nonlinear optimization parameters.

The raw GNSS data used to simulate the data for this study was generated using a commercially available SatNav–3.04 Toolbox [98], which is a GNSS constellation simulation toolbox. Simulation configuration inputs were defined for the generation of GPS signals and error sources, and several parameters were randomized for each simulation scenario including the simulation location (i.e., latitude, longitude, and height), the time of the GPS week, and the length of the flight. These inputs were selected at random in order to give each flight a different GNSS satellite constellation geometry and atmospheric effects. For more information on the generation of data, please refer to [99] for a more detailed description. The scale of the GPS error sources used in the simulation are listed in Table 7.1.

A GNSS error-source particularly important to this study was GNSS multipath
Table 7.1: Simulation parameters

<table>
<thead>
<tr>
<th>Error-Sources</th>
<th>Model Parameters</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ionospheric Delay</td>
<td>First order ionospheric effects mitigated with dual-frequency</td>
<td>linear scale factor randomly selected between [0.7,1]</td>
</tr>
<tr>
<td>Multipath</td>
<td>1.0 intensity: $\sigma = 8m, \tau = 2\text{min}$</td>
<td>linear scale factor randomly selected between [0,2]</td>
</tr>
<tr>
<td>Orbits</td>
<td>Orbits $\sigma = 5cm$</td>
<td>Description provided in paragraph below</td>
</tr>
<tr>
<td>Phase Ambiguity</td>
<td>Random initialization and phase breaks correlated with UAV attitude</td>
<td>likelihood varied from [0.008,0.02]</td>
</tr>
<tr>
<td>Receiver Clock Bias</td>
<td>Initial Bias $\sigma = 30ns, \delta\tau_b = 100ns$</td>
<td>N/A</td>
</tr>
<tr>
<td>Thermal Noise</td>
<td>$\sigma_{\rho} = 0.32m$, $\sigma_{\phi} = 0.16\lambda$</td>
<td>linear scale factor randomly selected between [0,1]</td>
</tr>
<tr>
<td>Tropospheric Delay</td>
<td>Percent of error assumed handled by broadcast correction</td>
<td>Modified Hopfield with linear scale factor randomly selected between [0,1.5]</td>
</tr>
</tbody>
</table>

errors. As such, for this work the multipath error was increased to simulate the GNSS-challenged environment of an urban-canyon. Multipath was modeled as a first order Gauss-Markov error source and with a $\sigma = 8$ meters and a time constant, $\tau$ of 2 minutes. To simulate an urban canyon, multipath was set as a function of the elevation of the satellite and added to the original multipath formed in the simulation, displayed in Eq. 7.1.

$$mp_{UC} = mp + \frac{1}{\sin(el^k) + \epsilon}$$  \hspace{1cm} (7.1)

where $mp_{UC}$ is the multipath in an urban canyon, $mp$ is the multipath modeled, $el^k$ is the elevation of the satellite, and $\epsilon$ is the offset.

Furthermore, to simulate a GNSS-challenged environment, a high elevation mask,
i.e. buildings in an *urban canyon*, was incorporated as displayed in Figure 7.1. The masks were held constant throughout the simulated flight and for all data sets. The left panel shows the elevation and azimuth of the satellites visible in one specific simulation. The blue line shows the progression of the azimuth and elevation of the satellites without a mask and the red line shows the satellites’ elevation and azimuth after the high elevation mask was imposed. The right panel shows what the PDOP was before and after the mask, with the blue representing the PDOP before a mask was applied and the red line representing the PDOP after a mask was applied.

For the cooperative ranging radio measurements, a zero-mean white-noise error of 5 cm was added to the true range, as commercially available UWB systems can achieve 2 cm ranging [100]. This simulation also included an orbit error model to represent the errors in the GNSS broadcast ephemeris. The satellite ephemeris errors were modeled by differencing the broadcast products provided by the International GNSS
Service (IGS) and the Center for Orbit Determination (CODE). A multi-sinusoidal model was fitted to the error and was based on the time of day. More information can be found in [99]. The UAV and UGV were assumed to start at arbitrary positions.

For the nonlinear optimization algorithm, there are a set of parameters that can be selected and adjusted based on the problem specification. These set of parameters are shown in Table 7.2. Changing each parameter has a different effect on the solution of the problem and the execution time of the nonlinear optimization.

Table 7.2: Parameters for nonlinear optimization algorithm

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm</td>
<td>Interior-Point</td>
<td>Others include: trust region, sqp, sqp legacy, active set</td>
</tr>
<tr>
<td>Check gradients</td>
<td>False</td>
<td>Compare gradients of objective to finite-differencing derivatives</td>
</tr>
<tr>
<td>Constraint Tolerance</td>
<td>$1e^{-5}$</td>
<td>Tolerance on the constraint violation</td>
</tr>
<tr>
<td>Diff. Max Change</td>
<td>Infinity</td>
<td>Maximum change in variable for finite-difference variables</td>
</tr>
<tr>
<td>Diff. Min Change</td>
<td>0</td>
<td>Minimum change in variable for finite-difference variables</td>
</tr>
<tr>
<td>Fin. Diff. Step Size</td>
<td>$\sqrt{\varepsilon}$</td>
<td>Step size factor for finite differences</td>
</tr>
<tr>
<td>Fin. Diff Type</td>
<td>Forward</td>
<td>Used to estimate gradients: Forward, central</td>
</tr>
<tr>
<td>Max Function Evaluation</td>
<td>$1e^{-5}$</td>
<td>Maximum number of function evaluations</td>
</tr>
<tr>
<td>Maximum Iterations</td>
<td>$1e^{3}$</td>
<td>Maximum number of iterations allowed</td>
</tr>
<tr>
<td>Optimality Tolerance</td>
<td>$1e^{-6}$</td>
<td>Termination tolerance on the first-order optimality</td>
</tr>
<tr>
<td>Constraint Gradient</td>
<td>Finite Differences</td>
<td>Calculation of gradient constraint functions</td>
</tr>
<tr>
<td>Objective Gradient</td>
<td>Finite Differences</td>
<td>Calculation of gradient objective function</td>
</tr>
<tr>
<td>Step Tolerance</td>
<td>$1e^{-10}$</td>
<td>Termination tolerance on $x$</td>
</tr>
</tbody>
</table>

