Collaborative Architectures for Relative Position Estimation of Ground Vehicles with UWB Ranging and Vehicle Dynamic Models

by

Benjamin Kai Jones

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
December 10, 2022

Keywords: Relative Position Estimation, Ultra-wideband Radios, Ground Vehicle Dynamics

Approved by

Scott Martin, Chair, Assistant Professor of Mechanical Engineering
David Bevly, Bill and Lana McNair Distinguished Professor of Mechanical Engineering
George Flowers, Professor, Dean of Graduate School

Abstract

The relative position to neighboring vehicles is critical to ongoing autonomy efforts including collision avoidance and path planning; therefore, it should not be fully dependent on an external reference such as GPS. This thesis presents methods for real-time relative positioning of ground vehicles by employing a network of on-board ultra-wideband (UWB) radios. The difficulties in range-based relative positioning, and the results from prior literature are described. Next, the proposed methods are derived which employ the kinematic bicycle model to constrain the estimated states to align with ground vehicle dynamics.

The initial methods do not require vehicle-to-vehicle (V2V) communication. However, cooperative methods are also explored which make use of the simultaneous ranging and communication capabilities of UWBs. Feedback of the tracked vehicle's dynamic states (velocity, yaw-rate, and steer angle) are analyzed for their impact on estimation quality. A geometrically-inspired consensus extended Kalman filter (CEKF) is also developed as a modification to both the prior work and the proposed vehicle-dynamic EKF (VehDynEKF).

The methods developed in this thesis improve upon prior literature results in accuracy and robustness in the presence of UWB measurement errors, unfavorable relative geometry, and dynamic maneuvers. While the CEKF shows improvement over the prior literature methods without additional sensors, it under performs the VehDynEKF proposed here. With only the use of UWB ranging and odometry, the VehDynEKF in this thesis can provide robust relative pose estimates to a neighboring vehicle. The estimate is affected by relative dynamics but maintains a mean error less than 2.5 meters in both simulation and experimental results without cooperative feedback. The lateral velocity of the vehicles is found to be a primary contributor to error; odometry including measurements of the estimating vehicle's lateral velocity significantly improves the results. Lastly, if the ego vehicle has access to the tracked

vehicle's longitudinal velocity, the mean error is refined to be less than 1 meter—sufficient for the majority of safety-critical applications.

Acknowledgments

First and foremost, I want to thank my best friend and fiancèe, Olivia, for her unwavering support every step of the way. You have been my main confidant throughout my graduate school career all while managing your own academic pursuits. This thesis would not have been possible without you and the love you provided me. I dedicate this thesis to you and the life we are embarking on together.

I want to thank my mother for the immense role she played in developing my curiosity and educational foundations. From the projects we did together to the MythBusters marathons, you provided an enthusiasm for education which was paramount in my academic pursuits. My father also deserves credit for the work ethic he portrayed to me at a young age. You taught me the importance of confidence and that humor always has a place (even if you're the main one who thinks it's funny). My parents never failed to check in on me to offer counsel and support during the graduate school process which can be admittedly grueling. I realize just how lucky that makes me. I would also like to extend my gratitude to the rest of my family who never failed to provide encouragement towards my engineering interests (even when they manifested themselves in strange or "nerdy" hobbies).

I owe a special thank you to Dr. David Bevly and Dr. Scott Martin for the opportunity to study within GAVLAB. The dedication to your students and the immense technical knowledge you share through your classes and research guidance is invaluable. The opportunities that GAVLAB provides for skill development and meaningful research are second to none, and I am honored to be associated with it. Similarly, I would like to thank Dr. George Flowers for his mentorship and excellent class. If I could choose a graduate school all over again, I'd be right back here.

I am also grateful to the many members of GAVLAB for their collaboration and for making the time in graduate school exciting. Tyler and Kyle made the many many MANY "track days" more palatable, and they both provided direct assistance in the development of the work presented here. John David was always willing to answer a hardware question (and there were many); "tech priest" is a very fitting title. Dr. Dan Pierce's prior work and guidance were imperative to the MECC paper which laid the foundation for this thesis. Dr. Howard Chen deserves recognition for his unceasing willingness to help, especially for the initial transition as a new graduate student (I am sorry this thesis does not contain any quaternions). And to the 2020 "DeadReckoner" colleagues, I am grateful for your friendships, as well as the daily crosswords and enticing conversations which always provided a much needed break. I wish I had spent more time getting out of the lab with you all for trivia or a round of golf.

Lastly, in case they ever manage to stumble across this one day, I'd like to thank the MSU Castle Boys for being great friends and a constant source of inspiration. Hail State.

Table of Contents

Abstract ii

Acknowledgments iv

List of Figures viii

List of Tables xiv

1 Introduction 1

1.1 Background and Motivation 1

1.2 Prior Work 6

1.3 Research Contributions 8

1.4 Thesis Outline 9

2 Ground-Vehicle Localization Technical Background 10

2.1 Coordinate Frames 10

2.1.1 Local Tangent-Plane Frame 11

2.1.2 Body Frame 12

2.1.3 Coordinate Transformation Matrices 13

2.2 Vehicle Odometry 18

2.2.1 IMU Mechanization 19

2.2.2 Ground Vehicle Dynamic Models 22

2.3 Ultra-wideband Radios (UWBs) 27

2.3.1 Measurement Methodology 28

2.3.2 Hardware Description 29

2.3.3 Experimental Analysis 31

3 Non-Cooperative Relative Positioning Algorithms 45

3.1 Geometric Relative Position Solution 45

3.2 Least Squares Relative Positioning . . . 48
3.3 Observability Difficulties (DOP) . . . 55
3.4 Extended Kalman Filter (EKF) Relative Positioning . . . 58
3.4.1 Constant Velocity Extended Kalman Filter . . . 60
3.4.2 Vehicle Dynamics Extended Kalman Filter (VehDynEKF) . . . 63
4 Cooperative Relative Positioning Algorithms . . . 71
4.1 Relative IMU Mechanization . . . 71
4.2 Vehicle Dynamic Model State Feedback . . . 78
4.3 Consensus Extended Kalman Filter (CEKF) . . . 80
5 Simulation Results . . . 86
5.1 Simulation Structure . . . 86
5.2 Non-Cooperative Results . . . 88
5.2.1 Scenario 1: S-Turn with Static Base Vehicle . . . 88
5.2.2 Scenario 2: Dynamic U-Turn . . . 93
5.2.3 Scenario 3: Figure-8 Maneuver . . . 99
5.3 Cooperative Results . . . 102
5.3.1 Scenario 1: Dynamic U-Turn . . . 103
5.3.2 Scenario 2: Figure-8 Maneuver . . . 114
5.4 Conclusion of Simulation Results . . . 120
6 Experimental Results . . . 122
6.1 Experimental Setup . . . 122
6.2 Scenario 1: Dynamic U-Turn . . . 124
6.3 Scenario 2: Random Driving . . . 129
6.4 Conclusion of Experimental Results . . . 135
7 Conclusions . . . 136
7.1 Recommendations of Future Work . . . 137
Bibliography . . . 139

List of Figures

1.1 Relative Positioning with GPS Obstacles 2

1.2 UWB Layout with 4 Ranges 7

2.2 NED and Vehicle Body Frames 12

2.4 Three Successive Elementary Rotations 14

2.5 Vehicle and Local NED Frames 18

2.6 IMU Mechanization Process 23

2.7 Kinematic Bicycle Model 24

2.8 RTK and Kinematic Bicycle Model Experimental Comparison 26

2.9 UWB Layout and Corresponding Range Measurements 27

2.12 Indoor Data Collection for Calibration 33

2.13 P440 Measurement Errors Pre-Calibration 35

2.14 P440 Measurement Errors Post-Calibration 36

2.15 P440 Individual Range Errors Post-Calibration 37

2.16 P440 Ranging Error Histogram 38

2.17 Experimental Vehicle and Sensor Setup 39

2.19 Satellite Trajectories for Double-Lane Change UWB Measurement Collection . . 40

2.20 P440 Double Lane-Change UWB Ranging Errors 41

2.21 P440 Data Collection with Figure-8 Trajectories 42

2.22 P440 Dynamic Figure-8 Ranging Errors with $3\sigma = 30$cm Bounds 43

2.23 UWB Ranges 101 $\rightarrow$ 103 Dynamic Figure-8 with Outliers 44

3.1 2D Components of Relative Pose 46

3.2 Geometric Solution Method 46

3.3 1000 Geometric Solutions at 0° Relative Bearing 47

3.4 1000 Geometric Solutions Showing Quadrant Ambiguity Faults 48

3.5 UWB Measurement Model Depiction 49

3.6 UWB Measurement Model Assuming UWB Baseline Intercepts COG 50

3.7 Nonlinear Least Squares Algorithm 53

3.8 Importance of Proper Initial Guess to Least Squares Quadrant Ambiguity . . . 54

3.9 Dilution of Precision with Varying Relative Poses 57

3.10 Least Squares Solution Quality for High vs Low DOP Geometries 58

3.11 Extended Kalman Filter Algorithm 61

3.12 Vehicle Dynamic EKF (VehDynEKF) Block Diagram 68

3.13 Simulated Static-Follower Encirclement Trajectory 69

3.14 Comparison of Non-Cooperative Results for Static Follower Encirclement Test . 70

4.1 Position in Multiple Reference frames (O as static and O' as mobile) 72

4.2 Relative Mechanization Errors with Tactical-Grade IMU Biases 77

4.3 Relative Mechanization EKF and Non-Cooperative Vehicle Dynamic EKF . . . 78

4.4 Two Geometrically-Related RPV Estimators 81

4.5 2-Vehicle Consensus Block Diagram with Veh. Dyn. CEKFs 84

5.1 Trajectory Generation Example 87

5.2 Simulated S-Turn Maneuver with Static Base 89

5.3 Single Run S-Turn RPV Estimation Results 89

5.4 S-Turn RPV Component Errors 90

5.5 Dynamic State Estimates of Each EKF 91

5.6 S-Turn Constant Velocity EKF 100 Monte Carlo Run RPV Errors 91

5.7 S-Turn Vehicle-Dynamics EKF 100 Monte Carlo Run RPV Errors 92

5.8 S-Turn Mean and Standard Deviation for Monte Carlo Errors 93

5.9 Dynamic U-Turn Trajectory Depiction 94

5.10 Single Run of Non-Cooperative Estimation Comparison 95

5.11 Dynamic U-Turn Estimate Snapshots 96

5.12 Dynamic U-Turn Constant Velocity EKF 100 Monte Carlo Run RPV Errors . . 97

5.13 Dynamic U-Turn Vehicle-Dynamics EKF 100 Monte Carlo Run RPV Errors . . 97

5.14 Dynamic U-Turn Mean and Standard Deviation for Monte Carlo Error Norms (Varied Y-Axis Scales) 98

5.15 Dynamic U-Turn Trajectory Depiction 100

5.16 Figure-8 Constant Velocity EKF 100 Monte Carlo Run RPV Errors 101

5.17 Figure-8 Vehicle-Dynamics EKF 100 Monte Carlo Run RPV Errors 101

5.18 Figure-8 Mean and Standard Deviation for Monte Carlo Error Norms (Varied Y-Axis Scales) 102

5.19 Cooperative U-Turn RPV Errors Through Run 104

5.20 U-Turn Cooperative Method Errors Sorted By Cumulative Lateral Velocity . . . 105

5.21 U-Turn Cooperative Method Errors Throughout Run with Ego Lateral Vel. Measurements 105

5.22 Error Norm Comparison of Sideslip-Estimated Lateral Velocity Feedback 107

5.23 U-Turn Consensus EKFs for Each Propagation Model (varying y-axis scales) . . 109

5.24 U-Turn Consensus EKFs for Each Propagation Model (Varying Y-Axis Scales) . 110

5.25 U-Turn Constant Velocity Consensus EKFs with Relative Heading Measurements 111

5.26 U-Turn Vehicle Dynamic Consensus EKFs with Relative Heading Measurements 111

5.27 Dynamic U-Turn Constant Velocity Method Comparisons 112

5.28 Figure-8 Vehicle-Dynamics EKF RPV Errors Through Run 114

5.29 Figure-8 VehDynEKF Results with v_L Feedback without Ego-Vehicle Lateral-Velocity Measurement 115

5.30 Figure-8 Cooperative Method Errors Sorted by Cumulative Lateral Velocity . . 115

5.31 Figure-8 Cooperative Method Errors Throughout Run with Ego Lateral Vel. Measurements 116

5.32 Figure-8 Cooperative Method Errors Sorted by Cumulative Lateral Velocity (with Ego Lateral Vel. Measurement) 117

5.33 Figure-8 Consensus EKFs for Each Propagation Model 118

5.34 Figure-8 Constant Velocity Consensus EKFs with Relative Heading Measurements 119

5.35 Figure-8 Vehicle-Dynamic Consensus EKFs with Relative Heading Measurements 119

5.36 Figure-8 Vehicle-Dynamic Cooperative Method Comparison 120

6.1 Experimental Passenger Vehicles: Lincoln MKZ and Kia Optima 123

6.2 KVH DSP 1760 123

6.3 Dynamic U-Turn Experimental Scenario Trajectories 125

6.4 Dynamic U-Turn Non-Cooperative Estimation Results 125

6.5 Dynamic U-Turn Non-Cooperative Error Norms 126

6.6 Dynamic U-Turn Feedback Results 126

6.7 Dynamic U-Turn Feedback Results VS Cumulative Lateral Velocity 127

6.8 Dynamic U-Turn Constant Velocity Consensus Comparison 128

6.9 Dynamic U-Turn Consensus Cumulative Error Norms 129

6.10 Random Maneuvers Experimental Scenario Trajectories 130

6.11 Random Maneuver Non-Cooperative Estimation Results 131

6.12 Random Maneuver Non-Cooperative Error Norms 131

6.13 Random Maneuver Feedback Results Through Run 132

6.14 Random Maneuver Feedback Results VS Cumulative Lateral Velocity 132

6.15 Random Maneuver Constant Velocity Consensus Cumulative Error Norms . . . 133

6.16 Random Maneuver Vehicle Dynamic EKF Consensus Results 134

6.17 Random Maneuver Vehicle Dynamic EKF Consensus Cumulative Error Norms . 134

List of Tables

2.1 PulsON P440 Operating Results from PII Selection 31
2.2 P440 Measurement Error Statistics Pre-Calibration 36
2.3 P440 Measurement Error Statistics Post-Calibration 37
2.4 S-Turn Following Ranging Error Statistics 41
5.1 Scenario 1: S-Turn with Static Base 100-Run Error Statistics 93
5.2 Scenario 2: Dynamic U-Turn 100-Run Error Statistics 98
5.3 Scenario 3: Figure-8 100-Run Error Statistics 102
5.4 Scenario 1: Dynamic U-Turn Cooperative Result Comparison 108
5.5 Scenario 2: Figure-8 Cooperative Result Comparison 117
6.1 Experimental Dynamic U-Turn Cooperative Result Comparison 127
6.2 Experimental Random Maneuver Cooperative Result Comparison 133

Chapter 1

Introduction

As vehicle research progresses further into driver-assistive and autonomous technology, tracking the relative pose of nearby vehicles is of increasing importance. The most critical requirement for autonomous vehicle platforms will always be safety. To ensure collision detection and avoidance, determining the proper relative pose to local operators is imperative. For this and other safety-critical operations, the minimum acceptable accuracy is often considered to be a two-dimensional relative position vector (RPV) error of 1 meter [1]. Beyond safety assurance, relative positioning is essential for numerous applications of interest for autonomous vehicles including search-and-rescue with multiple platforms, smart warehouses, and path duplication (platooning). The success of these applications and many more lies within accurate and robust localization to nearby participants. This thesis explores the problem of autonomous ground-vehicle relative positioning through the use of on-board sensors and local collaboration.

1.1 Background and Motivation

The Global Positioning System (GPS) introduced in 1993 became the first of many Global Navigation Satellite Systems (GNSS) to provide globally-available localization information within a few meters. Since then, advanced algorithms have been developed to refine the accuracy to the order of centimeters through differential GPS methods (DGPS) between receivers. If one of the receivers is at a known location, a highly accurate global solution can be provided, but it may also be employed with mobile receivers to provide precise relative positions. This is the most common method of ground-vehicle relative positioning and can

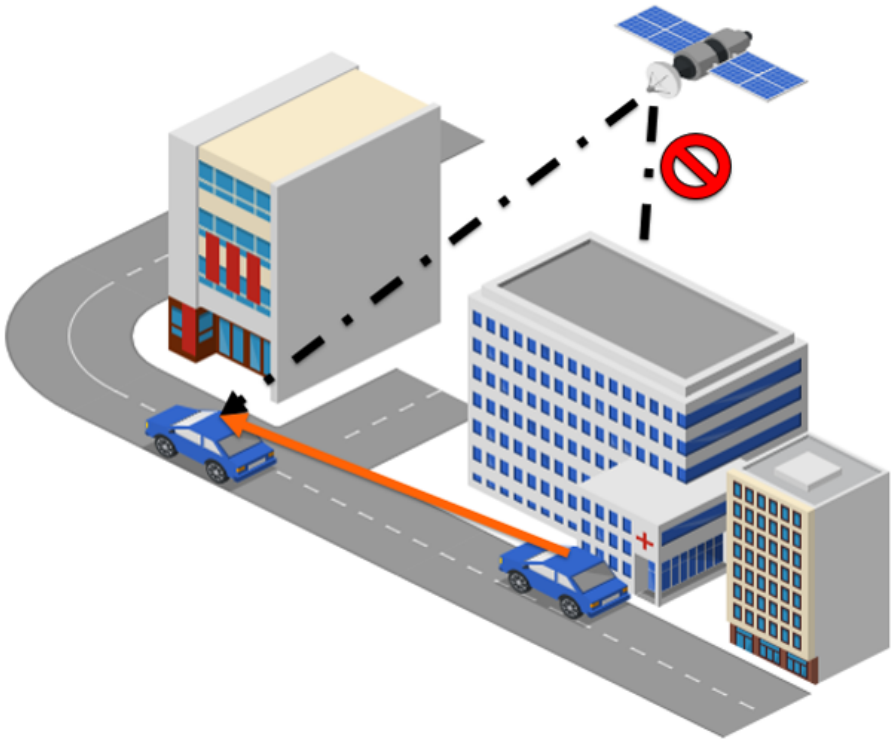


Figure 1.1: Relative Positioning with GPS Obstacles

achieve the 1 meter accuracy requirement for safety. However, as impressive and ubiquitous GNSS as solutions are, they are not without their shortcomings. The satellite signals originating from space are frequently blocked from passive sources including foliage, buildings, and tunnels or from intentional manipulation such as jamming or spoofing. A blockage due to structural interference is depicted in Figure 1.1. Due to these issues, GPS cannot be considered the sole solution of relative ground-vehicle positioning, and other methods independent of an external infrastructure should be explored.

The main goal of this thesis is to present onboard range-based estimation methods for tracking the relative position of ground vehicles without GPS. This problem practically consists of tracing a relative position vector between two vehicles through time. The two components of this problem are detection and tracking. Detection may be considered localization at an instant in time by processing current measurements, while tracking aims to predict the future path of the vehicle through a motion model [1]. This tracking step is

crucial; it helps to assist future detection with an estimate of the next likely location, to estimate additional states such as relative velocity, and to provide a path estimate through measurement outages or errors. The implementations presented in this thesis approach these tasks through cooperative solutions with relative ranging from on-board sensors and vehicle odometry. The platform performing the estimation is denoted as the "base" or "ego" vehicle, while the vehicle whose relative pose is being estimated will be referred to as the "tracked" vehicle.

In order to generate a relative position of the tracked vehicle in the ego vehicle's frame, this thesis makes use of multiple on-board range measurements. Many different sensors can provide this ranging estimate through different means. Camera-based techniques can estimate relative range through pixel-mapping image data to the physical world once the target vehicle has been identified [2, 3]; however, the image processing techniques required for proper identification are often complicated and computationally expensive. Additionally, vision-based solutions rely upon consistent line of sight within their limited field-of-view—requiring multiple camera systems to fully capture a vehicle's surroundings. Radio detection and ranging (RADAR) is a well-matured ranging solution based upon reflected elctromagnetic waves. It has already been implemented in middle-class commercial vehicles for collision warning and automatic cruise control (ACC) applications [1]. Although the technology has been around for over a century, research is ongoing including recent developments to generate ground-penetrating maps for localization in surveyed environments [4]. Light Detection and Ranging (LIDAR) operates similarly but utilizes reflections of light pulses to estimate relative range of objects in the vicinity. Their high accuracy and map-generating capabilities have made them a critical component of many budding autonomous vehicle efforts [5], but they are susceptible to environmental conditions and their cost is still relatively high compared to other sensing options. For an in-depth comparison on varying ranging sensor options as they relate to the vehicle tracking problem, see [1].

Ultra-wideband radios have gained interest in recent years for varying localization applications due to their robust and accurate ranging in many different scenarios where other sensors suffer. UWBs are more resistant to multi-path than GPS, provide peer-to-peer ranging without further processing, can be dispersed across platforms to provide multiple geometrically diverse measurements, and can even provide limited data along with their ranging measurements. Their relatively low cost and small form-factor have led to their inclusion in many localization algorithms. UWBs have been used for aiding DGPS solutions by providing a highly accurate base-length estimate [6–8]. They have also been used for emergency-responder navigation [9], as landmarks for ground-vehicle localization [10], and as an aiding source for range-only simultaneous localization and mapping (SLAM) [11].

This thesis also attempts to improve the UWB relative positioning results through cooperative methods. Collaboration for improved localization is of increasing interest as systems become increasingly connected. This cooperation between nodes may take many different forms including GPS location sharing, dynamic state transmission, or communication to nearby smart infrastructure. Although relative positioning has been achieved with on-board sensing alone, the inclusion of inter-vehicle communication presents ample opportunity for increased accuracy and robustness. The recent introduction of standardized vehicle-to-vehicle (V2V) communication structures in America, Europe, and Japan show the institutional interest in the collaborative possibilities of ground vehicles [1].

The simplest method for implementing cooperation is for the tracked-vehicle(s) to provide its dynamic states directly to the estimator. This can assist in propagating the previous state and prevent inaccurate solutions which may exist from the ranging alone. In Chapter 2, the dynamic parameters of a maneuvering ground vehicle are simplified to a few key components which can be broadcast to provide additional observability to the estimation process.

When local vehicles all have an interest in relative positioning to one another, the opportunity exists to leverage the local estimates into a more accurate solution for each

— this is the basis for mutually-beneficial cooperation. Collaborative localization methods can be divided into two main categories: centralized and decentralized. In a network of vehicles with on-board sensing along with relative measurements to one another, a centralized approach incorporates all of the measurement information to create an estimate of each vehicle's state vector and then broadcast this information back to each node in the network. These centralized algorithms represent the best-case scenario, but suffer from non-scalability due to limitations in communication bandwidth, processing bottlenecks, and proper time-syncing of transmitted measurements. It also introduces a single point of failure for the solution of a full network of vehicles.

Decentralized solutions attempt to overcome these problems by dividing the computational requirements and failure modes over a network of independent estimation algorithms. However, these decentralized methods introduce difficulties when the relative measurements are functions of multiple states. For instance, one of the most commonly seen applications of decentralized algorithms is for navigation solutions of independent platforms with relative measurements between one another. This application presents an issue where the cross-correlation terms in the covariance matrix are no longer being directly tracked. These terms are crucial for depicting the relation of errors between the states of multiple platforms. Methods for recovering these cross-correlations have been proposed — the most common being covariance intersection (CI) and inverse covariance intersection (ICI). For more information on these estimators see [12] and [13], respectively. These decentralized methods do not directly apply to the problem at hand, but they are the most common applications in literature.

The relative tracking problem in this thesis differs from these decentralized global-solution algorithms in a few critical ways. There are no global solutions tracked for each vehicle, and so the relative measurements are not functions of states from multiple estimators. Rather, the range measurements relate directly to the RPV solution of a given estimator. It will be shown in a later chapter that multiple vehicles estimating pose to one another can

be re-formulated to be considered as estimating a shared state. In problems with multiple estimations of shared states, a decentralized algorithm known as the Consensus Kalman Filter is often used for driving the solutions to a shared "consensus." This concept will be explored in proceeding chapters.

1.2 Prior Work

This thesis focuses on the generation of relative positions between locally-maneuvering ground vehicles with on-board sensors and collaboration with local neighbors. The majority of localization with UWBs has been restricted to employing measurements to local *landmarks* or *anchors* with known locations [14–16]. Therefore, their primary use is often for indoor or well-mapped local navigation where local anchors are feasible. For an in-depth overview of indoor positioning with UWBs, Alarifi et al. provides a detailed analysis [17].

Relative positioning without known landmarks makes the problem more difficult, and often UWBs are implemented to augment other existing solutions. An interesting new development from Stanford involves a tight-coupling of three UWBs and carrier phase DGPS solutions [18]. That work, published in May 2022, employed the three UWBs to create a narrow search space for resolving the carrier phase integer ambiguities with promising results. This shows an increasing use-case for UWBs as companion solutions to GPS. However, the work focused on simplistic UWB RPV estimation methods and only explored ideal relative geometries (one vehicle following closely behind the other).

In [19], UWB ranges and inertial measurements were used to estimate relative paths of UAVs. However, the path estimates were also dependent on magnetometers providing a global heading reference and sufficient relative dynamics as outlined in previous work by the same group [20]. One of the few sources in literature focusing on only on-board UWB sensors was presented recently by Ghanem et al [21, 22]. That work presents the first known application of a multi-UWB network to the ground vehicle relative positioning problem. A

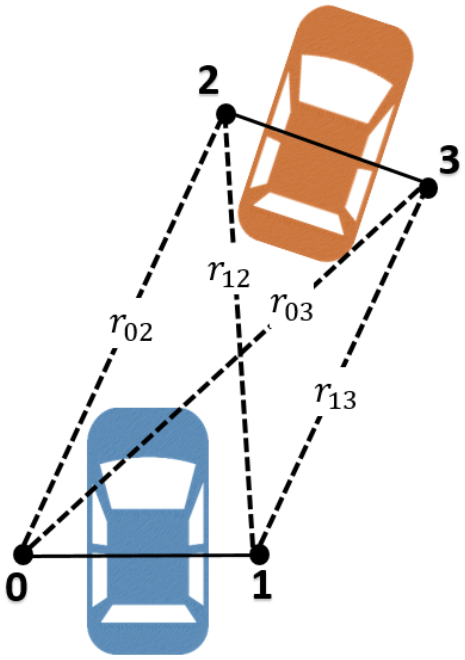


Figure 1.2: UWB Layout with 4 Ranges

configuration of four UWBs horizontally across the roof of the vehicles was employed and the same layout will be used here. This is seen in Figure 1.2.

Ghanem's work explores the difficulties in this problem through their initial results and observability analysis while employing a constant-relative-velocity Extended Kalman Filter. This method has shortcomings due to the limitations of the constant relative velocity assumptions resulting in instabilities and significant errors. Ghanem's solutions in [22] form a baseline comparison for all of the methods presented in this thesis. The results and error modes from Ghanem's method will be highlighted in more detail during the comparisons to methods proposed in this thesis.

While [22] provides the first UWB ground vehicle RPV estimation in literature, extensions were made with the application of vehicle dynamic models in [23]. However, these methods do not explore the possibility of cooperative aiding. This thesis leverages vehicle-to-vehicle information exchange through the UWBs communication capabilities in order to

examine potential cooperative benefits. Simultaneous ranging and communication is a useful aspect of certain UWB hardware and is further described in [24]. With this capability in mind, two methods of collaboration are explored: dynamic state feedback from tracked vehicles and consensus algorithms. The consensus condition applied to the decentralized relative position algorithms presented in this thesis originates from Oltafi-Saber who originally derived the decentralized Kalman filtering method specifically for distributed sensor networks [25, 26]. Since then, it has been applied to numerous applications including spacecraft relative motion [27], UAV swarms [28], and traffic estimation [29]. The geometrically-inspired consensus condition is varied from more typical consensus applications and is inspired from a recent implementation by Gong et. al [30]. In this prior work, the authors apply vector algebra to constrain the 2D relative positioning of aerial vehicles; however, the vehicles were also equipped with many other sensors (IMUs, pitot tubes, and magnetometers) and only a single range measurement was utilized on each vehicle. Nevertheless, this prior work highlights the successful application of the consensus condition in a less-conventional manner. A similar methodology will be employed here to the ground-vehicle RPV estimation.

1.3 Research Contributions

The calculation of relative poses between local ground-vehicles has been performed in literature with a variety of sensor suites. Compared to other on-board solutions, UWBs present the opportunity for reduced computational complexity, omni-directional performance, invariance to local weather conditions, and geometric measurement diversity. Prior applications of UWBs to ground-vehicle RPV estimation have shortcomings which this thesis improves upon. The primary contributions of this thesis are as follows:

- Derivation and application of a vehicle dynamics model towards ground-vehicle relative positioning with UWBs to provide a substantial improvement over existing methods in literature (previously presented at MECC [23])

- Design and comparison of extensions to [23] with numerous collaborative methods. Consideration was taken towards the communication bandwidth limitations of the UWB hardware
- A novel application of a geometrically-inspired Consensus Extended Kalman Filter to ground-vehicle relative pose estimation
- Comparison of proposed methods to prior literature in both simulation and experimentation

1.4 Thesis Outline

Chapter 2 of this thesis provides the technical foundations required to develop the ground vehicle relative-positioning algorithms. It highlights the major components: coordinate frames, vehicle odometry, kinematic models, and an analysis of the UWB-ranging hardware. In Chapter 3, the difficulties of the relative-positioning problems are presented and the non-cooperative algorithms are derived. Chapter 4 describes cooperative extensions to the base algorithms with consideration of data transmission and sensor-availability constraints. The simulation structure and associated results for all algorithms are presented in Chapter 5. Next, in Chapter 6, the simulation results are verified experimentally through a variety of trajectories. Lastly, Chapter 7 presents final conclusions and recommendations for future work.

Chapter 2

Ground-Vehicle Localization Technical Background

For the algorithms presented in this thesis, a foundation of coordinate frames, sensor models, vehicle odometry methods, and sensor analysis is required. These fundamental concepts are outlined in this chapter.

2.1 Coordinate Frames

Coordinate frames are a fundamental component of any localization solution. Even outside of technical applications, the statement "your position is [20, 15, -10]" is meaningless without some reference frame for context. A Cartesian coordinate frame is defined by three orthogonal axes intersecting at a defined origin. However, the potential options for valid frames are limitless and often up to the user's preferences. It is critical that the resolving frames are well-understood and outlined.

In navigation and tracking applications, numerous resolving frames are typically present since most solutions rely on the fusion of sensors which provide information in varying coordinate systems. Consider the Global Positioning System (GPS). GPS-derived solutions are calculated in an earth-centered, earth-fixed frame originating at the center of the earth, but these values are difficult to comprehend and are often transformed into a geodetic representation (latitude (Φ), longitude (λ), and height). Figure 2.1 depicts these frames along with a local navigation frame which will be described later.

Similarly, ground vehicles often employ mounted sensors such as cameras, inertial measurement units (IMUs), and LiDARs, which generate measurements in their own sensor-body frame. The vehicles are interested in solutions in a local navigation frame such as North-East-Down (NED). In order to fuse varying sensors together into a cohesive and useful solution, a

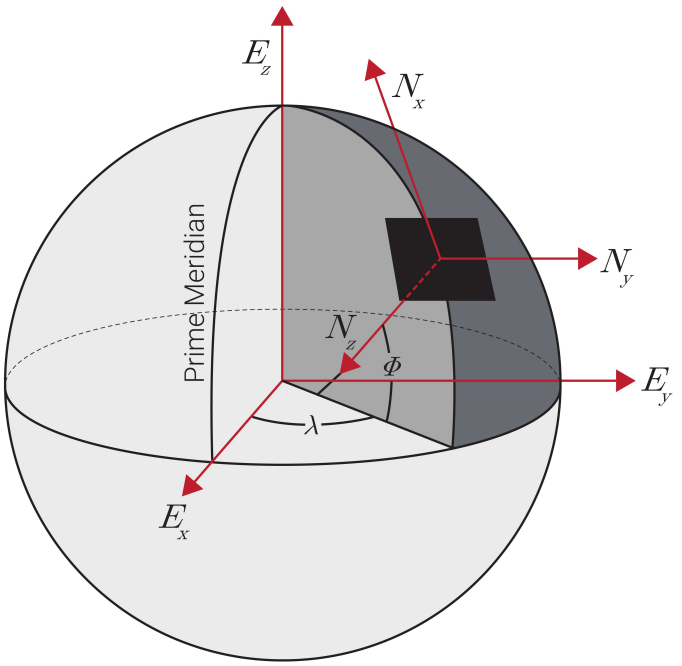


Figure 2.1: ECEF and LLA Coordinate Frames [31]

common frame is often selected into which each measurement is rotated and resolved. The frames used in this thesis and the mechanisms for relating them to each other are described in this section.

2.1.1 Local Tangent-Plane Frame

A local tangent-plane navigation frame is a commonly used to describe a frame which is valid for navigation with respect to a locally defined origin. This frame often aligns with some form of topographic directions (NED) and originates a coordinate system with a flat-earth assumption at a provided origin location. This origin is often chosen to be a local landmark or the first fixed position solution, and all subsequent navigation solutions are then with respect to this frame. The further the navigator strays from the origin, the more error is introduced due to the ellipsoidal nature of the Earth's surface. This local-frame assumption will be employed for the ground-vehicle trajectories presented in this thesis.

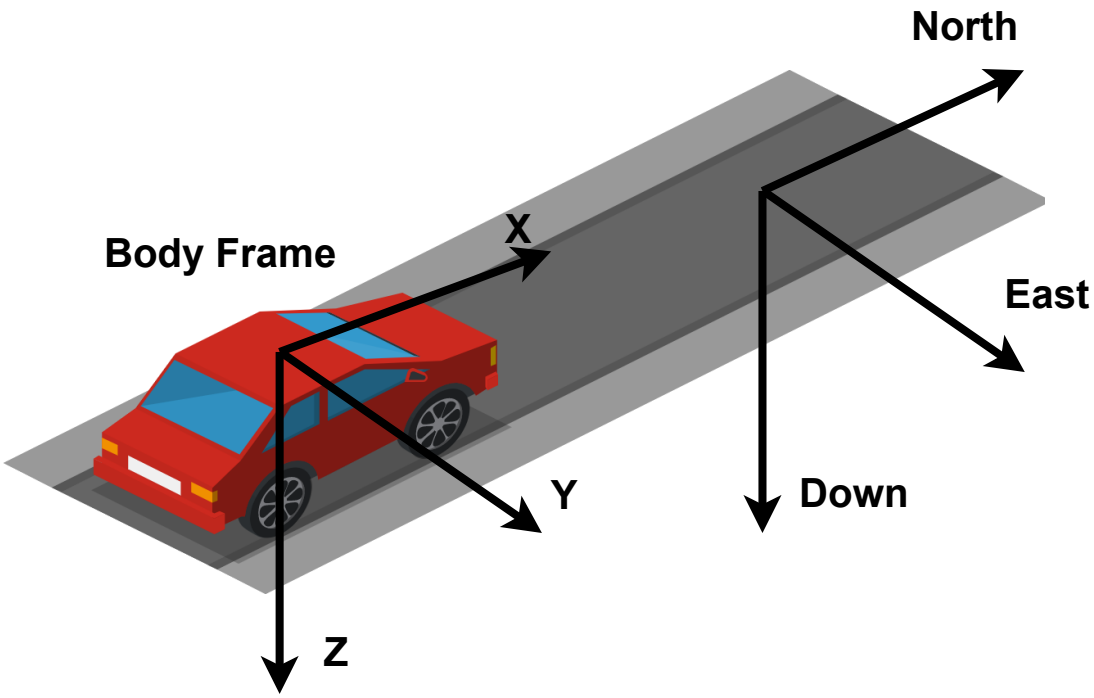


Figure 2.2: NED and Vehicle Body Frames

Note that all algorithms are derived in the ego vehicle's frame, so this NED frame is only used for visualization.

2.1.2 Body Frame

Body frame is a general term for any coordinate system defined to move with a body of interest. For a ground vehicle, the frame originates at the center of gravity (COG) with two common axis conventions: X (forward), Y (right), and Z (down) or X (right), Y (forward) and Z (up). These conventions are convenient as they align with the most commonly utilized local navigation solutions, NED and ENU (East-North-Up), when the Euler angles (described in the following section) are zero. The forward-right-down convention is selected for this thesis to align with the chosen local NED frame. Figure 2.2 depicts these frames.

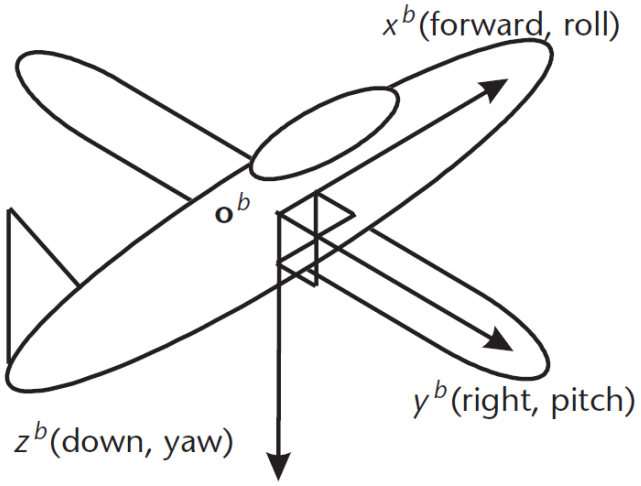


Figure 2.3: Aircraft Body-Frame Axis Convention [32]

2.1.3 Coordinate Transformation Matrices

The relationship between two 3D coordinate frames may be expressed in numerous different forms. One of the most useful representations, and the one which will be used in this thesis, is a coordinate transformation matrix consisting of 9 elements which relate common vectors in each frame. Commonly referred to as direction cosine matrices (DCMs) or rotation matrices, these matrices allow for rotation of a vector from one frame to another with simple matrix algebra. In Equation 2.1, the rotation matrix R_a^b rotates a vector resolved in the a frame, $\vec{x}^a$, into an equivalent vector resolved in the b frame, $\vec{x}^b$.

$$\vec{x}^b = R_a^b \vec{x}^a \tag{2.1}$$

In order to generate a rotation matrix, the composition of three sequential elementary rotations can be considered. The angles of these rotations are commonly referred to as Euler angles, but this general term encapsulates numerous conventions representing varying orders and axes of rotations. The Tait-Bryan convention (yaw, pitch, roll) shown in Figure 2.3 is frequently used in navigation, especially for aircraft, and will be described here.

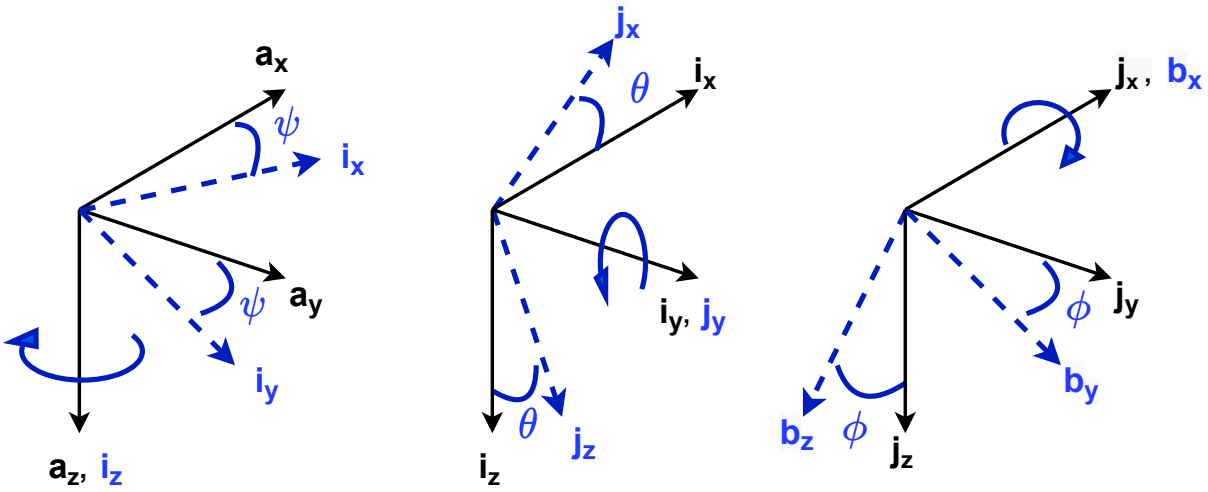


Figure 2.4: Three Successive Elementary Rotations

The first step in this convention is an elementary rotation about the original Z axis by the yaw angle, ψ. This rotation generates a new intermediate coordinate system, i, described by $X'Y'Z'$. The pitch rotation occurs next about the new Y' axis, i_y, by a magnitude of θ. Once more an intermediate frame, j, is considered and the final coordinate frame is derived by a roll about the X'' axis, j_x. These three elementary rotation steps are depicted in Figure 2.4.

The rotation matrices defining each of these elementary rotations are shown in Equations 2.2 to 2.4.

$$R_Z = R_a^i = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

$$R_Y = R_i^j = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \tag{2.3}$$

$$R_X = R_j^b = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \tag{2.4}$$

The resulting matrix is described by pre-multiplying the successive rotations as shown in Equation 2.5.

$$R_a^b = R_j^b(\phi) R_i^j(\theta) R_a^i(\psi) \tag{2.5}$$

This convention of employing intermediate axes for each successive rotation is often referred to as a body-fixed rotation sequence. A common alternative is to describe the rotations about the original axes, a space-fixed sequence. This distinction may seem critical, but in practice either leads to the same orientation [33]. Either is usable to generate the required matrices so long as the chosen convention is consistent and clear. The resulting rotation matrix for the Tait-Bryan convention defined here is shown in terms of Euler angle

components [32].

$$R_a^b = \begin{bmatrix} \cos(\theta)\cos(\psi) & \cos(\theta)\sin(\psi) & -\sin(\theta) \\ -\cos(\phi)\sin(\psi) + \sin(\phi)\sin(\theta)\cos(\psi) & \cos(\phi)\cos(\psi) + \sin(\phi)\sin(\theta)\sin(\psi) & \sin(\phi)\cos(\theta) \\ \sin(\phi)\sin(\psi) + \cos(\phi)\sin(\theta)\cos(\psi) & -\sin(\phi)\cos(\psi) + \cos(\phi)\sin(\theta)\sin(\psi) & \cos(\phi)\cos(\theta) \end{bmatrix} \tag{2.6}$$

The close relationship between Euler angles and rotation matrices is thus seen above. The angles can be utilized to compose the rotation matrix and likewise the rotation matrix may be decomposed into the more intuitive angles whenever necessary through [32]:

$$\phi = \arctan(R_a^b(2,3), R_a^b(3,3)) \tag{2.7}$$

$$\theta = -\arcsin(R_a^b(1,3)) \tag{2.8}$$

$$\psi = \arctan(R_a^b(1,2), R_a^b(1,1)) \tag{2.9}$$

There are a few key properties of rotation matrices to note. First, in order for rotation matrices to map equivalent vectors in between frames, it is required that rotation matrices do not impose any scaling of the vector through multiplication. That is, the norm of each row and column of a valid rotation matrix should have a magnitude of 1.

$$||R_{(:,i)}|| = 1 \quad \forall\ i \tag{2.10}$$

$$||R_{(j,:)}|| = 1 \quad \forall\ j \tag{2.11}$$

Additionally, all rotation matrices have columns which satisfy an orthogonal set. That is, each column is orthogonal to the others — this is necessary to guarantee the rotation yields values in an orthogonal coordinate frame. The combination of this property with those described in Equations 2.10 and 2.11 results in rotation matrices being considered orthonormal. All orthonormal matrices have the property of an inverse and transpose being

equivalent as shown in Equation 2.12.

$$(R_a^b)^{-1} = (R_a^b)^T \tag{2.12}$$

This property is valuable since the transpose of a matrix is simpler to calculate than the inverse. Additionally, the inverse of a rotation matrix describes the opposite rotation such that,

$$R_b^a = (R_a^b)^{-1} = (R_a^b)^T \tag{2.13}$$

For relative positioning of ground vehicles, a 2D assumption is considered sufficient for the majority of critical applications [1]. Therefore, throughout the remainder of this thesis, the focus will be on two-dimensional coordinate frames and the transformations between them. Two rotations of interest will be used extensively in the sections to follow: body-to-NED and multi-vehicle body-to-body. Both of these rotations have the same form: an elementary rotation about a shared Z-axis as in Equation 2.2. However, since the third dimension is not of interest and assumed to remain aligned with each other, the matrix may be simplified. The body-to-NED rotation is dependent on the heading measured clockwise from north while the body-to-body rotation is calculated from the difference in heading between the two vehicles. Equations 2.14 and 2.15 depict these relationships along with Figure 2.5 where C will be the chosen variable to represent the coordinate transformations from here on in this thesis.

$$C_{V_1}^N = R_Z(\psi)^T = \begin{bmatrix} \cos(\psi_{V_1}) & -\sin(\psi_{V_1}) \\ \sin(\psi_{V_1}) & \cos(\psi_{V_1}) \end{bmatrix} \tag{2.14}$$

$$C_{V_1}^{V_2} = R_Z(\Delta\psi)^T = \begin{bmatrix} \cos(\Delta\psi) & -\sin(\Delta\psi) \\ \sin(\Delta\psi) & \cos(\Delta\psi) \end{bmatrix} \tag{2.15}$$

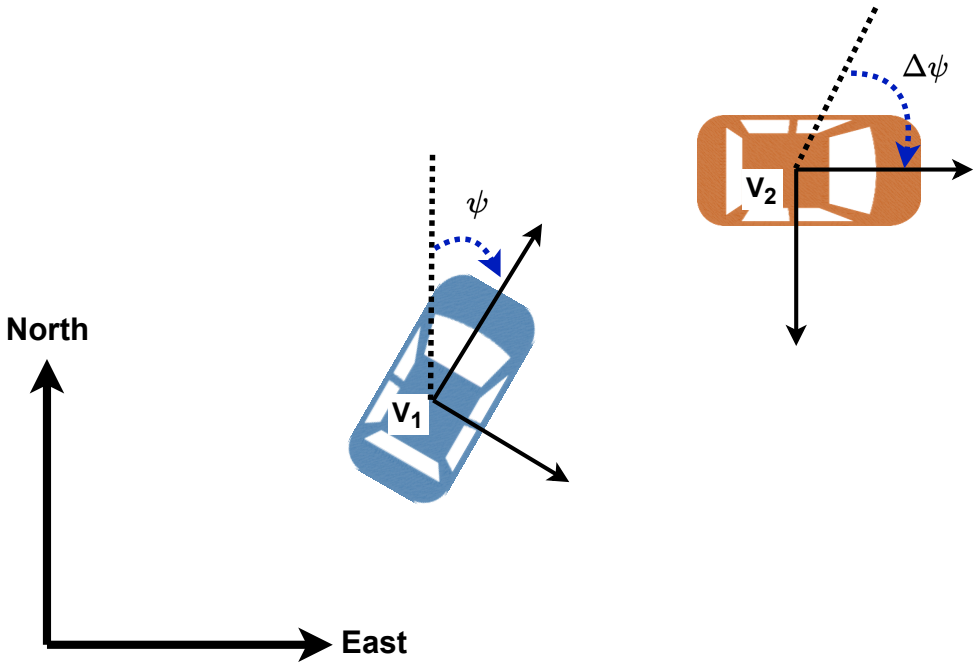


Figure 2.5: Vehicle and Local NED Frames

2.2 Vehicle Odometry

As mentioned in Section 1.1, two aspects are critical to the relative position tracking problem: the relative position of the target vehicle and estimates of the relative motion through time. This second aspect involves predictions of the future path of the vehicle through estimates of the vehicle's motion. The position and attitude of a vehicle over time is known as pose, and estimating the change in this pose over time through dynamic measurements is referred to as odometry. In addition to estimates of the tracked vehicle's odometry, knowledge of the ego (vehicle performing the on-board estimation) vehicle's dynamics is inherently required since the tracking presented in this thesis is calculated in the ego vehicle's moving body-frame. This necessitates defining the motion of the resolving frame. Solving for a vehicle's odometry has been well-studied using numerous methods which can often be categorized as either kinematic or model-based implementations.