The results generated for this approach were from 50 trials, with each trial having several parameters that were randomized, UAV latitude and longitude, UGV latitude
and longitude, and time of week. The elevation and azimuth mask were held constant throughout the trials. These inputs were selected to be random to give a wide range of GNSS satellite constellation geometry.
Chapter 8

Experimental Setup

In addition to the simulation work, this study includes the design and testing of a UAV and UGV, which includes the description of the on-board electronics, the circuit boards, sensors, and the experimental setup. A UGV and a UAV were built and instrumented to approximate the specifications in the simulation, to test whether the approach of using a ranging source to minimize the positioning accuracy of the UAV.

8.1 Unmanned Aerial Vehicle

The unmanned aerial vehicle was bought as an off-the-shelf hobby quadcopter, as shown in Figure 8.1. As it is says in the name, the quadcopter has 4 motors and 4 rotors that are in an ‘X’ configuration. The arms of the quadcopter are set 90 deg from each other. In this figure, the UAV is carrying the following sensors, a Ublox GPS receiver and antenna [101], Time Domain PulsON P-410 UWB ranging radio [100].

Using the Ublox GPS receiver and antenna, did not provide a very precise navigation solution. The solution had an error of 5 meters associated with it, Figure 8.2 when walking around with the UAV and having the UGV act as a base station.
The UAV was left on the ground for 10 minutes to be able to initialize before it was moved. There are only a few spots where there is a good navigation solution, green spots, but most of the solution is average solution, yellow. The actual path was a rectangle, with a long length and a short width, but the path from the solution does not resemble that. This test was done on campus, so the thought was that the buildings might have played a role in the poor performance.

Taking the UAV and base station to an open field did not provide better results, in fact the results were worse as shown in Figure 8.3. The UAV was flying around, while first being on the ground for an initialization. The solution’s path has an error of 20 meters.

The next trial was swapping out the Ublox receiver and antenna for a dual frequency Novatel receiver and antenna. This change is displayed in Figure 8.4.

A close up of the Novatel receiver and breakout board is shown in Figure 8.5. The breakout board provides power to the Novatel receiver, which is indicated by the top LED on the breakout board. The bottom LED indicates when the receiver has a satellite ‘fix’, meaning there are enough satellites in view for the receiver to compute a navigation solution. The breakout board houses an SD card slot, which saves the
Figure 8.2: Navigation solution using an Ublox GPS receiver and antenna, with an initialization time of 10 minutes

Figure 8.3: Navigation solution using an Ublox GPS receiver and antenna
raw GPS data from the receiver and the calculated UAV position. The data from the SD card is used in post processing.

While the new antenna adds more weight to the UAV, the navigation solution improves. The benefit here outweighs the negative aspect of the weight. Initially the UAV would fly around and collect and save the GPS data. Now the UAV needs to let the UGV know, where the UAV is, in XYZ coordinates. To send the data to the UGV, a Xbee modem [102] is used, shown in Figure 8.7. The data is sent to UGV every second. Other instruments/electronics on board are a NetBurner MOD541X [103] which is used with a breakout board to interface the different sensor data. The NetBurner has a SD card and data is stored on the card, from the IMU, and the Pulse Per Second (PPS) from the Novatel receiver, using the NetBurner firmware.

While the new antenna provided a better solution, this made the UAV heavier and unable to fly easily. Thus, the UWB radio, the breakout board, IMU, and the Netburner were removed from the UAV, shown in 8.9. This allowed the UAV to be lighter and more balanced allowing for an easier flying experience.

Figure 8.10 shows the block diagram of the UAV’s sensors and equipment.

The battery used for the UAV system is a 6.6 Volt Lithium Polymer battery.
Figure 8.5: Novatel receiver and breakout board

Figure 8.6: Navigation solution using a Novatel GPS receiver and antenna
Figure 8.7: Xbee modem on the UAV

Figure 8.8: Netburner on the UAV

Figure 8.9: Final UAV setup
The Novatel receiver and antenna sends the satellite positions and the UAV’s XYZ position to the UAV’s GPS board, which was designed and made in house. This data is saved on the SD card and the UAV’s position is sent to the Xbee Modem.

### 8.2 Unmanned Ground Vehicle

The UGV was put together in-house and sensors were installed on the UGV in different physical locations on the UGV. The UGV, Figure 8.11, is equipped with, a NovAtel Pinwheel antenna [104] with a NovAtel GPS receiver [105] and an Analog Device 16485 AMLZ IMU [106]. A NetBurner MOD541X is used with a breakout board to interface the different sensor data. The NetBurner has a SD card and data is stored on the card, from the IMU, the Pulse Per Second (PPS) from the GPS receiver, and a counter for time-tag alignment using the NetBurner firmware. The data from the GPS receiver is stored on a SD card located on the GPS breakout board.

The UGV contains a single board computer running Ubuntu 14.04. This on-board computer controls the speed controllers of each of the 4 wheels. Initially the UGV was controlled using a laptop with an Xbee modem, sending commands to the UGV’s Xbee Modem 8.12 and the on-board computer took those commands and sent them to the speed controllers, to move the UGV. Now, the UGV has been made to be an
Figure 8.11: UGV
autonomous system, implementing one of the UGV control strategies.

The NovAtel antenna is located at the top of the mast and the GPS breakout board and Novatel Receiver are located halfway up the mast (across from the UWB). The laptop sends the input over a Xbee modem [102] to the UGV’s Xbee modem. This input controls the speed controllers on the UGV.

Figure 8.13 shows the block diagram of the UGV’s sensors, equipment, and computer used aboard the UGV.

The battery used for the UGV system is a Lithium Polymer battery. Robot Operating System (ROS), setup with help from Nick Ohi, was used to control the UGV. The Novatel receiver and antenna sends the satellite positions and the UGV’s XYZ position to the UGV’s GPS board, which was designed and made in house. This data is saved on the SD card, and the UGV’s position is sent to the on-board computer. The power for the GPS receiver and UGV’s GPS board comes from the
autopilot board. The Xbee modem is connected to a serial port on the computer, as well as the left and right motors that control the UGV’s wheels.

8.3 Experimental Procedure

Once all the components were tested and the UAV was able to fly, multiple flight tests were conducted to test the *Locally Greedy* cooperative navigation approach. The procedure that was followed for an experimental data collection is listed below.

1. Determine a suitable testing location with an unobstructed view of the sky and decent amount of open field

2. Setup the base station GPS antenna and receiver and laptop to collect the base station data, as displayed in Figure 8.14

3. Place UAV in the middle of the field and connect battery to UAV’s GPS board, Figure 8.15

4. Connect the battery to the UGV and boot up the computer, Figure 8.16

5. Once UGV and UAV have a ‘fix’, plug in motor batteries for UAV

6. Start algorithm script for UGV movement and data collection
7. Once UGV starts moving, start flying UAV, Figure 8.17

8. Have UGV moving and UAV flying for a few minutes, collecting data

9. Land UAV and turn off algorithm script on UGV

10. Unplug motor and UAV GPS batteries, shutdown UGV computer and unplug remaining batteries

11. Transfer data from SD card onto laptop, and verify successful data acquisition
Figure 8.16: UGV computer and program being initialized before starting movement
8.4 Data Acquisition

The experimental demonstration is a simplified version of the simulation, the goal of the experimental demonstration is to show that the UGV strategy is feasible and the use of a cooperative UGV reduces the uncertainty of the UAV. The assumption in this part is the UAV has great positioning and knows where it is. This assumption is used only for calculating the movement of the UGV, not in the post processing approach when using the Differential GNSS EKF.

The previous sections describe the hardware side of the UAV and UGV, this section discusses the logic and the software written for the experimental setup. The information needed to calculate the UGV’s next move are the satellite’s XYZ ECEF position, the UGV’s XYZ position, and the UAV’s XYZ position. The satellite’s and UGV’s position are obtained from the Novatel antenna and receiver on the UGV and is sent by serial port to the computer on the UGV and saved to the SD card on the
GPS board. The UAV’s position is from the UAV’s Novatel antenna and receiver and sent to the UGV by the Xbee modem and is also saved on the SD card on the UAV’s GPS board. The Xbee modem on the UGV receives the data and sends it to the computer on-board the UGV. Two different software scripts were written to read the data coming over the two serial ports and were saved to separate files, one for UGV information and one for UAV information. The main script, takes in this info and check to see if there is a ‘fix’ for the UAV’s position. This means that the Novatel antenna has more than 4 satellites in view and the receiver is able to calculate the position of the UAV. If there is no fix, the loop breaks and starts from the beginning, getting new UAV data from the file. If there is a fix on the UAV’s position, the script moves on to extracting the UAV’s position from the file. Once that is completed, the same check is completed to see if the UGV’s position has a fix. If not, the loop breaks and starts from the beginning and grabs new UAV data. But if there is a fix, the UGV’s position and the satellites’ position are extracted from the data. The elevation from the UAV to each of the satellites is calculated, and an elevation and azimuth mask is enforced on the satellite data, and the satellites not fitting the criteria are removed. These satellites, UAV’s and the UGV’s position are input into the \textit{Locally Greedy} approach. The algorithm calculates the lowest PDOP and finds the UGV’s next location corresponding to the lowest PDOP. The main algorithm takes the UGV’s next location and sends a command to the 4 encoders (one on each wheel), to move the UGV to the desired location. This process is repeated over and over, until the end of the run, Figure 8.18.
8.5 Reference Solution Post-Processing

After the experiment, in order to process the data through the Differential GNSS Extended Kalman Filter, a few procedures need to happen beforehand. The text files from the SD cards were used to get the satellite information as well as the position of the UAV and UGV. The novaRinex Converter from the GPS Toolkit [107] (GPSTk) was used to convert the data. The converter opens and reads a binary Novatel file and converts the data to Receiver Independent Exchange (RINEX) format observation and navigation files. The Novatel Convert software was used to do the same process for the base station data to get the observation and navigation files. For the post processing analysis, RTKLIB [108] was used to compute the positioning solutions of the UAV and UGV. Using the RTKPOST software, a tool in the RTKLIB, the inputs are the RINEX file and it outputs the positioning solutions by various positioning modes, in this dissertation Ambiguity Fixed Real-time Kinematic was used. For the UAV, the ‘Rover’ was the UAV observation file, the ‘base station’ was the base station’s observation file. To process experimental data, the RINEX navigation message files
of GNSS are needed. These files are a daily GPS or GLONASS Broadcast Ephemeris File, which contain the unique GPS or GLONASS satellite ephemeris messages for each day. The Crustal Dynamics Data Information System (CDDIS) [109] provides these files. Executing the RTKPOST software gives the positioning solution for the UAV during the flight. The process is repeated for the UGV and is exactly the same except in the ‘Rover’ field, the UGV’s observation file is used. The positioning file is the UAV’s and UGV’s XYZ position during the flight. A script was written to extract the UAV and UGV’s satellite XYZ location.