Model-based methods incorporate knowledge of the operating vehicle or the surrounding environment to aid in the estimation process. For example, wheel encoders are commonly

employed on ground vehicles, but knowledge of the wheel radius is required to convert the number of revolutions into a ground speed estimate. Model-based implementations can be used in tandem with a variety of sensors and platforms: [34] incorporated non-holonomic ground-vehicle constraints to visual odometry methods while [35] employed UAV rotor dynamics with visual and inertial sensors.

Alternatively, kinematic approaches do not depend on vehicle parameters which typically vary between platforms. This method is commonly employed through accelerometer and gyroscope measurements from an IMU. The acceleration and angular velocity measurements from these sensors can be utilized to estimate the full 6 degrees of freedom (6-DOF) for any rigid body maneuvering in 3D space without requiring knowledge of the platform. The lever arm from the body's center of gravity is a required parameter, but this is typically well measured or can be found through estimation techniques such as transfer alignment [36–38]. Both model-based and kinematic methods will be explored in the following sections as they relate to vehicle odometry and the relative position tracking problem.

2.2.1 IMU Mechanization

Most inertial measurement units are comprised of three orthogonal single-axis accelerometers along with three orthogonal single-axis gyroscopes each meant to measure the specific force and angular rate along a single sensor body axis [32]. To make use of these sensor measurements, the resolving frame should first be selected along with the chosen method of depicting attitude. Here, the attitude will be described with the rotation matrix method described previously and the solutions will be resolved in an NED frame.

IMU mechanization is an example of a dead-reckoning technique. Dead reckoning is the process of finding the current position of an object by updating a previous position with estimates of motion over time. Therefore, all dead-reckoning techniques require initialization of the object's pose at the beginning of the algorithm. This initialization reference may come from various sources depending on the platform and use case. To obtain a global

solution from dead reckoning, a global reference is required for initialization—this is often the case for GPS/INS coupled solutions where the GPS solution is utilized for first position fix. If only a relative solution is desired, then the initial pose may be with respect to a nearby landmark or by taking the vehicle's initial position as the origin of a local navigation frame. The mechanization described here assumes the initial position to be the origin of the NED frame with a known initial attitude. It should be reiterated that this tangent-frame implementation with a static origin is valid for local solutions only. [32] provides an NED frame mechanization that is valid beyond local solutions by updating an origin that moves with along with the navigator. Although 2D solutions are the primary goal of this thesis, the general 3D mechanization is provided for thoroughness. Implementing zeros for the extraneous gyro and accelerometer measurements yields the equivalent solution to a simplified 2D mechanization.

Given initial values, the position, velocity, and attitude (PVA) states may be initialized as in Equations 2.16–2.18.

$$\vec{p}^{\,N}_{Nb}(1) = \begin{bmatrix} N_0 \\ E_0 \\ D_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.16}$$

$$\vec{v}^{\,N}_{Nb}(1) = \begin{bmatrix} v_N \\ v_E \\ v_D \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.17}$$

$$C^N_b(1) = (R^b_a(\phi_0, \theta_0, \psi_0))^T \tag{2.18}$$

Where the notation $\vec{p}^{\,N}_{Nb}$ has a subscript denoting the position vector of the body with respect to the local-tangent NED frame and a superscript expressing its resolution in said frame. This notation is critical to understand the following equations. The attitude is initialized

with the transpose of the rotation matrix shown in Equation 2.6 with initial Euler angles—the rotation matrix is updated directly beyond this point. After initialization is completed, the following steps are repeated for each successive inertial measurement received.

The previous attitude is updated with measurements from the gyroscopes along with compensation for the rotation rate of the Earth. In discrete time, a first-order approximation of this step is shown in Equation 2.19.

$$C_b^N(i+1) = C_b^N(i)(I + \Omega_{ib}^b \Delta t) - \Omega_{ie}^N C_b^N(i)\Delta t \tag{2.19}$$

This update assumes a small integration period, Δt, which represents the time step between consecutive IMU measurements. The Ω_{ib}^b and Ω_{ie}^N terms are skew-symmetric matrix forms of the input vectors — the gyroscope measurements and the rotation rate of the Earth, respectively. A skew-symmetric form of a vector is constructed according to Equation 2.20.

$$\Omega = \text{Skew}(\vec{\omega}) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{2.20}$$

The rotation rate of the Earth is considered constant about the Earth's Z-axis (connecting the two poles) at $\omega_E = 7.292115 * 10^{-5}$ radians per second. This may be rotated into the local frame through the latitude of the local frame origin, L_b:

$$\Omega_{ie}^N = \text{Skew}(\omega_E * [\cos(L_b) \quad 0 \quad -\sin(L_b)]^T) \tag{2.21}$$

The specific force measurements of the accelerometers are measured with respect to an inertial frame and resolved into the sensor's body frame. These measurements may then be rotated into the resolving frame with the average of the previous and newly-updated

attitude:

$$\vec{f}_{ib}^{N} = \frac{1}{2}\left[C_b^N(i) + C_b^N(i+1)\right]\vec{f}_{ib}^{b} \tag{2.22}$$

The velocity with respect to the local-tangent frame may then be updated. As was mentioned before, accelerometers measure *specific force*, $\vec{f}_{ib}^{b}$, which is the sum of the non-gravity forces per unit mass acting on a body with respect to an inertial frame. This means a perfect IMU in freefall would measure zero—contrary to the approximate 9.81 $\frac{m}{s^2}$ acceleration expected on Earth. Therefore, a gravity term must be added to the specific force in order to generate an estimate of the body's acceleration. This is commonly done with reference to The World Geodetic System 1984 (WGS84) datum which provides a simple gravity model as a function of latitude [39]. With this gravity compensation and a Coriolis term due to rotation of the Earth, Equation 2.23 updates the velocity states.

$$\vec{v}_{Nb}^{N}(i+1) = \vec{v}_{Nb}^{N}(i) + \left[\vec{f}_{ib}^{N} + \vec{g}_b^N - 2\Omega_{ie}^{N}\vec{v}_{Nb}^{N}(i)\right]\Delta t \tag{2.23}$$

Finally, the position with respect to the local-tangent frame origin may be updated:

$$\vec{p}_{Nb}^{N}(i+1) = \vec{p}_{Nb}^{N}(i) + \left[\vec{v}_{Nb}^{N}(i+1) + \vec{v}_{Nb}^{N}(i)\right]\Delta t \tag{2.24}$$

A flowchart summarizing an epoch of this process is shown in Figure 2.6.

2.2.2 Ground Vehicle Dynamic Models

Vehicle models are highly useful approximations of physical truth and are commonly employed to produce simulations, generate control commands, or improve estimation results. Ground-vehicle systems often employ kinematic methods to improve their odometry solutions. These can consist of providing known motion constraints to an IMU-based implementation through pseudo-measurements as in [40–42], or the constraints can simplify the

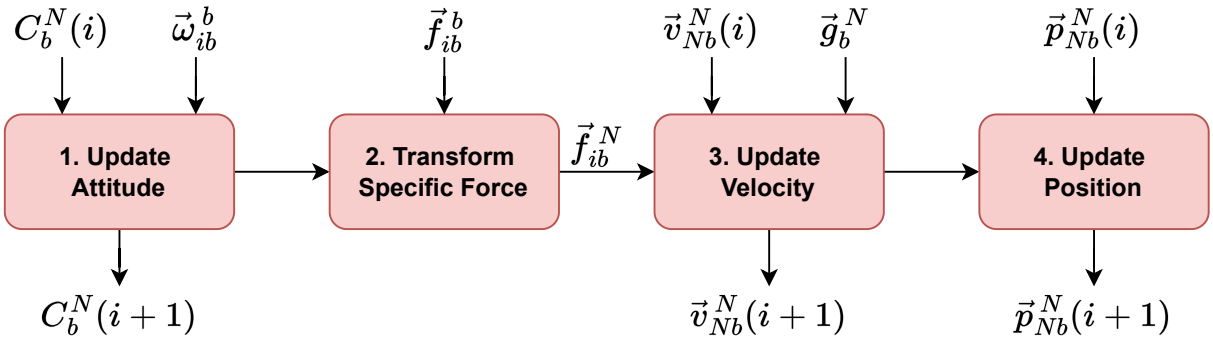


Figure 2.6: IMU Mechanization Process

required measurements to predict the motion. Numerous models of varying complexity have been implemented throughout literature. One of the simplest and most common models, the kinematic bicycle model, employs geometric relationships to predict motion without considering the forces governing said motion [43]. This model allows for fewer sensor measurements to provide an odometry solution, and key components of this model will also be useful to assist in the estimation process in proceeding chapters.

Kinematic Bicycle Model

The first major assumption of this model is the reduction of a four-wheeled vehicle into two by combining the rear and front tires into one. Assuming only front-wheel steering, the resulting depiction mimics a bicycle and is shown in Figure 2.7.

Perpendicular lines drawn from the surface of each tire intercept at a point, O, which is known as the instantaneous point of rotation for the vehicle. R depicts the radius of this rotation by connecting the COG and this stationary point. δ represents the deflection at the wheels due to a steering input. L is known as the wheelbase and is the distance between the two axles, and β is the sideslip angle. From these definitions, the geometric relationships governing the vehicle's motion may then be described. First, the following two trigonometric

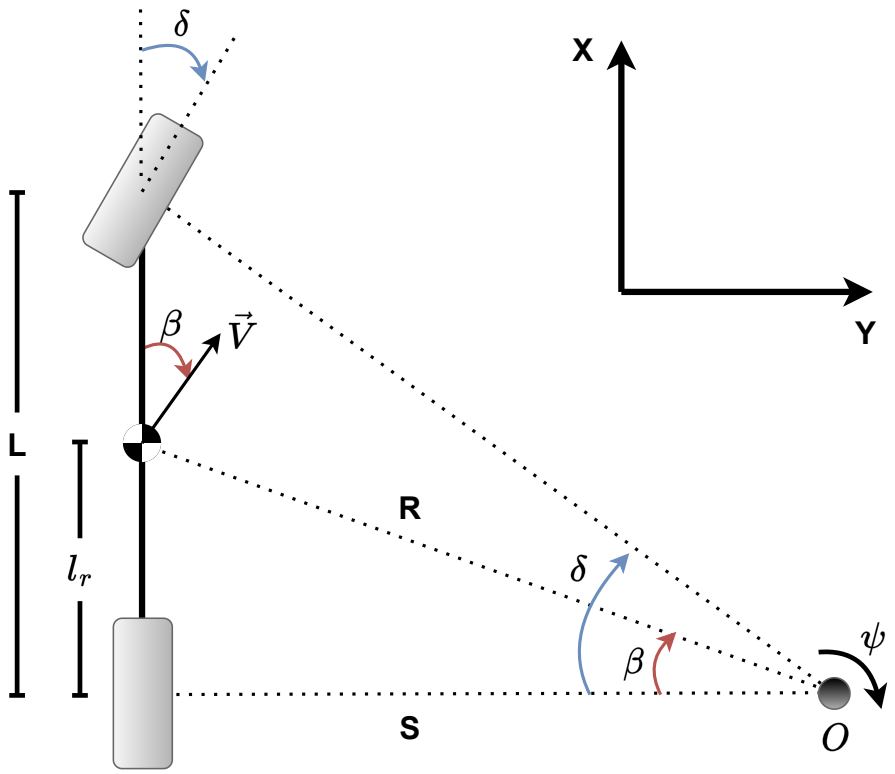


Figure 2.7: Kinematic Bicycle Model

relationships need to be considered:

$$S = \frac{L}{\tan(\delta)} \tag{2.25}$$

$$R = \frac{S}{\cos(\beta)} = \frac{L}{\tan(\delta)\cos(\beta)} \tag{2.26}$$

The angular velocity of the vehicle may then be described by the speed, $|\vec{V}|$, and this radius:

$$\dot{\psi} = \frac{|\vec{V}|}{R} \tag{2.27}$$

and so:

$$\dot{\psi} = \frac{|\vec{V}|\tan(\delta)cos(\beta)}{L} \tag{2.28}$$

The sideslip angle can be related to other known geometric values as:

$$\beta = \arctan\left(\frac{l_r \tan(\delta)}{L}\right) \tag{2.29}$$

With X and Y as local-frame position components of the vehicle, the component velocities and yaw rate of the COG, are summarized with Equations 2.30–2.32

$$\dot{X} = |\vec{V}|\cos(\psi + \beta) \tag{2.30}$$

$$\dot{Y} = |\vec{V}|\sin(\psi + \beta) \tag{2.31}$$

$$\dot{\psi} = \frac{|\vec{V}|\tan(\delta)\cos(\beta)}{L} \tag{2.32}$$

The sideslip angle term, β, relates the body's longitudinal and lateral velocities at an instant in time while ψ denotes the heading of the vehicle measured from the local X-axis. The sum of these values is considered the vehicle's course. Depending on the application, it may useful to instead find the velocity components in the body frame as seen in Equation 2.33.

$$\vec{v}^b = \begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} |\vec{V}|\cos(\beta) \\ |\vec{V}|\sin(\beta) \end{bmatrix} \tag{2.33}$$

Although simplistic, this model is quite useful in practice at low speeds since this model assumes no side-slip at the tires—an assumption that becomes less accurate at higher speeds and higher lateral accelerations [43]. This is clear from the relationship shown in Equation 2.34 which depicts the lateral forces required at the tires required to turn at a given radius and speed.

$$F_{lat} = \frac{mV^2}{R} \tag{2.34}$$

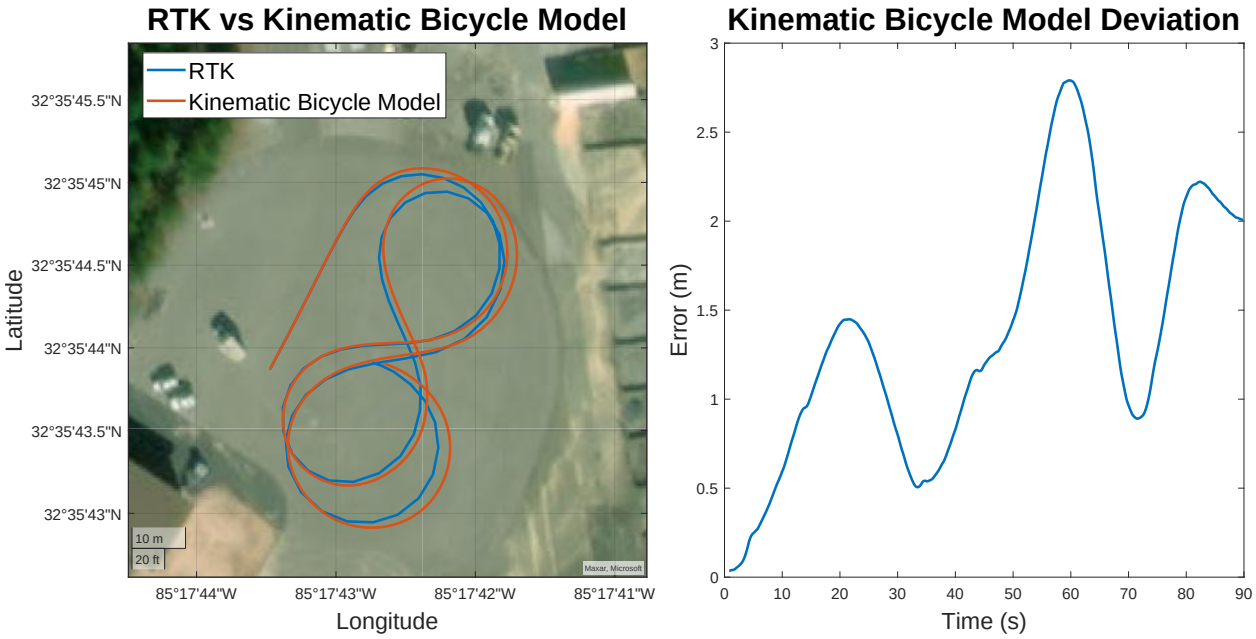


Figure 2.8: RTK and Kinematic Bicycle Model Experimental Comparison

This thesis focuses on low-speed relative trajectories. Therefore, the simulation described in a later chapter makes use of this model as its basis. Through this model, the odometry of a ground vehicle in a local-frame can be estimated with only two sensors. A example comparison of this odometry method to a real-time kinematic (RTK) ground truth from an Auburn test vehicle with wheel encoder and steer angle sensors is shown in Figure 2.8.

Even with the assumptions made within the kinematic bicycle model, the accuracy remains within a few meters after a 1.5-minute test. Note the max speed achieved during this test was roughly 13.4 miles per hour, thus the low-speed requirement was well preserved.

Dynamic Bicycle Model

Although it will not be detailed here, it is worth noting the existence of a more sophisticated ground vehicle model known as the dynamic bicycle model. At higher speeds, this model's consideration of forces at the tires more accurately predicts the lateral motion of the vehicle. With well-known vehicle parameters, the dynamic bicycle model is quite accurate

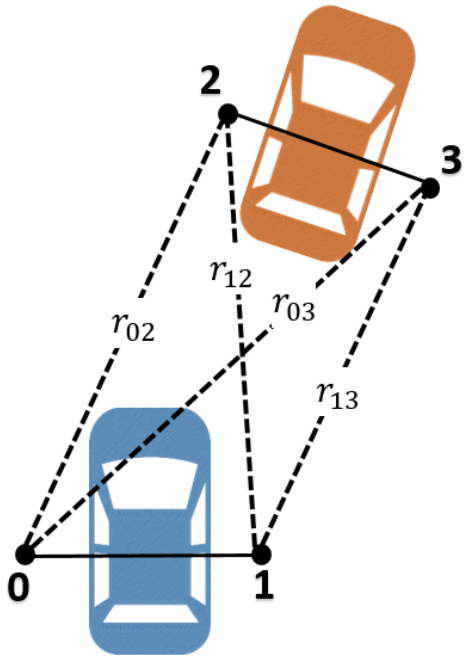


Figure 2.9: UWB Layout and Corresponding Range Measurements

under most circumstances. This model is therefore one of the most widely used in literature. [43] provides a thorough derivation and analysis of this model while a comparison of these kinematic and dynamic versions may be seen in [44].

2.3 Ultra-wideband Radios (UWBs)

As mentioned, the estimation algorithms presented in this thesis are reliant upon inter-vehicle ranging measurements. UWB node-to-node ranging is one of the most-commonly employed relative measurements in collaborative estimation techniques. UWBs are employed in this thesis for all ranging measurements between the locally-operating vehicles, and each vehicle is assumed to be equipped with two UWBs positioned horizontally across the vehicle's roof. This results in each vehicle producing four ranges with a given neighbor as seen in Figure 2.9. Note the sensor spacing is exaggerated for clarity. UWB technology, performance, and the chosen hardware model will be described in this section.

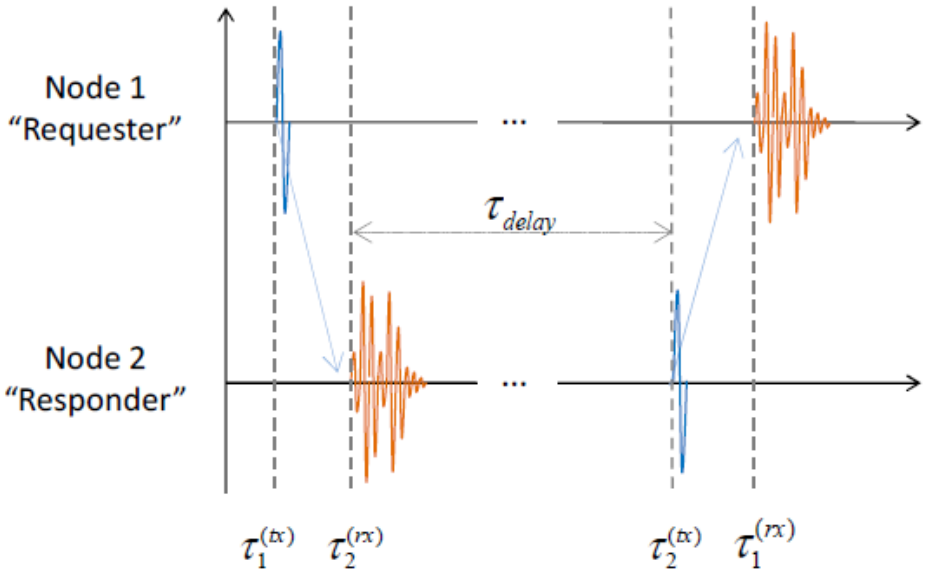


Figure 2.10: Two-Way Time-of-Flight (TW-TOF) Process [45]

2.3.1 Measurement Methodology

The most common implementation of UWB technology generates peer-to-peer measurements with a two-way time-of-flight (TW-TOF) protocol. This method removes the need for accurate time synchronization between nodes required by time of arrival (TOA) techniques. Figure 2.10 depicts a simplified illustration of this process. The process begins with the requesting node sending a pulse (blue) at a recorded time, τ_1^{tx}. The time of arrival of the request at the responder, τ_2^{rx}, is found from a leading edge detection (LED) algorithm and recorded. The proceeding orange signals seen after this point are duplicated due to multipath effects which can be mitigated with successful LED. The responder then forms a response signal after a highly precise delay interval, τ_{delay}, and generates its own waveform to be broadcast back to the requester node. This packet contains the delay interval which will be required for the range calculation. At the requester node, the leading edge is once more calculated to generate the TOA for the response, τ_1^{rx}. With the three necessary values, τ_1^{rx}, τ_1^{tx}, and τ_{delay}, collected at the requesting node, the range may be calculated by

Figure 2.11: TimeDomain (TDSR) PulsON 440 Development Board [46]

Equation 2.35.

$$\text{Range} = c * \frac{1}{2}(\tau_1^{rx} - \tau_1^{tx} - \tau_{delay}) \tag{2.35}$$

2.3.2 Hardware Description

The methodology described above is a general outline of the TW-TOF procedure utilized by modern UWBs. However, the exact signal structures and processing methods are more complex and vary between UWB models and manufacturers. The methodology described here is based upon the selected hardware: a TimeDomain PulsON 440 (P440). This module, shown in Figure 2.11, is one of the top-performing and well-recognized UWB modules at the time of writing. Note that the manufacturer, TimeDomain, has undergone acquisitions since the creation of this module. The P440 is being improved upon but remains available under a new company name: TDSR [46].

The following description is a summary of more-detailed information in [47]. Communication between the P440s is performed by sending a train (packet) with hundreds or thousands of pulses. The transmission rate and phase of these pulses are determined by a pseudo-random code. Thus, a responding node must have knowledge of this code in order

to lock onto this pulse train. The train of pulses also may contain user data which offers the ability to use one sensor for both communication and ranging simultaneously. This creates an even higher-incentive for the use of UWBs in the presented collaborative algorithms which inherently rely upon data transmission to operate. Up to 1024 bytes may be included in each packet of a ranging request. However, only 136 bytes can be sent per packet without increasing packet size or transmission time.

It should be noted that the short-pulse nature of the signals are inherently broadband. This is verified in a Fourier analysis of an ideal impulse over an infinitesimally short duration which would have infinite bandwidth [48]. This results in in the highest possible bandwidths and lowest center frequencies for the UWB signals. This improves multipath rejection considerably; however, it also means each pulse fully occupies the entire operating bandwidth. The result is that the common method of frequency-based channelization is not possible. Instead, the P440s pseudo-random noise encoding on each pulse train allows for the creation of parallel channels [9].

The individual pulses are very low power. To counteract this, the transmissions are encoded such that blocks of pulses are integrated to form a single communication "symbol." This summation of pulses increases the signal-to-noise ratio (SNR) of the received transmission. The number of pulses which are summed for each symbol may be selected by the user by modification of the *pulse integration index* (PII). These PII values are based on powers of two, so a PII of 4 results in 2^4 or 16 pulses per symbol. The more pulses integrated into each symbol, the higher the SNR which results in a further operating range. However, this comes at a trade-off with update rate of the ranging network. A table describing the expected ranges (in free space) and measurement rates is recreated from data sheets in Table 2.1 [47].

Note that UWBs often operate in a network format where only a single range is being generated at a given instance in time. The result is that for the ranging network in this thesis, only one of the four desired ranges will be present at a given instance in time. An

Table 2.1: PulsON P440 Operating Results from PII Selection

PII	Max Range (m)	Range Measurement Rate (Hz)	Data Rate (kbps)
4	66	123.5	612.48
5	101	102.0	308.48
6	145	72.5	144.64
7	207	47.2	76.80
8	260	27.5	39.04
9	410	14.9	19.20

advertised ranging measurement rate is therefore divided among the nodes in the network. For example, at the chosen PII value of 9, the update rate for r_{02}, seen in Figure 2.9, would be approximately 14.9/4 or 3.725 Hz. However, there exist methods for dynamically switching connectivity (ALOHA protocol) and research is being conducted to generate the ranges concurrently [49]. Regardless of the update rate, the expected accuracy is approximately 2 cm if properly calibrated as documented on the data sheet.

2.3.3 Experimental Analysis

The selected UWBs advertise an accuracy on the order of a few centimeters up to 400 meters of range. Before analyzing the results on the vehicle platforms, the hardware needs to undergo calibration. The calibration process built-in to the system's software is for a single node-to-node calibration. This successfully accounts for the electrical delays between a two node range request. However, this method distributes the calculated delays between the two nodes evenly and cannot determine the individual error in each. For example, a calibration between nodes 1 and 2 does not provide information on the errors internal to node 2 and it is not valid for any other range such as between node 2 and node 3 [50]. Although it is possible to keep a generate a reference table of errors between node pairs, solving for the individual delays of each node is far more advantageous; it allows for delay errors to be removed in software before the ranges are published and for new connections to be made between nodes

without re-calibration. For the four-node network implemented here, the delay for each node will be calculated in a fully-networked fashion with a least-squares technique.

The premise of this technique is that the only errors in a TW-TOF measurement implemented in the TimeDomain radios should be the electrical delays in each node along with random noise. The resulting measurement model is then seen to be:

$$\tilde{r}_{12} = r_{12} + \epsilon_1 + \epsilon_2 + \nu \tag{2.36}$$

With knowledge of the true ranges between the radios along with a sufficient connectivity of a network, the delays embedded in each radio, ϵ, may be calculated. This can be done from a least squares estimation technique which minimizes the residual between the measurements, $\vec{y}$, and the estimated value as a function of states, $h(\vec{x})$. In a linear model, this form is simplified to a linear a mapping matrix, H. Also, since the true ranges are known and constant, the measurement vector may be simplified to be measurements of the errors instead:

$$\tilde{\vec{y}} = \vec{y} - \vec{r} \tag{2.37}$$

where $\vec{r}$ is a vector consisting of the true ranges between each node pair. Selecting the states of interest to be the errors in each node, the final form of the measurement error model is:

$$\tilde{\vec{y}} = H \begin{bmatrix} \epsilon_0 \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix} + \nu \tag{2.38}$$

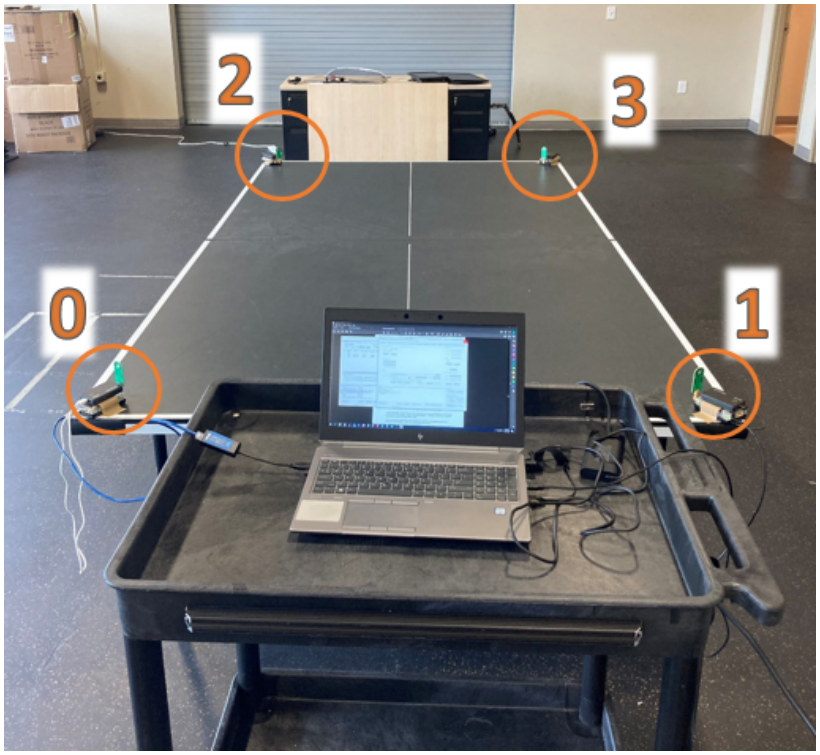


Figure 2.12: Indoor Data Collection for Calibration

If the model is properly observable (H is full-rank and invertible), the optimal estimates of the unknown states, x, can be found with:

$$\vec{x} = (H^T H)^{-1} H^T \vec{y} \tag{2.39}$$

The measurements for this calibration were made indoors across a flat rectangular surface (ping pong table) with the true spacing hand measured. This setup is shown in Figure 2.12

As shown in Figure 2.9, only 4 ranges are useful in the estimation—r_{01} and r_{23} are static and assumed to be well-measured. This results in an H matrix mapping the errors to the

measurement errors for a single set of to be of the form:

$$H\vec{x} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \epsilon_0 \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix} \tag{2.40}$$

However, with only the four range measurements needed for estimation, the least-squares calibration is unobservable as the H matrix in Equation 2.40 is not full-rank and is non-invertible regardless of how many measurements are taken. Even though they will not be used in estimation, the extra ranges r_{01} and r_{23} provide observability to the node errors; therefore, the measurements and H matrix are extended to include all 6 possible range measurements. The final measurement form which will be utilized to estimate the errors is summarized in Equations 2.41 and 2.42.

$$\tilde{\vec{y}} = H\vec{x} + \nu \tag{2.41}$$

$$\begin{bmatrix} \tilde{r}_{01} - r_{01} \\ \tilde{r}_{02} - r_{02} \\ \tilde{r}_{03} - r_{03} \\ \tilde{r}_{12} - r_{12} \\ \tilde{r}_{13} - r_{13} \\ \tilde{r}_{23} - r_{23} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \epsilon_0 \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix} + \nu \tag{2.42}$$

The errors are found by solving Equation 2.39 from a few minutes of static data (approximately 350 measurements of each range) with H and $\tilde{\vec{y}}$ concatenated accordingly. This solution is optimal if and only if a few requirements are met: there are infinite measurements, the measurement model is completely accurate, and the noise on the measurements is zero-mean and Gaussian. Here, all of the ranges are collected in a single batch and are considered

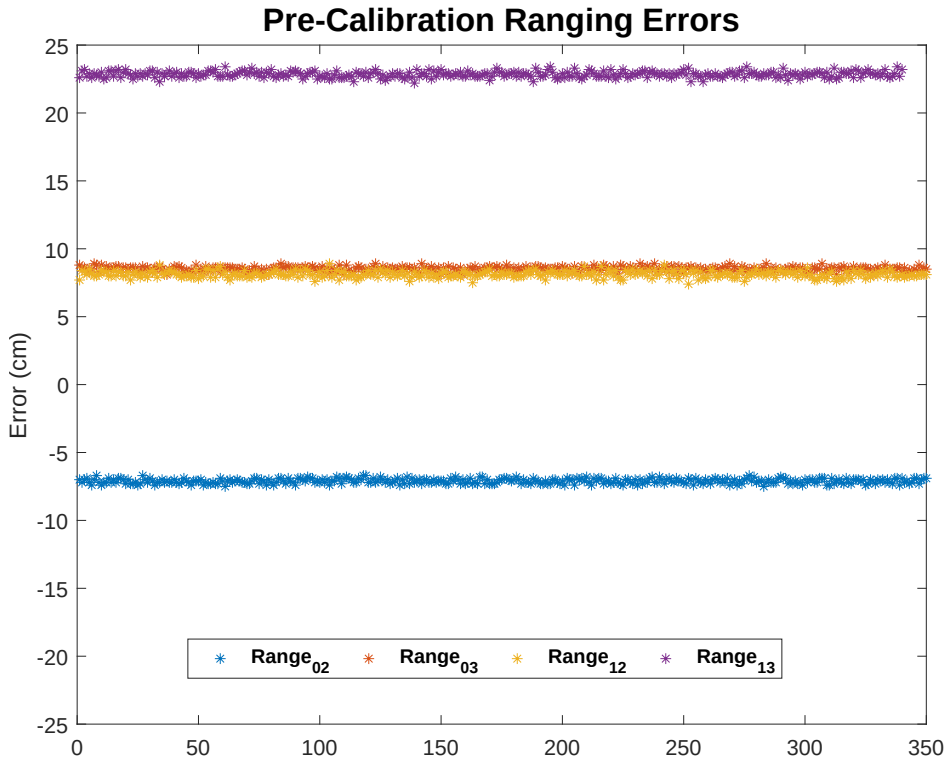


Figure 2.13: P440 Measurement Errors Pre-Calibration

to be equivalent in accuracy (dropout range measurements of 0 meters were removed before processing the results). This means no weighting of the measurements is required and the estimates may be made in post-processing since the error states are considered constant. Modifications for dynamic states and weighted estimations exist and form the basis for the Kalman filter which will be described in a later chapter.

Figure 2.13 shows the errors in the ranging network before calibration of the biases. It can be clearly seen that the network contains significant errors. This is most likely due to previous attempts to calibrate the network through the single node-to-node method. The errors are also non-zero mean, but their precision is quite similar to the TDSR documented value of 0.3 cm [47]. The mean and standard deviation of this set of data is described in Table 2.2. Note that the P440 node naming convention defines the 4-node network from 100 to 103. Therefore, the ranges may be referred to with either convention (e.g: r_{02} is the range from 100 to 102).

Table 2.2: P440 Measurement Error Statistics Pre-Calibration

Range	μ (cm)	σ (cm)
100 to 102 (r_{02})	-7.11	0.17
100 to 103 (r_{03})	8.52	0.16
101 to 102 (r_{12})	8.12	0.23
101 to 103 (r_{13})	22.85	0.22

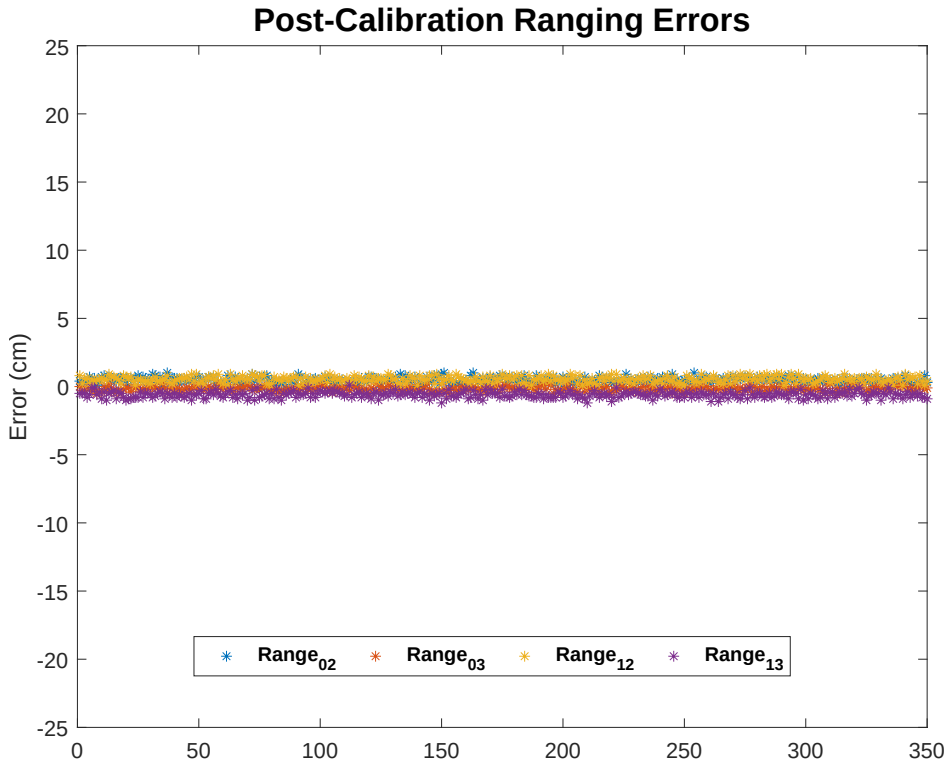


Figure 2.14: P440 Measurement Errors Post-Calibration

Following the estimation of the delay errors, the biases were loaded into the software of each UWB to compensate for their measurement delays. A new set of data was then recorded with these updated biases. Figure 2.14 shows the improvement of the measurements. The errors are now nearly zero-mean and maintain similar noise characteristics to the pre-calibration results. This indicates a static bias was properly removed. The new error statistics are shown in Table 2.3.

It should be noted that UWB errors have been shown to not be truly Gaussian in nature [48]. The result is a bimodal distribution due to quantization errors in the measurement

Table 2.3: P440 Measurement Error Statistics Post-Calibration

Range	μ (cm)	σ (cm)
100 to 102	0.45	0.22
100 to 103	-0.15	0.16
101 to 102	0.53	0.21
101 to 103	-0.60	0.22

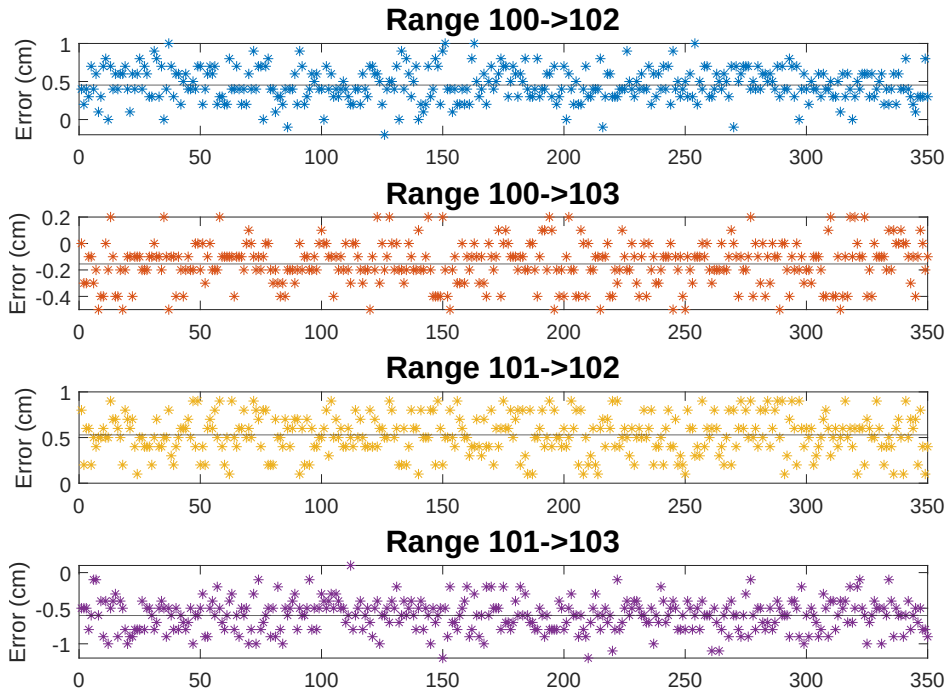


Figure 2.15: P440 Individual Range Errors Post-Calibration

methodology. It can be difficult to see, but close inspection of the individual range measurements in Figure 2.15 reveals this trend, but it is further highlighted in the Histogram in Figure 2.16. In this thesis, the Gaussian assumption will be implemented with a larger standard deviation to encapsulate both modes.

The errors calculated up to this point were found in a favorable environment: static, LOS, and indoors. The hardware will be implemented outdoors on moving vehicles, so outdoor data is taken to consider performance in this environment. The outdoor performance data of the UWBs were analyzed in comparison to an RTK-corrected GPS solution on two maneuvering vehicles. The experimental setup is seen in Figure 2.17.