Table 8.1: Parameters for RTKPOST

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positioning Mode</td>
<td>Kinematic</td>
<td>Carrier–based Kinematic positioning</td>
</tr>
<tr>
<td>Frequencies</td>
<td>L1+2</td>
<td>L1 and L2 Dual-frequency</td>
</tr>
<tr>
<td>Filter Type</td>
<td>Combined</td>
<td>Smoother combined solution with forward and backward filter solutions</td>
</tr>
<tr>
<td>Elevation Mask</td>
<td>5</td>
<td>Elevation mask angle (Degrees)</td>
</tr>
<tr>
<td>Satellite Ephemeris/Clock</td>
<td>Broadcast</td>
<td>Broadcast ephemeris</td>
</tr>
<tr>
<td>Solution Format</td>
<td>X/Y/Z-ECEF</td>
<td>X/Y/Z components of ECEF</td>
</tr>
</tbody>
</table>

The next step is to import the files into MATLAB to be able to create double–differences between the carrier–phase and pseudorange L1 and L2 data. Six data files, UAV and UGV position, observation, and satellite files were imported into MATLAB. The observation files contain the L1 and L2 carrier–phase and pseudorange data. The first step to creating the double–differences is to align the time tag for all the data files. The time tag indicates when the measurement was recorded to the text file. The time tags that match up are saved, and the rest is removed from the data set. Due to the fact that the UWB radios are not in use, because of the weight constraint on the UAV, a range measurement between the UAV and UGV had to be formed. This measurement was formed by taking the norm of the difference between
the UAV position and UGV position at every time–step. Then the elevation angle of the satellites, with respect to the rover, was calculated. To simulate a GNSS-challenged environment, a high elevation mask, i.e. buildings in an urban canyon, was incorporated as shown in Figure 8.19. An azimuth criteria was included in the elevation mask. If the satellite was below an elevation angle, but between a certain azimuth angle, then the satellite was not removed. This was to simulate that the satellites at the end of the street of an urban canyon would still be visible even at a low elevation angles. The masks were held constant throughout the flight and for all data sets. The left panel shows the elevation and azimuth of the satellites visible in one specific simulation. The blue line shows the progression of the azimuth and elevation of the satellites without a mask and the red line shows the satellites’ elevation and azimuth after the high elevation mask was imposed. The right panel shows what the PDOP was before and after the mask, with the blue representing the PDOP before a mask was applied and the red line representing the PDOP after a mask was applied.

After the elevation and azimuth mask were applied, the corresponding L1 and L2 carrier–phase and pseudorange satellite data were removed for that time–step. To form the single difference and the double–difference measurements, the same procedure was followed as in 5.3.
Figure 8.19: Left panel: Elevation and Azimuth before and after high Elevation mask was applied
Right panel: PDOP before and after high Elevation mask was applied
Chapter 9

Results

Portions of this chapter were described in a 2016 ION GNSS+ Conference Proceedings article (Sivaneri and Gross, 2016) [1] and a 2017 AST Journal Paper (Sivaneri and Gross, 2017) [2]

The results are divided into the three different phases, first the results of having one UAV and one UGV in simulation, then the results from the path planning approach and having multiple UGVs, and finally the results from the experimental demonstration.

9.1 One UAV and One UGV

Each of the 25 generated flight data sets were run through estimation filters both with and without the aiding from the UGV in order to characterize the performance of including the cooperative strategy. In addition, both the undifferenced GNSS EKF and a double-differenced EKF were employed on each data set. Likewise, both of the cooperative UGV trajectory design strategies were tested for each data set in order to uncover any differences between the two approaches. Figure 9.1 indicates the process used in the simulation for the UAV and UGV.

The first few figures detail a specific example of one simulation trial in which the
Figure 9.1: Simulation overview of the UAV with the cooperative UGV
UGV’s path and the reduction in PDOP are shown. For example, Figure 9.2 shows the path that the UGV takes when employing both the Locally Greedy path (left panel) and the Regionally Optimal (right panel) path planning algorithm.

The green * is where the UGV starts, the black O is where the Locally Greedy path ends, and the yellow x is where the Regionally Optimal path ends. The orange line indicates the Locally Greedy path, and the black line indicates the Regionally Optimal path. The blue * indicates where the UAV starts, and the red X is the end of the UAV path. The blue line shows the entire path of the UAV. The Locally Greedy UGV moves toward the optimal location in the East direction and moves back and forth in the East direction, while also moving in the North direction. Whereas the Regional Optimal strategy moves toward the optimal geometric location and then moves along the constraint boundary. This is expected, as shown in Figure 6.1, the largest reduction of PDOP matches the path of the UGV with Regionally Optimal strategy employed for this particular scenario.

Figure 9.3 shows the PDOP over the entire flight for having no cooperative UGV, the UGV employing the Locally Greedy path planning algorithm, and the UGV em-
ploying the Regionally Optimal approach. The PDOP for the Regionally Optimal algorithm is reduced further than that of the Locally Greedy path planning approach, and both are significantly lower than the case of including no cooperative UGV. As shown in Fig. 9.4, this extra cooperative ranging measurement and PDOP reduction leads to more accurate, smoother, and more consistent positioning performance.

To summarize the result for the undifferenced GNSS EKF, Figure 9.5 shows the median 3D position error for each strategy over the 25 flight simulation trials. In general, the solution improves when using cooperative ranging from a UGV, and the Regionally Optimal approach provides the best solution with a nearly 40% error reduction.

Next, in Figure 9.6 summarizes the median errors of differential GNSS EKF with and without cooperative UGV aiding are shown, where both integer ambiguity fixed and float solutions are shown.

As expected, in all three cases, the carrier–phase ambiguity integer fixed solution
Figure 9.4: Example flight trial of the UAV’s positioning errors, for both filter designs, without a UGV, with a *Locally Greedy* UGV, and with a *Regionally Optimal* UGV.

Figure 9.5: 3D position error of median of data sets with the undifferenced GNSS EKF
performs better when compared to the carrier-phase ambiguity integer float solution. Also, having a UGV significantly improves the solution with the Regionally Optimal UGV path planning strategy proving to be the best approach. Table 9.1 summarizes both the median and average 3D position error for each strategy over the 25 flights. These metrics are shown alongside the PDOP and % of Carrier-Phase Biases successfully integer fixed.

In Table 9.1 it is clear having a cooperative UGV employing the Locally Greedy strategy is better than having no UGV, and that having a UGV that employs the Regionally Optimal strategy is the best scenario in terms of positioning performance. This is consistent in both the undifferenced GNSS EKF and the differential GNSS EKF. Of particular significance, in Table 9.1, it is evident that the 3D position error can be reduced from a median error of 1.17 m to 10-cm when using a cooperative UGV. Additionally, when comparing the average to the median performance over the 25 flights, the use of the cooperative UGV makes the positioning performance much more consistent despite the GNSS-challenged conditions. That is, with the UGV,
<table>
<thead>
<tr>
<th>Filter Type</th>
<th>Strategy</th>
<th>Med. Error (m)</th>
<th>Avg. 3D Error (m)</th>
<th>Avg. PDOP</th>
<th>Avg. % Fixed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Undiff.</td>
<td>None</td>
<td>1.17</td>
<td>2.49</td>
<td>4.66</td>
<td>N/A</td>
</tr>
<tr>
<td>Undiff.</td>
<td>Greedy</td>
<td>0.77</td>
<td>0.79</td>
<td>2.33</td>
<td>N/A</td>
</tr>
<tr>
<td>Undiff.</td>
<td>Optimal</td>
<td>0.70</td>
<td>0.71</td>
<td>1.58</td>
<td>N/A</td>
</tr>
<tr>
<td>Diff. Float</td>
<td>None</td>
<td>0.64</td>
<td>1.86</td>
<td>4.66</td>
<td>N/A</td>
</tr>
<tr>
<td>Diff. Fixed</td>
<td>None</td>
<td>0.23</td>
<td>1.55</td>
<td>4.66</td>
<td>99.94</td>
</tr>
<tr>
<td>Diff. Float</td>
<td>Greedy</td>
<td>0.30</td>
<td>0.42</td>
<td>2.35</td>
<td>N/A</td>
</tr>
<tr>
<td>Diff. Fixed</td>
<td>Greedy</td>
<td>0.19</td>
<td>0.30</td>
<td>2.35</td>
<td>99.98</td>
</tr>
<tr>
<td>Diff. Float</td>
<td>Optimal</td>
<td>0.20</td>
<td>0.28</td>
<td>1.56</td>
<td>N/A</td>
</tr>
<tr>
<td>Diff. Fixed</td>
<td>Optimal</td>
<td>0.10</td>
<td>0.14</td>
<td>1.56</td>
<td>99.99</td>
</tr>
</tbody>
</table>

Table 9.1: Summary of positioning performance with and without UGV cooperative navigation, for both control strategies and both filter designs.

the average and median 3D errors are very similar, where there is a large discrepancy between the average and the median performance without a UGV. Further, the benefit of the cooperative UGV is clear in the average PDOP, which is significantly reduced by cooperative UGV. In particular, the average PDOP reduction over the 25 flights was 50% with the *Locally Greedy* strategy and a PDOP reduction of nearly 2/3 with respect to the GNSS-only value whenever the UGV is steered to the *Regionally Optimal* location. Finally, while a marginal impact, it also is promising that whenever the cooperative UGV is present, the percentage of epochs that are successfully integer ambiguity fixed is greater with a UGV when compared to not using a UGV. It should be noted that this metric would be more favorable for the cooperative UGV if a more stringent ambiguity ratio test were used.

In addition to presenting the statistical summary of the 25 flights, Figure 9.7 shows the 3D positioning performance for each of the 25 flights for both filters and all three scenarios, no UGV, *Locally Greedy*, and *Regionally Optimal*, as well as all three filter types (i.e., undifferenced, differential float, and differential fixed). As shown in Figure 9.7, having a UGV consistently helps in reducing the 3D positioning error of the UAV for the majority of flight trials. In nearly all cases, the addition of the cooperative UGV makes a significant impact, while there are a few cases that its
impact is negligible. In the flights that the UGV benefit was negligible, the GNSS-only geometry was likely favorable, as is evident in that the fact that positioning error is already low without the cooperative UGV in these instances. Likewise, as suggested in the motivational Monte Carlo study of the potential PDOP reduction from a single ranging source that is shown in Figure 5.1, the worse the GNSS-only performance, the more beneficial the updates from the cooperative UGV. This is evident in particular, in flight trials #19 and #24.

It is worth also noting that there are practical advantages and disadvantages of each approach on the implementation–level. For example, undifferenced GNSS EKF requires less communication and the computational performance less than the differential GNSS EKF (i.e., when including the LAMBDA method, etc.). However, the differential GNSS EKF has been shown to be more precise in all cases (i.e., Locally Greedy UGV, Regionally Optimal UGV, and No UGV). Likewise, with respect to the two cooperative planning strategies of the UGV, the Regionally Optimal performed better in both filters, but the computational time to compute a grid around the UGV

Figure 9.7: 25 flights of the 3 strategies and the undifferenced and the double–differenced Float and Fixed 3D Error.
is much greater than compared to the *Locally Greedy* strategy. As such, depending on the required accuracy and computational constraints of the application, one configuration may be a better fit than others.