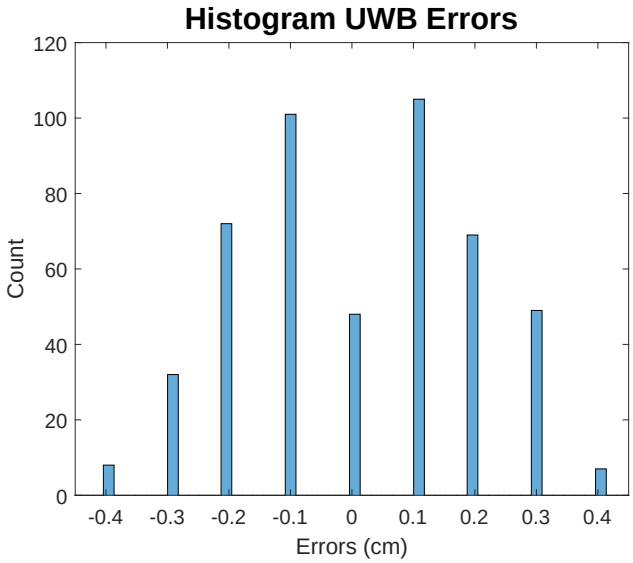
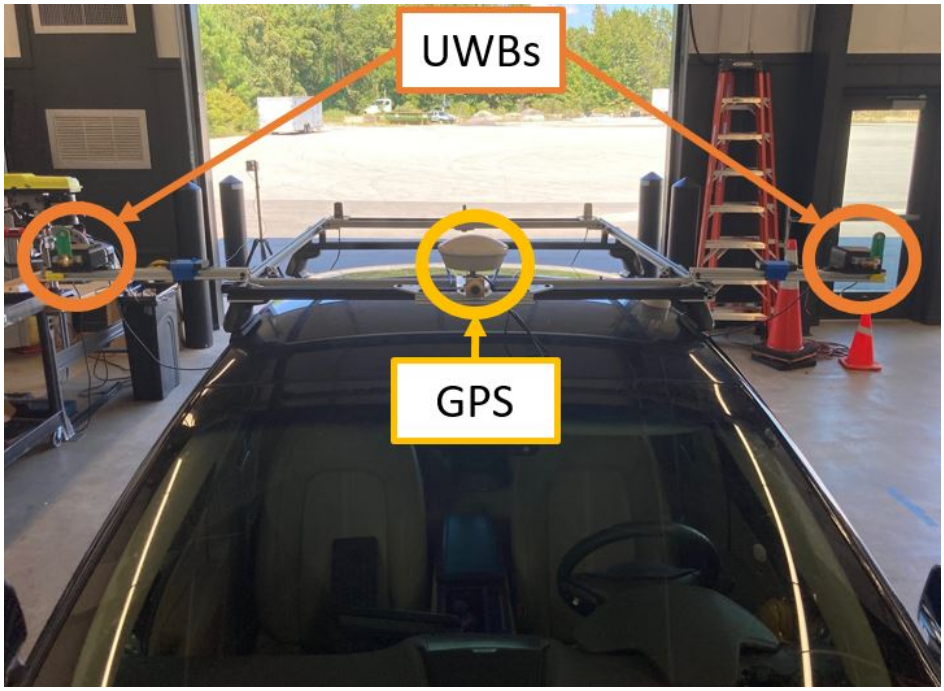


Figure 2.16: P440 Ranging Error Histogram

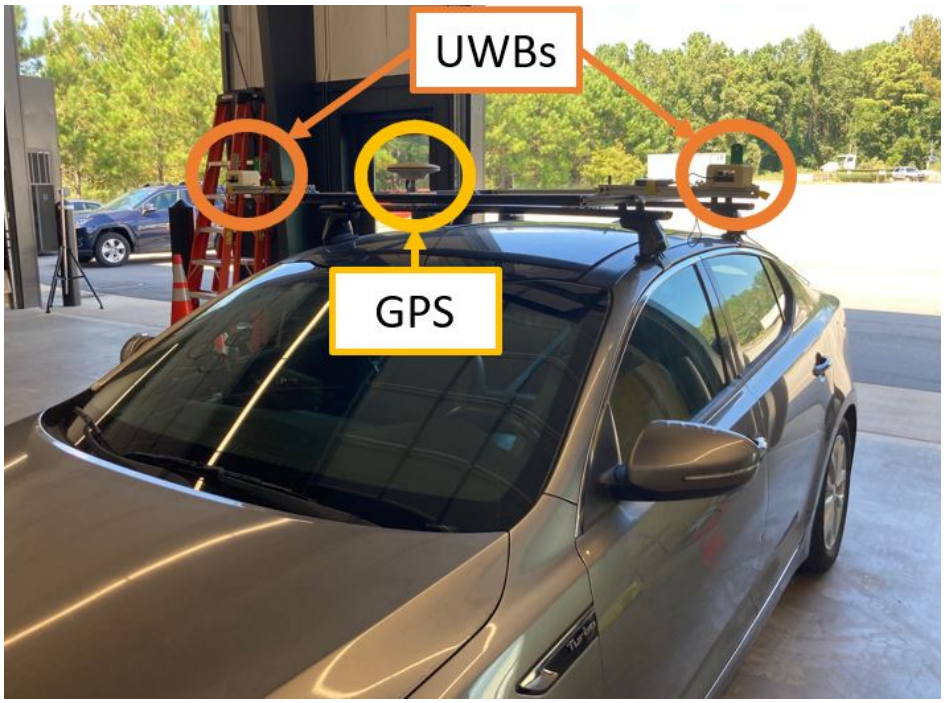
The ranges were collected in the same layout that will be utilized in the experimental results. Each vehicle's RTK antenna was placed along the baseline of its respective UWB nodes. A GPS/INS solution was also utilized to provide a heading estimate of each vehicle. Post-processing of the measurements was performed to time-sync the ranges, headings, and RTK positioning. This data was then converted to relative position vectors in the local NED frame before being rotated into the follower frame according to transpose of the rotation matrix in Equation 2.14. With a body frame relative position vector and relative heading of the two vehicles, RTK-level ranges may be extracted and utilized as truth for comparison.

The ranging accuracy in a testing environment is expected to be poorer than the indoor calibration results. This is due to a number of factors. First, although the UWB technology is resistant to multipath, it still may cause some errors. This often shows itself in Fresnel effects which result in destructive interference of the LOS signal due to reflections off surrounding surfaces. This is depicted in Figure 2.18.

Although the UWBs send and receive signals from all directions (omnidirectional), there is a slight reduction in performance when their antennas are not facing each other directly (in each other's antenna boresight). Given the orientation of the UWB nodes in the vehicles, this may frequently arise during many reasonable maneuvers of the vehicles. TDSR estimates



(a) Lincoln MKZ (71.5 Inch UWB Baseline)



(b) Kia Optima MKZ (67.25 Inch UWB Baseline)

Figure 2.17: Experimental Vehicle and Sensor Setup

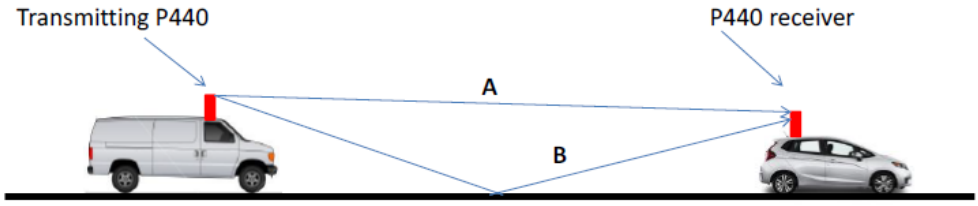


Figure 2.18: P440 Transmitted Signal (A) Destructively Interfered by Signal (B) [47]

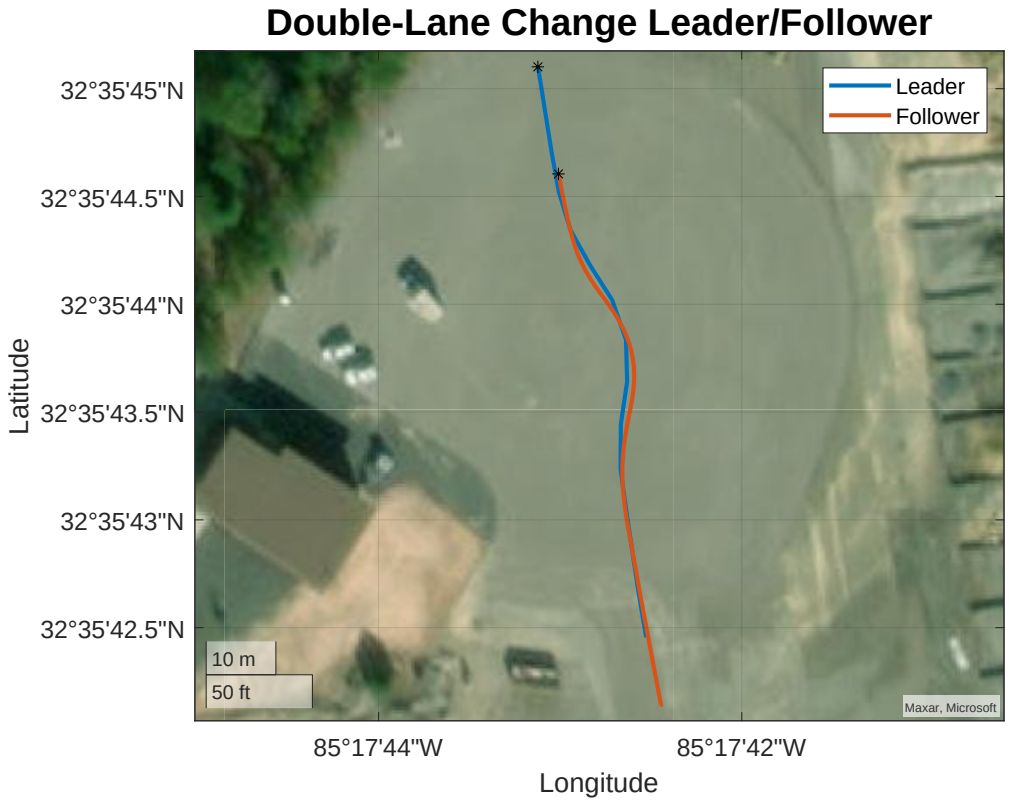


Figure 2.19: Satellite Trajectories for Double-Lane Change UWB Measurement Collection

this to be around 2 cm of added errors. Other factors which degrade the performance include onboard temperature, relative dynamics, and errors due to increasing range [47].

The goal of the following outdoor experimental analysis is to generate a reasonable error bound for the UWB measurements to be utilized in the simulation and experimental estimation algorithms. The first test was a leader-follower format starting from rest and then performing a double lane-change maneuver. The path is shown in Figure 2.19.

The errors in ranges during this maneuver are shown in Figure 2.20. The relative dynamics in this run are relatively low. It is noticeable how the error shapes loosely follow

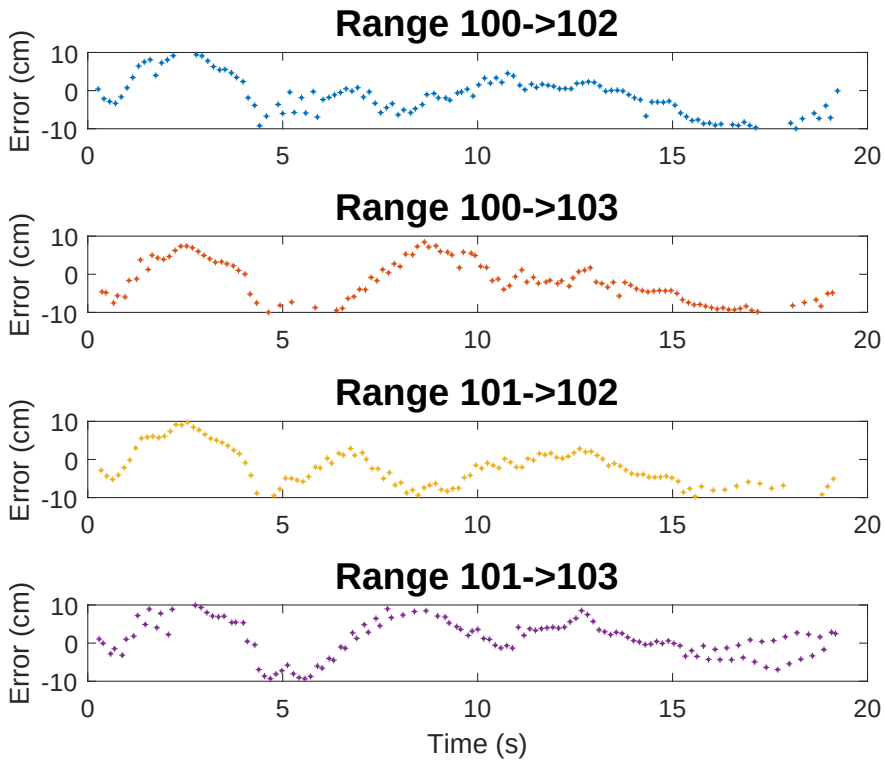


Figure 2.20: P440 Double Lane-Change UWB Ranging Errors

the dynamics of the vehicles. Although the errors are increased over the benign static test, the errors remain close to the expected outdoor accuracy of a few centimeters. The standard deviation of the errors are all around 5 cm as well. Table 2.4 shows a summary of the ranging errors for each node pair.

Table 2.4: S-Turn Following Ranging Error Statistics

Range	μ (cm)	σ (cm)
100 to 102	-1.79	5.71
100 to 103	-2.97	5.87
101 to 102	-3.12	5.60
101 to 103	1.65	5.25

A more dynamic test was performed for the second experiment. A figure-8 maneuver was performed by the two vehicles at varying follow distances. This resulted in numerous sections where the vehicles were in non-optimal orientations for ranging and the errors increased as a result. The maneuver is shown in Figure 2.21.

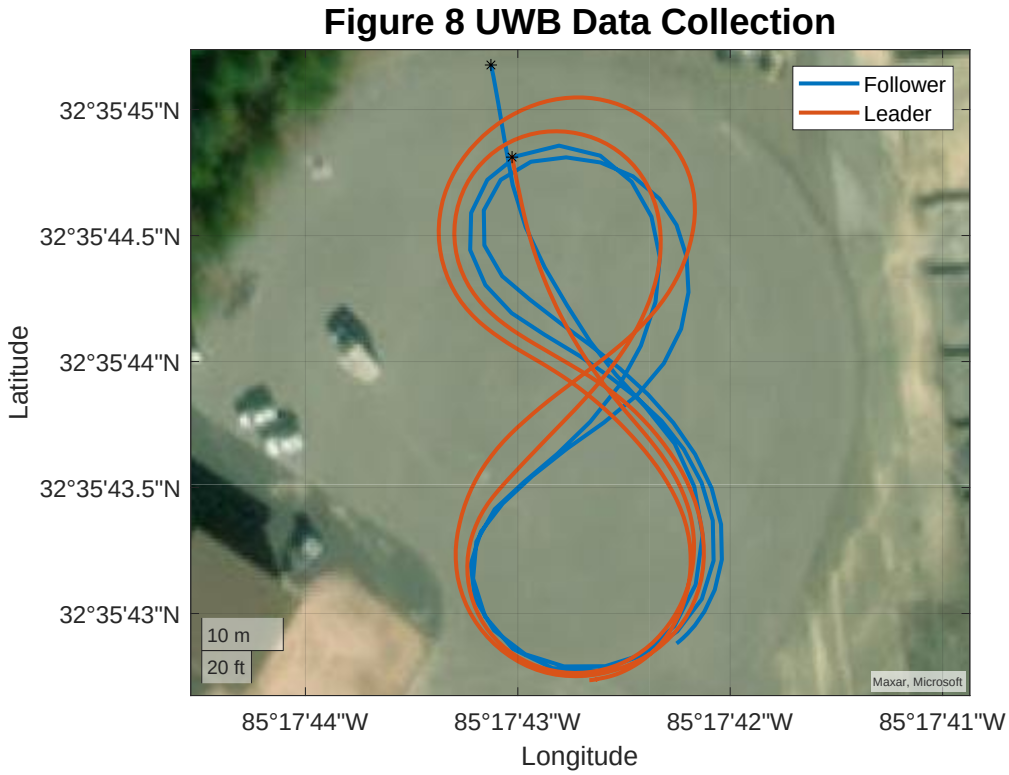


Figure 2.21: P440 Data Collection with Figure-8 Trajectories

The errors from the figure-8 test along with the errors bounds are shown in Figure 2.22. Rather than trying to model the errors as functions of range, relative geometry, and environmental effects, sufficient 3σ bounds are selected in order to approximate the errors as zero mean with a Gaussian distribution. The goal is to have a cautious but reasonable value to encapsulate both the ideal and non-ideal ranging conditions. A 3σ bounding means that 99.7% of the ranging errors should be within these bounds. From analysis of these tests, the bounds were selected to be $3\sigma = 30$cm. It can be seen that the errors lie within these bounds for nearly all of the calculated errors. This selected standard deviation (i.e. $\sigma = 10$cm) is therefore implemented for generation of the UWB measurements in the simulation environment as well as for the measurement variance. Both of these applications will be described in more detail in later sections.

There is one more important aspect to modeling and utilizing the UWB ranges: outlier rejection. Like all sensors, the P440s may sometimes have a measurement fault resulting in a

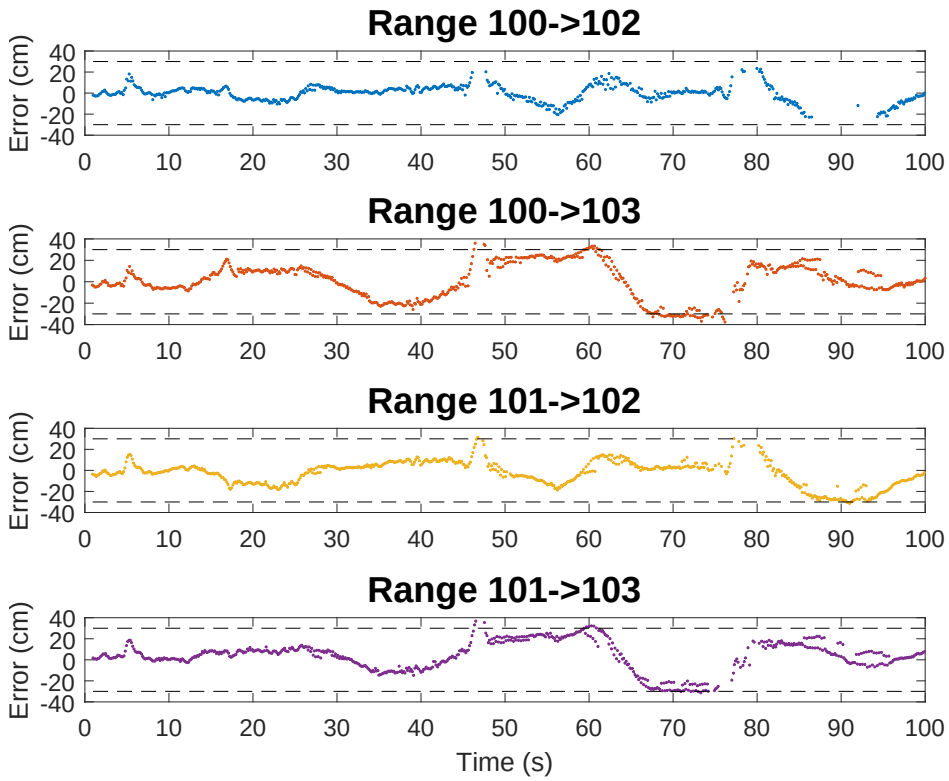


Figure 2.22: P440 Dynamic Figure-8 Ranging Errors with $3\sigma = 30$cm Bounds

largely erroneous measurement which could severely damage relative positioning estimates. An example of one set of ranges before outlier rejection is shown in Figure 2.23. These measurements were removed in post-processing before calculation of the errors; however, in real-time runs, an online rejection criteria must be defined. Such a method is implemented in the estimation algorithms and will be described along with the algorithms in future sections.

Even though these experimental errors are substantially larger than the indoor static test, the previous works utilizing Decawave TREK1000 UWBs frequently experienced errors up to half a meter during dynamic runs with standard deviations over 20 centimeters [21]. The TimeDomain P440s implemented here present a hardware improvement for this relative positioning problem. Previous results from prior work will be recreated with this new hardware for a fair comparison. Now that the technical foundations have been built, the next chapter will derive the relative positioning algorithms.

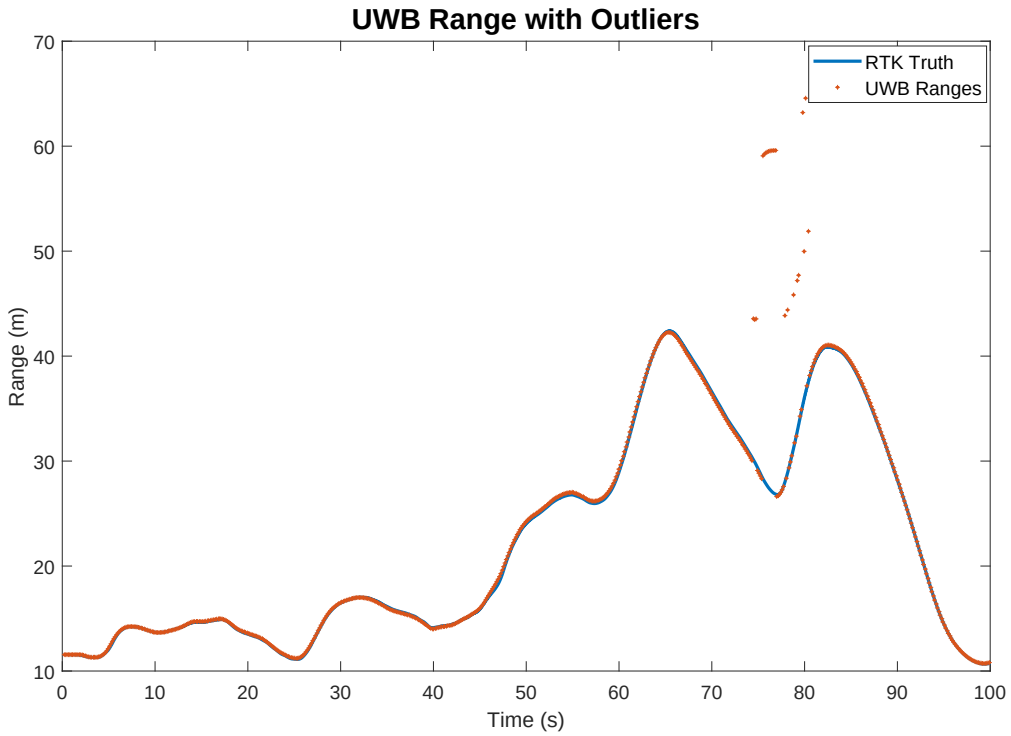


Figure 2.23: UWB Ranges 101 → 103 Dynamic Figure-8 with Outliers

Chapter 3

Non-Cooperative Relative Positioning Algorithms

The non-cooperative algorithms will be described in a two-vehicle network with the vehicle being estimated denoted the "leader" and the vehicle performing the tracking denoted the "follower." This represents a common use case for autonomous vehicle relative position tracking and is primarily used here for clarity. Figure 3.1 displays the states necessary to describe the relative pose over time of two locally maneuvering ground vehicles: a two component relative position vector (RPV), $\vec{R}_F$, and the relative heading between the two vehicles, $\psi_{L|F}$.

3.1 Geometric Relative Position Solution

With the four available ranges between each vehicle, the simplest method to solve the relative position is through direct geometric relationships. Figure 3.2 shows the required ranges and angles to calculate for this solution method.

The interior angles from the vectors formed from base UWB 0 to UWBs 2 and 3 are α and β, respectively. They each may be calculated from the law of cosines by:

$$\alpha = \arccos\left(\frac{r_{02}^2 + a^2 - r_{12}^2}{2ar_{02}}\right) \tag{3.1}$$

$$\beta = \arccos\left(\frac{r_{03}^2 + a^2 - r_{13}^2}{2ar_{03}}\right) \tag{3.2}$$

These angles may then be used to construct two position vectors in the follower's frame between base 0 and the two leader nodes: $\vec{r}_{02}$ and $\vec{r}_{03}$. These vectors may be shifted to

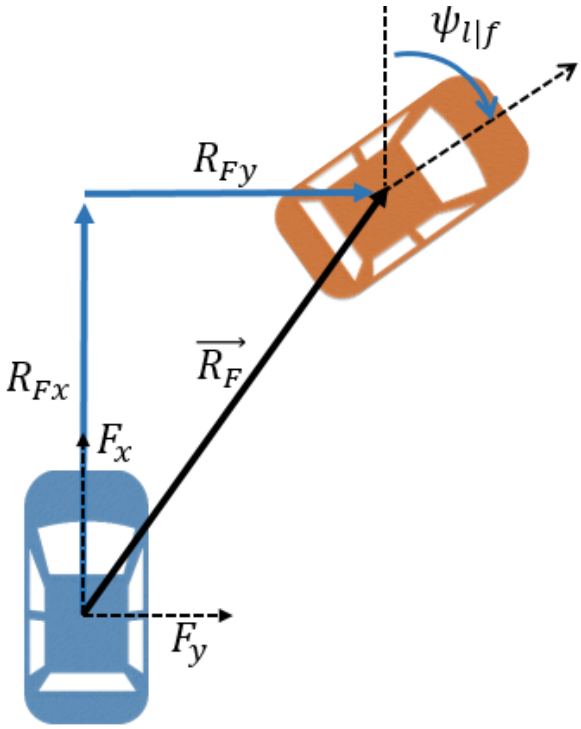


Figure 3.1: 2D Components of Relative Pose

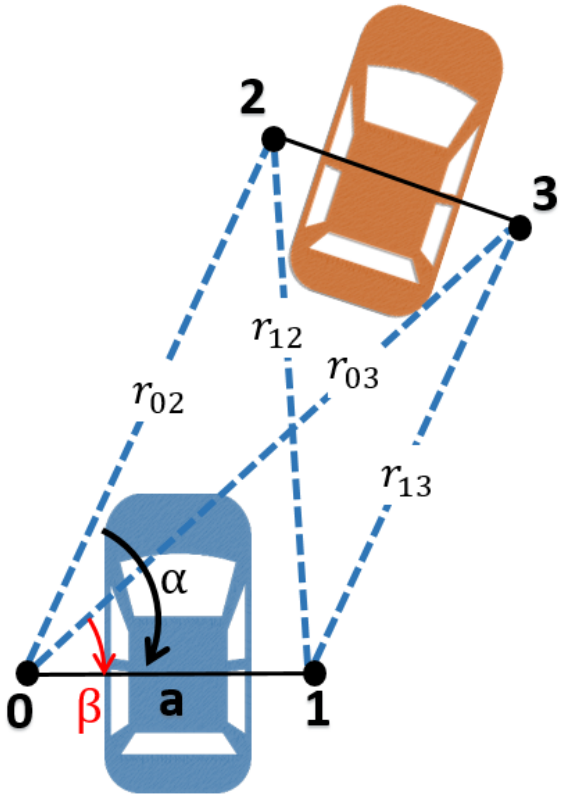


Figure 3.2: Geometric Solution Method

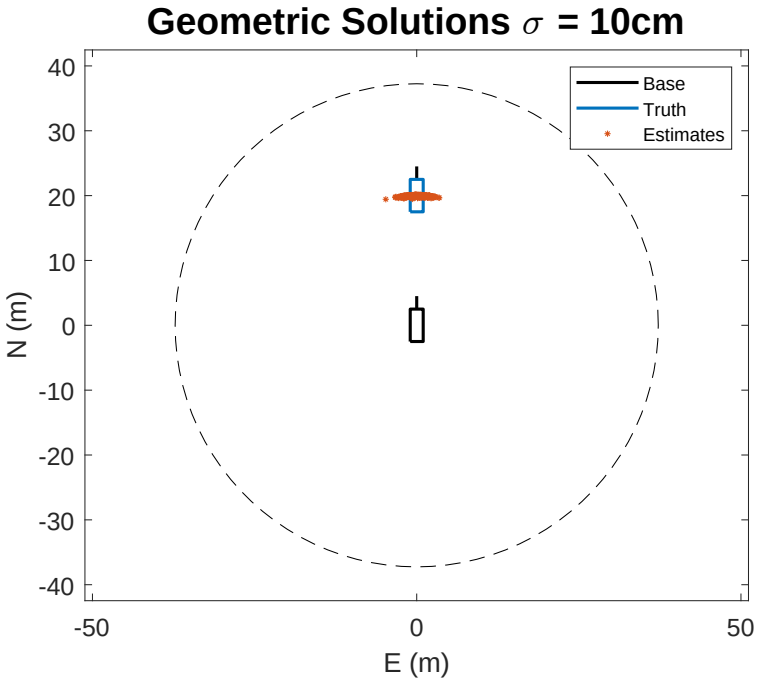


Figure 3.3: 1000 Geometric Solutions at 0° Relative Bearing

instead originate from the middle of the base on the follower.

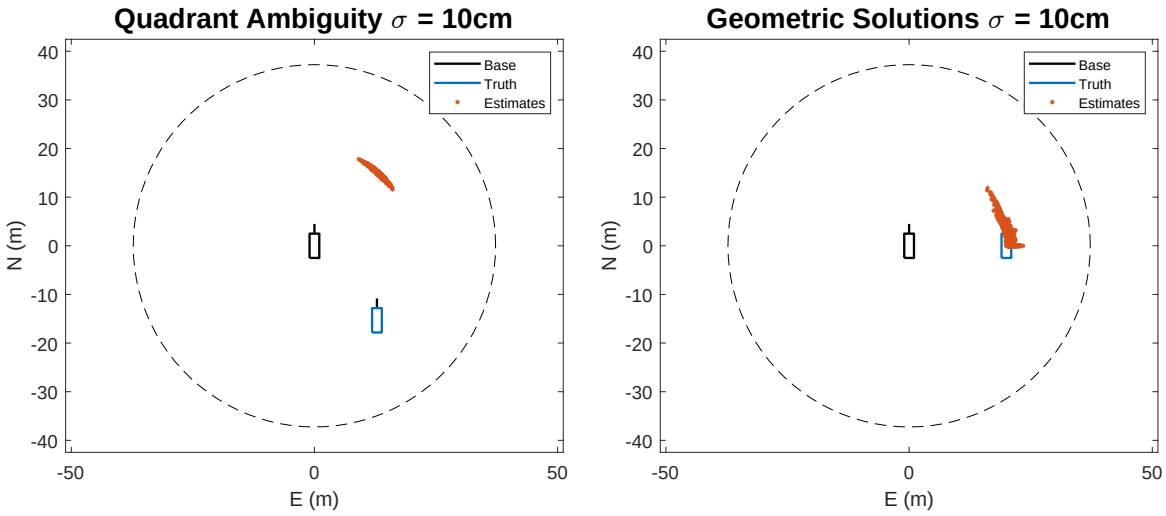
$$\vec{r}_{a2}^{F} = \begin{bmatrix} r_{02}\sin(\alpha) \\ r_{02}\cos(\alpha) - \frac{a}{2} \end{bmatrix} \tag{3.3}$$

$$\vec{r}_{a3}^{F} = \begin{bmatrix} r_{03}\sin(\beta) \\ r_{03}\cos(\beta) - \frac{a}{2} \end{bmatrix} \tag{3.4}$$

The relative position vector between the center of the UWB baseline on the vehicles can then be considered the average of these two vectors:

$$\vec{R}^{F} = \frac{\vec{r}_{a2}^{F} + \vec{r}_{a3}^{F}}{2} \tag{3.5}$$

This solution is valid and accurate in the presence of accurate, time-synced ranges. Simulated geometric results with the nominal UWB errors described in Section 2.3.3 are shown in Figure 3.3.



(a) Geometric Solution Quadrant Ambiguity

(b) Geometric Solutions at 90° Relative Bearing

Figure 3.4: 1000 Geometric Solutions Showing Quadrant Ambiguity Faults

A major drawback of this method is that the geometric relationships are unable to calculate solutions in the 3$^{\text{rd}}$ and 4$^{\text{th}}$ quadrant (behind the follower vehicle). This results in larger and single-sided solution errors when the relative bearing is approaching these quadrants in addition to complete inaccuracies when in these unobservable positions. This issue is depicted in Figure 3.4.

3.2 Least Squares Relative Positioning

The geometric solution has clear quadrant ambiguities and lacks a method to account for the accuracy of the measurements because each range is assumed perfectly accurate. A non-linear least squares technique may be implemented instead which iterates upon the solution from a manually selected initialization. [22] began with this method as their initial solution. The non-linear least squares algorithm employs the Gauss-Newton iterative method to reach a converging solution while considering the relative accuracy of each measurement. The format is similar to the one used in the calibration of the P440 UWBs. First, the nonlinear measurement model which maps the states to the measurements needs to be

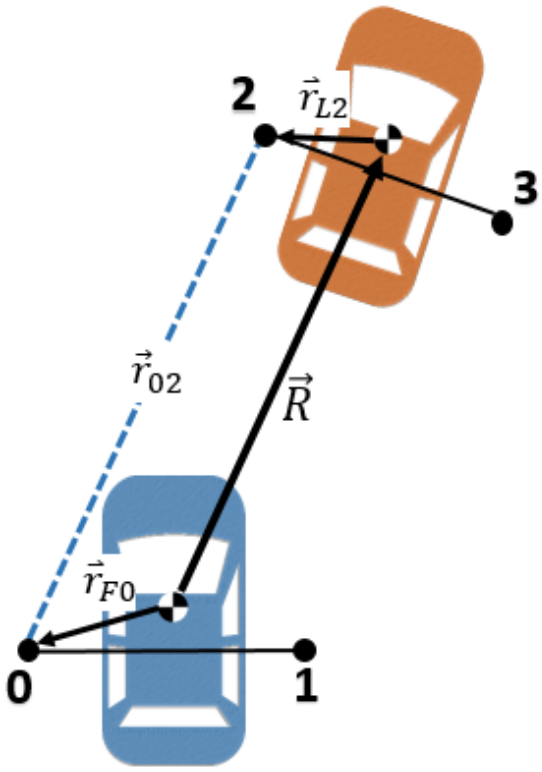


Figure 3.5: UWB Measurement Model Depiction

formed. Figure 3.5 depicts the relationship between a single range measurement and the relative position vector.

In order to achieve enough geometric diversity to calculate the relative pose, the UWBs must be offset from each vehicle's COG; therefore, the positions of the UWBs in their respective vehicle frames, r_{F0} and r_{L2}, must be known *a priori*. The UWBs measure the magnitude of vector between them, $\vec{r}_{02}$. The relationship between $\vec{r}_{02}$ and the position vector of interest is given by:

$$\vec{r}_{02}^{F} = \vec{R}^{F} + C_{L}^{F}\vec{r}_{L2}^{L} - \vec{r}_{F0}^{F} \tag{3.6}$$

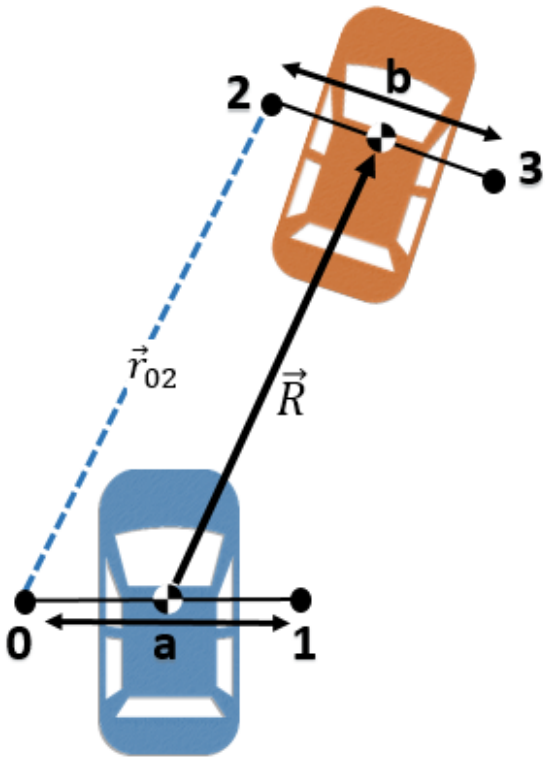


Figure 3.6: UWB Measurement Model Assuming UWB Baseline Intercepts COG

The nonlinear measurement model is the norm of this vector. For an arbitrary UWB pair, γ_1 and γ_2, this measurement model is:

$$r = h(\vec{x}) = ||\vec{R}^F + C_L^F \vec{r}_{L\gamma_1}^L - \vec{r}_{F\gamma_2}^F|| \tag{3.7}$$

In this thesis, the RPV is estimated between the baseline centers of the UWB pairs on each vehicle. This implicitly assumes that the UWBs are spaced along the COG and have no offset from the body's Y-axis (i.e. their offset from the vehicle's COG is only in the Y direction). This allows for simplification of the UWB position vectors with given baselines between the UWBs on the follower and leader as a and b, respectively, as seen in Figure 3.6

This assumption simplifies the derivation, but is not required in a true implementation. So long as the body frame vector is known, the UWB measurements may be related to any point on the body. The body frame vectors for all the UWBs under this assumption are

then:

$$\vec{r}^{F}_{F_{0,1}} = \begin{bmatrix} F_x \\ \pm F_y \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{a}{2} \end{bmatrix}, \begin{bmatrix} 0 \\ \frac{a}{2} \end{bmatrix} \tag{3.8}$$

$$\vec{r}^{L}_{L_{2,3}} = \begin{bmatrix} L_x \\ \pm L_y \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{b}{2} \end{bmatrix}, \begin{bmatrix} 0 \\ \frac{b}{2} \end{bmatrix} \tag{3.9}$$

This simplifies the range measurement model from:

$$r_{02} = \text{norm}\left(\begin{bmatrix} R_x + \cos(\psi_{L|F})L_x - \sin(\psi_{L|F})L_y - F_x \\ R_y + \sin(\psi_{L|F})L_x + \cos(\psi_{L|F})L_y - F_y \end{bmatrix} \right) \tag{3.10}$$

to:

$$r_{02} = \text{norm}\left(\begin{bmatrix} R_x + \frac{b}{2}\sin(\psi_{L|F}) \\ R_y - \frac{b}{2}\cos(\psi_{L|F}) + \frac{a}{2} \end{bmatrix} \right) \tag{3.11}$$

This measurement model is fundamental to not only the least squares algorithm but to all algorithms henceforth in this thesis. Utilizing the measurement model for comparison to the generated measurements yields the residuals to be minimized:

$$\hat{z} = \tilde{y} - h(\hat{x}) \tag{3.12}$$

A linearized model of this measurement model is required for calculating the gain used in the state iteration. It is found by taking the Jacobian of this model with respect to the

three pose states: R_x, R_y, and $\psi_{L|F}$. The linearized model is seen in Equation 3.13.

$$H = \frac{\partial h(\vec{x})}{\partial \vec{x}} = \begin{bmatrix} \frac{\partial r_{02}}{\partial R_x} & \frac{\partial r_{02}}{\partial R_y} & \frac{\partial r_{02}}{\partial \psi_{L|F}} \\ \frac{\partial r_{03}}{\partial R_x} & \frac{\partial r_{03}}{\partial R_y} & \frac{\partial r_{03}}{\partial \psi_{L|F}} \\ \frac{\partial r_{12}}{\partial R_x} & \frac{\partial r_{12}}{\partial R_y} & \frac{\partial r_{12}}{\partial \psi_{L|F}} \\ \frac{\partial r_{13}}{\partial R_x} & \frac{\partial r_{13}}{\partial R_y} & \frac{\partial r_{13}}{\partial \psi_{L|F}} \end{bmatrix} \tag{3.13}$$

The general form of the partial derivatives are the same for each range. The only differences are the signs of a and b depending on which UWBs the measurements were made between. For the range estimate $\hat{r}_{02}$, a single row of the linearized measurement matrix, H, is constructed by Equations 3.14–3.16.

$$\frac{\partial \hat{r}_{02}}{\partial R_x} = \frac{R_x + \frac{b}{2}\cos(\psi_{L|F}) - \frac{a}{2}}{\hat{r}_{02}} \tag{3.14}$$

$$\frac{\partial \hat{r}_{02}}{\partial R_y} = \frac{R_y + \frac{b}{2}\sin(\psi_{L|F})}{\hat{r}_{02}} \tag{3.15}$$

$$\frac{\partial \hat{r}_{02}}{\partial \psi_{L|F}} = \frac{\frac{b}{2}R_y\cos(\psi_{L|F}) - \frac{b}{2}R_x\sin(\psi_{L|F}) + \frac{ab}{4}\sin(\psi_{L|F})}{\hat{r}_{02}} \tag{3.16}$$

With the H matrix constructed, the last remaining component is the relative weighting of the measurements based on their expected variance. This is done through a measurement covariance matrix, R, which is diagonal if the measurements are independent of each other. In this implementation, each measurement is assumed to be uncorrelated with similar errors ($1\sigma = 10$ cm), so R is a diagonal matrix with each element equivalent as seen in Equation 3.17.

$$R = \begin{bmatrix} \sigma^2_{r_{02}} & 0 & 0 & 0 \\ 0 & \sigma^2_{r_{03}} & 0 & 0 \\ 0 & 0 & \sigma^2_{r_{12}} & 0 \\ 0 & 0 & 0 & \sigma^2_{r_{13}} \end{bmatrix} = \begin{bmatrix} .1^2 & 0 & 0 & 0 \\ 0 & .1^2 & 0 & 0 \\ 0 & 0 & .1^2 & 0 \\ 0 & 0 & 0 & .1^2 \end{bmatrix} \tag{3.17}$$

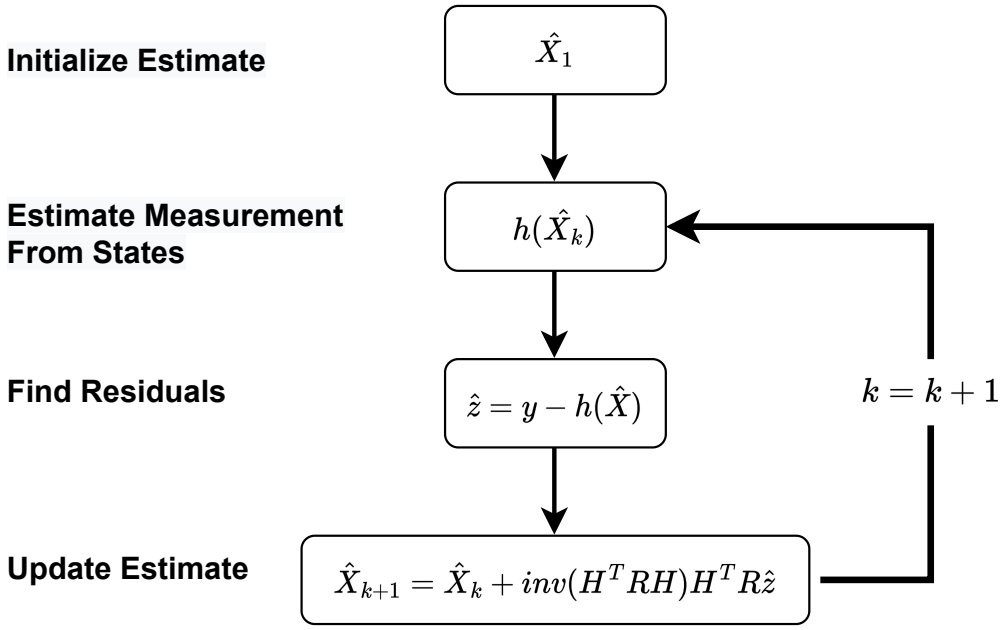


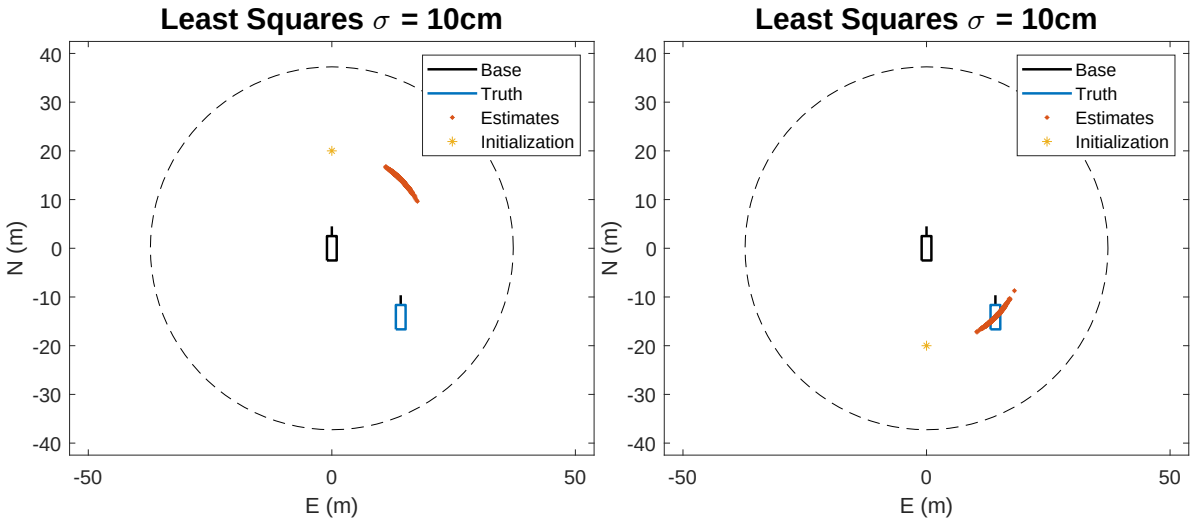
Figure 3.7: Nonlinear Least Squares Algorithm

Since all measurements have the same noise characteristics in this case, each is weighted equally and the result is equivalent to omitting R from the algorithm. The iteration on the state estimate is then performed by:

$$\hat{X}_{k+1} = \hat{X}_k + (H^T R H)^{-1} H^T R \hat{z} \tag{3.18}$$

A summary of this nonlinear least squares method is shown in Figure 3.7. This process is repeated until a sufficient convergence criteria is met or the maximum allotted iterations have been performed.

Upon convergence, the covariance matrix representing the estimated error in each state may be extracted from the measurement matrix at the final step. This is a symmetric matrix with the diagonal representing the main values of interest: the estimated error variance in each state. The off-diagonal terms represent their cross-correlations. Equation 3.19 shows



(a) Least Squares Improperly Initialized Ahead of Base

(b) Least Squares Properly Initialized Behind Base

Figure 3.8: Importance of Proper Initial Guess to Least Squares Quadrant Ambiguity

the error covariance matrix.

$$\hat{P} = (H^T R H)^{-1} = \begin{bmatrix} \sigma_{R_x}^2 & \rho_{x,y}\sigma_{R_x}\sigma_{R_y} & \rho_{x,\psi}\sigma_{R_x}\sigma_{\psi_{L|F}} \\ \rho_{x,y}\sigma_{R_x}\sigma_{R_y} & \sigma_{R_y}^2 & \rho_{y,\psi}\sigma_{R_y}\sigma_{\psi_{L|F}} \\ \rho_{x,\psi}\sigma_{R_x}\sigma_{\psi_{L|F}} & \rho_{y,\psi}\sigma_{R_y}\sigma_{\psi_{L|F}} & \sigma_{\psi_{L|F}}^2 \end{bmatrix} \tag{3.19}$$

If all the measurements are of equivalent accuracy, a single variance may be applied as a scaling factor rather than including the full R matrix:

$$\hat{P} = \sigma^2 (H^T H)^{-1} \tag{3.20}$$

The least squares solution is an improvement over the geometric method in regards to quadrant ambiguity. Although the ambiguity still exists, it can be overcome more readily. This quadrant ambiguity is due to the presence of multiple non-global minima—the least squares algorithm will regress to whichever is closer to the initial estimate. The quadrant ambiguity can therefore be overcome provided that the initial estimate is closer to the truth than its ambiguous pair, as shown in Figure 3.8.

The above solutions assume all four ranges are available instantaneously. A minimum of 3 unique measurements are required at a given time for the least squares solution to be solvable (for H to be full rank). This is similar to the geometric solution and is one of the main drawbacks of attempting a least squares solution for this relative positioning problem. In Chapter 2, it was described that a network of UWBs provide measurements sequentially, thus all 4 measurements are not available simultaneously. One method of overcoming this would be to hold onto previous measurements until enough have been received to perform the estimation. If the vehicle's are static, this is a valid option and is how all algorithms presented here are initialized. However, as the relative speed or yaw rate increases, this introduces substantial errors. This requires the use of a method which can provide state estimates from the individual measurements as they arrive. Such methods include Kalman Filtering which is explored in the Section 3.4. Before deriving the Kalman Filter algorithms, there are inherent difficulties to the estimation structure which need to be explored.

3.3 Observability Difficulties (DOP)

Beyond the quadrant ambiguity problem, the relative pose solution quality is generally dependent on the relative geometry of the vehicles. This is because different orientations result in H matrices of varying observability. The RPV components of the linearized measurement matrix, H, seen in Equations 3.14 and 3.15 are the unit vectors between the UWBs used in a given measurement. Essentially, the more diverse the unit vectors for the four ranges are, the more observable the states are from the measurements and the more accurate the solutions become. Different orientations lead to varying unit-vector diversity.

This nonlinear least squares process is analogous to how GPS solutions are formed. In common GPS solutions, nonlinear least squares is utilized with a measurement matrix consisting of the estimated unit vectors between the receiver and each satellite [51]. This leads to the measurement matrix often being referred to as the *geometry matrix*. Quantifying the effect of satellite geometry on the resulting position solution is possible through a dilution of

precision (DOP) analysis. Although this method is most frequently seen in GPS implementations, it is also applies to any solution which depend on geometry [52, 53]. DOP analysis will be utilized here to depict problematic orientations for UWB relative positioning. Similar results were shown in [21], but are re-created here.

As mentioned, the measurement matrix, H, provides the geometric relationships for the vehicles' current positions and is the critical component of the estimation accuracy. DOP analysis separates errors due to measurement quality from errors due to geometry. Therefore, it is calculated by excluding the measurement variance from covariance calculation in Equation 3.19. This is seen in Equation 3.21.

$$DOP = (H^T H)^{-1} = \begin{bmatrix} a & \# & \# \\ \# & b & \# \\ \# & \# & c \end{bmatrix} \tag{3.21}$$

The DOP for each state is then the square root of the corresponding diagonal element:

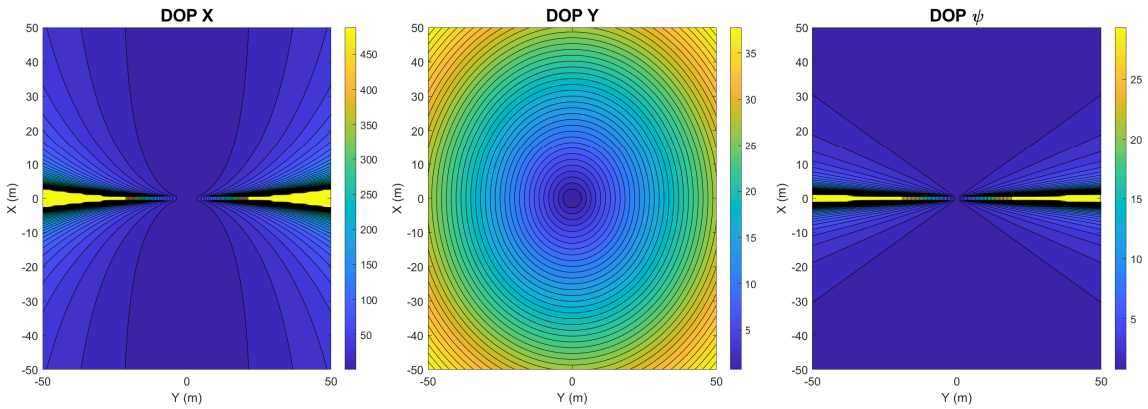
$$DOP_{R_x} = \sqrt{a} \tag{3.22}$$

$$DOP_{R_y} = \sqrt{b} \tag{3.23}$$

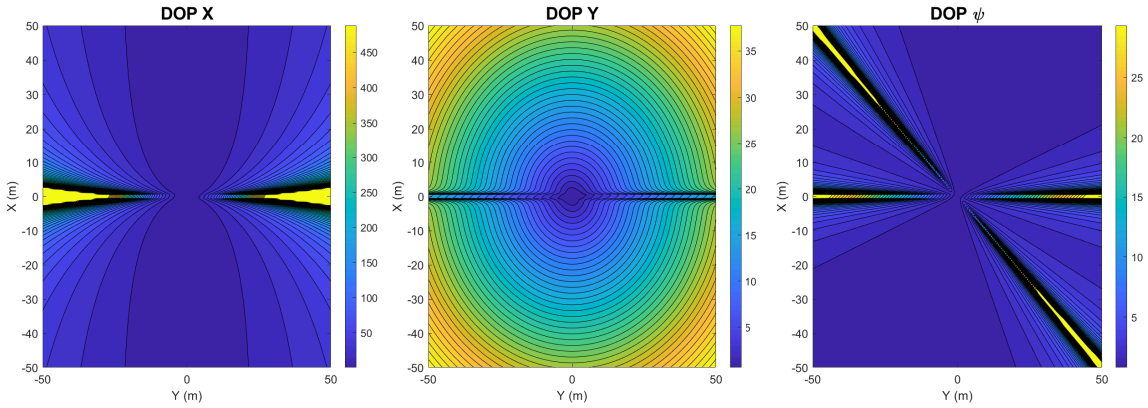
$$DOP_{\psi_{L|F}} = \sqrt{c} \tag{3.24}$$

The DOPs were calculated assuming all four measurements were available and the UWBs were separated by 72 in (6 ft) across each vehicle's roof. The DOP was found for all locations within 50 meters of the vehicle with varying relative headings. The results are summarized in Figure 3.9

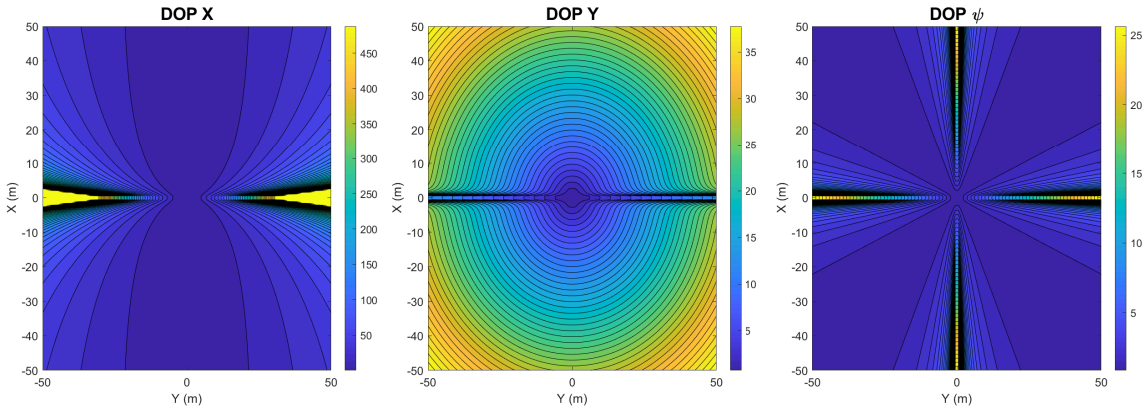
The DOP plots in Figure 3.9 provide valuable insight into the expected observability of the RPV given a relative orientation. The dilution of precision in the Y-axis of the follower's body frame, DOP_y, is more or less consistent regardless of orientation, but is highly dependent on the range. The accuracy in the X direction is seen to be higher in



(a) Dilution of Precision with 0° Relative Heading

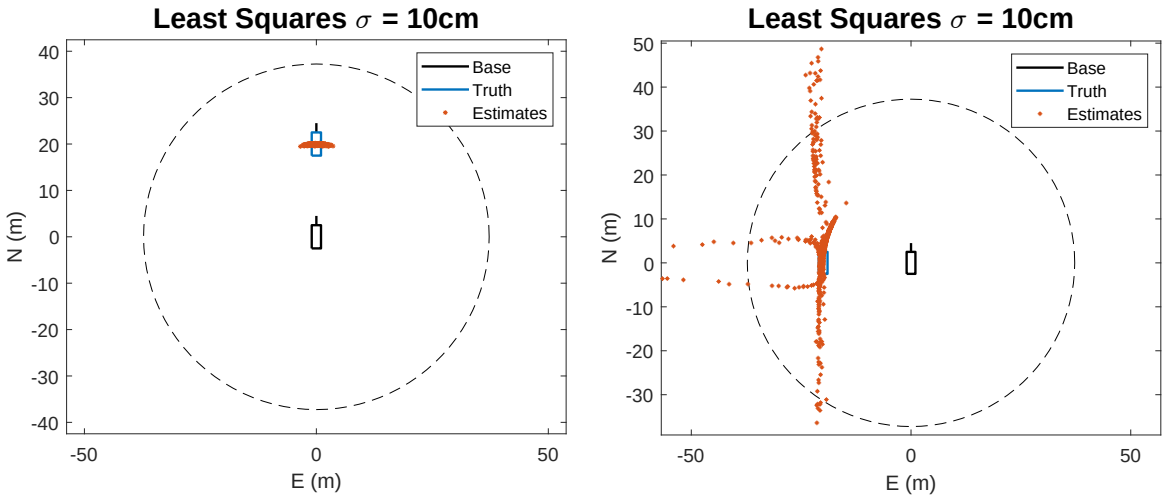


(b) Dilution of Precision with 45° Relative Heading



(c) Dilution of Precision with 90° Relative Heading

Figure 3.9: Dilution of Precision with Varying Relative Poses



(a) LS Solution for Low-DOP Relative Pose (b) LS Solution for High-DOP Relative Pose

Figure 3.10: Least Squares Solution Quality for High vs Low DOP Geometries

front of and behind the estimating vehicle, but there are large areas of uncertainty when the vehicles are side by side. This difference is due to the UWBs being situated along the Y-axes of the vehicles. The DOP_x and DOP_y characteristics would swap if the UWBs were aligned along the longitudinal track of the vehicles instead. An example comparison of an area with high and low DOP is depicted in Figure 3.10. These locations correspond to the DOP plots in Figure 3.9a with a relative heading of 0. Notice how much worse the solutions are in Figure 3.10b with the same noise on the range measurements. Note that in these results both solutions were initialized within the proper quadrant to avoid the ambiguity problem.

Assuming these poorly-observable relative geometries will not arise in locally maneuvering vehicles is invalid. Therefore, methods to propagate the previous states through these areas is required to attain a robust relative positioning solution.

3.4 Extended Kalman Filter (EKF) Relative Positioning

Kalman Filtering was first developed by Rudolf E. Kalman in 1960 [54] and has seen extensive adoption in numerous fields. It forms the basis for the majority of localization algorithms and was even used in the NASA Apollo Missions [55]. In essence, the Kalman Filter is a recursive algorithm which processes a stream of measurements to update state

estimates similar to recursive least squares. However, Kalman Filter techniques include an additional step: a state propagation *time update* which utilizes a system model for updating the estimates and their covariances between measurements. In a linear system with properly modeled statistics, the result is an optimal balancing of uncertainty in the system model and the measurement noise. Many sub-optimal derivatives of this algorithm have been created for application to nonlinear systems. The most common is known as the Extended Kalman Filter (EKF), which is presented below. For detailed analyses or intuitive applications of the Kalman Filter, many sources exist including [56, 57]. Here, the discretized form of the EKF will be briefly described.

An initialization of the state and state covariances needs to be formed: $\vec{x}_1$ and P_1, respectively. Then, the two steps of the filter, the time and measurement updates, can be performed from these initial estimates.

The measurement update step is essentially equivalent to a single update of a recursive least squares estimator. It consists of first calculating a *Kalman gain*, K, based on the linearized measurement matrix, H, along with the uncertainties in the states and measurements: P and R. The superscripts, $-$ and $+$, denote variables before or after correction, respectively.

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \tag{3.25}$$

For the UWB relative positioning problem, H is a single row of Equation 3.13 constructed according to the corresponding range measurement. Each state is then corrected through the measurement residuals with the resulting gain matrix:

$$\vec{x}_k^+ = \vec{x}_k^- + K_k(\tilde{y} - h(\vec{x}_k^-)) \tag{3.26}$$

Lastly, the state covariance is corrected to show the change in confidence due to this newly-processed measurement:

$$P_k^+ = (I - K_k H)\, P_k^- \tag{3.27}$$

The time update is then utilized to propagate the state and corresponding uncertainty from time k to time $k+1$. For a nonlinear system, the nonlinear model is employed to update the states with the previous values and any inputs to the system, while a linearized form, Φ, is necessary to update the covariance matrix. This is summarized in Equations 3.28 and 3.29.

$$\vec{x}_{k+1}^- = f(\vec{x}_k^+, \vec{u}) \tag{3.28}$$

$$P_{k+1}^- = \Phi P_k^+ \Phi^T + Q_d \tag{3.29}$$

The matrix Q is a discretized process noise matrix which is often partially based on known sensor or model errors and then hand-tuned. The composition of this matrix varies for the different implementations presented in the following sections. This EKF process is then repeated with measurement updates performed when measurements are available and time updates performed intermediately. The integration step length of the filter is either pre-determined or set by frequency of input measurements. The shorter this step is, the more accurate the linearization will be. Figure 3.11 provides an overview of the EKF algorithm process.

3.4.1 Constant Velocity Extended Kalman Filter

The first application of Kalman Filter-based dynamic RPV estimation with multiple UWBs was presented in [21]. This work provided a detailed overview of the problem with least squares techniques and designed a constant velocity model to handle the least squares estimation issues described previously. Constant velocity dynamic models are commonly

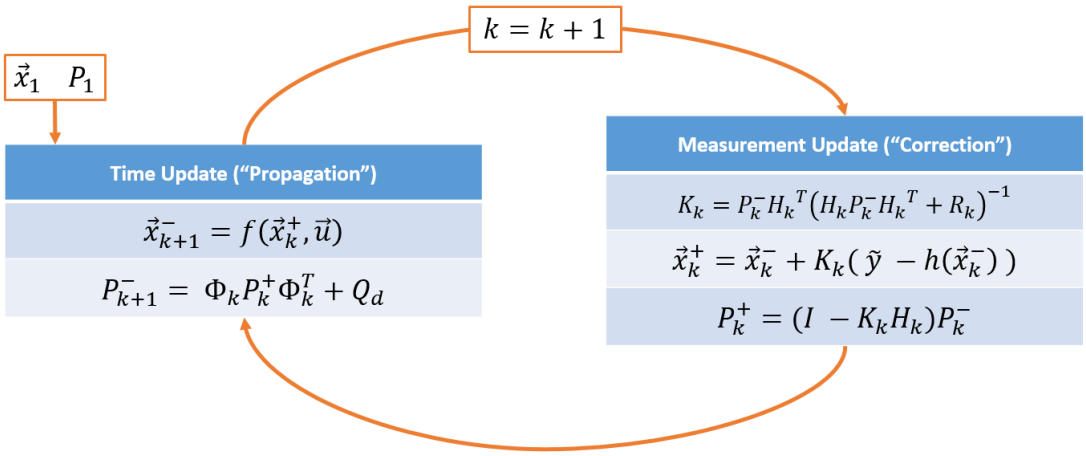


Figure 3.11: Extended Kalman Filter Algorithm

used in estimation algorithms to smooth measurements, especially in platforms where odometry is not available or the dynamics may be difficult to model [58–60]. In [21], the state vector consisted of the three pose states and their derivatives. Although direct measurements of the derivatives are not available, the coupling to the change in position through the dynamic model allows for the EKF algorithm to approximate the relative velocity. With this estimate, the implementation aims to propagate the state vector through measurement outages and unobservable areas. [21] modified the general constant velocity model to govern the relative yaw-rate by a decaying exponential function with a time constant, τ. This is a purely linear model, so the time update portion of the EKF does not require linearization for the covariance propagation. The prior work's time update is illustrated in Equation 3.30 with modifications to match the notation used in this thesis.

$$\begin{bmatrix} R_x \\ R_y \\ \psi_{LF} \\ \dot{R}_x \\ \dot{R}_y \\ \dot{\psi}_{LF} \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & e^{-\frac{\Delta t}{\tau}} \end{bmatrix} \begin{bmatrix} R_x \\ R_y \\ \psi_{LF} \\ \dot{R}_x \\ \dot{R}_y \\ \dot{\psi}_{LF} \end{bmatrix}_k \tag{3.30}$$

In a constant velocity implementation, the acceleration is assumed to be 0 mean. This is rarely the case, so the process noise matrix, Q, may be tuned in relation to the expected variance in the acceleration. Higher values of this variance causes the covariance estimate to grow at a faster rate, and thus the constant velocity model is trusted less as a result. Consequentially, more trust is placed in the measurements to deviate the state estimates from the dynamic model. The author of [21], Ghanem, provided the tuning parameters utilized for their results. This will be utilized in all implementations of this method during comparison to the new algorithms. The discretized form is shown in Equation 3.31.

$$Q_{CV} = \begin{bmatrix} 0.01 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.005 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.002 \end{bmatrix} \Delta t \tag{3.31}$$

As was mentioned in Chapter 3, the UWBs are subject to measurement outages characterized by significant outliers. [32] outlines a method based on innovation filtering to reject these measurements in real time. An innovation test for a single UWB range measurement is found by:

$$\chi = \sqrt{\delta z^T C^{-1} \delta z} \tag{3.32}$$

where:

$$C = HPH^T + R \tag{3.33}$$

$$\delta z = \tilde{y} - h(\vec{x}) \tag{3.34}$$

Comparing this to a pre-determined threshold of χ allows for exclusion of measurements which are erroneous. This is an implementation of a chi-squared test. With perfect tuning of the filter's statistics and truly Gaussian measurement errors, setting the threshold to 3 results in accepting 99.7% of the range measurements and rejecting the clear outliers. This technique is implemented in all filters presented in this thesis.

This prior work in [21] forms the baseline comparison for all of the estimation algorithms presented in this thesis. The proposed methods in this thesis aim to improve upon the algorithm in [21]. Results from the proposed algorithms are compared to results from [21] in the following chapters.

3.4.2 Vehicle Dynamics Extended Kalman Filter (VehDynEKF)

A modification to the algorithm in [21] was proposed and presented at the 2022 Modeling, Estimation, and Control Conference, (MECC) [23]. The goal of the work was to improve the relative positioning results by shifting away from a constant velocity state propagation to one governed by vehicle dynamic models. The measurement model, and thus the measurement update of the EKF, remains the same.

Rather than assumed constant, the derivative of the relative position and heading between the vehicles can be described by the velocity and rotation rates of the two vehicles as in Equations 3.35 and 3.36.

$$\dot{\vec{R}} = C_L^F \begin{bmatrix} v_{Lx} \\ v_{Ly} \end{bmatrix} - \begin{bmatrix} v_{Fx} \\ v_{Fy} \end{bmatrix} + \omega_F \times \begin{bmatrix} R_x \\ R_y \end{bmatrix} \tag{3.35}$$

$$\dot{\psi}_{L|F} = \omega_L - \omega_F \tag{3.36}$$

The time derivative of the heading of the two vehicles is simply the difference in the yaw-rates of the two vehicles, ω_F and ω_L, since their body-frame Z-axes remain aligned under the 2D assumption. The final term in Equation 3.35 represents apparent motion of the leader

with respect to the follower due to rotation of the follower frame. Since the yaw-rate of the follower frame is about the body frame Z-axis, the cross product term may be simplified:

$$\dot{\vec{R}} = C_L^F \begin{bmatrix} v_{Lx} \\ v_{Ly} \end{bmatrix} - \begin{bmatrix} v_{Fx} \\ v_{Fy} \end{bmatrix} + \omega_F \begin{bmatrix} R_y \\ -R_x \end{bmatrix} \tag{3.37}$$

Note that the rotation matrix C_L^F is an application of Equation 2.15. Applying this equation with the relative heading of the vehicles yields:

$$C_L^F = \begin{bmatrix} \cos(\psi_{L|F}) & -\sin(\psi_{L|F}) \\ \sin(\psi_{L|F}) & \cos(\psi_{L|F}) \end{bmatrix} \tag{3.38}$$

The constant velocity model does not consider these separate sources of relative motion and attempts to lump them together into two constant terms, $\dot{R}_x$ and $\dot{R}_y$. This vehicle dynamics-based EKF has additional constraints on the propagation through this relative motion model. From the equations presented above, the estimate of the relative position over time can improved by employing the vehicles' velocities and yaw-rates. The follower's velocity, $\vec{v}_F$, can be estimated through onboard odometry methods as described in Chapter 2. Here, the no side-slip assumption (utilized in the results in Figure 2.8) is employed so that only two sensors are required for the follower's odometry: a wheel encoder and a yaw-gyro. This means the lateral velocity of the follower, v_{Fy}, is considered to be 0. The process noise matrix can incorporate additional uncertainty in the propagation due to this assumption and will be described shortly.

If there is no communication between the vehicles, the estimating/following vehicle lacks knowledge of two necessary parameters: the lead vehicle's velocity ($\vec{v}_L$) and yaw rate(ω_L). Instead of estimating these states directly, the kinematic bicycle model can be employed to relate them and reduce the number of required states. Recalling Equation 2.32, the yaw rate

may be shown as:

$$\omega_L = \frac{|\vec{V}| \tan(\delta) \cos\left(\frac{l_r \tan(\delta)}{L}\right)}{L} \tag{3.39}$$

Utilizing the same no side-slip assumption applied to the follower vehicle, the velocity becomes purely longitudinal, and the yaw rate representation may be simplified to:

$$\omega_L = \frac{v_{Lx} \tan(\delta)}{L} \tag{3.40}$$

The wheelbase, L, varies between vehicles, and it is unreasonable to assume that neighboring vehicles know its value. To avoid estimating the wheelbase and in order to make the estimator vehicle-agnostic, the wheel deflection and base length may be compacted into an *effective steer angle*:

$$d_L = \frac{\tan(\delta)}{L} \tag{3.41}$$

The dynamics of the lead vehicle have thus been simplified to only two states: v_{Lx} and d_L. This also has the benefit of closely linking the yaw rate and velocity of the lead vehicle. Along with odometry measurements on the follower, proper estimation of these additional states can provide more accurate state propagation. This improved time update should increase the resiliency of the estimator in the presence of measurement outages and high-DOP relative geometry. A summary of the nonlinear state propagation equations with the appropriate assumptions made here is shown in Equations 3.42 and 3.43.

$$\dot{\vec{R}} = C_L^F \begin{bmatrix} v_{Lx} \\ 0 \end{bmatrix} - \begin{bmatrix} \tilde{v}_{Fx} \\ 0 \end{bmatrix} + \tilde{\omega}_F \begin{bmatrix} R_y \\ -R_x \end{bmatrix} \tag{3.42}$$

$$\dot{\psi}_{L|F} = v_{Lx} d_L - \tilde{\omega}_F \tag{3.43}$$

The tilde notation denotes a measurement of the value exists (e.g. $\tilde{\omega}_F$ is a measurement from the follower yaw-rate gyroscope). With this relative dynamics model and vehicle dynamics constraints, the vehicle dynamics EKF can be constructed. The state vector consists of the three states required to define the relative pose along with the lead vehicle states derived from the bicycle model:

$$\vec{X} = \begin{bmatrix} R_x \\ R_y \\ \psi_{L|F} \\ v_{Lx} \\ d_L \end{bmatrix} \tag{3.44}$$

The required linearization is found from the partial derivative of each state's update equation with respect to the state vector. These first order partial derivatives form the Jacobian, A. After discretization, the resulting state-transition matrix is shown in Equation 3.45.

$$\Phi = (I + A\Delta t) = \begin{bmatrix} 1 & \omega_f & -v_{Lx}\sin(\psi_{L|F}) & \cos(\psi_{L|F}) & 0 \\ -\omega_f & 1 & v_{Lx}\cos(\psi_{L|F}) & \sin(\psi_{L|F}) & 0 \\ 0 & 0 & 1 & d_L & v_{Lx} \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \Delta t \tag{3.45}$$

Again, two more simplifications were made while applying the bicycle model to this algorithm: the lateral velocities, v_{Fy} and v_{Ly}, are zero. This is rarely the case during vehicle motion, but similar to the constant velocity model, process noise may be added to these values to allow for the estimates to deviate from the model. Also, the lead vehicle's velocity is assumed to be constant, so process noise is modeled on $\dot{v}_L$ as well. These additions along with the sources on the odometry measurements and state process noise yield 6 total sources

of process noise:

$$Q = \text{diag}\left(\begin{bmatrix} \nu_{v_{F_x}}^2 & \nu_{v_{F_y}}^2 & \nu_{\omega_F}^2 & \nu_{\dot{v}_{L_x}}^2 & \nu_{v_{L_y}}^2 & \nu_{\dot{d}_L}^2 \end{bmatrix}\right) \tag{3.46}$$

The variances of these process noise sources are found through a combination of calculation and tuning. For instance, the error in the follower's gyroscope measurement can be estimated from static experimental runs, while the variance in the effective steer angle is tuned heuristically. Summarizing the state-propagation equations with these additions:

$$\dot{\vec{R}} = C_L^F \begin{bmatrix} v_{L_x} \\ 0 + \nu_{v_{L_y}} \end{bmatrix} - \begin{bmatrix} \tilde{v}_{Fx} + \nu_{v_{F_x}} \\ 0 + \nu_{v_{F_y}} \end{bmatrix} + (\tilde{\omega}_F + \nu_{\omega_F}) \begin{bmatrix} R_y \\ -R_x \end{bmatrix} \tag{3.47}$$

$$\dot{\psi}_{L|F} = v_{L_x} d_L - (\tilde{\omega}_F + \nu_{\omega_F}) \tag{3.48}$$

$$\dot{v}_{L_x} = 0 + \nu_{\dot{v}_{L_x}} \tag{3.49}$$

$$\dot{d}_L = 0 + \nu_{\dot{d}_L} \tag{3.50}$$

Since these sources of process noise are all additive, the mapping of these process noise values to the states is linear. The mapping matrix is:

$$B_w = \begin{bmatrix} 1 & 0 & R_y & 0 & -\sin(\psi_{L|F}) & 0 \\ 0 & 1 & -R_x & 0 & \cos(\psi_{L|F}) & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.51}$$

The discretized process noise matrix, Q_d, is found from Equation 3.52, though the resulting matrix is too large to visualize here.

$$Q_d = B_w Q B_w^T \Delta t \tag{3.52}$$

A block diagram summarizing this vehicle dynamics EKF is shown in Figure 3.12.

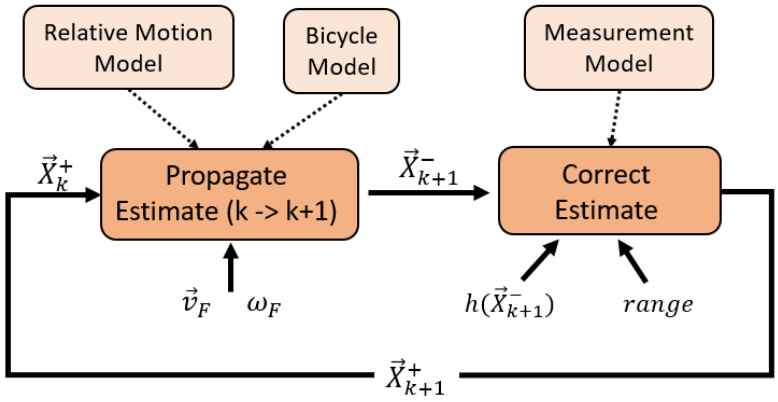


Figure 3.12: Vehicle Dynamic EKF (VehDynEKF) Block Diagram

These two non-cooperative methods are now well-defined and will be compared to one another in simulation and experimentation in the following chapters. The main improvement of this vehicle dynamics model is the ability to better propagate through difficult relative geometries (high DOP) than the constant velocity method. This was shown in the author's previous work [23]. Although further comparisons between methods will be made in the following chapters, this direct comparison is useful.

The full-path estimation of the leader vehicle encircling a static follower is shown in Figure 3.13.

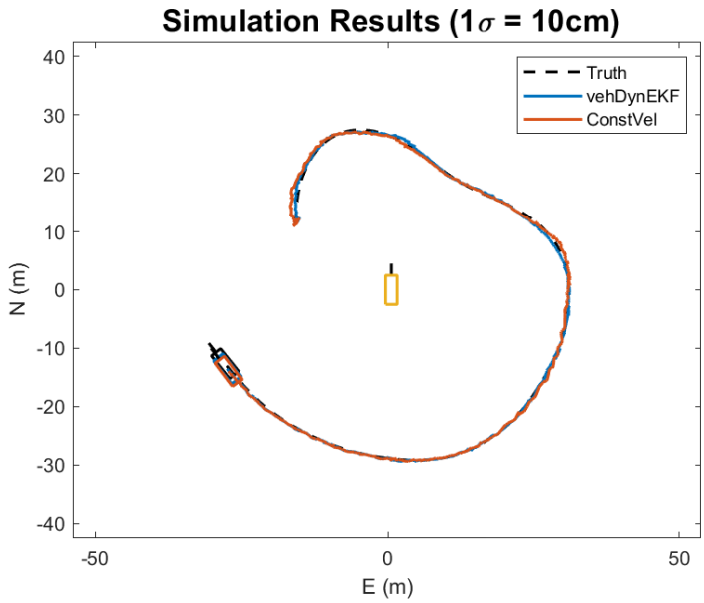
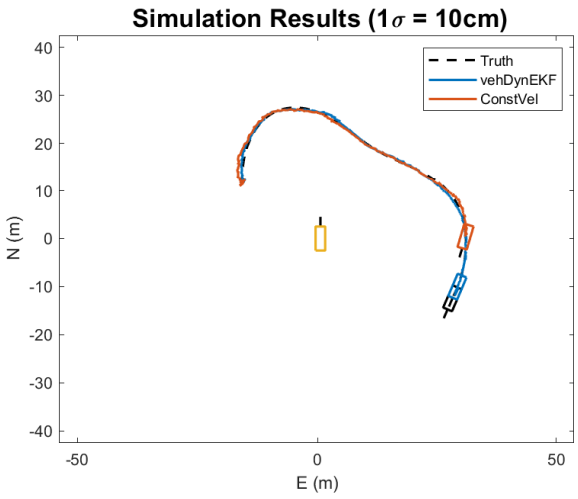
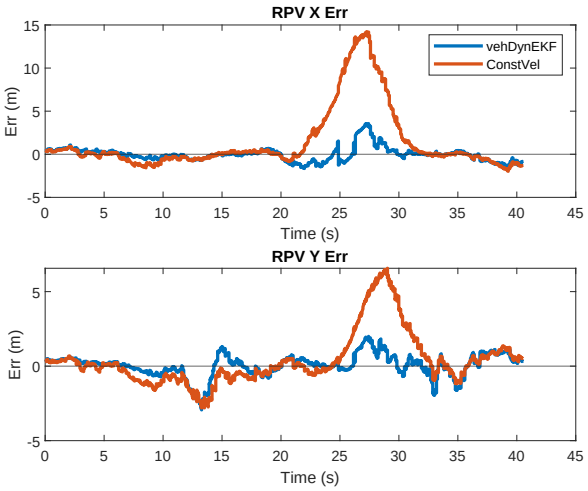


Figure 3.13: Simulated Static-Follower Encirclement Trajectory

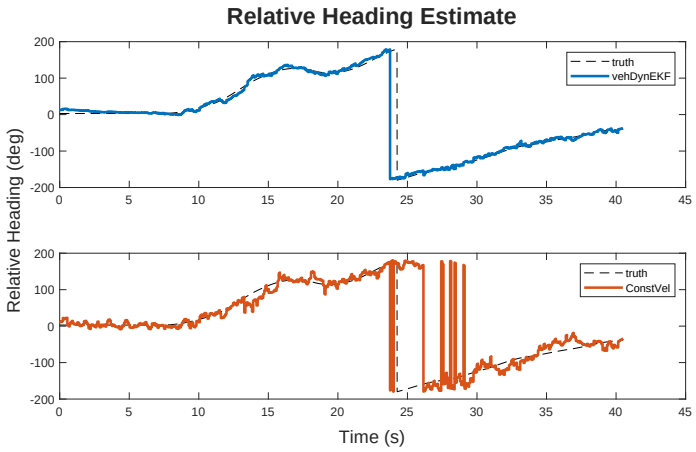
Although the overall results look similar, the constant velocity method significantly lags the estimate through the unobservable region, as seen in Figure 3.14. Both methods have increased errors through the high-DOP region shown in 3.14a, but the improvement of the vehicle dynamics EKF is evident. Although it is arguably less important than the RPV components, the VehDynEKF also performs better in estimating the relative heading as seen in Figure 3.14c. The non-cooperative methods shown in this chapter form the basis for the collaborative extensions derived in the following chapter.



(a) Snapshot of High DOP Section



(b) Relative Position Vector Estimation Error Comparison



(c) Relative Heading Accuracy Comparison

Figure 3.14: Comparison of Non-Cooperative Results for Static Follower Encirclement Test

Chapter 4

Cooperative Relative Positioning Algorithms

As the computing capabilities and connectivity of autonomous vehicles increase, solutions employing information from multiple nodes within a network are increasingly common. This chapter explores cooperative methods for improving the UWB RPV estimation solutions.

4.1 Relative IMU Mechanization

In the majority of collaborative algorithms, inertial measurements are assumed to be available on each member of the network: [12, 61–67]. The most straightforward way to utilize IMUs for relative positioning is to perform a full mechanization for each with relative measurements providing corrections to the IMU errors of the vehicles; however, depending on the dimensionality of the mechanization, the state vector can be quite large. A full 6-DOF mechanization typically consists of a 15-state vector for each vehicle as described in [32]:

$$\vec{X} = \begin{bmatrix} \Psi^i_{ib} & v^i_{ib} & r^i_{ib} & b_a & b_g \end{bmatrix}^T_{15x1} \tag{4.1}$$

Each component above is a 3-dimensional vector corresponding to the attitude, velocity, position, and accel and gyro biases, respectively. Tracking the states of two vehicles would result in a state vector with a dimension of 30. For the 2-dimensional scenario on which this thesis focuses, a simplified state vector in a local NED frame can be constructed:

$$\vec{X} = \begin{bmatrix} \psi & v_N & v_E & r_N & r_E & b_{ax} & b_{ay} & b_{gz} \end{bmatrix}^T_{8x1} \tag{4.2}$$

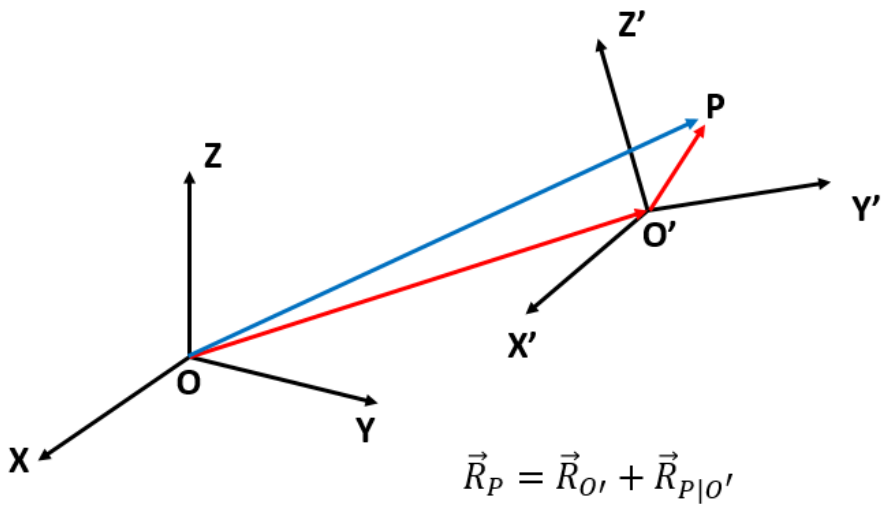


Figure 4.1: Position in Multiple Reference frames (O as static and O' as mobile)

Even with the reduced-state vector, a two-vehicle mechanization would result in 16 states to track. Since relative positioning is the primary goal of this thesis, a relative IMU mechanization is developed. [19] explored a 3D relative mechanization for multiple UAVs with UWBs to achieve promising results. However, the algorithm employed magnetometers to provide an accurate rotation of the IMU measurements into a common, static local frame. The goal here is to perform the estimation in real time in the estimating vehicle's moving frame. Although a 2D relative mechanization in a mobile reference frame is not novel, the author was unable to find a well-related source, thus the derivation is presented here.

As described in Chapter 2, the initial step in IMU mechanization is selecting the resolving coordinate frame. The most logical choice is to perform the mechanization within the follower frame, as this is the most commonly chosen frame for the relative position vector. However, this poses difficulties since the estimating (follower) vehicle's body frame is rotating and translating through time. This yields centripetal, Coriolis, and Euler acceleration terms [32] which account for the apparent motion of the leader due to motion of the follower's frame. Figure 4.1 depicts a particle's position with respect to both an inertial and moving reference frame.

[33] provides a compact formulation for the acceleration of the a body in a moving reference frame with each component well-described. The notation has been modified to fit the specific use case here in Equation 4.3.

$$\vec{a}^{i}_{iL} = \vec{a}^{i}_{iF} + \vec{a}^{F}_{FL} + \dot{\vec{\omega}}^{F}_{iF} \times \vec{r}^{F}_{FL} + \vec{\omega}^{F}_{iF} \times (\vec{\omega}^{F}_{iF} \times \vec{r}^{F}_{FL}) + 2\vec{\omega}^{F}_{iF} \times \vec{v}^{F}_{FL} \tag{4.3}$$

The left-hand side of this equation, $\vec{a}^{i}_{iL}$, is the leader's acceleration with respect to an inertial frame, resolved in an inertial frame. The leader's IMU measures this acceleration directly, but it is resolved in the vehicle's body frame. Similarly, $\vec{a}^{i}_{iF}$ is the body-resolved acceleration for the follower. The yaw rate of the follower's frame is $\vec{\omega}^{F}_{iF}$. It is assumed that a local NED frame, N, may be considered inertial for local trajectories; therefore, the rotation of the follower's 2D frame may be directly measured by the yaw-rate gyro. Lastly, $\vec{r}^{F}_{FL}$ and $\vec{v}^{F}_{FL}$are the relative position vector and relative velocity vector, respectively. The primary variable of interest is the relative acceleration of the leader with respect to the follower's frame, $\vec{a}^{F}_{FL}$, as this can be integrated to the get the relative position over time. The accelerations measured in the leader's frame are then rotated into the follower's frame to combine all terms into a consistent coordinate system.

Rewriting Equation 4.3 to solve for $\vec{a}^{F}_{FL}$ and substituting in the necessary rotation to utilize the leader's IMU measurements yields Equation 4.4.

$$\vec{a}^{F}_{FL} = C^{F}_{L}\vec{a}^{L}_{iL} - \vec{a}^{F}_{iF} - \dot{\vec{\omega}}^{F}_{iF} \times \vec{r}^{F}_{FL} - \vec{\omega}^{F}_{iF} \times (\vec{\omega}^{F}_{iF} \times \vec{r}^{F}_{FL}) - 2\vec{\omega}^{F}_{iF} \times \vec{v}^{F}_{FL} \tag{4.4}$$

Note the acceleration terms, $\vec{a}^{L}_{iL}$ and $\vec{a}^{F}_{iF}$, are measured directly from the accelerometers, and the rotation rate of the follower's frame, $\vec{\omega}^{F}_{iF}$, is measured directly from the follower's yaw gyroscope. The term representing the angular acceleration of the follower's frame, $\dot{\vec{\omega}}^{F}_{iF}$, is more difficult as it is not measured from an IMU. The simplest estimate of this value can be found from a numerical derivative of the yaw rate across two time steps.

This relative mechanization can then be utilized as a time update for an EKF with a state vector of the form:

$$\vec{x} = \begin{bmatrix} R_x \\ R_y \\ \psi_{L|F} \\ V_x \\ V_y \end{bmatrix} \tag{4.5}$$

This cooperative algorithm does not make use of the vehicle dynamics and instead has a form closer to the constant velocity implementation by including estimates of the relative velocity in the follower's frame: V_x and V_y. However in this algorithm, acceleration is not assumed to be zero mean and is instead measured by onboard sensors. The nonlinear equations for updating these states are seen in Equations 4.6–4.10.

$$\dot{R}_x = V_x \tag{4.6}$$

$$\dot{R}_y = V_y \tag{4.7}$$

$$\dot{\psi}_{L|F} = \omega_L - \omega_F \tag{4.8}$$

$$\dot{V}_x = \vec{a}^F_{FL}(1) \tag{4.9}$$

$$\dot{V}_y = \vec{a}^F_{FL}(2) \tag{4.10}$$

The inputs to the system are the accelerometer and gyroscope measurements of the two vehicles along with the analytical derivative of the follower's yaw rate: a_F, a_L, ω_F, ω_L, and $\dot{\omega}_F$. The state transition matrix, A, and the input-mapping matrix, B, can be found from the Jacobians of the nonlinear equations with respect to the states and inputs, respectively.

These are seen in Equation 4.11 and 4.12.

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ \omega_F^2 & \dot{\omega}_F & -a_{L_y}\cos(\psi_{L|F}) - a_{L_x}\sin(\psi_{L|F}) & 0 & 2\omega_F \\ -\dot{\omega}_F & \omega_F^2 & -a_{L_x}\cos(\psi_{L|F}) - a_{L_y}\sin(\psi_{L|F}) & -2\omega_F & 0 \end{bmatrix} \tag{4.11}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \\ -1 & 0 & 2V_y + 2R_x\omega_F & \cos(\psi_{L|F}) & -\sin(\psi_{L|F}) & 0 \\ 0 & -1 & 2R_y\omega_F - 2V_x & \sin(\psi_{L|F}) & \cos(\psi_{L|F}) & 0 \end{bmatrix} \tag{4.12}$$

Assuming there is uncorrelated noise on all of the input measurements, Q is a diagonal matrix with 7 elements tuned according to the expected measurement noise. It should be noted that this model does not attempt to estimate the biases which are present in inertial sensors. Depending on the application and the selected hardware, estimation of these biases may or may not be necessary. These biases typically consist of a static (turn-on) bias and a dynamic (walking) bias, but the largest error source is due to the static portion [32]. Although they are not estimated, constant biases are introduced to each accelerometer and gyroscope measurement along with the noise. The measurement model for the IMUs is therefore:

$$\tilde{\vec{a}} = \vec{a} + \vec{b}_a + \vec{\nu}_a \tag{4.13}$$

$$\tilde{\omega} = \omega + b_g + \nu_g \tag{4.14}$$

The constant biases are generated by a random value with a standard deviation derived from a lower-end tactical grade IMU. Table 4.1 illustrates the typical ranges for various grades.

Table 4.1: Typical IMU Bias Characteristics [32]

IMU Grade	Accelerometer Bias (m/s^2)	Gyro Bias (deg/hr)
Marine	10^{-4}	0.001
Aviation	$3 * 10^{-4} - 10^{-3}$	0.01
Intermediate	$0.001 - 0.01$	0.1
Tactical	$0.01 - 0.1$	$1 - 100$
Consumer	> 0.1	> 100

With these biases included, a comparison of the relative IMU mechanization to the true relative position for a sample run is shown in Figure 4.2. The error grows with time as expected for a biased-IMU mechanization.

To quantify the performance of the relative mechanization EKF, a comparison can be made to the vehicle dynamics EKF results from the static-follower test depicted in Figure 3.13. The relative mechanization algorithm was implemented on the same scenario with identical range errors. The resulting RPV error norm for the two methods are shown in Figure 4.3.