### 9.1.1 Nonlinear Optimization

This section discusses the results from the nonlinear optimization algorithm that was implemented as one of the UGV control strategies. Unlike the results in the previous section, these results only look at the effect of PDOP when different methods are implemented. As displayed in the previous results, there is a direct correlation with PDOP and the average 3D error, the higher the PDOP the larger the average 3D positioning error and the lower the PDOP the smaller the average 3D positioning error, as shown in Table 9.1. The following results in the next sections take advantage of this fact and focuses on the changes in PDOP. The parameters used for the nonlinear optimization are discussed in Table 7.2. Before delving too far into this new algorithm, it needs to determine how this method stands up against the *Regionally Optimal* approach. Figure 9.8 shows the percent reduction of PDOP from using the nonlinear optimization and the *Regionally Optimal* approach. This is for 50 trials and each trial is run for 10 time-steps. The x-axis shows the sum of the PDOP for one trial but for 10 time-steps. The figure shows that using the nonlinear optimization always has some benefit when comparing it to the *Regionally Optimal* approach. In some cases, there is a large percent reduction, close to 70%, and some cases there is a very little benefit, 2%. While in some cases there is not a huge benefit, there is still a benefit to using the nonlinear optimization process. Computing the run time of the 50 trials, it was shown that the nonlinear optimization approach was **four times** faster than the *Regionally Optimal* approach, which is a brute force method.

Knowing that the nonlinear optimization is a good tool to use and it performs better than the *Regionally Optimal* and *Locally Greedy* approach, the next step is to
examine the path planning approach. The number of time–steps to ‘plan ahead’ was selected to be 10. Figure 9.9 shows the PDOP reduction of having a 10-step path planning approach versus having a one-step path planning approach. The x-axis is the sum of the PDOP for 10 time–steps at that specific trial run. For the 10-step approach, the nonlinear optimization takes in 10 different satellite geometries, 10 different UAV locations, and 10 UGV positions and while taking the constraints into account, chooses which direction to move toward for 10 steps. The PDOP is calculated as if the UGV were at these 10 different locations. For the one step path planning, the UGV’s movement is calculated at each time–step for 10 different time–steps. The

<table>
<thead>
<tr>
<th>Case</th>
<th>Mean Sum PDOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regionally Optimal</td>
<td>28.88</td>
</tr>
<tr>
<td>Nonlinear Optimization with No UAV Error</td>
<td>22.45</td>
</tr>
</tbody>
</table>
nonlinear optimization figures out which direction to move toward, moves there, then starts the whole process again. The results show that there is not a significant loss of performance when employing path planning, which is a good sign. Table 9.3 shows the mean sum PDOP comparison.

Table 9.3: Mean PDOP comparison for 50 data sets: nonlinear optimization one-step versus 10-step path planning

<table>
<thead>
<tr>
<th>Case</th>
<th>Mean sum PDOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nonlinear Optimization: 1 Step</td>
<td>22.56</td>
</tr>
<tr>
<td>Nonlinear Optimization: 10 Steps</td>
<td>22.69</td>
</tr>
</tbody>
</table>

The next figure, Figure 9.10, indicates how the UGV moves compared to the two different path planning methods. The UGV for both methods start off at the bottom left of the graph and move towards the top right. For reference the UAV is located at [0,0]. The 10-step path planning, the blue *line*, takes a sharper approach whereas
the one-step path planning, the red line, has a wider angle toward the UAV.

While this shows a promising result, there is one issue that might arise. For the path planning approach, the assumption is that the UGV knows exactly where the UAV is at all times. While for the 10-step path planning approach, this may not be the case, even though it knows the path of the UAV. There are certain external factors that may push the UAV off course such as wind, obstacles, or changing goals. The UGV knows exactly where the UAV is at the first time-step but may differ after that. To take this into account, a random growing error was associated with the UAV’s position for time-steps 2-10. This change in theory was processed to see what type of effect it may have on the PDOP values. This result is shown in Figure 9.11. This percent reduction shows that when there is an error associated with the UAV’s positioning, the PDOP value increases, approximately by as low as 2% and as high as 16%. The x-axis shows the sum of the PDOP, for the UAV with no UAV positioning.
While this is shows an increase in PDOP, is there any way that this error can be reduced? Since it is known that the UAV has a growing error associated with it, a weight-based cost function is applied to the PDOP constraint function. This procedure is shown in Chapter 6.3 in Eq. 6.28. Figure 9.12 confirms that this weighting approach is beneficial and can recover some of the increase in PDOP due to the UAV’s positioning error. The x-axis shows the sum of the PDOP, for the UAV with UAV positioning error with weighting the PDOP, for 1 trial and for 10 time–steps. Again, in some instances there is a minimal effect, around 1%, but there are cases that get as much as 5% reduction in PDOP.

In Figure 9.13, the percent reduction from the UAV with no positioning error from the UAV with positioning error and with weighting, there is less of a reduction. Before the weighting, there was a max reduction of 17%, now the max reduction is
Figure 9.12: PDOP reduction of nonlinear optimization between UAV’s position with error but weighting and UAV’s position with error but no weighting
Figure 9.13: PDOP reduction of nonlinear optimization between UAV with no positioning error and UAV’s position with error and with weighting

15%. The mean reduction was 5.5% with no weighting, and now after weighting the PDOP constraint function, the mean reduction drops to 2.1%. The x-axis shows the sum of the PDOP, for the UAV with no UAV positioning error, for one trial and for 10 time–steps. Table 9.4 shows the mean sum PDOP over the 50 data sets for the different cases of having the UAV with no error, with error and no weighting, and error with weighting. As shown in the figures, UAV with no error has the lowest PDOP, then the UAV with error and weighting, followed by UAV with error but no weighting.

Table 9.4: Mean sum PDOP comparison for 50 data sets: 10-step path planning nonlinear optimization cases

<table>
<thead>
<tr>
<th>Case</th>
<th>Mean sum PDOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>UAV with No Error</td>
<td>22.69</td>
</tr>
<tr>
<td>UAV Error and No Weighting</td>
<td>24.03</td>
</tr>
<tr>
<td>UAV Error and with Weighting</td>
<td>23.21</td>
</tr>
</tbody>
</table>
Finally, for the last figure in this subsection, it shows that there is still a large amount of reduction when using the 10-step path planning approach while having UAV positioning error and weighting compared to the Regionally Optimal approach. The maximum PDOP reduction without UAV positioning error was around 70% and now the maximum sum PDOP reduction is around 45%. The mean sum PDOP reduction before the UAV error was added was 27% and now the mean sum PDOP reduction is 19%.

9.1.2 Incorporating Other Satellite Systems

As stated in the literature review, Chapter 2, there are other satellite systems that were launched and are being launched by other countries. With technology advancing at a rapid rate, there are GNSS antennas and receivers that are able to track these
Figure 9.15: Number of satellites, GPS + Galileo vs. GPS only, visible over one flight

different satellite system, GLONASS, Galileo, and BeiDou. One may ask, instead of employing another vehicle, such as an unmanned ground vehicle, why not include another satellite system instead? It is known that having more satellites will improve satellite geometry, and reduce the PDOP, but will it have the same effect as having a UGV? In simulation, this was considered. Since Galileo resembles GPS L1 and L1 wavelengths, it was used in this study. Using the undifferenced GNSS EKF with the UAV having Galileo and GPS with no UGV is compared with the undifferenced GNSS Extended Kalman Filter using the UAV with only GPS and having the UGV. The first figure, Figure 9.15, shows the difference in satellites visible when adding in Galileo to the simulation.

As shown in the figure, there are more than 2x the amount of satellites when you incorporate Galileo and GPS, the blue line, when comparing it to GPS only, the red line. Note, that the number of satellites for the GPS only case does not include the UGV as a satellite. When adding in the UGV as an extra ‘satellite’ to the GPS only
Figure 9.16: PDOP comparison for GPS + Galileo vs. GPS + UGV, for one flight case, the Galileo plus GPS scenario has at least twice the amount of satellites over the entire data set. Taking the same data set, next look at the PDOP history, Figure 9.16, comparing similar parameters, except this time, the UGV is included with the GPS only situation. Meaning the UGV is located where it would be most beneficial for the UAV based on the nonlinear optimization algorithm and is included in the calculation of PDOP.

In the beginning, the PDOP for the multiple vehicle case performs better than the multiple satellite configuration but that is short lived as the multiple satellite configuration performs better, in terms of PDOP. But after less than halfway through the data set, the multiple vehicle configuration has a greater performance, having a PDOP of 1.7 compared to the PDOP of 2.1 for the multiple satellite configuration. The PDOP difference increases towards the end, with the difference between the
PDOPs being approximately 0.7. As with the previous cases, a Monte Carlo was tested to observe if this benefit was consistent or a fluke. For this Monte Carlo, 50 different situations were tested, and the results are shown in the table below, Table 9.5.

Table 9.5: Results for comparison of multi-satellite systems vs. multi-vehicles

<table>
<thead>
<tr>
<th>Case</th>
<th>Mean PDOP</th>
<th>Average Satellites Visible</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPS + Galileo</td>
<td>3.06</td>
<td>8</td>
</tr>
<tr>
<td>GPS + UGV</td>
<td>2.04</td>
<td>5</td>
</tr>
</tbody>
</table>

These results show that even though you have more satellites, it is not an indication of having better PDOP than having a cooperative ranging source. This confirms the need for being able to have the UGV be able to optimize its’ location to best help the UAV. This supports the data that was found in Chapter 5.1 and Figure 5.1, that adding in a ranging source does not mean a significant reduction in PDOP, the maximum benefit is from having the UGV find the best location and move toward that location.

9.2 Multiple UGVs

With the previous results showing that adding on cooperative vehicle improves PDOP and improves the positioning accuracy of the UAV, this section observes what happens when there are two cooperative UGVs helping the UAV. In order to evaluate the benefit or determent from having a second cooperative UGV, as with the previous two previous results, the change in PDOP will be the deciding factor. For this simulation case, the UGVs start at two different random positions and will do a decentralized approach. Figure 9.17 shows the effect of employing two UGV compared to one UGV. The figure shows the sum of PDOP for 10 steps and this is repeated for 50 data sets. The maximum percent PDOP reduction achieved is 60% with a mean PDOP
Figure 9.17: % PDOP reduction of two UGVs from one UGV

reduction of 16%. Table 9.6 shows the mean sum PDOP over the 50 data sets, which shows that having two UGVs is more beneficial than only using one UGV.

Table 9.6: Mean PDOP comparison for 50 data sets: one UGV versus two UGVs

<table>
<thead>
<tr>
<th>Case</th>
<th>Mean Sum PDOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>One UGV</td>
<td>21.15</td>
</tr>
<tr>
<td>Two UGVs</td>
<td>19.40</td>
</tr>
</tbody>
</table>

As mentioned in the literature review 2, there are two main types of control when having multiple vehicles, decentralized and centralized. Figure 9.17 used the decentralized optimization movement, where each vehicle decides on its own how and where to move. The centralized approach is known for being more expensive and having one central vehicle tell the other vehicles where to move. Figure 9.18 shows the benefit of having a decentralized approach compared to the centralized. The x-axis represents the sum of PDOP for 10 time-steps, and this was repeated for 50 data sets. The y-axis represents the % PDOP reduction when using the decentralized method compared to the centralized method. While there is a small benefit using the decentralized approach when observing the reduction in PDOP, another benefit is
Figure 9.18: % PDOP reduction of two UGVs: decentralized from centralized

processing time. The decentralized approach is 15% faster than the decentralized approach. [71] cites that other advantages come from the decentralized approach not needing a measurement from the other vehicles, whereas the centralized approach needs to have constant communication to the other vehicles.