Even though the non-corrected relative mechanization errors grow quickly, the relative mechanization provides valuable estimates of the relative dynamics when the UWB measurements are employed alongside them. Since this method employs IMUs, it has the additional benefit of being platform agnostic. Although this method seems promising for relative positioning, the requirement of transmitting high update rate IMU measurements (≈100 Hz) may be overbearing for a collaborative approach. The UWBs utilized in this thesis are capable of transmitting data, but the limited data rate is impractical for transmitting IMU data effectively without additional dedicated hardware. This is especially true if there are multiple vehicles locally maneuvering and the number of relative positions being tracked grows. With the selected UWB configuration (PII = 9), the maximum data rate is 19.2 kilobits per

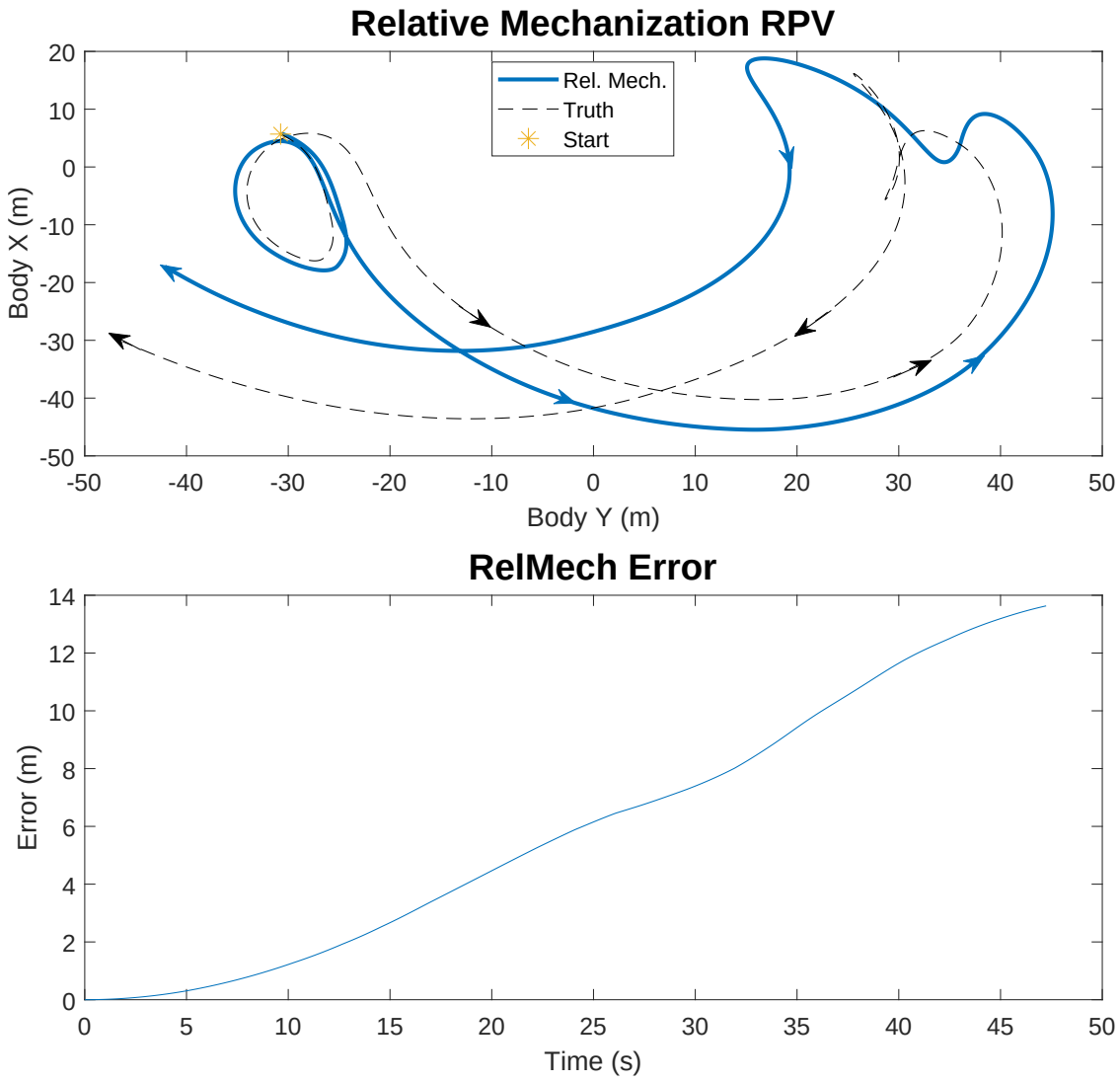


Figure 4.2: Relative Mechanization Errors with Tactical-Grade IMU Biases

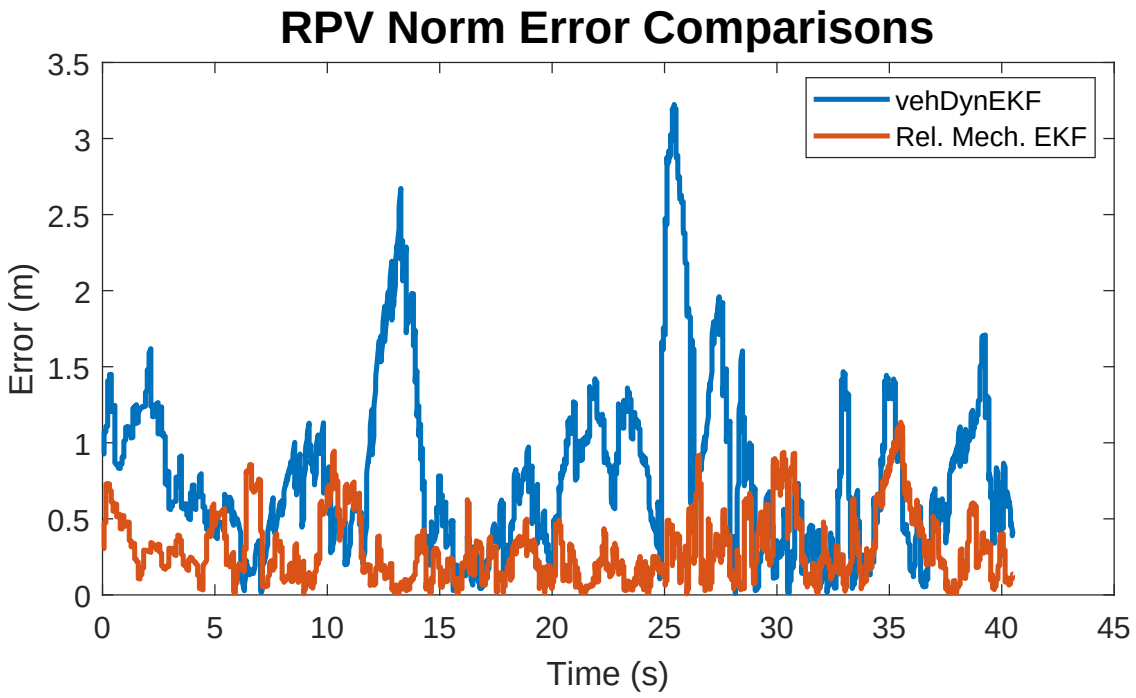


Figure 4.3: Relative Mechanization EKF and Non-Cooperative Vehicle Dynamic EKF

second. Equation 4.15 shows how a full 6-axis IMU with `float32` (32 bits) format operating at 100 Hz reaches the maximum of this data transmission rate.

$$\text{kbps}_{IMU} = 32\frac{\text{bits}}{\text{axis}} \ * \ 6 \text{ axes} * 100 \ \frac{\text{meas}}{\text{s}} = 19,200\frac{\text{bits}}{\text{s}} \tag{4.15}$$

It should be noted that only 3-axes of the IMUs are required for this 2D mechanization, but even with this reduction, adding another node in the network would then reach the threshold. Additionally, once the data is received, it must be processed by the estimation algorithms by the on-board hardware. Therefore, this algorithm presents scaling challenges and will not be explored as thoroughly as the other methods for the results presented in this thesis.

4.2 Vehicle Dynamic Model State Feedback

The vehicle dynamics EKF described in the previous chapter seeks to improve RPV estimation quality by providing a more sophisticated model for the propagation of the RPV

over time. The results shown in the following chapter highlight the error reduction in a variety of non-cooperative test cases. The inclusion of this dynamic model also provides an avenue for highly-impactful collaboration. Recall the crucial relationship between lead vehicle speed and yaw rate (previously shown in Equations 3.39–3.41):

$$\omega_L = v_{Lx} d_L \tag{4.16}$$

The above relationship was useful to simplify the number of states estimated to describe the change in the RPV due to the lead vehicle's motion. This also means that significant information can be provided to the filter by the feedback of a few states from the lead vehicle. Any or all of the three dynamic variables ($\omega_L/v_{Lx}/d_L$) may be transmitted to the estimating vehicle. The longitudinal velocity could be generated from wheel encoders or a rough speedometer estimate, while the effective steer angle is a more involved measurement since it requires knowledge of the deflection at the tires along with the length of the wheelbase. On the other hand, a yaw-gyro is exceedingly common in autonomous vehicle applications. As with any EKF measurement, the measurement mapping matrices are required for implementation. For the velocity and effective steer angle, the measurements are directly related to the states, so their measurement matrices are:

$$H_{v_{Lx}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{4.17}$$

$$H_{d_L} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.18}$$

The yaw-rate measurement is a function of both the velocity and steer angle, so the mapping matrix is found from the partial derivative with respect to each:

$$H_{\omega_L} = \frac{\partial\, \omega_L}{\partial \vec{X}} = \begin{bmatrix} 0 & 0 & 0 & d_L & v_{Lx} \end{bmatrix} \tag{4.19}$$

Any combination of these measurements can be useful to the relative position estimators and will be explored in the simulation and experimental results. It should be noted that this direct state feedback capability is not feasible in the constant velocity EKF and represents a further improvement over that method. Further, unlike the relative IMU mechanization, the data sizes are substantially smaller. The feedback measurements also do not require integration to be useful. If communication bandwidth is limited, logic can be implemented to only send these feedback terms when deemed necessary, such as in non-observable regions or when a large change occurs.

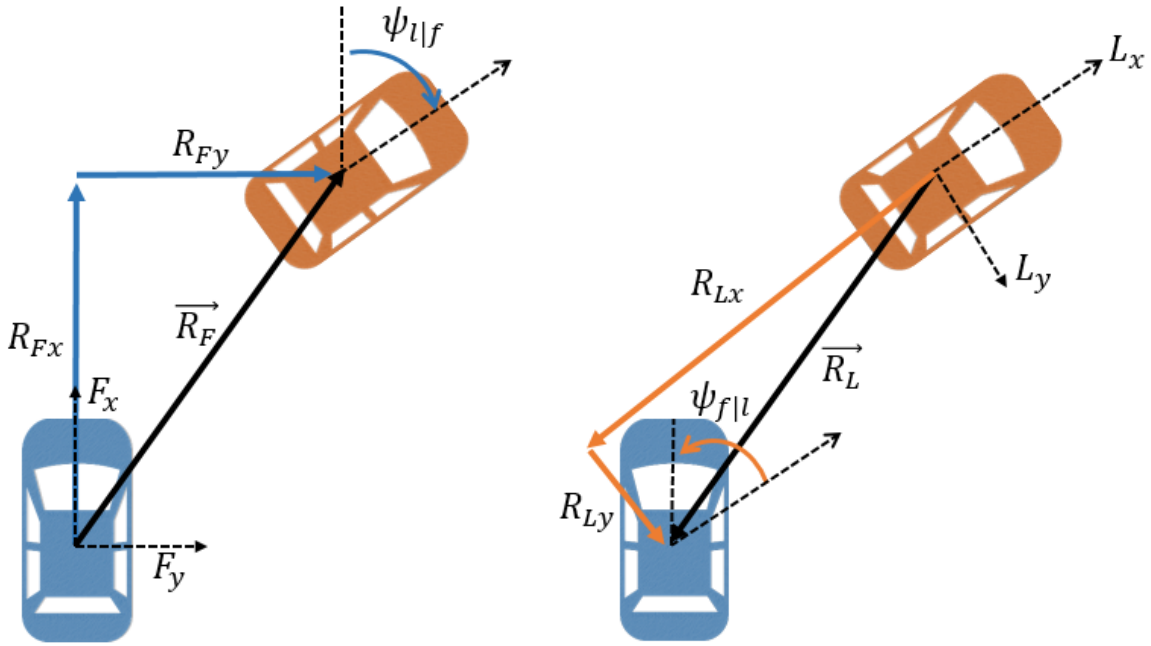
4.3 Consensus Extended Kalman Filter (CEKF)

Up until this point, all of the algorithms assume only one vehicle was tracking the pose of another. It is rational to assume other vehicles would also be interested in this information. Extending this problem to multiple vehicles tracking relative poses to one another creates opportunities for cooperation. As was mentioned in Chapter 1, ground vehicles estimating their relative positions to one another may be re-formulated to be estimating a shared state. With this modification, a consensus condition may be employed to achieve a more optimal estimate for each node in the network. For instance, consider two locally-operating vehicles each estimating the relative position to one another as seen in Figure 4.4.

Although these estimators are tracking two different vectors independently, $\vec{R}_F$ and $\vec{R}_L$, these two vectors are simply the inverses of one another resolved in different frames. The conversion between the frames is dependent on another existing state, the relative heading ψ. These relationships are shown in Equations 4.20 and 4.21.

$$\vec{R}_F = -C_L^F \vec{R}_L \tag{4.20}$$

$$\psi_{L|F} = -\psi_{F|L} \tag{4.21}$$



(a) Estimator 1 on Follower (b) Estimator 2 on Leader

Figure 4.4: Two Geometrically-Related RPV Estimators

With a simple negation and conversion between frames, the two estimators are estimating the same state and can cooperate with one another to converge their estimates to a common *consensus*. The class of decentralized estimators that achieve consensus between one another are known as Consensus Kalman Filters (CKF). As in a traditional Kalman Filter, the sub-optimal extension to a nonlinear system is known as a Consensus Extended Kalman Filter (CEKF).

The CEKF process for a single node's time update remain unchanged. The measurement update process is modified to account for the neighboring estimates. This is seen in Equations

4.22–4.24. Note the subscript i refers to an update for a specific node in the network.

$$K_i = \left(P_i + C_i \sum_{j \in N_i} (P_{j,i} - P_i) \right) H_i^T \left(R_i + H_i P_i H_i^T \right)^{-1} \tag{4.22}$$

$$P_i = F_i P_i F_i^T + C_i \sum_{r \in N_i} (P_{r,i} - P_i) F_i^T + F_i \sum_{s \in N_i} (P_{i,s} - P_i) C_i^T + C_i D_i C_i^T + K_i R_i K_i^T \tag{4.23}$$

$$\vec{x}_i^+ = \vec{x}_i^- + K_i \left(\tilde{y} - h(\vec{x}_i^-) \right) + C_i \sum_{j \in N_i} (\vec{x}_i^- - \tilde{x}_i^-) \tag{4.24}$$

At steady-state, the consensus update term, $C_i \sum_{j \in N_i} (\vec{x}_i^- - \tilde{x}_i^-)$, leads each individual estimator to a globally-averaged consensus for the shared states. The terms, $P_{i,j}$ and the like, represent edge-covariance matrices depending on the connectivity of the network, while N_i denotes the first neighbors of a given node. These edge-covariance matrices are tracked separately and require interaction with second neighbors along with their local covariance estimates. For specifics on their calculations and a full derivation of the optimal CEKF, look to [68].

The main takeaway from the measurement update equations is that the optimal consensus estimation method requires exchange of each node's covariance matrix and estimated states. It was shown in [68] that this results in a non-scalable algorithm where the computational complexity of the covariance update is $O(n^2)$ with n being the number of nodes in the network. The sub-optimal CKF was also introduced by Oltafi which utilizes the assumption of a small consensus gain, C_i, to neglect sharing of covariance estimates. Although the optimal CEKF does generally perform better, successful tracking has been achieved with the sub-optimal form as well [27]. This sub-optimal method will be employed here. The omission of covariance transmission leads to the sub-optimal CEKF gain calculation and covariance update matching a traditional EKF. The only remaining difference from a traditional EKF is the additional term applied to the state correction as seen in Equation 4.25. This makes it exceptionally easy to augment a candidate EKF into a sub-optimal CEKF regardless of

the filter's time or measurement model forms.

$$\vec{x}_i^{+} = \vec{x}_i^{-} + K_i\left(\tilde{y} - h(\vec{x}_i^{-})\right) + C_i \sum_{j \in N_i} (\vec{x}_i^{-} - \tilde{x}_i^{-}) \tag{4.25}$$

The final term, $\tilde{x}_i^{-}$, represents the state estimate derived from a consensus condition with relationships to the other estimator(s). The consensus gain is a tunable parameter for each estimator, but prior literature results have lead to the commonly used form:

$$C_i = \gamma \frac{P_i}{1 + ||P_i||_F} \tag{4.26}$$

where P_i is the covariance of the local estimator. The choice of γ is free under the condition it is greater than 0. A higher value results in a faster consensus, but may result in unstable error dynamics [68]. In general, it should be kept small to guarantee convergence and maintain the small gain assumption (often on the order of the discretization time step). The subscript F denotes the Frobenius norm of a given node's covariance matrix. It is found by the square root of the sum of all the squared values of a matrix. It may be calculated directly by:

$$||P_i||_F = \sqrt{trace(P_i^T P_i)} \tag{4.27}$$

This geometric consensus condition was inspired by an application of a sub-optimal CEKF to aerial vehicles with numerous sensors by Gong et al. [30]. In this work by Gong, each of the three aerial vehicles were estimating 2-D relative pose to one other node and were equipped with single UWBs along with IMUs, pitot tubes, and magnetometers. The three aerial vehicles utilized vector algebra to create a global consensus condition. To the author's knowledge, this thesis is the first application of a geometric condition CEKF to locally-maneuvering ground vehicle relative positioning.

For a multi-vehicle network, potential benefits of a consensus condition include: reduced transmission requirements compared to relative mechanization (similar to odometry

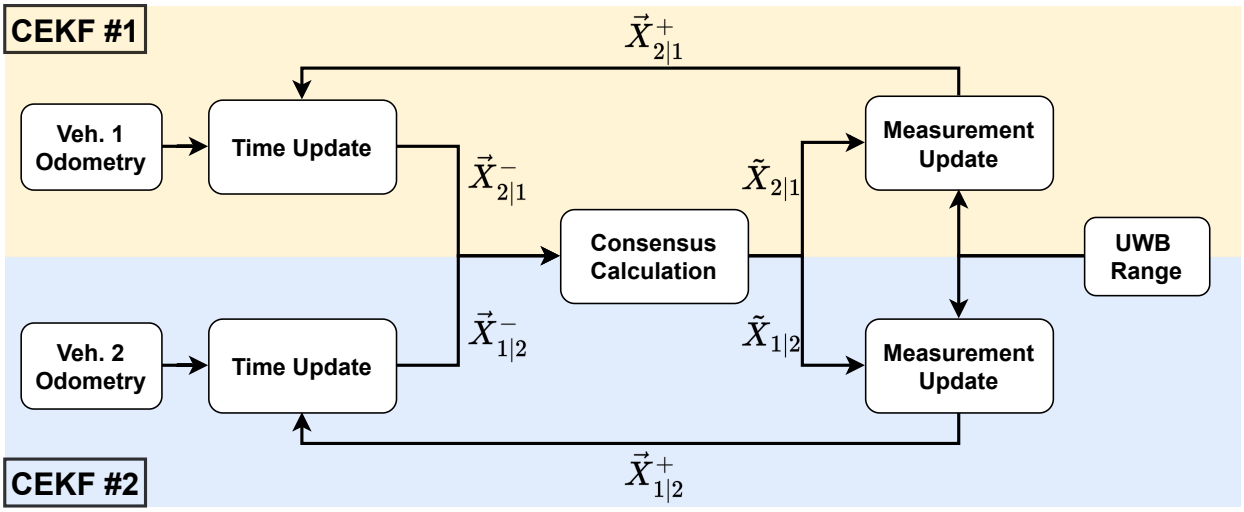


Figure 4.5: 2-Vehicle Consensus Block Diagram with Veh. Dyn. CEKFs

feedback), error averaging, and method-agnostic fusion capabilities. Method-agnostic fusion means that not all vehicles must be utilizing the same RPV estimation method; for instance, vehicle 1 may estimate vehicle 2 with the vehicle dynamic EKF with onboard odometry while vehicle 2 may estimate vehicle 1 with a vision-based tracking solution. This consensus condition can allow for leveraging the strengths and weaknesses of multiple methods together with minimal changes to each algorithm. If there were more than two vehicles interested in relative positions to one another, the consensus condition could make use of all the estimator's solutions and provide further agreement among the network, but here only two-vehicle networks will be considered.

In this thesis, two-vehicle consensus modifications will be applied and compared within both the constant velocity and vehicle dynamic frameworks. A block-diagram for the geometrically-inspired Consensus Extended Kalman Filter for two vehicles is shown in Figure 4.5.

Even without the consensus correction, the block diagram illustrates that the filters in the two-vehicle CEKF algorithm do not have complete independence from one another since they make use of the same UWB ranges at an instance in time. This is another factor

leading to a sub-optimal result. The consensus updates will be explored alongside the other cooperative algorithms in the following chapters.

Chapter 5

Simulation Results

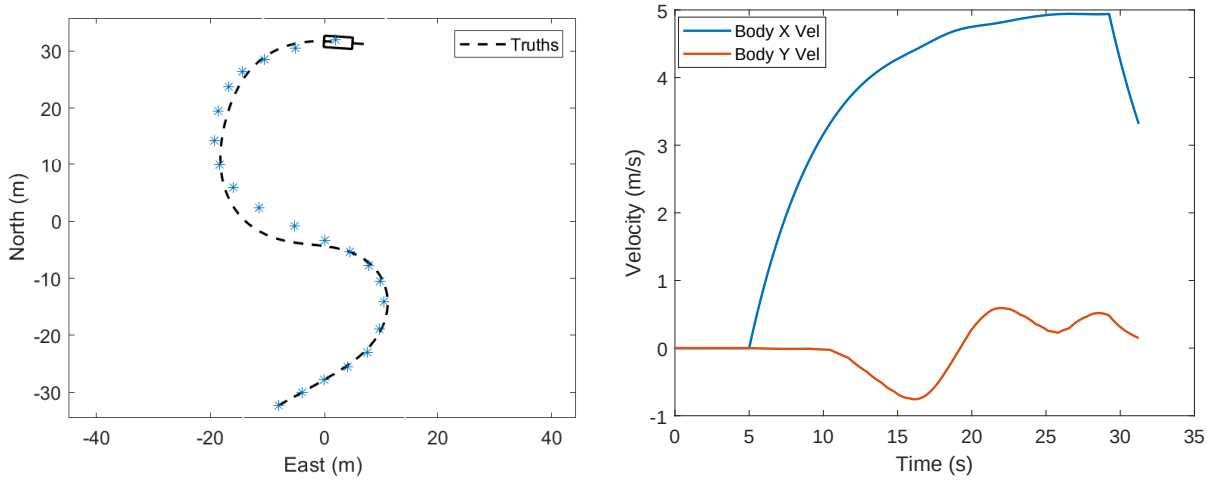
5.1 Simulation Structure

The simulation was generated according to the models described in the previous sections. The dynamics of the vehicles are governed by the full kinematic bicycle model from Chapter 2 but are not constrained by the zero side-slip angle assumption employed in the estimation algorithms. This means the dynamic model for the trajectories and the state-propagation model in the vehicle dynamics EKF are not identical. This more closely resembles an experimental implementation where the estimator model does not fully encapsulate the ground-vehicle physics. It is common in sub-optimal estimation algorithms for the modeled and true dynamics to differ when the full model may be unknown or difficult to describe.

The kinematic bicycle model is traditionally employed with velocity and steer-angle inputs. However, since accelerations are desired for some of the algorithms presented here, Equations 2.30 to 2.32 are differentiated to generate the governing dynamic equations. The dynamics are controlled through acceleration and the rate of change of the steer angle instead, reducing singularities and allowing for generation of simulated IMU accelerations. This more closely resembles reality and also results in improved estimation accuracy once the vehicles begin moving since the velocity-derived equations would predict an infinite acceleration when starting from rest otherwise. The acceleration-based equations are shown here:

$$\ddot{X} = |\dot{V}|\cos(\psi+\beta) - |V|\sin(\psi+\beta)(\dot{\psi}+\dot{\beta}) \tag{5.1}$$

$$\ddot{Y} = |\dot{V}|\sin(\psi+\beta) - |V|\cos(\psi+\beta)(\dot{\psi}+\dot{\beta}) \tag{5.2}$$



(a) Trajectory (b) Generated Body Velocities

Figure 5.1: Trajectory Generation Example

where:

$$\dot{\psi} = \frac{|\vec{V}| \tan(\delta) cos(\beta)}{L} \tag{5.3}$$

$$\beta = \arctan\left(\frac{l_r \tan(\delta)}{L}\right) \tag{5.4}$$

$$\dot{\beta} = \frac{L L_r \dot{\delta} \sec^2(\delta)}{L^2 + L_r^2 \tan^2(\delta)} \tag{5.5}$$

These derivatives are used to update the vehicle states at 100 Hz. The trajectories are determined by a coarse waypoint-following control law where the points are manually selected at the start of a run. After waypoint selection, the vehicles accelerate according to a first-order model up to the desired speed. This velocity is maintained and the steering commands are generated from a heading error control law with a velocity-moderated lookahead distance. This results in a rough following of the desired paths, but since specific trajectories are not the focus of this thesis, realistic dynamics are more important than the exact paths taken. An example of the trajectory generation and resulting velocities is shown in Figure 5.1.

The UWB measurements are constructed at the nominal update rate of the P440 hardware configuration: 15 Hz. As mentioned previously, this equates to each range having an

independent update rate of ≈3 Hz. The values are generated from the nonlinear measurement model seen in Equation 3.7 using the vehicle states at given instant in time. The UWBs are assumed to be spaced 6 feet (36 inches) apart and to have the midpoint of their baseline coincident with the vehicle's COG.

The vehicles always start from rest to allow for the least-squares initialization process to convergence. The initial covariance is set by multiplying the calculated DOP at the initial location by the UWB measurement variance. Also, all of the results are performed with the wheel encoder and gyro odometry method that will be employed in the experimental results. This means the lateral velocity of the ego vehicle is unmeasured and set to zero in the state propagation step (i.e. the zero side-slip angle assumption described in Chapter 2). To compensate, the follower lateral velocity uncertainty process noise parameter is increased.

5.2 Non-Cooperative Results

5.2.1 Scenario 1: S-Turn with Static Base Vehicle

The first simulated scenario is a static base vehicle estimating the RPV to a dynamic vehicle performing an "s-turn" maneuver. This is the most benign experiment presented in this work as the maneuvering vehicle remains in low-DOP relative pose geometries throughout the run. The generated path is depicted in Figure 5.2.

For a direct comparison, the two non-cooperative algorithms were tested on UWB ranges corrupted with the same errors. The results for a single-run are shown in Figure 5.3. The accuracy of each algorithm is quantified by comparing the RPV estimate to the true trajectory at each instance in time. The RPV-component errors for this run can be seen in Figure 5.4

From inspection of the RPV errors, another source of error for the algorithms can be seen—relative dynamics. The errors grow when the relative velocity between the two vehicles change such as during a turn. Since each non-cooperative algorithm does not have direct measurements of the relative velocities, the EKF estimates for them are generated from their

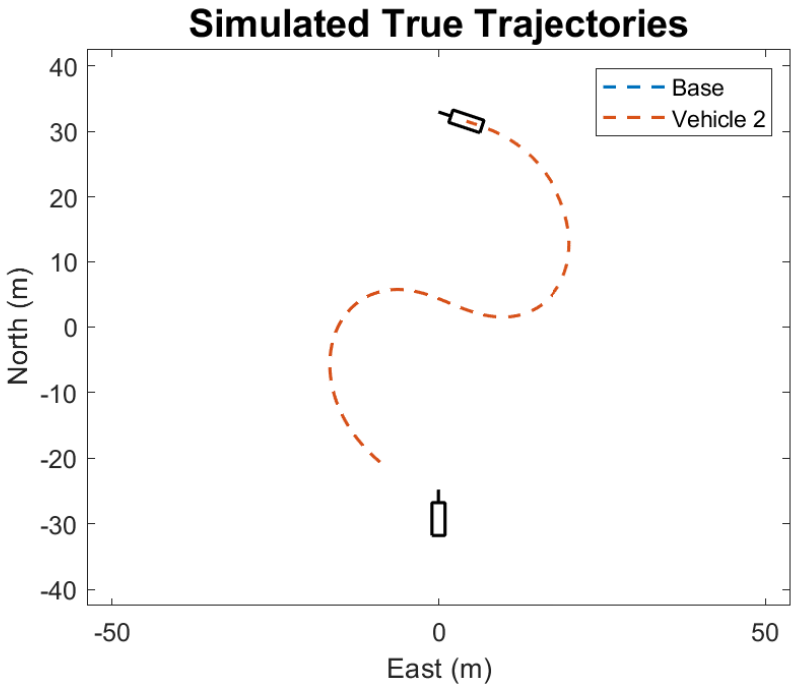


Figure 5.2: Simulated S-Turn Maneuver with Static Base

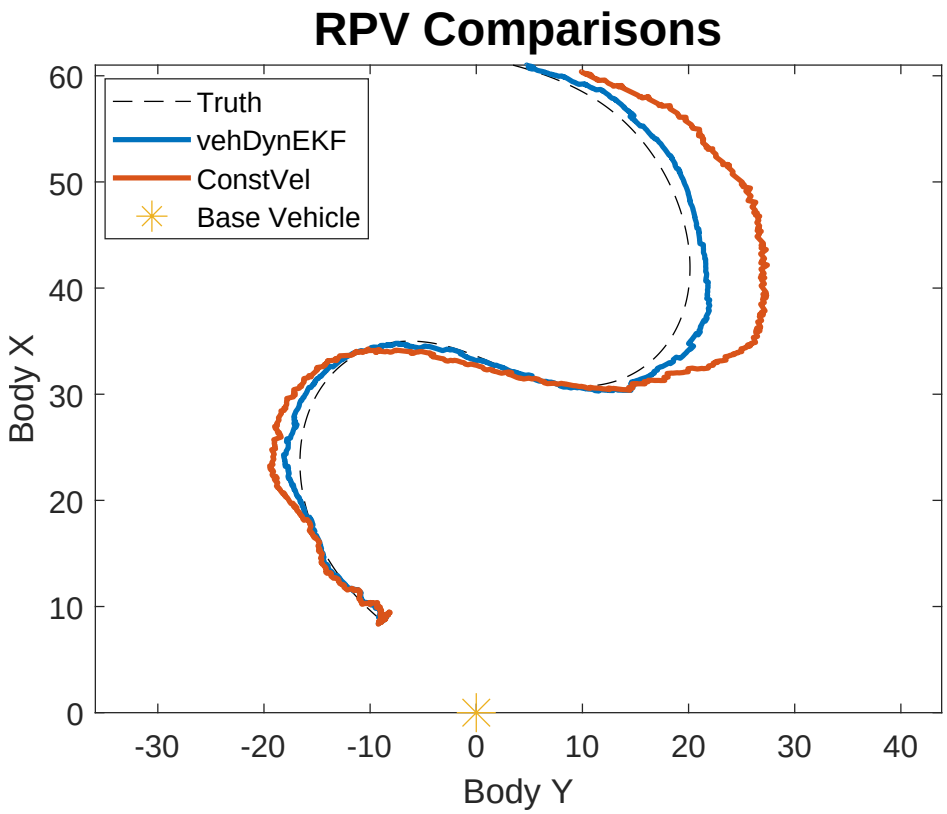


Figure 5.3: Single Run S-Turn RPV Estimation Results

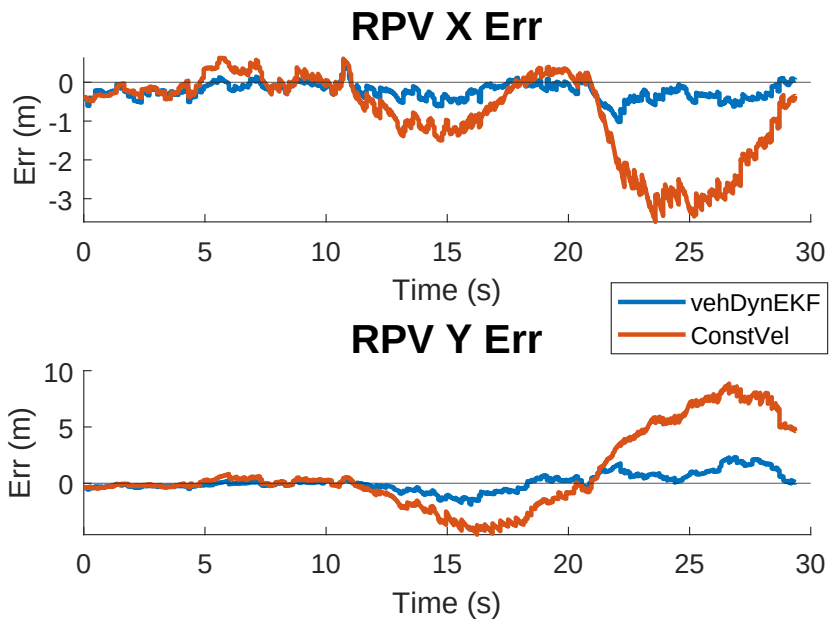


Figure 5.4: S-Turn RPV Component Errors

correlation with the consecutive range measurements. This leads to a lagging estimate in the relative velocities for the constant velocity EKF and a lagging estimate of speed and effective steer angle for the vehicle dynamics EKF. Recall that the VehDynEKF algorithm does not estimate the relative velocities directly but rather the leader's components of the relative motion model. The vehicle dynamics EKF estimates its dynamic states more quickly and accurately, however, which is one reason for the improved results. This is shown in Figure 5.5.

To perform a statistical analysis of the errors encountered in each algorithm, a Monte Carlo simulation of 100 runs was performed with the nominal 10 cm of random error added to the simulated ranges. The errors of the RPV components were plotted with the mean 3σ bounds at each time-step from the covariance of EKFs. The Monte Carlo sample covariance bounds were calculated as well. These values are shown in Figure 5.6 for the constant-velocity EKF.

The impact of the relative dynamics and resulting lagging relative velocity estimate are clear. Although there are differences in each Monte Carlo run due to varying errors on the

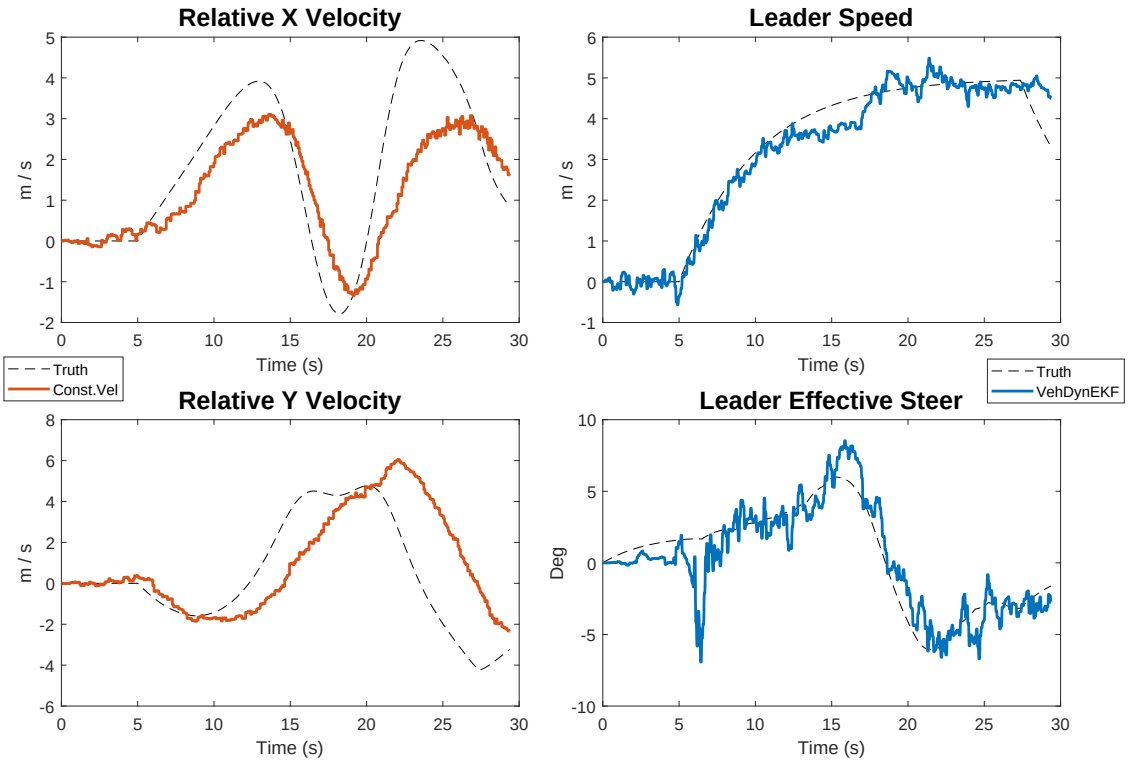


Figure 5.5: Dynamic State Estimates of Each EKF

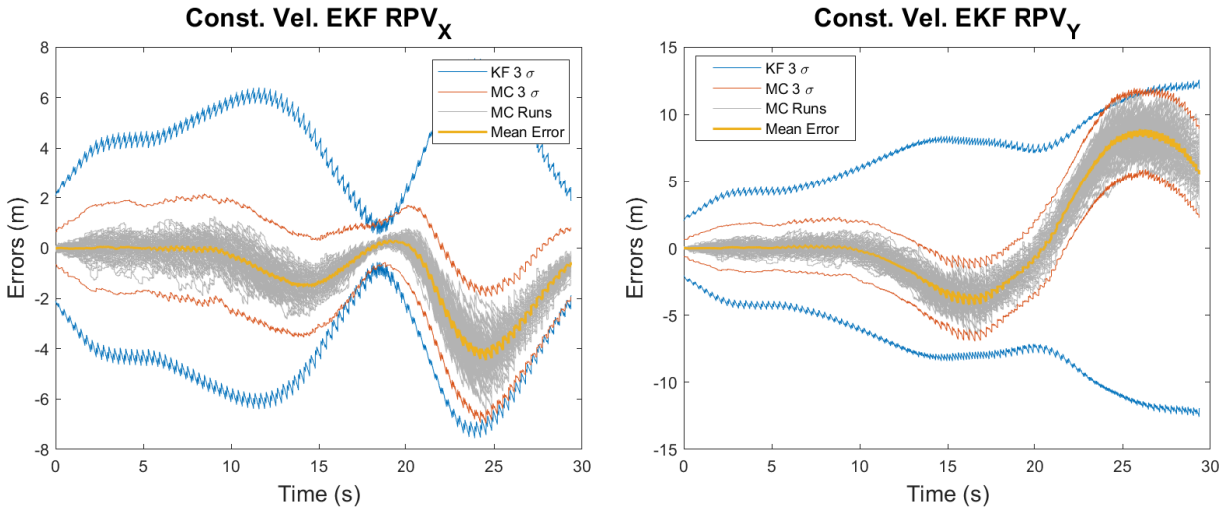


Figure 5.6: S-Turn Constant Velocity EKF 100 Monte Carlo Run RPV Errors

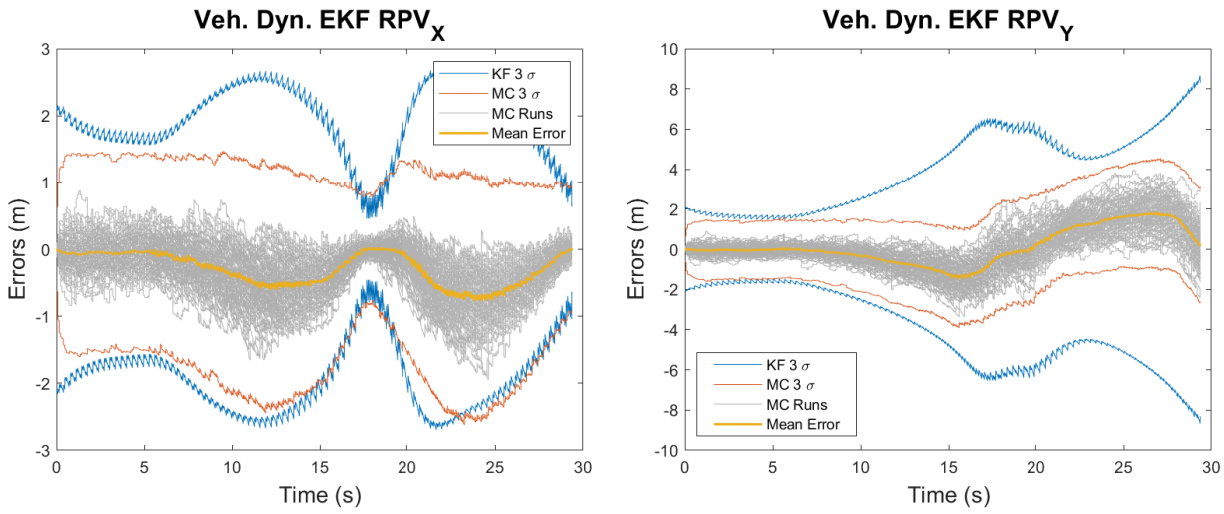


Figure 5.7: S-Turn Vehicle-Dynamics EKF 100 Monte Carlo Run RPV Errors

UWB observables, the errors are not zero-mean and follow a similar trend for each run, especially in the y-direction. Figure 5.7 shows the same plot for the vehicle dynamics EKF.

Comparisons of these figures show clear improvements between this EKF and the constant-velocity method from [21]. However, similar to the constant velocity method, the errors are not truly zero-mean due to the relative dynamics. To combine the two-dimensional RPV results into one scalar value, the norm of the RPV error at a given point in time is calculated by the Euclidean distance between the truth and estimated point. These error norms were found for each Monte Carlo run and were utilized to calculated the expected error mean and standard deviation for each filter's estimate. Figure 5.8 shows the mean of the results at each time-step while the vertical bars represent one standard deviation in the Monte Carlo error results at that instant in time.

In this scenario, the range between the vehicles grows with time, and the expected error tends to grow as a result—particularly in the base vehicle's y-axis. The DOP plots presented in Chapter 3 agree with this finding as the UWB measurements provide less observability of the RPV with increasing distance. A summary of the error statistics for this non-cooperative scenario is shown in Table 5.1.

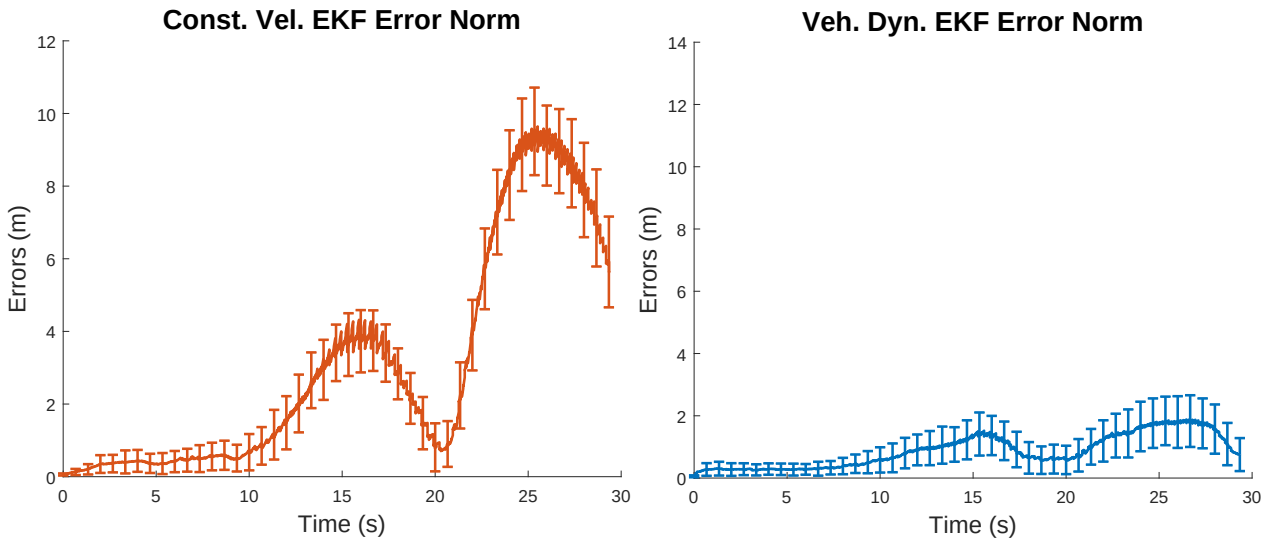


Figure 5.8: S-Turn Mean and Standard Deviation for Monte Carlo Errors

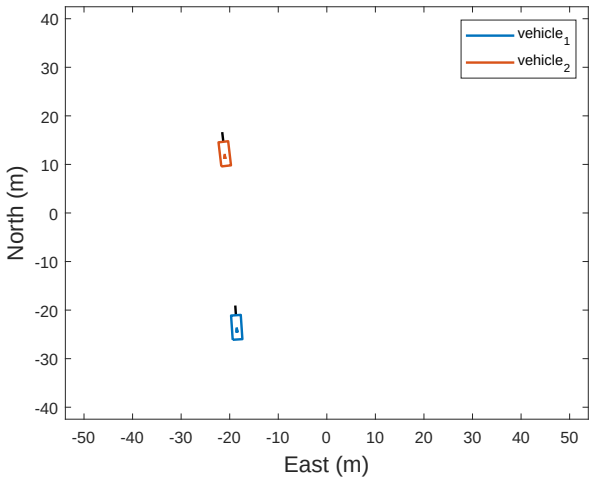
Table 5.1: Scenario 1: S-Turn with Static Base 100-Run Error Statistics

Algorithm		Mean	Std. Dev.
Const. Vel. EKF	X Error (m)	-1.06	1.29
	Y Error (m)	1.10	3.86
	ψ Error (deg)	-5.84	39.39
	RPV Error Norm (m)	3.03	3.11
Veh. Dyn. EKF	X Error (m)	-0.29	0.37
	Y Error (m)	0.07	1.00
	ψ Error (deg)	0.14	4.27
	RPV Error Norm (m)	0.88	0.71

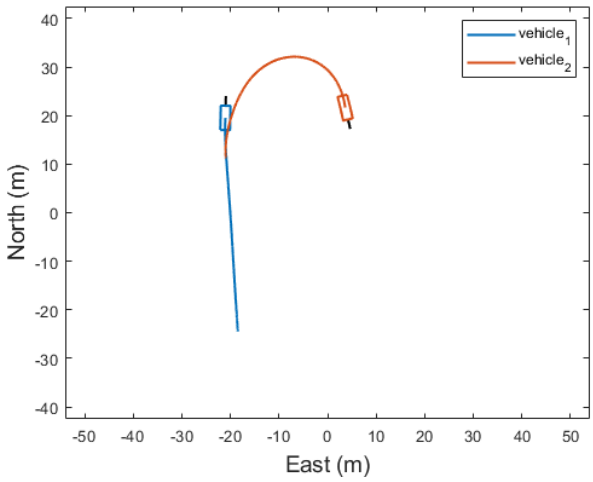
5.2.2 Scenario 2: Dynamic U-Turn

Ghanem [21] showed in their results that there is a problem area when the two vehicles pass one another. This was encountered during a U-turn in their experimental data set where the estimator locked onto the ambiguous solution. This maneuver is recreated in simulation to verify their findings and to compare the proposed vehicle dynamics EKF. The trajectory of the vehicles is outlined in the snapshots shown in Figure 5.9.

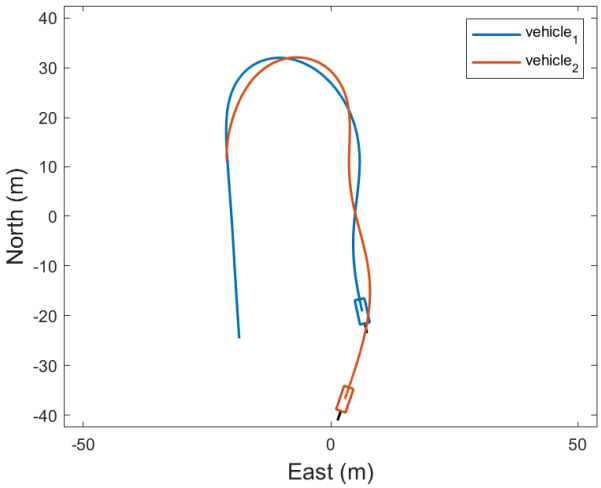
For this run, the mean range between the vehicles was 32 meters. Each filter is given the same UWB ranges with equivalent error statistics. A comparison for a single run is shown in Figure 5.10. It can be seen from the single-run results that the constant-velocity EKF



(a) Starting Locations



(b) Vehicles Pass Each Other



(c) End of Run

Figure 5.9: Dynamic U-Turn Trajectory Depiction

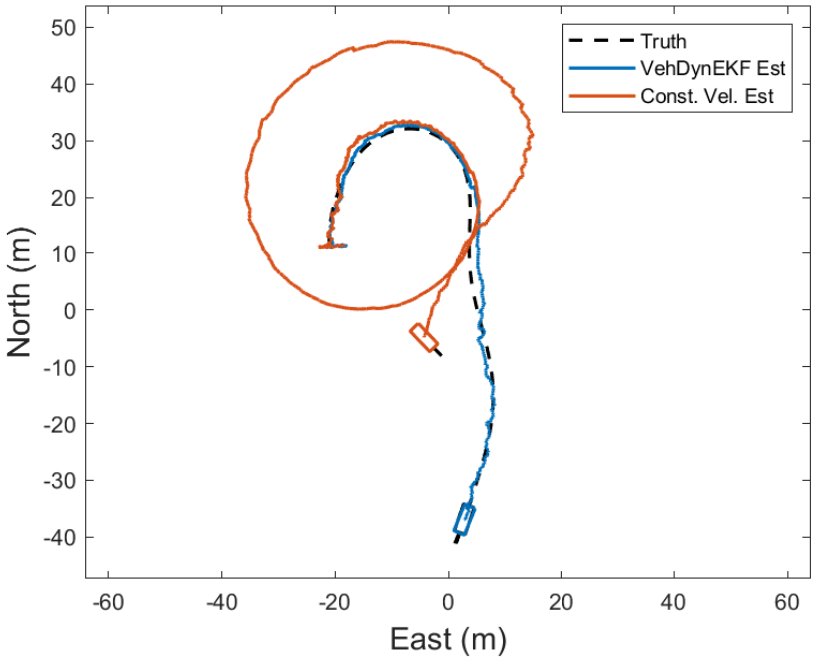
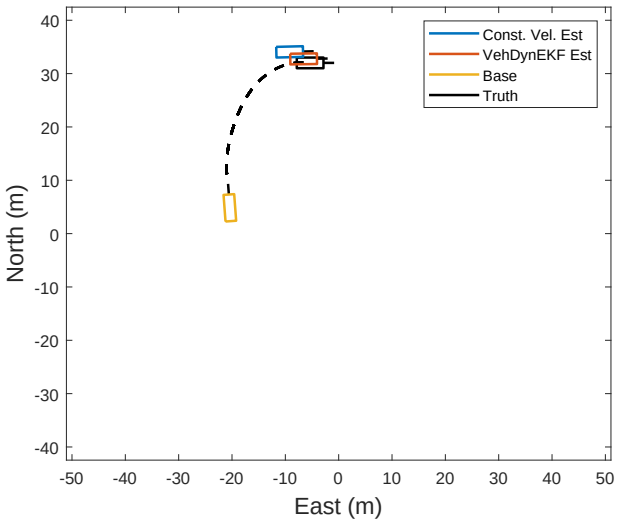


Figure 5.10: Single Run of Non-Cooperative Estimation Comparison

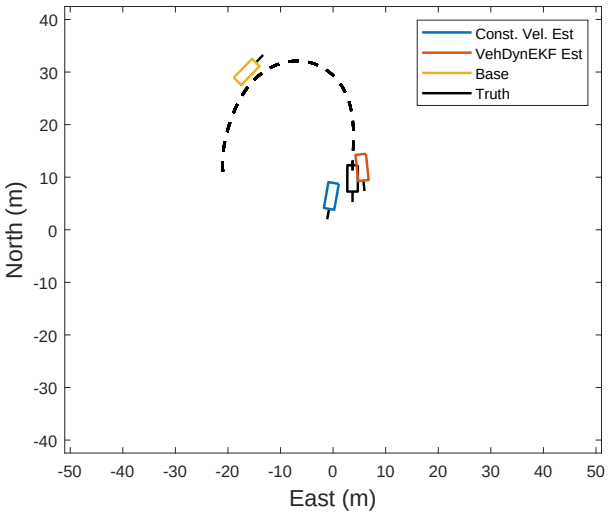
solution begins to break down during the unobservable relative pose when the vehicles are side-by-side. Figure 5.11 highlights 3 different segments of the trajectory to visualize the errors more directly. In Figure 5.11a, when the tracked vehicle is at the top of its turn, the estimators both have a decent solution of the relative pose. In Figure 5.11b, the solutions of both estimators have degraded as the vehicles pass one another; however, by the end of the run in Figure 5.11c, the vehicle-dynamics EKF has continued tracking the vehicle and returned to the truth while the constant-velocity EKF has converged to the geometrically extraneous solution.

As in the previous scenario, 100 Monte Carlo simulations were performed to quantify the error statistics for this case. Once again, the impact from the relative dynamics is still present as the errors are not zero-mean for either algorithm. From inspection of Figures 5.12 and 5.13, the reduction in RPV errors in the vehicle dynamics EKF are evident.

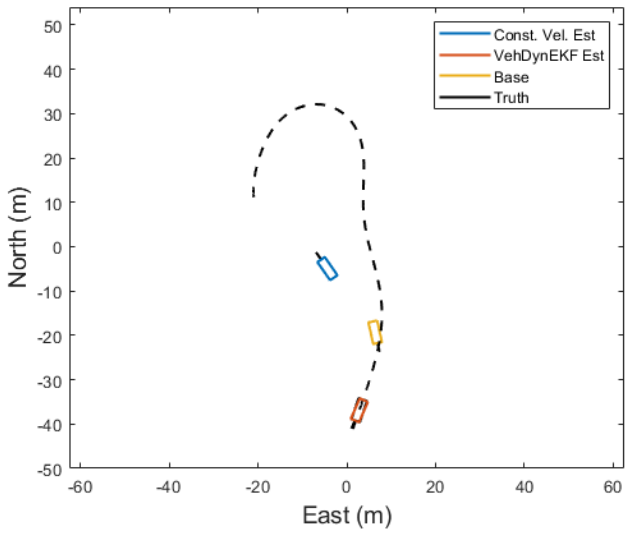
The error norms and standard deviations were calculated as before. A comparison of these two non-cooperative EKFs for this scenarios are shown in Figure 5.14. It is clear from



(a) Top of U-Turn



(b) After Passing Each Other



(c) End of Run

Figure 5.11: Dynamic U-Turn Estimate Snapshots

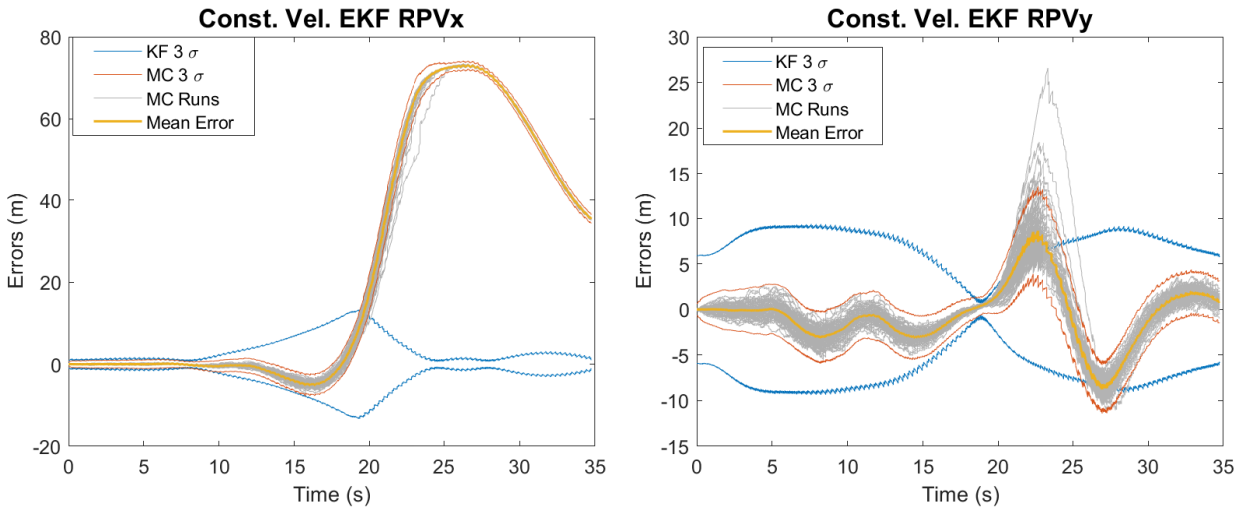


Figure 5.12: Dynamic U-Turn Constant Velocity EKF 100 Monte Carlo Run RPV Errors

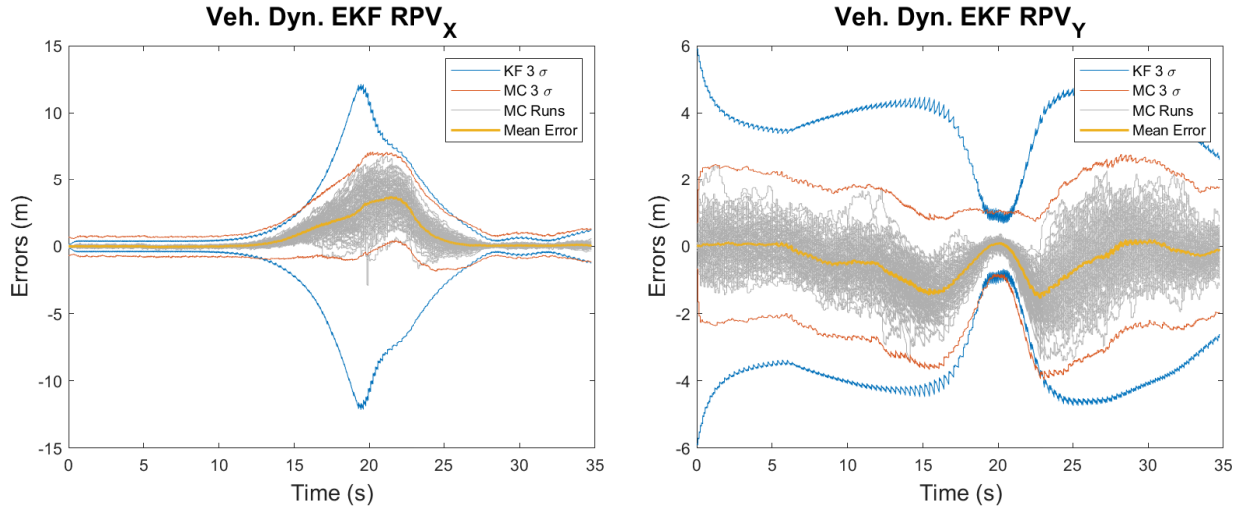


Figure 5.13: Dynamic U-Turn Vehicle-Dynamics EKF 100 Monte Carlo Run RPV Errors

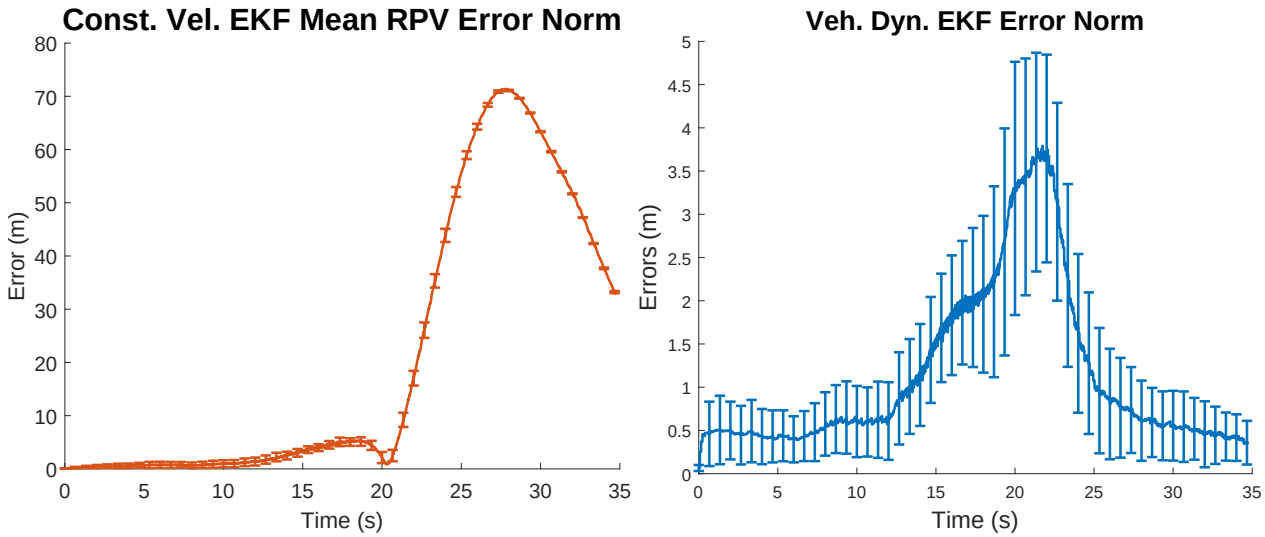


Figure 5.14: Dynamic U-Turn Mean and Standard Deviation for Monte Carlo Error Norms (Varied Y-Axis Scales)

these results that the constant velocity method lacks robustness, and certain trajectories can lead to filter instability. This finding agrees with the results from Ghanem's prior work with similar trajectories [21].

The errors for the vehicle dynamics EKF were also higher for this scenario compared to the last due to the increased relative dynamics. However, the errors were still below a meter on average in each axis and remain bounded by the filter's covariance estimate. Table 5.2 summarizes the errors for this scenario.

Table 5.2: Scenario 2: Dynamic U-Turn 100-Run Error Statistics

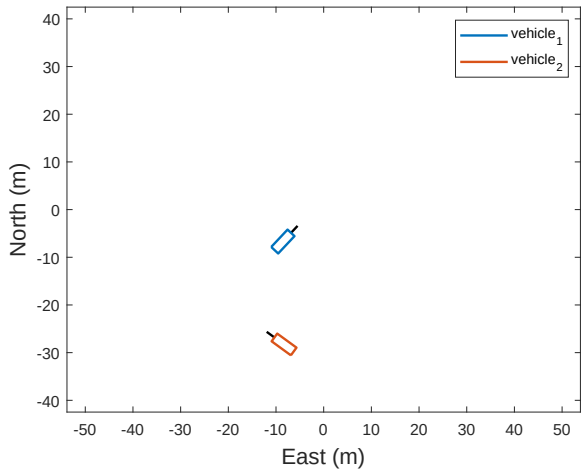
Algorithm		Mean	Std. Dev.
Const. Vel. EKF	X Error (m)	-19.03	27.29
	Y Error (m)	-0.27	3.05
	ψ Error (deg)	-3.05	66.80
	RPV Error Norm (m)	20.97	25.97
Veh. Dyn. EKF	X Error (m)	0.77	1.40
	Y Error (m)	-0.41	0.79
	ψ Error (deg)	-1.32	4.40
	RPV Error Norm (m)	1.25	1.33

5.2.3 Scenario 3: Figure-8 Maneuver

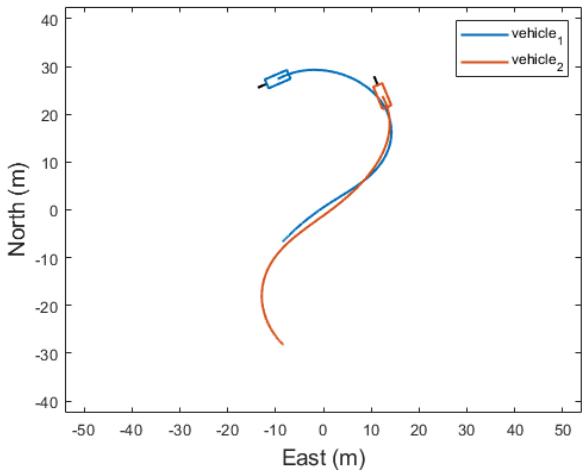
The previous two scenarios are specific situations to exhibit a variety of scenarios of the estimators to show performance in both ideal and edge-cases. A scenario to encompass many of these aspects into one run is a figure-8 trajectory. This scenario consists of a mixture of favorable and unfavorable relative geometries. Figure 5.15 highlights components of the paths taken by the vehicles in this scenario. The range between the vehicles varied in this scenario, but the average was 21.49 meters.

Note that the vehicles approach the unobservable relative-geometry regions throughout the turns, but never fully pass each other in opposite directions (as they did in scenario 2). This prevents the constant velocity estimates from completely breaking down for this run. Nevertheless, the resulting errors show similar trends as before. The two non-cooperative methods have similar non-zero-mean errors as the vehicles have higher relative dynamics during the turns, but the vehicle dynamic EKF maintains substantially lower errors throughout. Figures 5.16 and 5.17 depict the 100-run Monte Carlo results for each filter. Note that the vehicle-dynamic EKF remains bounded by the estimated covariance bounds.

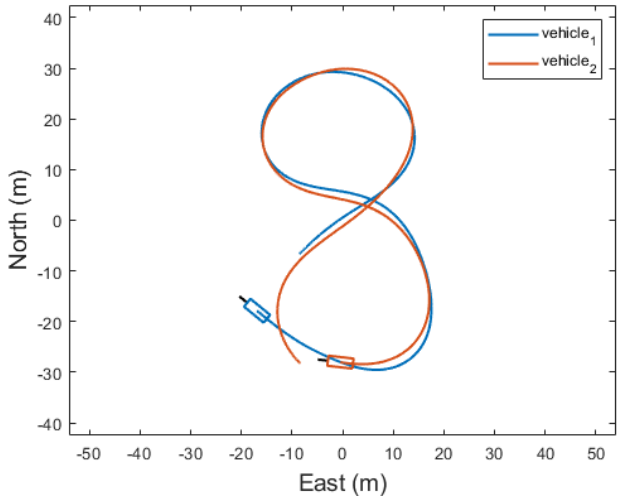
The error norms for this run are shown in Figure 5.18. The y-axis limits are different on the left and right plots to show more detail on the lower errors of the vehicle-dynamic EKF results. Since the relative range of the two vehicles was relatively constant (between 18–24 meters) during this run, the error spikes are more dependent on the relative dynamics as mentioned previously. Table 5.3 shows the error statistics for this scenario.



(a) Starting Locations



(b) Approaching Unobservable Region



(c) End of Run

Figure 5.15: Dynamic U-Turn Trajectory Depiction

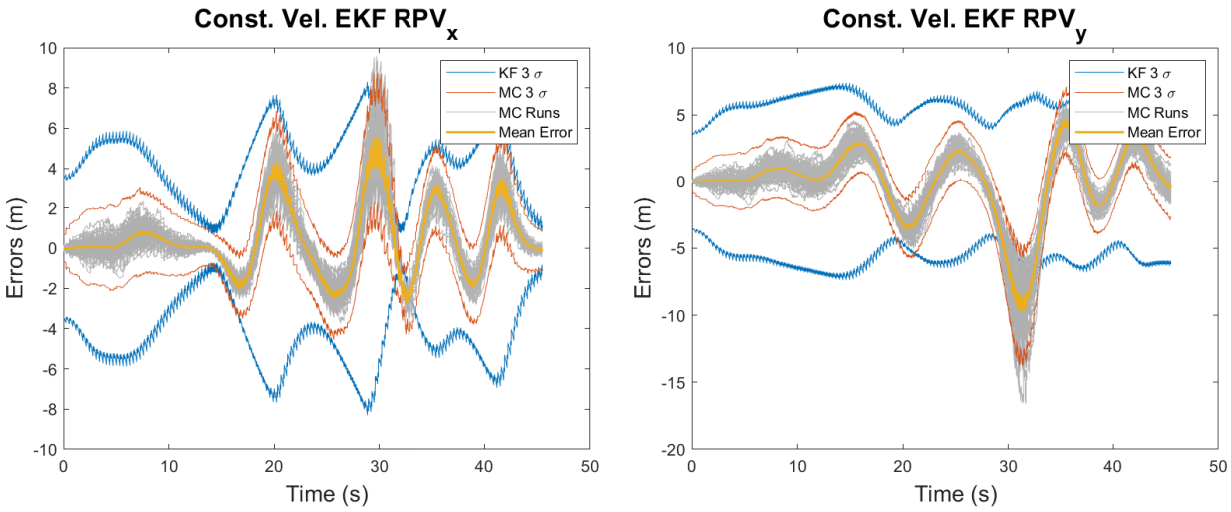


Figure 5.16: Figure-8 Constant Velocity EKF 100 Monte Carlo Run RPV Errors

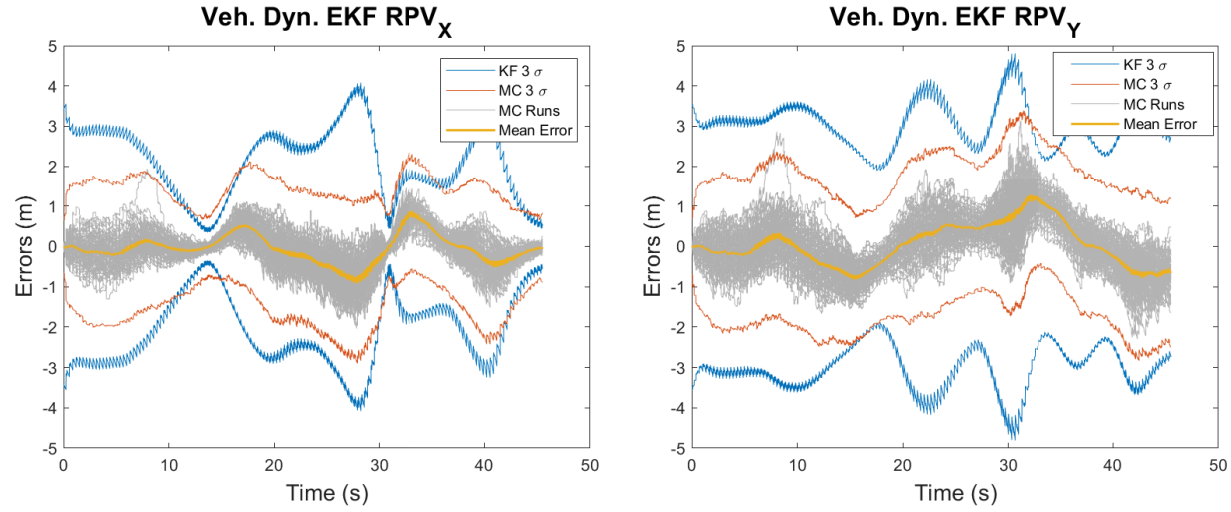


Figure 5.17: Figure-8 Vehicle-Dynamics EKF 100 Monte Carlo Run RPV Errors

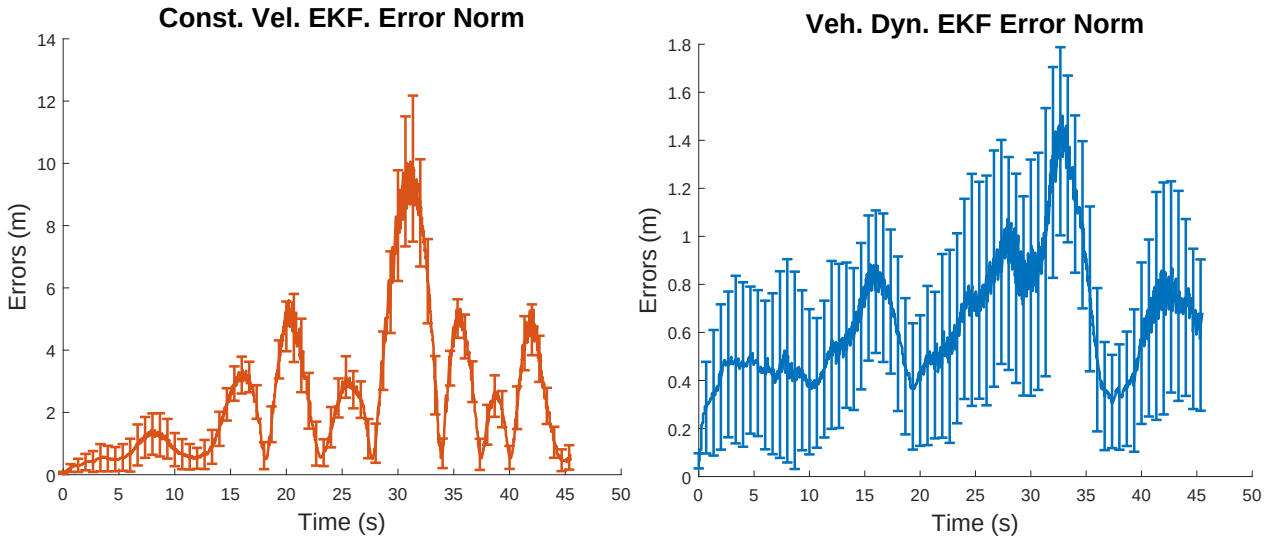


Figure 5.18: Figure-8 Mean and Standard Deviation for Monte Carlo Error Norms (Varied Y-Axis Scales)

Table 5.3: Scenario 3: Figure-8 100-Run Error Statistics

Algorithm		Mean	Std. Dev.
Const. Vel. EKF	X Error (m)	0.47	1.76
	Y Error (m)	-0.12	2.89
	ψ Error (deg)	-10.79	55.88
	RPV Error Norm (m)	2.47	2.38
Veh. Dyn. EKF	X Error (m)	-0.06	0.43
	Y Error (m)	0.06	0.61
	ψ Error (deg)	-0.69	4.09
	RPV Error Norm (m)	0.61	0.44

5.3 Cooperative Results

Rather than repeating *all* of the previously-shown non-cooperative scenarios, only the two dynamic scenarios were selected to highlight key points of analysis for the cooperative methods. Combining all of the potential options described in the previous chapter in various forms, the following collaborative methods were considered:

1. VehDynEKF with Feedback of tracked-vehicle states (longitudinal velocity, v_{Lx}, effective steer angle, d_L, yaw rate, ω_L)

   (a) v_{Lx}

(b) d_L

(c) ω_L

(d) v_{Lx} and d_L

(e) v_{Lx} and ω_L

(f) d_L and ω_L

(g) v_{Lx} , d_L and ω_L

2. Consensus EKF with constant velocity model

3. Consensus EKF with vehicle model

The feedback terms all have random noise applied to them on varying orders of magnitude: $\sigma_{v_L} = 0.1$, $\sigma_{d_L} = 0.01$, and $\sigma_{\omega_L} = 0.001$. Since it is assumed the UWBs are passing this information along with the range measurements, these feedback measurement updates happen alongside the range updates at 15 Hz. The consensus algorithms will be explored separately.

5.3.1 Scenario 1: Dynamic U-Turn

The same dynamic U-turn scenario was performed to compare the cooperative and non-cooperative results. A Monte Carlo analysis with 100 runs was again performed for each collaborative algorithm. The mean norm error over time is shown for each method in Figure 5.19.

Although there are improvements in accuracy with the inclusion of state feedback terms, there is not a clear best option across the entire run as slightly different error dynamics arise for each case. This mainly arises due to the assumptions made in the time update of the estimation algorithm. Recall that the lateral velocity is assumed to be zero for both the ego and the tracked vehicle. Therefore, feedback of dynamic states has the effect of increasing confidence in a model that does not fully encapsulate the system dynamics. Although this

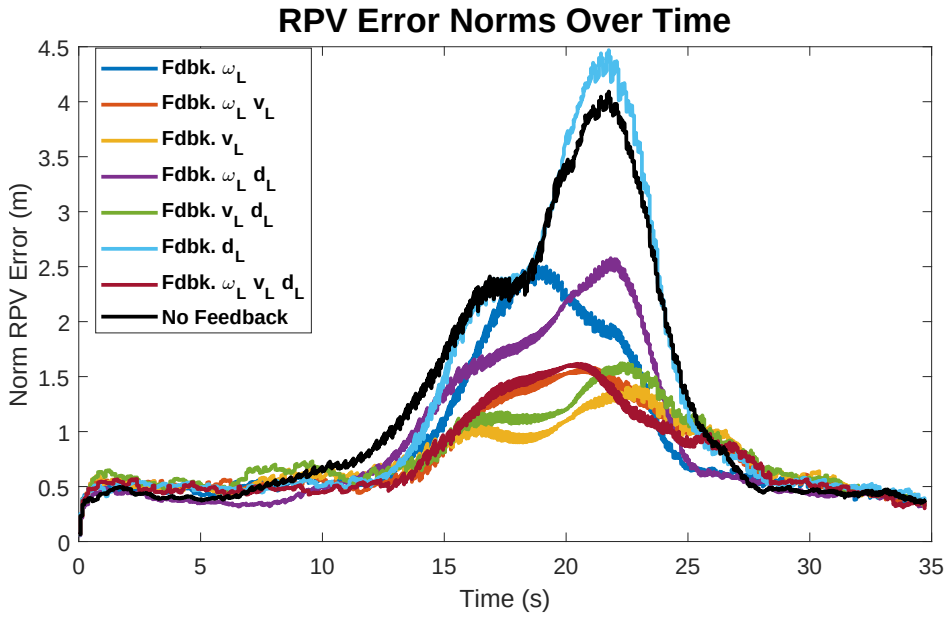


Figure 5.19: Cooperative U-Turn RPV Errors Through Run

feedback can help accuracy since the model is a decent approximation of truth, it can also lead to increased errors in some situations. In particular, the error increases when the vehicles perform turns and the zero-lateral-velocity assumption is violated. To quantify this effect, the absolute sum of the vehicles' respective lateral velocities was calculated and the errors were sorted with respect to this. Figure 5.20 shows the resulting trends. Note that a moving average smoothing operation was performed to enhance readability.

The trend in this plot provides support for the predicted error being significantly correlated with the lateral velocities of the vehicles. Regardless of the employed method, the accuracy generally suffers as the lateral velocity assumption is violated. If the ego vehicle had access to a sufficient lateral velocity measurement, such as from a side-slip sensor or an error-corrected IMU solution, then only the tracked vehicle's dynamics must be subject to the zero-lateral-velocity assumption. The accuracy could be improved significantly as a result. The same feedback methods were re-run assuming the ego vehicle had access to a noisy ($\sigma = 0.1\frac{m}{s}$) measurement of its lateral velocity. The results, shown in Figure 5.21, demonstrate clear improvements with the feedback terms which include a velocity estimate from the tracked vehicle.

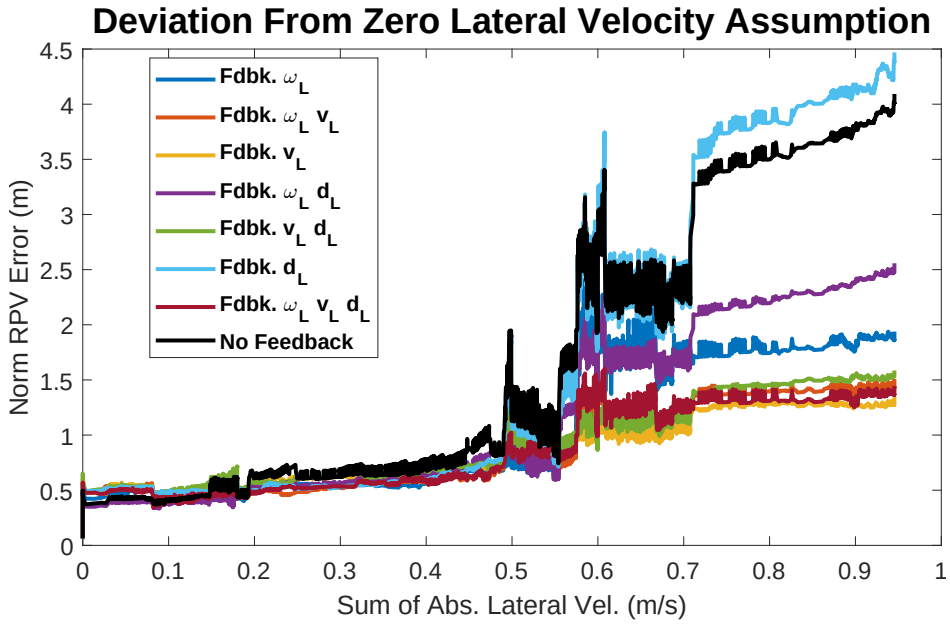


Figure 5.20: U-Turn Cooperative Method Errors Sorted By Cumulative Lateral Velocity

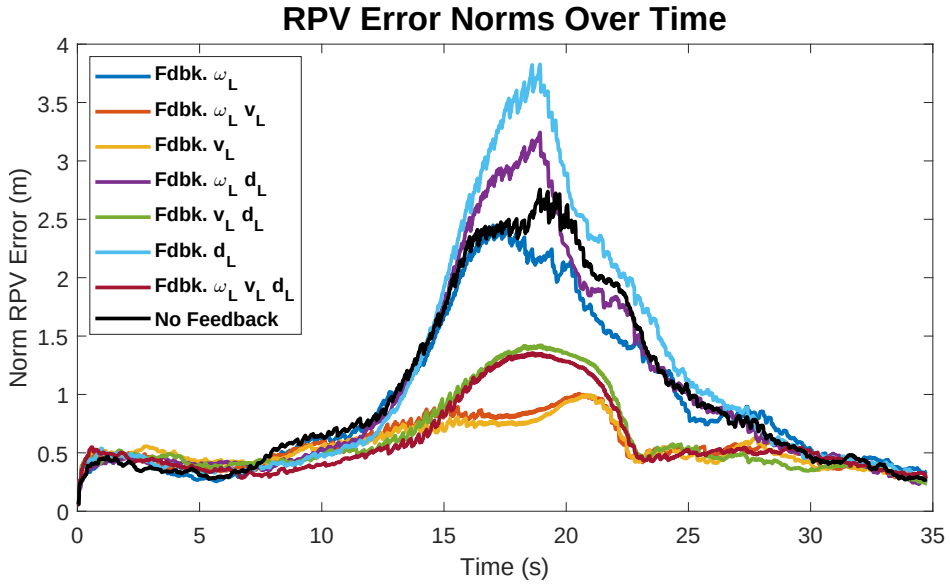


Figure 5.21: U-Turn Cooperative Method Errors Throughout Run with Ego Lateral Vel. Measurements

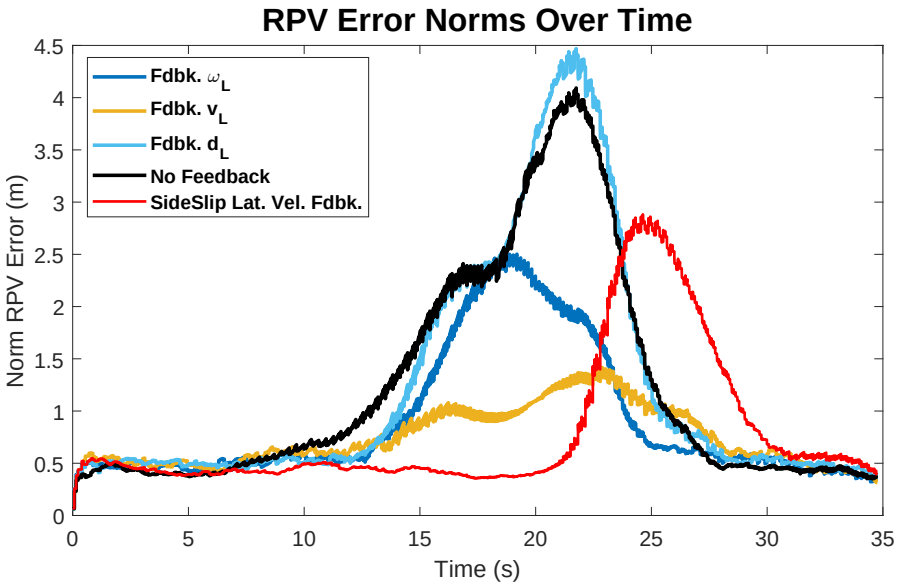
There is an outlying feedback method which should be highlighted before moving forward. Notice that the feedback methods including effective steer angle (d_L) generally perform worse than their counterparts. The light blue line which represents only the effective steer angle feedback is even worse than the baseline approach. By inspecting the derivation of the kinematic bicycle model in Chapter 2, it can be seen that the input steer angle is geometrically related to the side slip experienced by the vehicle. However, the estimator still is restricted by the zero-lateral-velocity assumption. In the Kalman Filter framework, this leads to a high confidence in the current estimate of the tracked vehicle's steer angle and a corresponding resistance to change of this estimate based on measurements from the UWB ranging. The results in the estimator struggling to deviate from the zero-side-slip model and leads to an increase in error. To highlight this, an additional test was performed where the effective steer feedback was related to a sideslip, β, which was then used to provide a direct estimate of the lead vehicle's longitudinal and lateral velocities through:

$$V_x = |V| \cos(\beta) \tag{5.6}$$

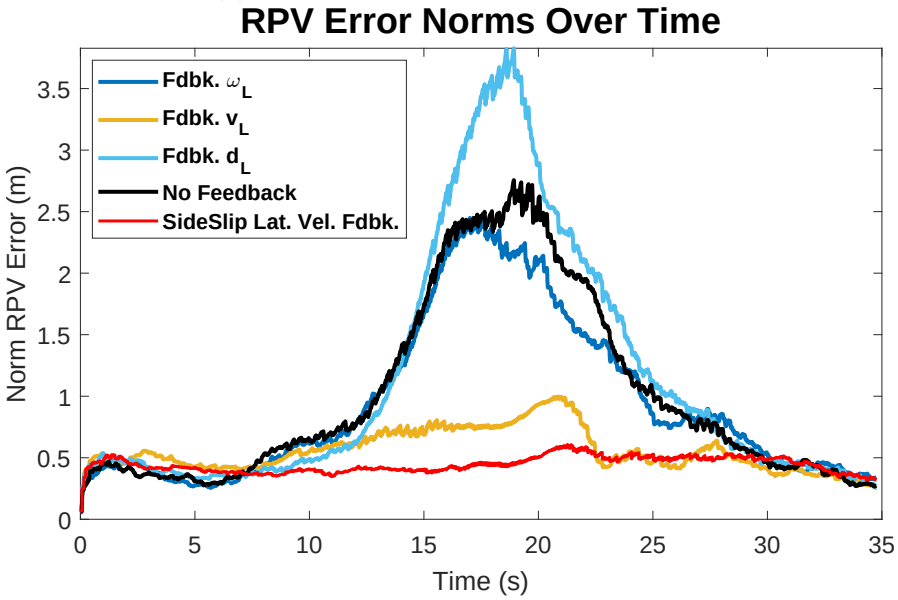
$$V_y = |V| \sin(\beta) \tag{5.7}$$

This modified feedback method was employed on the scenario and is compared to a few of the other methods in Figure 5.22.

The red lines in these figures represent this modified feedback method. The results without a lateral velocity on the base vehicle show a reduced error over the effective steer feedback method (light blue), but it performs worse than the other options. A lag is also present. These aspects are seen since the following vehicle still is restricted to the zero-lateral-velocity assumption and that vehicle experiences sideslip later in the run. When the following vehicle also has a lateral velocity measurement for itself, the best potential results arise since the estimator now matches the model directly. This is shown with the sideslip lateral velocity feedback method in Figure 5.22b.



(a) Without Lateral Velocity on Base



(b) With Lateral Velocity on Base

Figure 5.22: Error Norm Comparison of Sideslip-Estimated Lateral Velocity Feedback

The results are promising but there are a few key issues with employing this method in practice. In particular, the accuracy experienced here is mostly due to the estimator dynamics matching the true dynamics perfectly — something which is not true with employing a simple kinematic model on real vehicles. Secondly, the equation for sideslip generation from input steer (Equation 2.29) requires knowledge of the lead vehicle's COG offset from the rear axle. This is more involved than knowledge of the wheelbase which is required for the effective steer feedback and also has potential to vary with loading conditions. For these reasons, this feedback method will not be explored further in experimentation.

For comparison of the methods with a single value, the root mean squared error (RMSE) was taken of each method's RPV error norm. The results for both the lateral and non-lateral ego vehicle measurement cases are shown in Table 5.4.

Table 5.4: Scenario 1: Dynamic U-Turn Cooperative Result Comparison

RMSE of Cooperative RPV Estimates (m)		
Fdbk. Method	No Ego V_y	With Ego V_y
None	1.77	1.42
v_L	0.93	0.70
d_L	1.81	1.63
ω_L	1.17	1.28
v_L and d_L	1.03	0.81
v_L and ω_L	0.95	0.71
d_L and ω_L	1.17	1.43
v_L, d_L, and ω_L	0.96	0.76

Within the consensus implementation, the assumption is that both vehicles are interested in relative positioning to one another. Therefore, rather than only analyzing the error of only one estimator, the cumulative errors of the network are the metric. Figure 5.23 shows a comparison of the traditional algorithms versus one with consensus between the two-vehicle network with a consensus hyper-parameter of $\lambda = 0.1$.

For this run, the consensus algorithm does not present a significant change in results for the vehicle-dynamic EKF network, but there is a noticeable improvement in the RPV mean error for the constant velocity case. Recalling the results from the previous section,

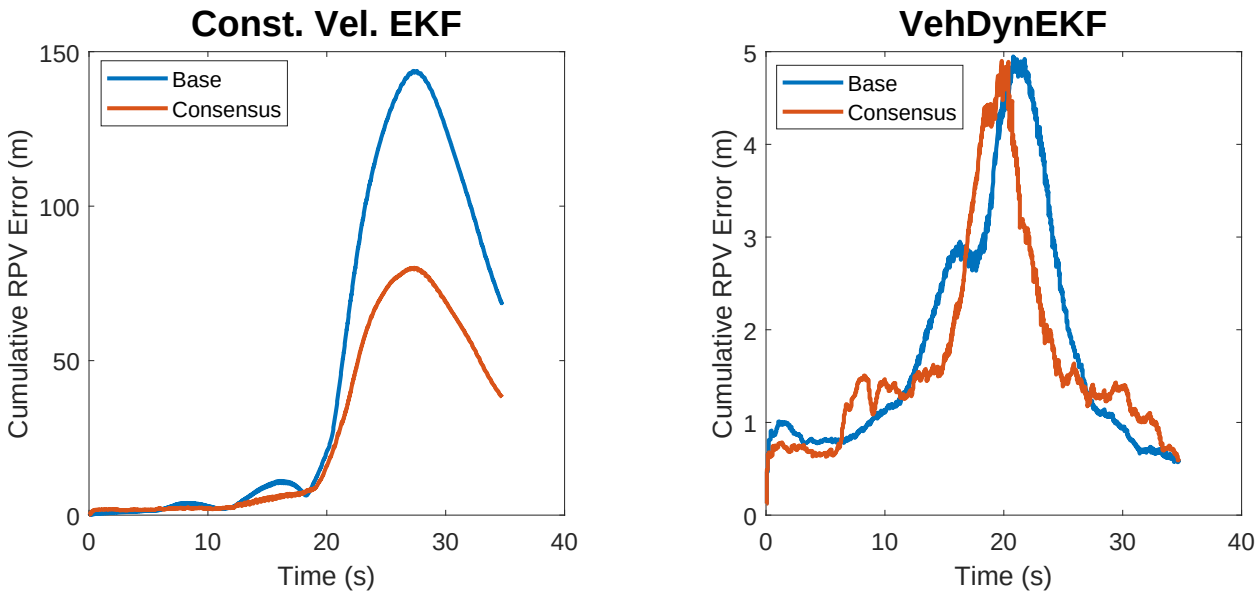


Figure 5.23: U-Turn Consensus EKFs for Each Propagation Model (varying y-axis scales)

the constant velocity method went unstable through the unobservable region for all 100 runs of the trajectory. Here, with a consensus constraint, a portion of the runs managed to track through the this section of the path. This is shown in Figure 5.24 which can be compared to Figure 5.12. Only the RPV_x component is shown as this is the component which suffers the most from the unobservable region. Although this consensus method does provide the constant velocity method an opportunity to pass through observability-faults, it is certainly not guaranteed. Of the 100 runs, 69 managed to re-converge to the proper quadrant solution while passing through the unobservable region.

One more aspect of this consensus application should be highlighted. Recalling from the previous sections, the constant velocity method struggled to track the relative heading of the two vehicles. This is particularly problematic because the consensus condition requires the relative heading when applying the RPV consensus condition in Equation 4.20. In [30], which first applied this geometric consensus condition, the author avoided this issue by using a magnetometer on each vehicle for relating their RPV estimates to a common frame. To explore this effect, the constant velocity consensus was performed again assuming the relative heading was measured ($\sigma = 0.01$rad), such as from a magnetometer on each vehicle.

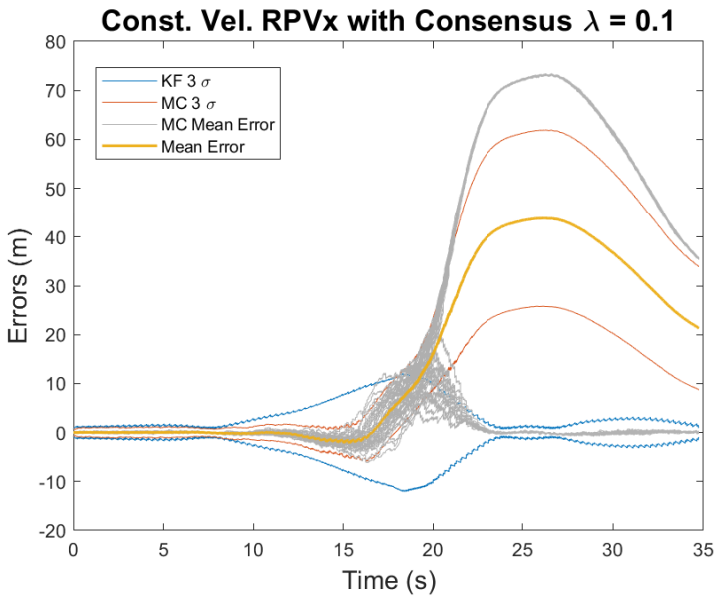


Figure 5.24: U-Turn Consensus EKFs for Each Propagation Model (Varying Y-Axis Scales)

This relative heading is also a state of the estimator, so if it is available for the consensus conditions, one should assume it could also be used as a direct measurement outside of a consensus framework. Therefore, the consensus will be compared to a constant velocity implementation which is aided by a relative heading measurement update. This allows for a fair comparison of how the consensus may assist the estimation and is shown in Figure 5.25. The same experiment was performed for the vehicle-dynamic EKF in Figure 5.26.

The improvement of the relative heading addition to the constant velocity method is clear. An even further improvement is seen from applying a consensus update along with the relative heading corrections. Figure 5.27 shows the improvements of these additions for a single run of the constant velocity method. Since the interest lies in the error of the full network, both vehicle positions are estimated and shown here.

The improvement seen from inclusion of relative heading measurements with the consensus condition in the constant velocity method is less noticeable for the vehicle dynamic method. Since the vehicle-dynamic EKF manages to track the relative heading quite well in all scenarios, the relative heading component of the consensus condition is not the primary

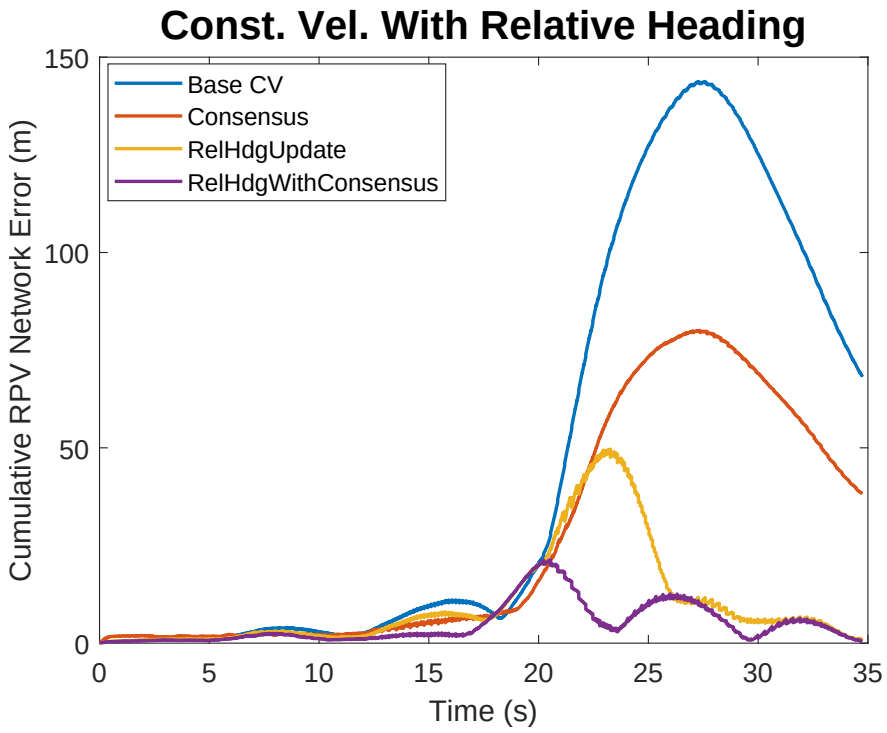


Figure 5.25: U-Turn Constant Velocity Consensus EKFs with Relative Heading Measurements

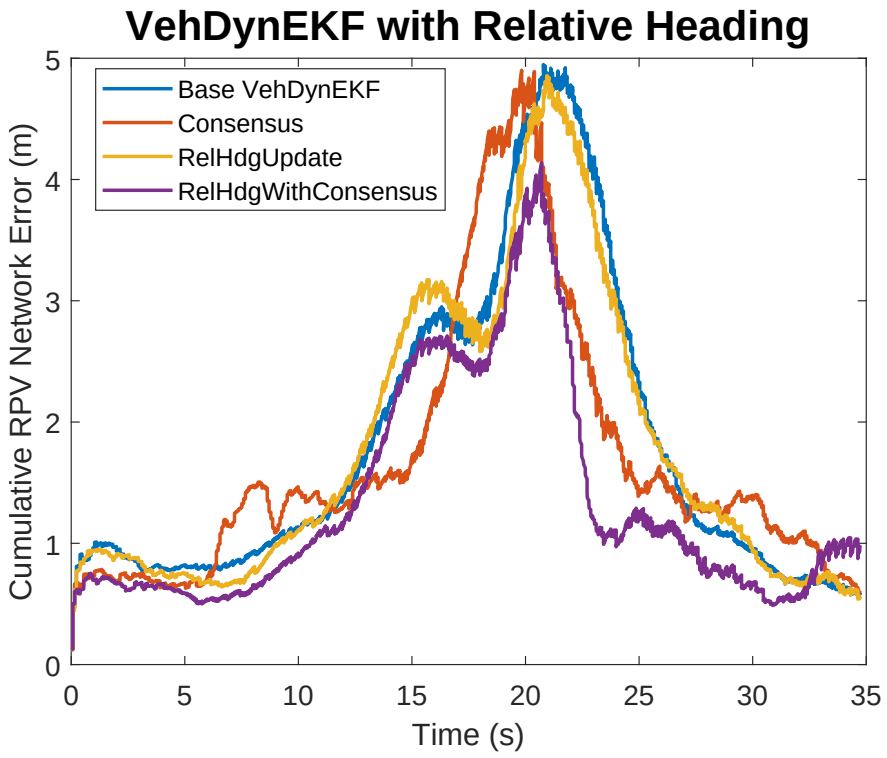
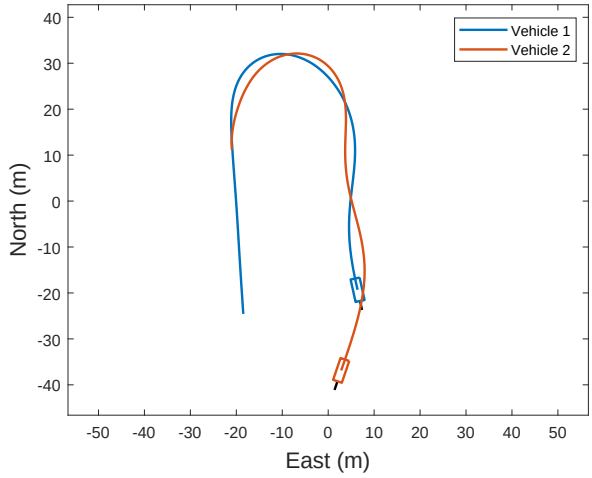
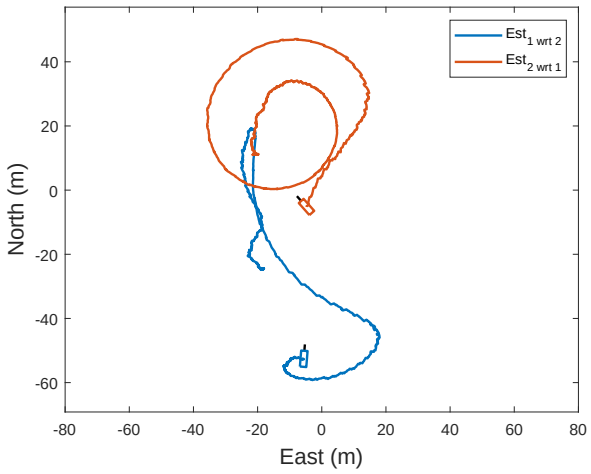


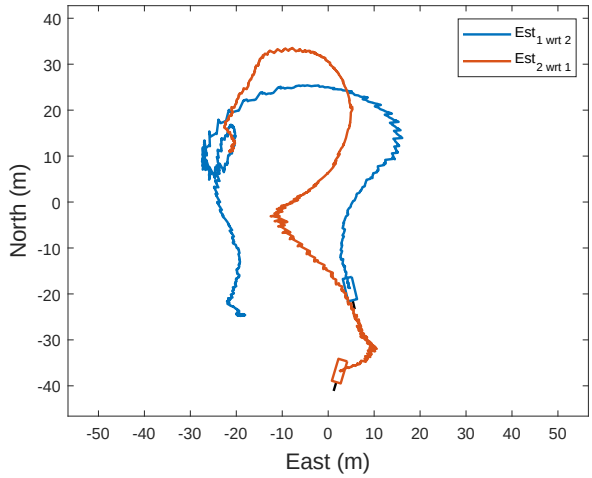
Figure 5.26: U-Turn Vehicle Dynamic Consensus EKFs with Relative Heading Measurements



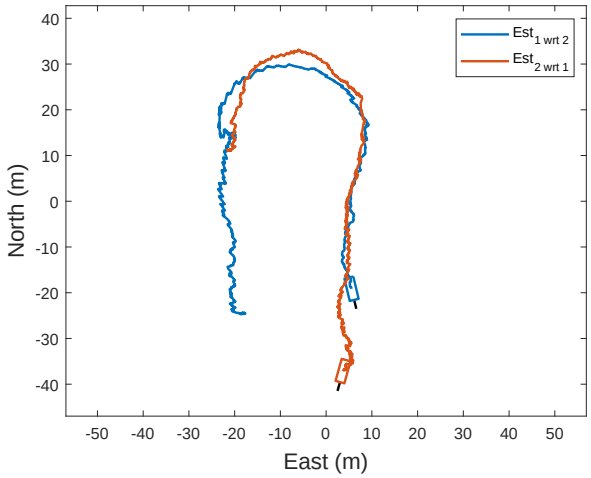
(a) True Paths



(b) Base Const. Vel. EKF



(c) Const. Vel. EKF with Rel. Hdg.



(d) Const. Vel. with Rel. Hdg. and Consensus

Figure 5.27: Dynamic U-Turn Constant Velocity Method Comparisons

source of error. This is verified in Figure 5.26 where the inclusion of a relative heading measurement provides little change compared to the base vehicle-dynamic EKF. The inclusion of the relative heading measurement along with the consensus condition does represent an improvement, but it is less pronounced than the constant velocity application.

This scenario has shown that the consensus condition could be helpful to the constant velocity method if on-board odometry is unavailable for the vehicle dynamic method to be employed. With the application of a relative heading measurement, the consensus shows an even further improvement. Whether or not the consensus addition should be applied will depend on how it performs in more diverse runs, such as when the filter is not predisposed to instability. The next scenario will examine this.

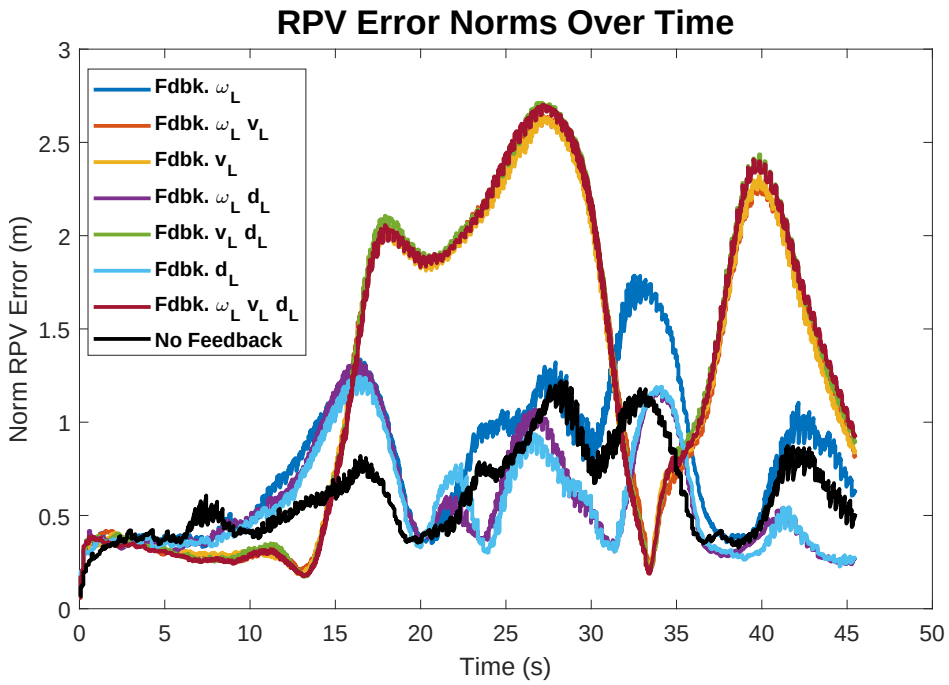


Figure 5.28: Figure-8 Vehicle-Dynamics EKF RPV Errors Through Run

5.3.2 Scenario 2: Figure-8 Maneuver

In this run, the vehicles are performing alternating turns frequently; therefore, the lateral velocity assumption is violated even more than in the previous scenario. The RPV errors for the varying methods in this scenario are shown in Figure 5.28. From initial inspection of these results, it can be seen that some of the feedback options actually degrade the accuracy compared to the non-cooperative solution. Figure 5.29 shows the RPVx and RPVy solutions from the longitudinal-velocity feedback. In comparison to the same non-cooperative results in Figure 5.17, the solution degradation is apparent. Not only do the errors grow, but also the filter no longer properly bounds the estimate errors.

Sorting the errors by the cummulative lateral velocities provides more insight to the error trends in the case without the ego vehicle's lateral velocity measurement. This is depicted in Figure 5.30.

Notice that for this figure-8 scenario the velocity feedback options have the highest error with respect to the cumulative lateral velocity. In the previous scenario, these same

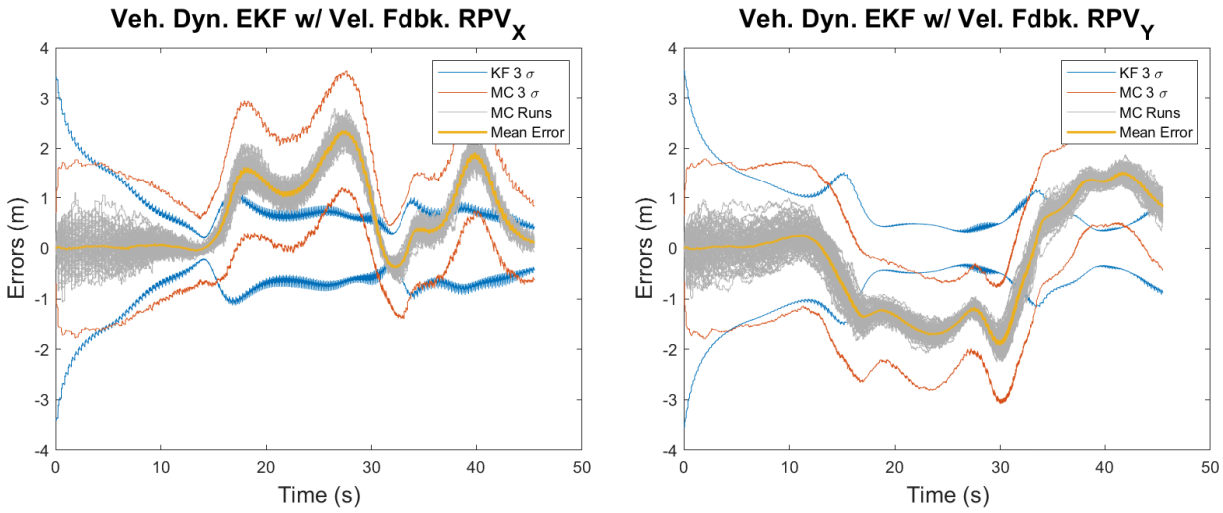


Figure 5.29: Figure-8 VehDynEKF Results with v_L Feedback without Ego-Vehicle Lateral-Velocity Measurement

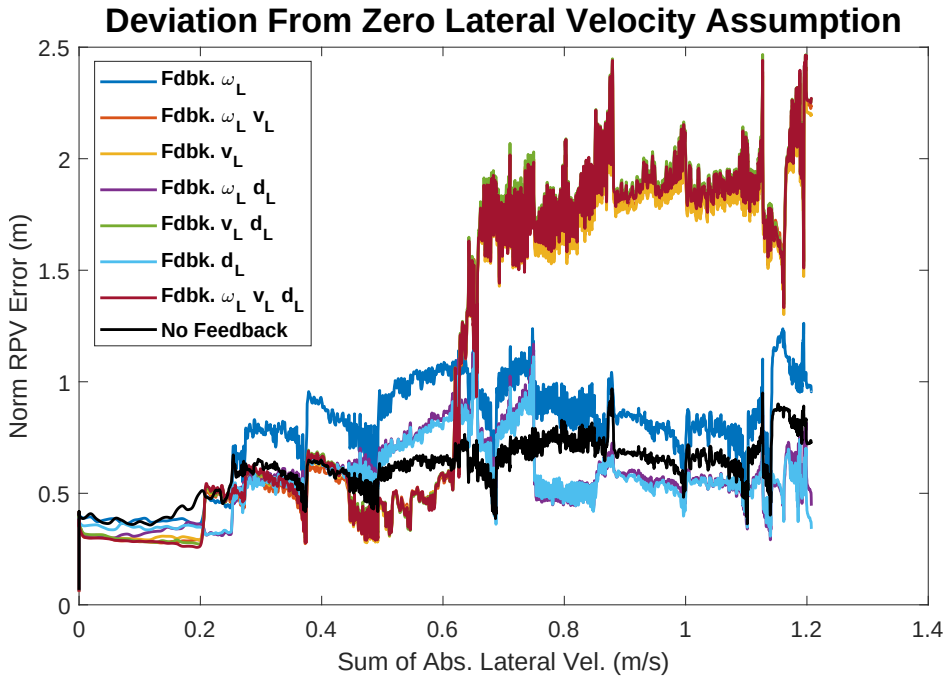


Figure 5.30: Figure-8 Cooperative Method Errors Sorted by Cumulative Lateral Velocity

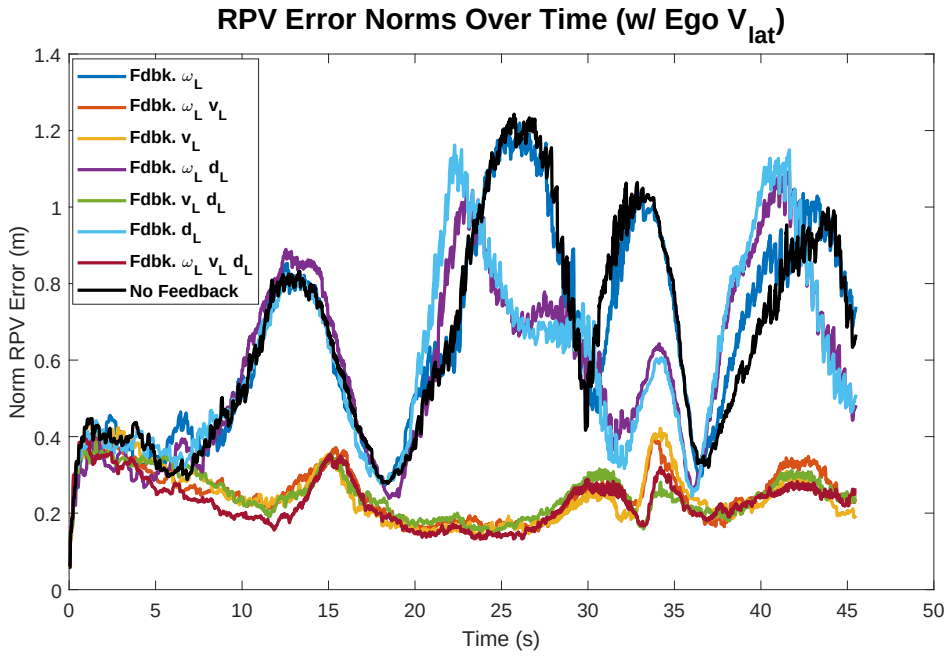


Figure 5.31: Figure-8 Cooperative Method Errors Throughout Run with Ego Lateral Vel. Measurements

feedback methods showed some of the lowest sensitivity to lateral velocity. This is likely to do with the differences in the ego vehicle's dynamics in each test. In the dynamic u-turn, the ego only performed one large turn. Meanwhile, in the figure-8 run, the direction of turning was oscillatory and the side-slip angle has changing signs. Nevertheless, including a measurement of the lateral velocity of the ego vehicle once more significantly improves the estimation shown in Figure 5.31.

Figure 5.32 sorts the errors with respect to the cumulative lateral velocity once more. Comparison to the same plot without the ego vehicle's lateral velocity measurement, Figure 5.30, shows that the feedback options with the highest correlation of error with lateral velocity are now the lowest: the velocity feedback terms. Consequentially, these methods are also the best performing as previously shown in Figure 5.31. Table 5.5 summarizes the feedback results with and without the ego vehicle's lateral velocity measurement.

The consensus cumulative error results for the two-vehicle network are depicted in Figure 5.33. Similar to the previous scenario, the consensus condition presents an improvement

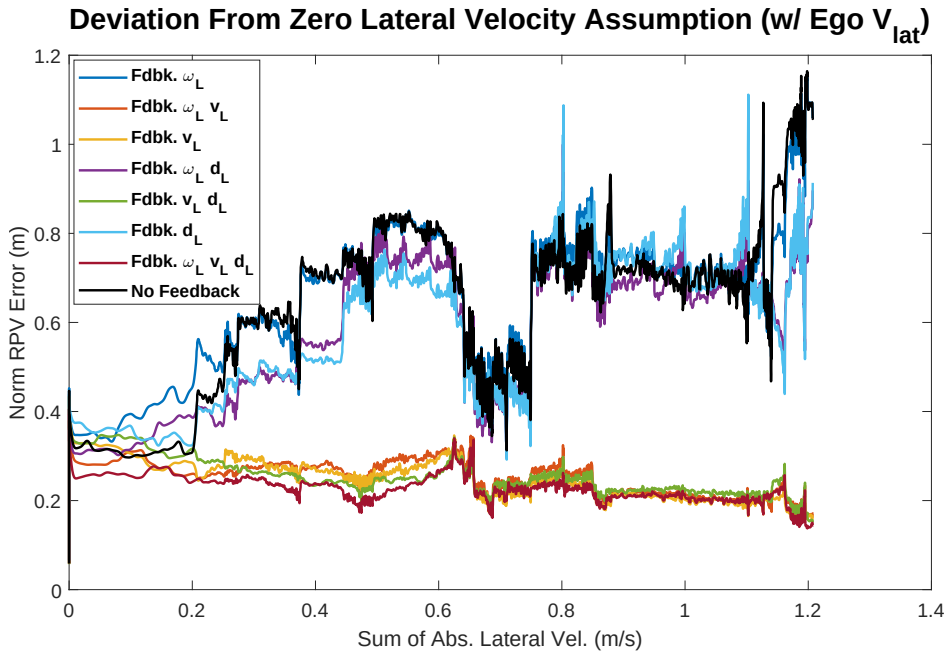


Figure 5.32: Figure-8 Cooperative Method Errors Sorted by Cumulative Lateral Velocity (with Ego Lateral Vel. Measurement)

to the constant-velocity method even without the relative heading measurement. The vehicle-dynamic EKF consensus results also demonstrate a clear improvement in this scenario.

Inclusion of a relative heading measurement is depicted for the constant velocity method and Figure-8 maneuver in Figure 5.34. As in the dynamic U-turn scenario, the inclusion of a relative heading measurement represents a significant improvement. The consensus algorithm improves upon this even further. These results along with the previous scenario provide a

Table 5.5: Scenario 2: Figure-8 Cooperative Result Comparison

RMSE of Cooperative RPV Estimates (m)		
Fdbk. Method	No Ego V_y	With Ego V_y
None	0.76	0.75
v_L	1.50	0.30
d_L	0.69	0.72
ω_L	0.95	0.79
v_L and d_L	1.54	0.302
v_L and ω_L	1.51	0.306
d_L and ω_L	0.72	0.71
v_L, d_L, and ω_L	1.53	0.28

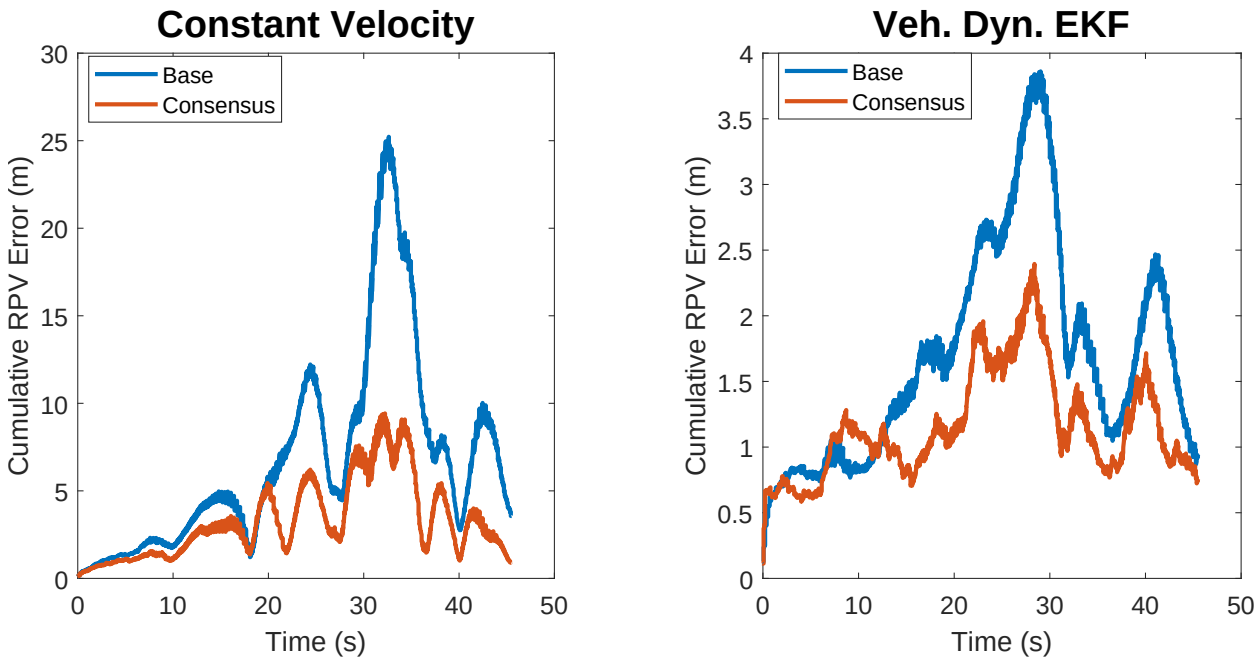


Figure 5.33: Figure-8 Consensus EKFs for Each Propagation Model

strong case for the inclusion of a consensus condition to the constant velocity algorithm if on-board odometry is unavailable.

The relative heading inclusion to the vehicle-dynamic EKF once more provides little change over the base algorithm. Unlike the previous U-turn scenario, the consensus algorithm does show more significant improvements over the base methods. Comparisons of the consensus results with the relative heading feedback is shown in Figure 5.35.

Since the vehicle-dynamic EKF has two distinct cooperative methods, feedback and consensus constraints, the two are compared with the same cumulative network norm in Figure 5.36. The consensus condition's improvement is notable, but it is slightly outperformed by the top performing feedback methods. The selection between these methods will be determined by their performance during experimental tests in the following chapter.

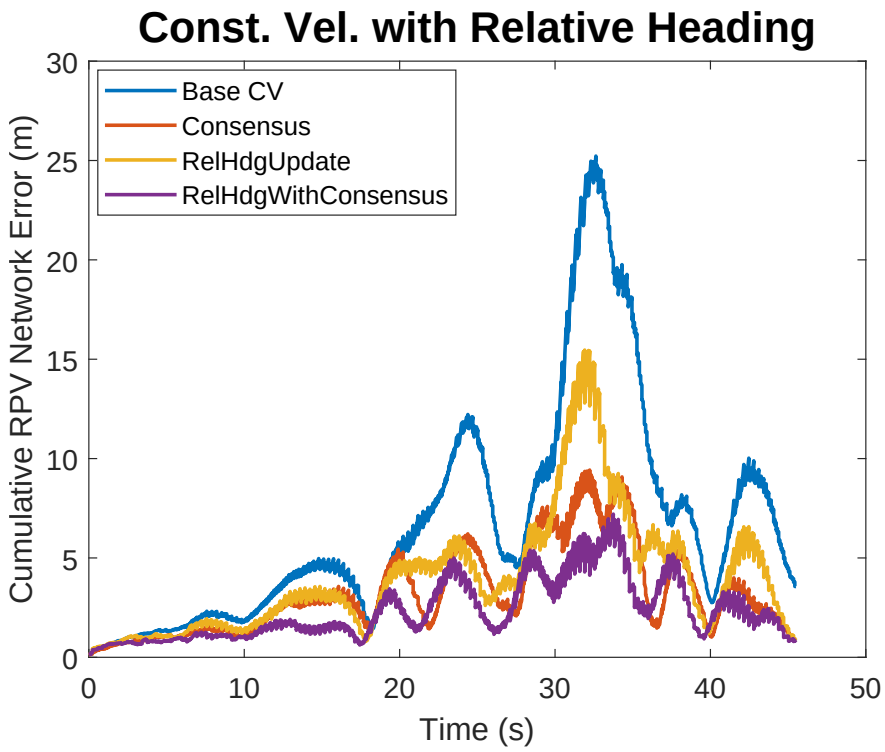


Figure 5.34: Figure-8 Constant Velocity Consensus EKFs with Relative Heading Measurements

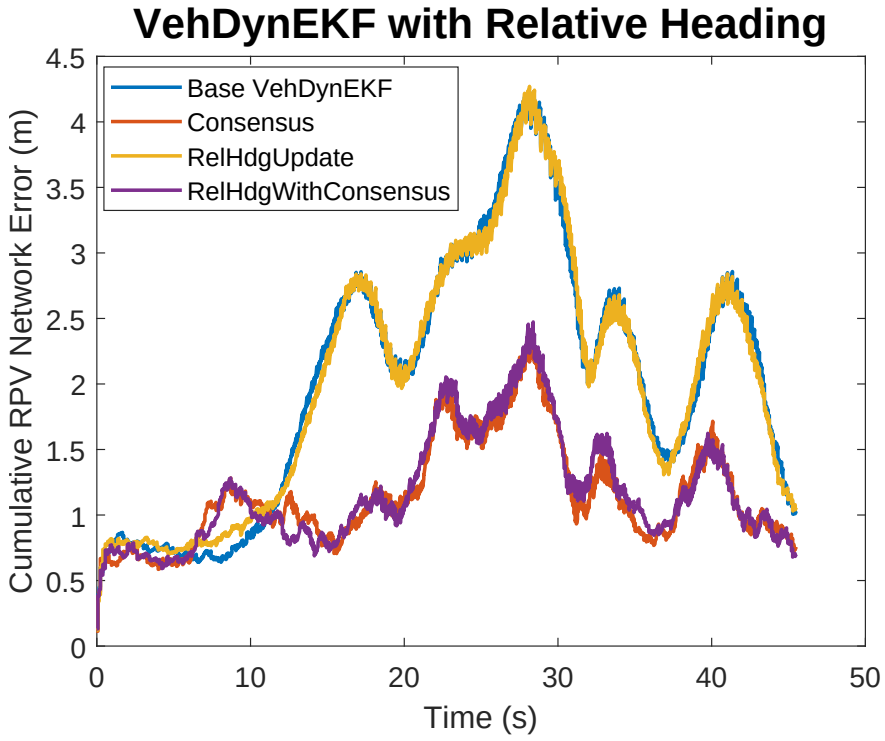


Figure 5.35: Figure-8 Vehicle-Dynamic Consensus EKFs with Relative Heading Measurements

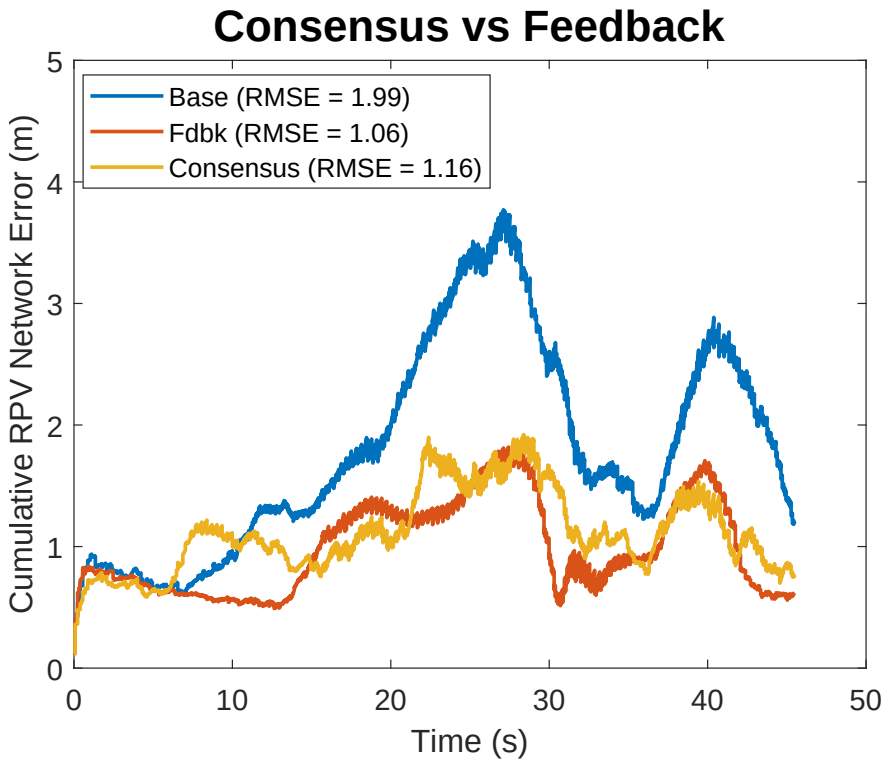


Figure 5.36: Figure-8 Vehicle-Dynamic Cooperative Method Comparison

5.4 Conclusion of Simulation Results

The simulation analysis performed in this chapter outlined the performance of both cooperative and non-cooperative methods for ground-vehicle relative positioning with UWBs. The main conclusions are as follows:

1. Without communication between vehicles, the proposed VehDynEKF substantially outperforms the constant velocity method in both high and low-observability relative geometries

2. Feedback of the tracked vehicle's dynamics provides useful information for the VehDynEKF and can improve the estimation results. However, without a measurement of the ego vehicle's lateral velocity, the potential exists for the some feedback methods to degrade the solution instead. This is due to feedback measurements providing a highly confident solution in a model that does not fully encapsulate the true dynamics of the

system. In particular, the zero-lateral-velocity assumption is a large source of error, so inclusion of a measurement of the ego vehicle's lateral velocity improves the impact of the feedback terms and provides the highest estimation accuracy. This could also be a by-product of the dynamic model used in the simulation generation, so it will be re-analyzed in the experimental section.

3. If no on-board odometry is available, the consensus constraint can improve the constant-velocity model if the vehicles may communicate with one another. The consensus provides the opportunity for the constant velocity method to sometimes overcome the major pitfalls experienced through unobservable areas, but this is not guaranteed. Further improvement is seen if a relative heading measurement is available to aid in the consensus construction since the traditional constant velocity method often does not track this state well. However, the results still underperform the VehDynEKF.

4. The VehDynEKF also experiences a benefit from a consensus constraint in some scenarios, but it is not required to overcome observability issues. If the vehicles are capable of communication and each are employing the VehDynEKF, the dynamic states can shared instead which offers slightly improved performance. These two cooperative methods will be compared in experimental applications.

Chapter 6

Experimental Results

6.1 Experimental Setup

The experimental data was collected on two passenger vehicles, a retrofitted drive-by-wire Lincoln MKZ and a Kia Optima. The UWBs were situated across the roof-rack of each vehicle with a GPS antenna bisecting their baselines. The aim for the spacings of the UWBs was to match the simulation of 6-foot baselines, but the resulting separations were 71.5 and 67.25 inches, respectively. The antenna signals were fed into a splitter with one output connected to an RTK-capable receiver and one connected to a GPS/INS solution utilized for heading. The two vehicles are shown in Figure 6.1 (a more detailed depiction of the UWB placement was previously shown in Figure 2.17).

The odometry of the estimating vehicle (MKZ) was obtained with a wheel encoder and gyroscope. The bicycle model mechanization was employed as was done in simulation. The MKZ has embedded encoders in each wheel. The average of the two rear wheels was taken as the longitudinal velocity. The gyro was a KVH DSP 1760 fiber optic single-axis gyroscope shown in Figure 6.2.

As explored in the simulation results, lateral velocities on the ego vehicle may improve the estimation significantly. This lateral measurement is not as easily generated as a longitudinal estimate. Although it does not align with the sensor suite of this thesis, this lateral velocity was generated by a high-quality GPS/INS system as proof of concept for its inclusion. Due to hardware constraints, the steer feedback is omitted from the experimental results, but prior simulation results showed it to be the least impactful option.

The TimeDomain P440s were configured with a PII value of 9 corresponding to a 14.9 Hz network update rate with an approximate range of 410 meters. Each range measurement

Figure 6.1: Experimental Passenger Vehicles: Lincoln MKZ and Kia Optima

Figure 6.2: KVH DSP 1760

has an individual update rate of around 3.75 Hz. The cooperative feedback terms from the tracked vehicle were not configured to pass through the UWBs in real-time, but they are synced in post-process to be delivered alongside the range measurements. This results in the cooperative updates having the same 14.9 Hz update rate as the UWBs — as in the simulations.

Two scenarios were chosen for the experimental analysis. The first is the dynamic U-turn maneuver to again examine the observability issues in this relative geometry. In the second scenario, the two vehicles moved around the track randomly to see how well the algorithms can perform in a variety of ranges, maneuvers, and relative geometries. The consensus condition is applied to the constant velocity method in both scenarios, but the VehDynEKF is only explored for the second scenario.

6.2 Scenario 1: Dynamic U-Turn

For this test, the vehicles pass by one another imposing the unobservability condition through the turn. The mean range between the vehicles was 21.4 meters. It should be noted and the UWBs had a dropout rate of ≈4%, but all of those were confined to the range between UWB 1 and 2 which had a dropout rate of ≈16%. The paths of the vehicles are depicted in Figure 6.3.

A comparison of the non-cooperative algorithms can be seen in Figure 6.4. In this figure, the relative positioning estimates were transformed into an local frame solution through the position and heading information of the GPS/INS system. The RPV error norms are depicted in Figure 6.5.

It was seen in simulation that a measurement of the ego vehicle's lateral velocity significantly improved estimation quality for the feedback and non-feedback cases. Both are explored here and are shown in Figure 6.6 with the same y-axis scales. The euclidean error norms are improved for each feedback method.

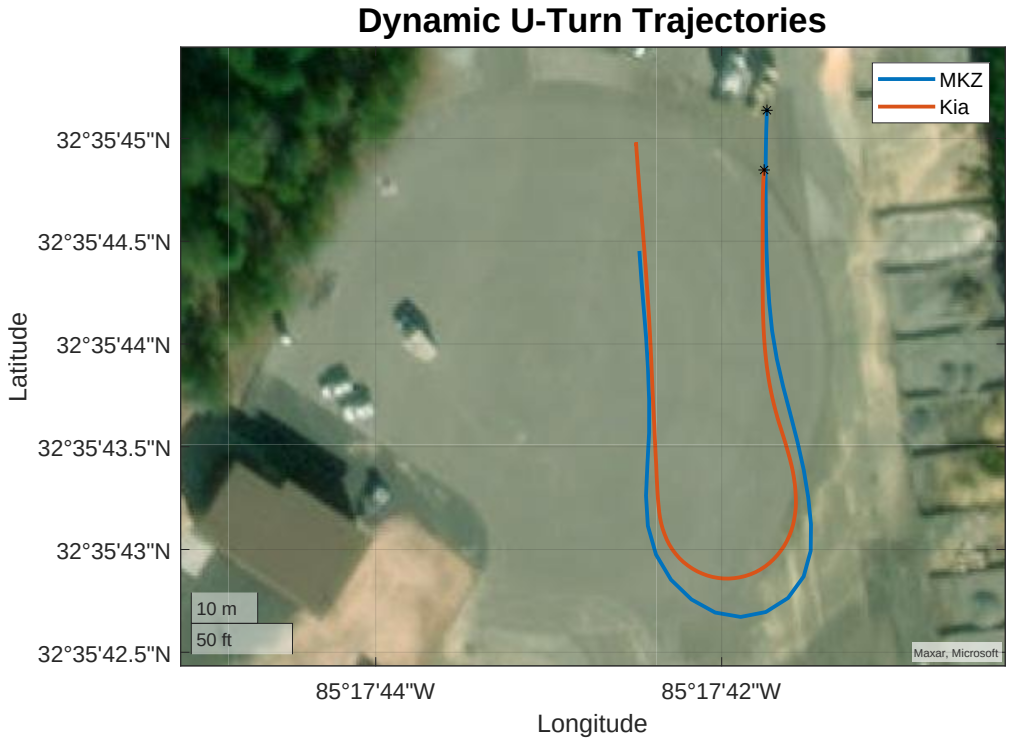


Figure 6.3: Dynamic U-Turn Experimental Scenario Trajectories

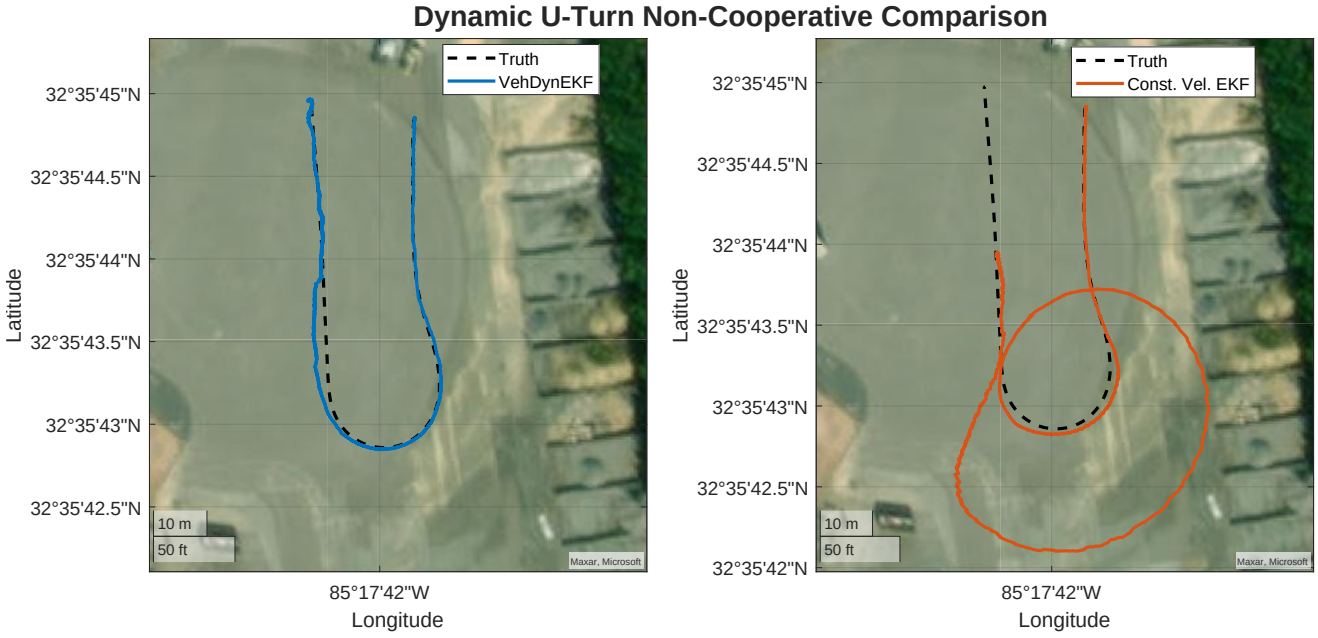


Figure 6.4: Dynamic U-Turn Non-Cooperative Estimation Results

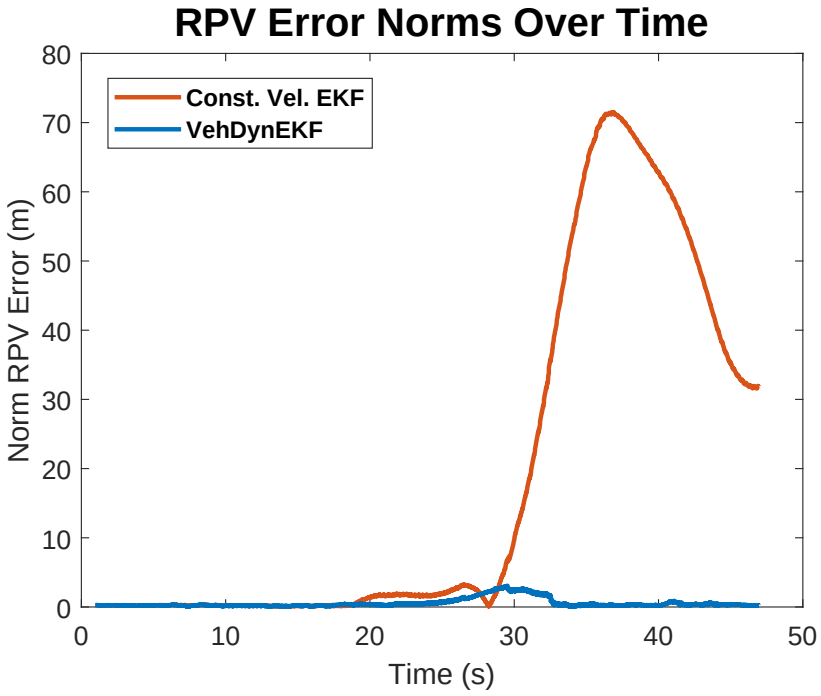


Figure 6.5: Dynamic U-Turn Non-Cooperative Error Norms

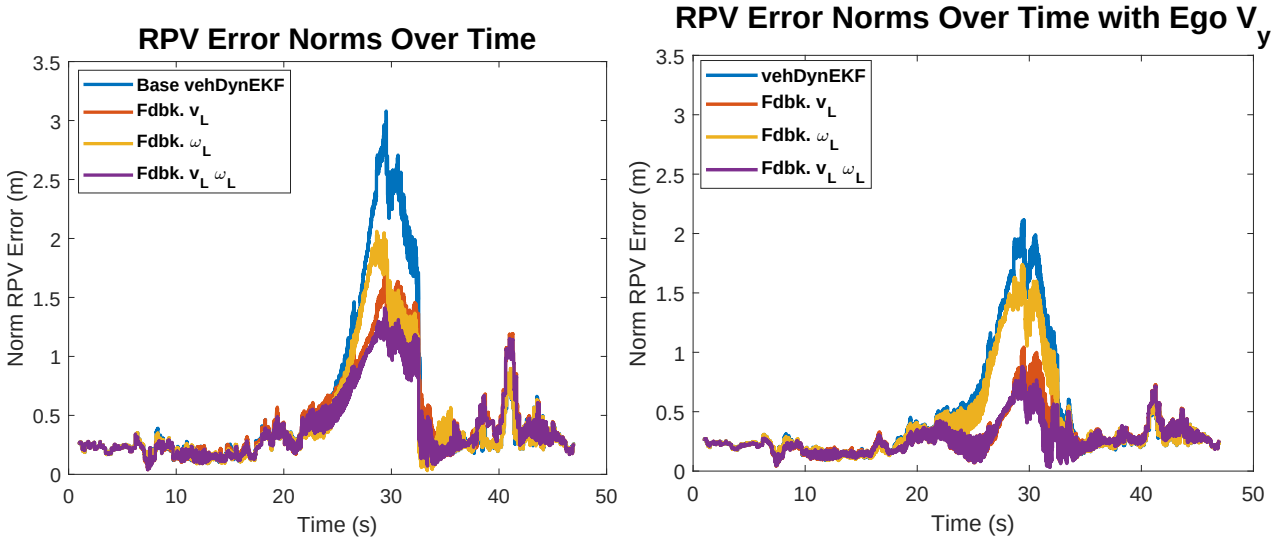


Figure 6.6: Dynamic U-Turn Feedback Results

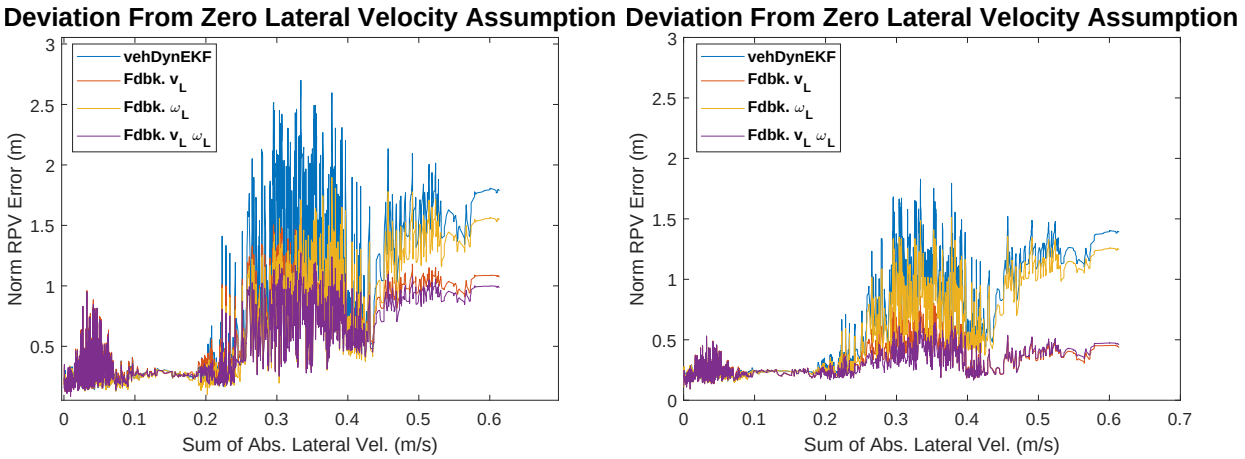


Figure 6.7: Dynamic U-Turn Feedback Results VS Cumulative Lateral Velocity

The results of the previous chapter predicted that the relative positioning errors are correlated with the violation of the zero-lateral-velocity assumption employed in the estimation algorithm. Figure 6.7 sorts the errors with respect to the cumulative lateral velocity in the network over time. This figure shows that, with the inclusion of the ego vehicle's lateral velocity, the feedback methods become less sensitive to the lateral velocity of the vehicles. This is especially true for the methods employing longitudinal velocity feedback.

Table 6.1 summarized the experimental results of the dynamic-state cooperative methods. The longitudinal velocity feedback proves to be the most useful dynamic state to be shared between vehicles, but the inclusion of both velocity and yaw rate provides the best results. The inclusion of lateral velocity measurements on the Ego vehicles refines the solutions substantially and allows for the state feedback to be more impactful. These results concur with those seen in the simulation studies.

Table 6.1: Experimental Dynamic U-Turn Cooperative Result Comparison

RMSE of Cooperative RPV Estimates (m)		
Fdbk. Method	No Ego V_y	With Ego V_y
None	0.86	0.62
v_L	0.59	0.34
ω_L	0.63	0.53
v_L and ω_L	0.51	0.31

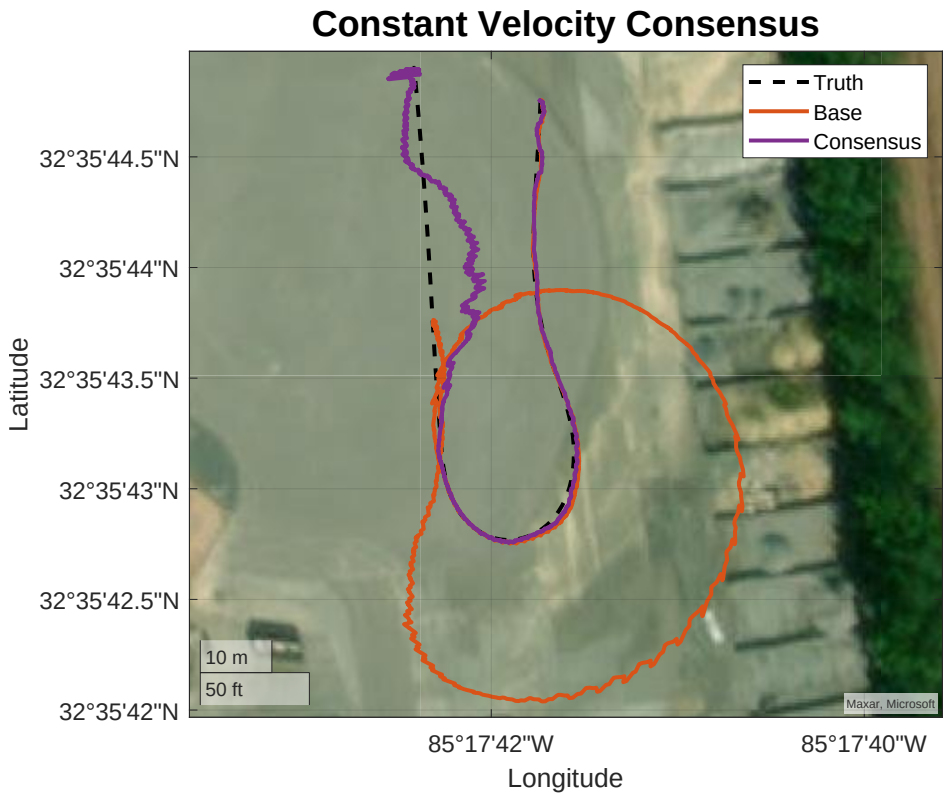


Figure 6.8: Dynamic U-Turn Constant Velocity Consensus Comparison

The consensus constraints were tested with the constant velocity model in this scenario to compare to the simulation results. In simulation, the consensus condition showed promise in aiding the constant velocity filter through the poor observability areas. As in simulation, the base form of the constant velocity EKF was unable to track the vehicle as they pass one another. When the consensus condition is applied, the estimate is able to recover for this scenario as shown in Figure 6.8. This agrees with the simulation results.

Simulation results showed how the inclusion of a relative heading also increases the likelihood of propagating through this region. Figure 6.9 compares these options as a cumulative error across the consensus network. Interestingly, the consensus condition performs better when the relative heading is excluded from the algorithm—this was not the case in simulation. This may be due to biases in the relative heading solutions generated from the differenced GPS/INS solutions; the relative heading measurements were unbiased and low

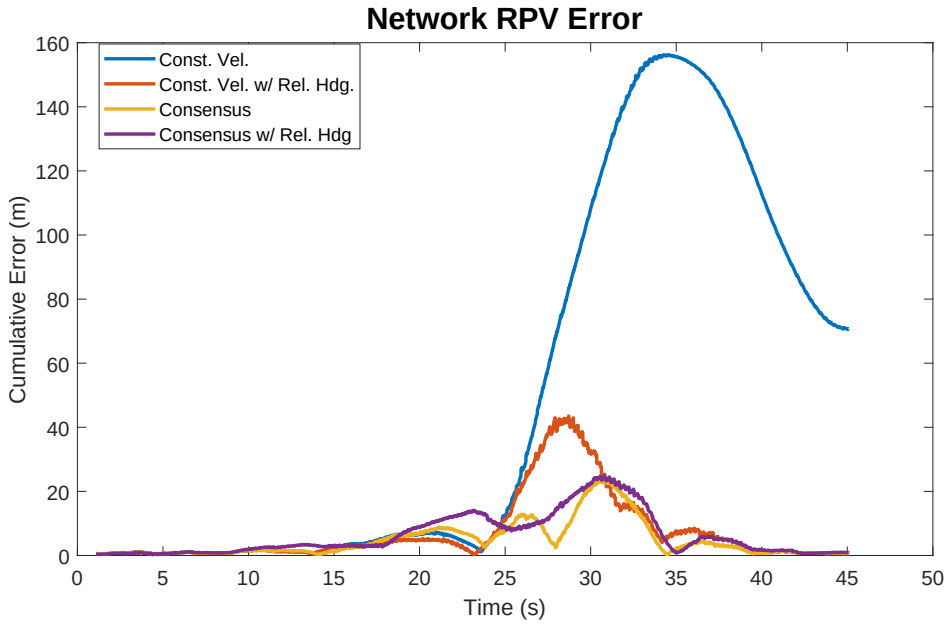


Figure 6.9: Dynamic U-Turn Consensus Cumulative Error Norms

noise in simulation. This scenario once again shows the potential improvement of a consensus condition application to a constant velocity implementation, but it still consistently underperforms the VehDynEKF.

6.3 Scenario 2: Random Driving

This scenario consisted of the two vehicles wandering the testing pad randomly. The drivers of the vehicles were instructed to speed up and slow down sporadically while also performing dynamic turns. The range between the vehicles varied from 14.1 meters to 58.2 meters with a mean of 31.9 meters. The ego vehicle had a maximum velocity of 7 $\frac{m}{s}$ and an average of 3.9 $\frac{m}{s}$. Meanwhile, the tracked vehicle had a maximum velocity of 10.5 $\frac{m}{s}$ and an average of 4.2 $\frac{m}{s}$. The paths of the vehicles are depicted in Figure 6.10.

This trajectory provided many opportunities for poor observability. The constant velocity method's solution broke down quickly in the run as a result. Meanwhile, the vehicle-dynamic EKF experienced spikes in error during the turns and further ranges, but remained

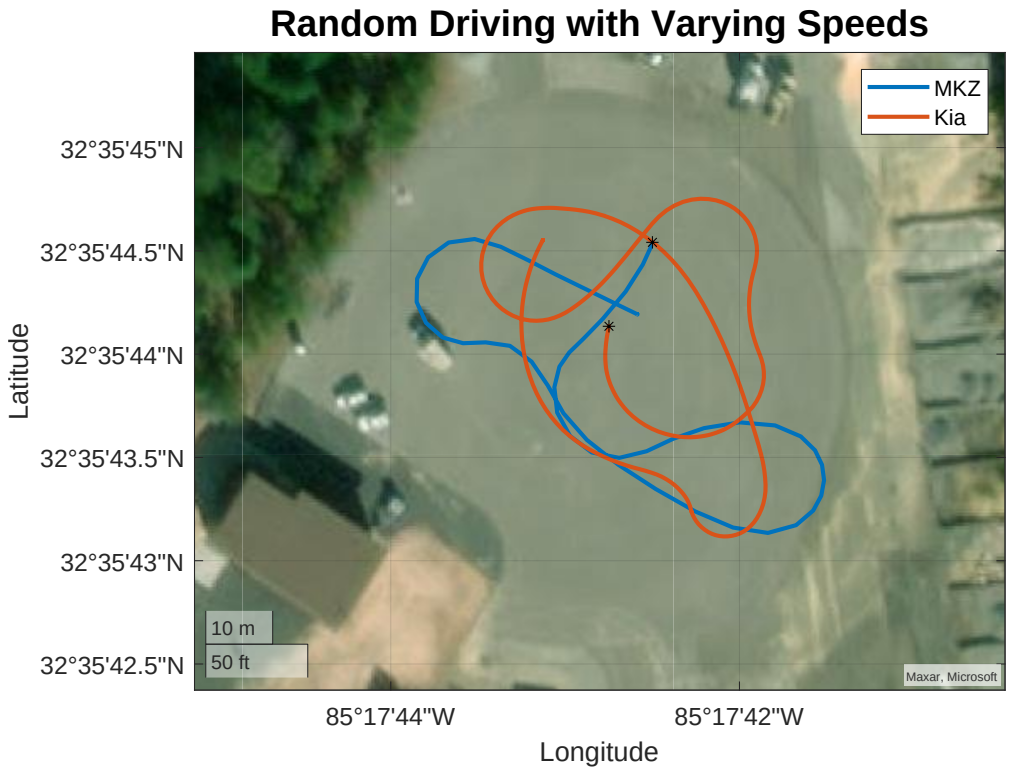


Figure 6.10: Random Maneuvers Experimental Scenario Trajectories

a consistent solution throughout. These results are shown on a geographical map in Figure 6.11. The RPV errors for this run are depicted in Figure 6.12

Improvements to the VehDynEKF solution are seen once more from the inclusion of lateral velocity of the ego vehicle. Again, the most beneficial feedback term continues to be the speed of the tracked vehicle. Figure 6.13 shows RPV error norms throughout the run. The error's dependency on lateral velocity remains true and can be seen in Figure 6.14.

Table 6.2 shows the promising results for the UWB RPV estimation algorithms presented in this thesis. The variable dynamics, range, and geometries experienced in this scenario result in an increased error in the estimation, but the results remain on the order of a few meters or less for all vehicle dynamic based methods.

The consensus application to the constant velocity model is once more explored. The error norms throughout the run are shown in Figure 6.15. The clear improvements seen in the previous scenario are missing in these results. Although the consensus methods outperform

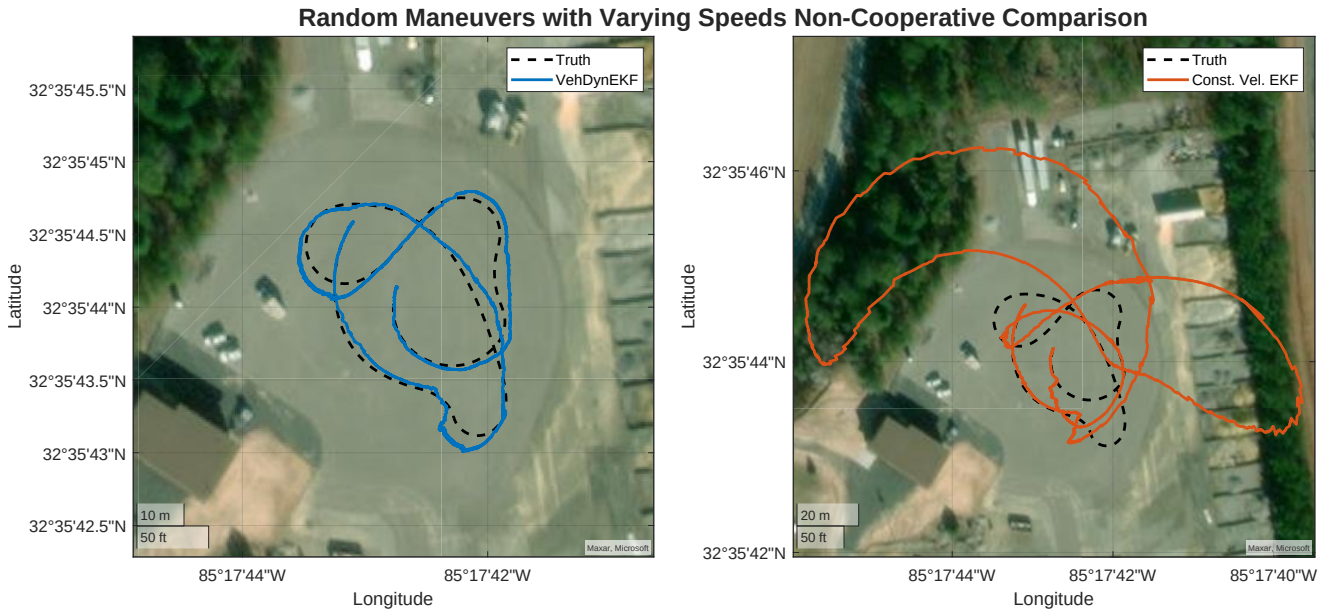


Figure 6.11: Random Maneuver Non-Cooperative Estimation Results

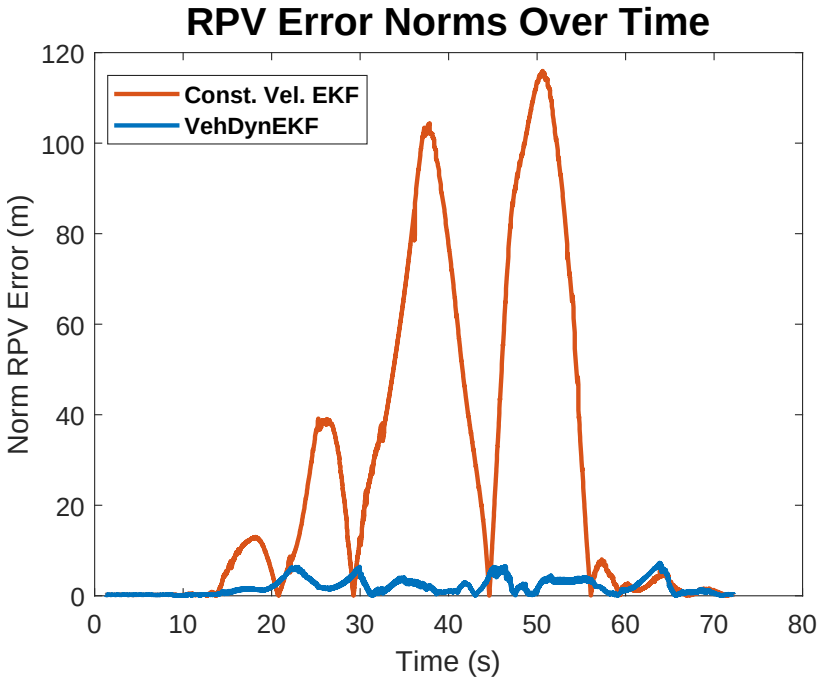


Figure 6.12: Random Maneuver Non-Cooperative Error Norms

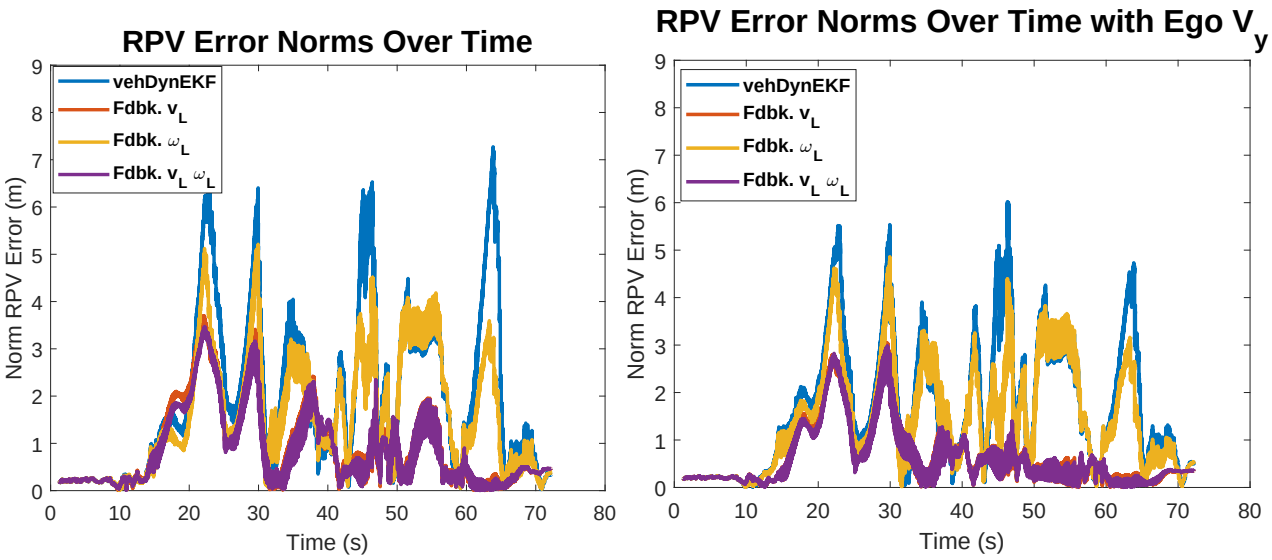
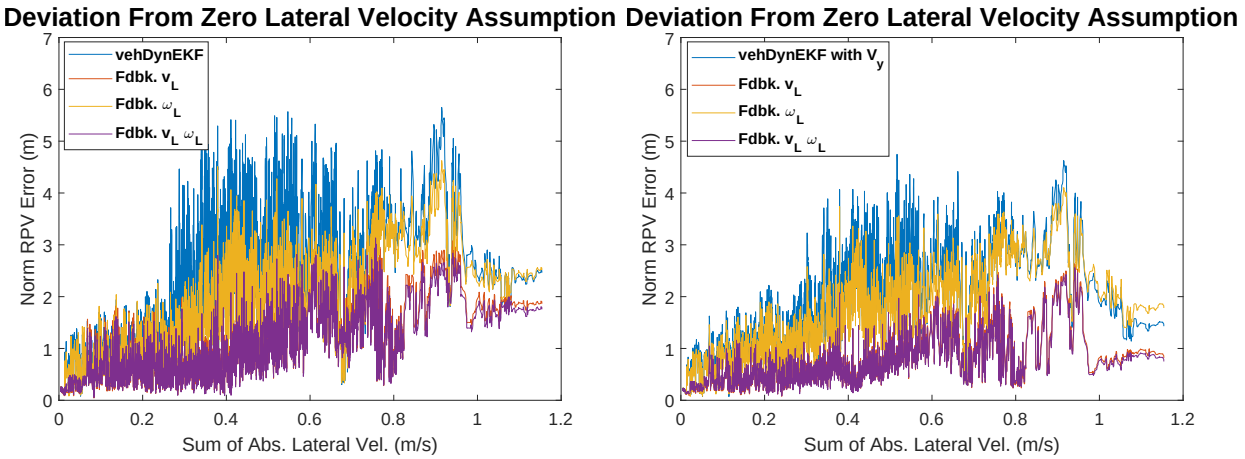


Figure 6.13: Random Maneuver Feedback Results Through Run



(a) Without Lateral Velocity Measurement (b) With Lateral Velocity Measurement

Figure 6.14: Random Maneuver Feedback Results VS Cumulative Lateral Velocity

Table 6.2: Experimental Random Maneuver Cooperative Result Comparison

RMSE of Cooperative RPV Estimates (m)		
Fdbk. Method	No Ego V_y	With Ego V_y
None	2.59	2.16
v_L	1.25	0.89
ω_L	1.99	1.80
v_L and ω_L	1.16	0.86

the base model in portions of the run, they also degrade the solutions in some areas. As mentioned previously, this scenario is a stress-test for the feasibility of on-board UWB relative positioning. In such a demanding scenario, these results show that the constant velocity model cannot provide robust solutions even with the two-vehicle consensus application.

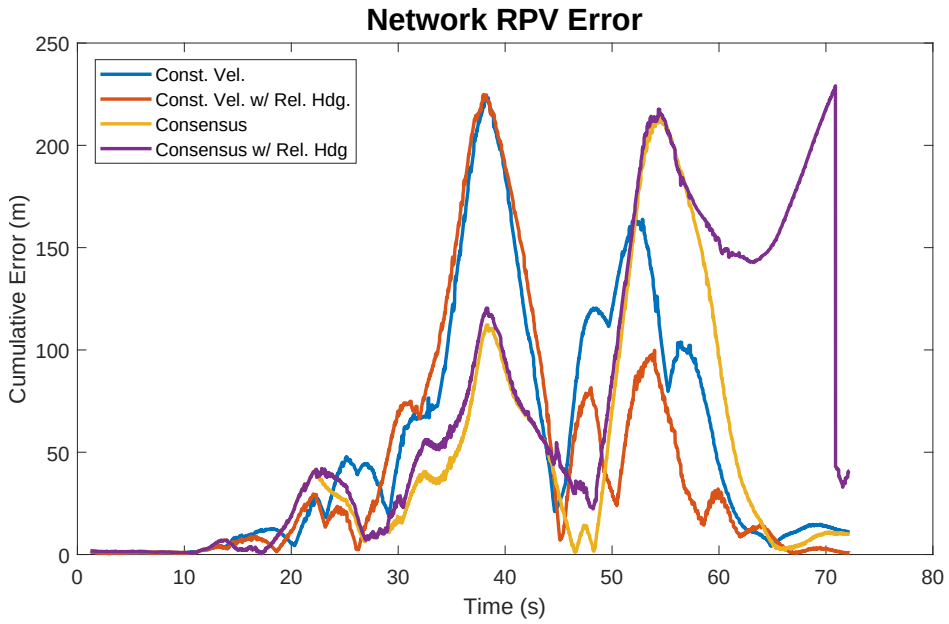


Figure 6.15: Random Maneuver Constant Velocity Consensus Cumulative Error Norms

Lastly, the consensus condition was applied to the VehDynEKF. Figure 6.16 compares the estimation results of velocity feedback and the consensus conditions for one of the vehicles in the network. Meanwhile, Figure 6.17 compares the cumulative network errors.

The consensus results in these figures do not show the same positive trend that was seen in simulation. Although the consensus condition improves the estimation throughout

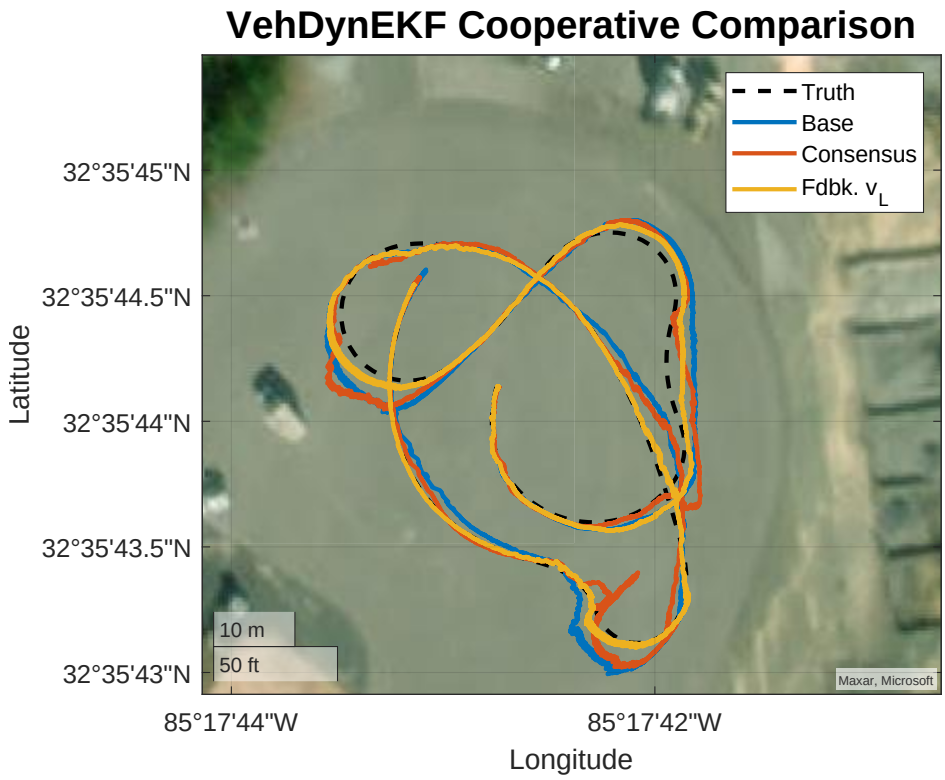


Figure 6.16: Random Maneuver Vehicle Dynamic EKF Consensus Results

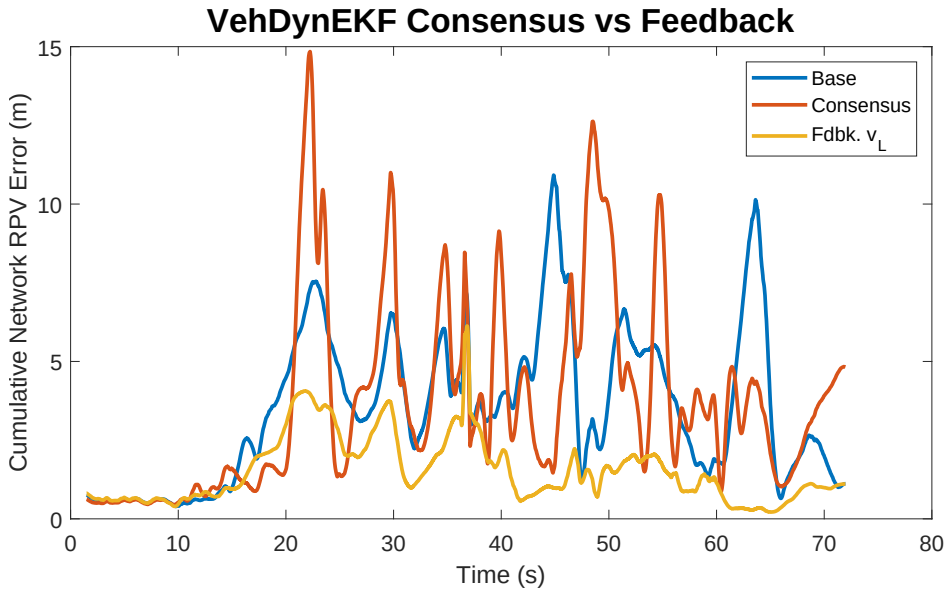


Figure 6.17: Random Maneuver Vehicle Dynamic EKF Consensus Cumulative Error Norms

portions of the run, it also degrades the results in some areas. This deviation from simulation results may be due to non-perfect time syncing for the consensus updates and the physical deviation of from the kinematic bicycle model employed in the simulation. Contrarily, the velocity feedback outperforms both the base and consensus methods substantially even without the lateral velocity measurements on the base vehicle. Therefore, if the vehicles are able to communicate with one another and odometry is available, dynamic state feedback offers more accuracy and robustness than the consensus condition and should be employed instead.

6.4 Conclusion of Experimental Results

The experimental results presented in this chapter validate the findings from simulation in Chapter 5. The proposed vehicle-dynamic EKF is more accurate and robust than the constant velocity model in all explored scenarios. With the application of cooperative methods to the VehDynEKF, the feedback methods including longitudinal speed measurements provide the largest improvement. This provides further benefit if a measurement of the lateral velocity on the base vehicle is available. The application of a consensus condition to the constant velocity method provides the opportunity for tracking through difficult relative geometry areas, but an accuracy improvement is not guaranteed in more dynamic scenarios.

Chapter 7

Conclusions

This thesis explored various methods for relative positioning of ground vehicles with on-board UWB range measurements. After providing necessary technical background, Chapter 3 outlined the difficulties of range-based relative positioning with an observability analysis through DOP calculations. The relevant non-cooperative algorithms from prior work were then outlined. An extension to this prior work was described through the application of vehicle dynamic models and the inclusion of on-board odometry to the base vehicle. In Chapter 4, modifications were made to each algorithm to explore the effect of cooperative methods. All of the algorithms were compared to one another in both simulation and experimental results in Chapters 5 and 6.

It was shown in Chapters 5 and 6 that the VehDynEKF described in Chapter 4 is superior in both accuracy and robustness to the constant velocity method in prior works. When the ego vehicle does not have access to a lateral velocity measurement, the RMSE of the proposed VehDynEKF estimator is under 3 meters. Feedback of dynamic states from the tracked vehicle can reduce this to below 2 meters. With the inclusion of lateral velocity measurements and feedback of the tracked vehicle's speed, the VehDynEKF is capable of a mean error below the 1 meter goal for collision avoidance during real-world trajectories with varying dynamics. However, the errors are not zero-mean and are affected by the relative geometry between the vehicles.

Consensus modifications to the non-cooperative EKFs were explored. The consensus conditions showed improvement to the VehDynEKF algorithm in simulation, but in experimental analysis the results were less promising. Consensus applications to the constant

velocity method present a significant opportunity for overcoming the instability issues experienced when the vehicles are side by side. This represents a improvement over prior work in [21] without the inclusion of any additional sensors. However, this two-vehicle consensus is not sufficient to guarantee the constant velocity EKF will remain stable throughout dynamic runs.

This thesis presents a recommendation for improved ground vehicle relative position estimation with only on-board sensors. With only UWB ranges and odometry, the application of the dynamic models through the VehDynEKF in Chapter 4 can provide robust relative pose estimates to a local vehicle. This was shown to be successful with only a wheel encoder and yaw-gyro, but if the odometry method includes lateral velocity measurements, knowledge of the tracked vehicle's longitudinal speed provides a relative position estimate accurate enough for most safety-critical applications.

7.1 Recommendations of Future Work

There exist a variety of potential extensions to this work. The following list contains suggestions for iterating upon the work presented here:

- Test the algorithm in more realistic driving scenarios to explore the sensitivity to cut-ins and multi-path interference of local objects.
- Extend upon recent work by [18] to employ the algorithms presented here with their carrier phase positioning methods. The author believes that thes work presented here would allow for a more refined and robust search space for the integer ambiguities.
- Bridge the gap between odometry and UWB sensor interfaces to implement simultaneous ranging and communication for implementing the dynamic state feedback methods in real-time systems.
- Numerous methods for refining the UWB hardware layout could be explored:

1. Additional UWBs could eliminate the quadrant ambiguity and reduce observability issues. However, with more UWBs in the network, the update rate of each decreases. A study on these trade offs could prove useful.
2. A more omni-directional antenna may allow for better range quality as the vehicles approach the unobservable region
3. Reconfiguring the UWBs to implement an ALOHA random-access network protocol which is not restricted to the sequential order of range requests. This would allow for use of the most opportunistic measurements and may alleviate the impact of poorly-performing nodes.

- Explore the spatial effects on UWB measurements along with their multi-modal error patterns to create a more refined error model than presented here.
- Extend the consensus network to more than two vehicles. The more actors involved, the more averaging can take place and the constant velocity method may be able to further overcome its fault areas.
- This thesis showed the potential for consensus estimation of locally-operating ground vehicles, but many aspects of this consensus problem remain unexplored: (dynamic network switching, relative uncertainty [69]).

Bibliography

[1] Fabian de Ponte Müller. Survey on Ranging Sensors and Cooperative Techniques for Relative Positioning of Vehicles. *Sensors*, 17(2):271, January 2017.

[2] G.P. Stein, O. Mano, and A. Shashua. Vision-based ACC with a single camera: bounds on range and range rate accuracy. In *IEEE IV2003 Intelligent Vehicles Symposium. Proceedings (Cat. No.03TH8683)*, pages 120–125, Columbus, OH, USA, 2003. IEEE.

[3] Ki-Yeong Park and Sun-Young Hwang. Robust Range Estimation with a Monocular Camera for Vision-Based Forward Collision Warning System. *The Scientific World Journal*, 2014:1–9, 2014.

[4] Matthew Cornick, Jeffrey Koechling, Byron Stanley, and Beijia Zhang. Localizing Ground Penetrating RADAR: A Step Toward Robust Autonomous Ground Vehicle Localization. *Journal of Field Robotics*, 33(1):82–102, January 2016.

[5] Santiago Royo and Maria Ballesta-Garcia. An Overview of Lidar Imaging Systems for Autonomous Vehicles. *Applied Sciences*, 9(19):4093, September 2019.

[6] Yuhang Jiang, Mark Petovello, and Kyle O'Keefe. Augmentation of Carrier-Phase DGPS with UWB Ranges for Relative Vehicle Positioning. page 12, 2012.

[7] Christian J. Campos-Vega, Dan Kamarath, John David Sprunger, Scott Martin, and David Bevly. Precise Relative Positioning in GPS-Degraded Environments using a DGPS/UWB Navigation Filter. pages 2526–2543, September 2021. ISSN: 2331-5954.

[8] Jason N Gross, Yu Gu, and Brandon Dewberry. Tightly-Coupled GPS/UWB-Ranging for Relative Navigation During Formation Flight. page 12.

[9] Steve Huseth, Brandon Dewberry, and Robert McCroskey. Pulsed-RF Ultrawideband Ranging for the GLANSER GPS-denied Emergency Responder Navigation System. page 9.

[10] Jacob Hartzer and Srikanth Saripalli. Vehicular Teamwork: Collaborative localization of Autonomous Vehicles. *arXiv:2104.14106 [cs]*, April 2021. arXiv: 2104.14106.

[11] Yang Song, Mingyang Guan, Wee Peng Tay, Choi Look Law, and Changyun Wen. UWB/LiDAR Fusion For Cooperative Range-Only SLAM. *arXiv:1811.02854 [cs]*, November 2018. arXiv: 1811.02854.

[12] Hamid Mokhtarzadeh and Demoz Gebre-Egziabher. Cooperative Inertial Navigation. *Navigation*, 61(2):77–94, 2014. Number: 2.

[13] Benjamin Noack, Joris Sijs, Marc Reinhardt, and Uwe D. Hanebeck. Decentralized data fusion with inverse covariance intersection. *Automatica*, 79:35–41, May 2017.

[14] Anton Ledergerber, Michael Hamer, and Raffaello D'Andrea. A robot self-localization system using one-way ultra-wideband communication. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3131–3137, Hamburg, Germany, September 2015. IEEE.

[15] Justin Cano, Saad Chidami, and Jerome Le Ny. A Kalman Filter-Based Algorithm for Simultaneous Time Synchronization and Localization in UWB Networks. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1431–1437, Montreal, QC, Canada, May 2019. IEEE.

[16] F.R. Fabresse, F. Caballero, and A. Ollero. Decentralized simultaneous localization and mapping for multiple aerial vehicles using range-only sensors. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6408–6414, Seattle, WA, USA, May 2015. IEEE.

[17] Abdulrahman Alarifi, AbdulMalik Al-Salman, Mansour Alsaleh, Ahmad Alnafessah, Suheer Al-Hadhrami, Mai Al-Ammar, and Hend Al-Khalifa. Ultra Wideband Indoor Positioning Technologies: Analysis and Recent Advances. *Sensors*, 16(5):707, May 2016.

[18] Adyasha Mohanty, Asta Wu, Sriramya Bhamidipati, and Grace Gao. Precise Relative Positioning via Tight-Coupling of GPS Carrier Phase and Multiple UWBs. *IEEE Robotics and Automation Letters*, 7(2):5757–5762, April 2022.

[19] Charles Champagne Cossette, Mohammed Shalaby, David Saussie, James Richard Forbes, and Jerome Le Ny. Relative Position Estimation Between Two UWB Devices With IMUs. *IEEE Robotics and Automation Letters*, 6(3):4313–4320, July 2021.

[20] Mohammed Shalaby, Charles Champagne Cossette, James Richard Forbes, and Jerome Le Ny. Relative Position Estimation in Multi-Agent Systems Using Attitude-Coupled Range Measurements. *IEEE Robotics and Automation Letters*, 6(3):4955–4961, July 2021. arXiv:2207.07993 [cs].

[21] Ehab Ghanem. *2-D Relative Position and Attitude Determination for Land Vehicles using Multiple UWB Ranges*. PhD thesis.

[22] Ehab Ghanem, Kyle O'Keefe, and Richard Klukas. Estimating Vehicle-to-Vehicle Relative Position and Attitude using Multiple UWB Ranges. page 7.

[23] Ben Jones, Kyle Thompson, Dan Pierce, Scott Martin, and David Bevly. Ground-Vehicle Relative Position Estimation with UWB Ranges and a Vehicle Dynamics Model. *MECC 2022*.

[24] Hessam Mohammadmoradi, Milad Heydariaan, and Omprakash Gnawali. SRAC: Simultaneous Ranging and Communication in UWB Networks. In *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 9–16, Santorini Island, Greece, May 2019. IEEE.

[25] R. Olfati-Saber. Distributed Kalman Filter with Embedded Consensus Filters. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 8179–8184, Seville, Spain, 2005. IEEE.

[26] R. Olfati-Saber. Distributed Kalman filtering for sensor networks. In *2007 46th IEEE Conference on Decision and Control*, pages 5492–5498, New Orleans, LA, USA, 2007. IEEE.

[27] Jingwei Wang, Eric A. Butcher, and Tansel Yucelen. Extended Kalman Filter and Observability Analysis for Consensus Estimation of Spacecraft Relative Motion. In *2018 AIAA Guidance, Navigation, and Control Conference*, Kissimmee, Florida, January 2018. American Institute of Aeronautics and Astronautics.

[28] E. D'Amato, I. Notaro, M. Mattei, and G. Tartaglione. Attitude and position estimation for an UAV swarm using consensus Kalman filtering. In *2015 IEEE Metrology for Aerospace (MetroAeroSpace)*, pages 519–524, Benevento, Italy, June 2015. IEEE.

[29] Ye Sun and Daniel B. Work. A distributed local Kalman consensus filter for traffic estimation. In *53rd IEEE Conference on Decision and Control*, pages 6484–6491, Los Angeles, CA, USA, December 2014. IEEE.

[30] Baichun Gong, Sha Wang, Mingrui Hao, Xujun Guan, and Shuang Li. Range-based collaborative relative navigation for multiple unmanned aerial vehicles using consensus extended Kalman filter. *Aerospace Science and Technology*, 112:106647, May 2021.

[31] Reference frames and how they are used in inertial navigation · VectorNav.

[32] Paul D Groves. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, second edition, 2013.

[33] Jerry Ginsberg. *Engineering Dynamics*. Cambridge University Press, 2008.

[34] Julian Jordan and Andreas Zell. Kinematic model based visual odometry for differential drive vehicles. In *2017 European Conference on Mobile Robots (ECMR)*, pages 1–7, Paris, September 2017. IEEE.

[35] Oyindamola Omotuyi and Manish Kumar. UAV Visual-Inertial Dynamics (VI-D) Odometry using Unscented Kalman Filter. *IFAC-PapersOnLine*, 54(20):814–819, 2021.

[36] Sinpyo Hong, Yu-Shin Chang, Sung-Kee Ha, and Man-Hyung Lee. Estimation of Alignment Errors in GPS/INS Integration. page 9.

[37] Alan M. Schneider. Kalman Filter Formulations for Transfer Alignment of Strapdown Inertial Units. *Navigation*, 30(1):72–89, March 1983.

[38] Jr. Sutherland, Gelb Arthur A., and Arthur. The Kalman Filter in Transfer Alignment of Airborne Inertial Guidance Systems:. Technical report, Defense Technical Information Center, Fort Belvoir, VA, October 1968.

[39] Anon