9.3 Experimental Results

The results from the experimental testing are from four data sets of having the UAV flying and the UGV moving based on the cooperative strategy. The experimental results were evaluated after using the Differential GNSS EKF on the data sets. The results compare when having a cooperative ranging source and without the aid of the UGV. The first figure, generated from RTKLIB, details the path of the UAV and the UGV, Figure 9.19, for one experimental test. The green line is the UGV’s path, starting near the base station and ending near the top right of the figure. The blue
Figure 9.19: Path of UAV (blue) and UGV (green)

line is the UAV’s path, starting on the left side of the figure and ending near the top right. The path of the UGV takes the shape of an arc, while the UAV is flying near it.

Figure 9.20 shows the PDOP of three scenarios, the blue line shows the PDOP with no elevation mask, red line shows the PDOP with an elevation mask, and the green line shows the PDOP with an elevation mask and a cooperative ranging source for the same test. The PDOP is the least when no mask is applied, then PDOP with an elevation mask and cooperative ranging source, and the PDOP is the highest when no cooperative ranging source is used with an elevation mask. There are some time–steps where the PDOP with the cooperative ranging source is lower than the PDOP when there is no elevation mask applied.

For the same data as the previous two plots, Figure 9.21 shows the East, North, Up position error of the UAV with UGV cooperative ranging, blue line and without UGV cooperative ranging, red line. The top plot is the UAV’s positioning error in the East direction, the middle plot is the UAV’s positioning error in the North direction, and the bottom plot is the UAV’s positioning error in the Up direction. As expected, the positioning error is the largest in the Up direction for both cases. The UGV
cooperative ranging is the most helpful in the Up direction, as it reduces the error by as much as 5 meters.

To summarize the result for the differential GNSS EKF, Table 9.7 shows the average 3D position error of the UAV with and without a cooperative UGV and the mean PDOP for the four experimental tests. As expected, and displayed in the simulation work, the UAV with the UGV cooperative ranging performed better than the UAV without the UGV. And that same trend follows for the mean PDOP as well.

Table 9.7: Results for all experimental data

<table>
<thead>
<tr>
<th>Case</th>
<th>Avg. 3D Error (m)</th>
<th>Mean PDOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>UAV with No UGV</td>
<td>1.14</td>
<td>3.81</td>
</tr>
<tr>
<td>UAV with UGV Cooperative Ranging</td>
<td>0.22</td>
<td>2.50</td>
</tr>
</tbody>
</table>
Figure 9.21: East, North, Up position error (m) for the UAV with UGV cooperative ranging and without UGV cooperative ranging
Chapter 10

Conclusions and Future Work

10.1 Conclusion

Navigating a UAV in an urban canyon will often lead to large positioning errors due to fewer satellites in view. In order to reduce this error, cooperative navigation of multiple vehicles are often used. In this dissertation, a cooperative ranging UGV was employed to improve in the navigation accuracy of the UAV. Several cooperative navigation strategies and estimation filters were formulated and one cooperative navigation strategy was experimentally validated.

A cooperative navigation strategy has been developed and evaluated in simulation and has shown to dramatically improve the positioning performance of a UAV in an urban canyon. This dissertation has described two different filter designs and three different cooperative planning techniques. The results show that having a single cooperative ranging source, a UGV, that is strategically located can tremendously help the positioning of the UAV. By employing an elevation and azimuth mask to simulate an urban canyon, it has been shown that the more challenged the GNSS environment, the more benefit the use of a cooperative ranging UGV has. It was determined through simulated flight data that the positioning error of the UAV is
reduced more when employing the differential GNSS EKF compared to the undiffer-
enced GNSS EKF and the constrained nonlinear optimization performs as the best
cooperative navigation strategy, followed by Regionally Optimal, and lastly Locally
Greedy.

Another aspect of this study was to employ a path planning approach for the UGV
to cooperatively help the UAV that leverages the path of the UAV and satellites’
ephemeris data. In this study, it was shown that using the path planning approach
showed little difference in PDOP reduction compared to the one-step ahead method.
Also considered in this study was the benefit of using multiple cooperative ranging
UGVs to help the UAV’s positioning. This analysis showed that there was an im-
provement to using multiple cooperative ranging UGVs compared to only having one
cooperative ranging UGV. This study employed another satellite system, Galileo, to
compare the benefit of having more high elevation satellites to having a cooperative
UGV. Having more satellites did not help the positioning error of the UAV more than
using the cooperative ranging UGV.

To test the feasibility of a cooperative navigation algorithm, an experimental test
was conducted. The cooperative navigation algorithm, Locally Greedy, was tested in
real-time and a differential GNSS EKF was used to estimate the error of the UAV
in post-processing. This experimental test showed that having a cooperative ranging
UGV helps reduce the positioning error of the UAV.

10.2 Future Work

There are many opportunities to further the research presented in this dissertation.
For the simulation, the elevation cutoff is constant for the whole flight and multipath
is based on the elevation angle of the satellite. Future work would include having a
model of the city block to be able to have an accurate elevation mask and an accurate
multipath estimation. Currently the nonlinear optimization cooperative navigation strategy only evaluates the reduction of PDOP, and while this is an accurate evaluation, future work would include using the two estimation filters to evaluate the positioning error of the UAV. This would be further extended to the path planning approach and when using multiple cooperative ranging UGVs.

On the experimental platform, a UWB and an IMU on the UAV and a UWB on the UGV would prove to be beneficial. In this study, only the Locally Greedy approach was tested on the UGV. Future work would test out the other two cooperative algorithms of the UGV to show which was the most beneficial in experimental data.
Bibliography


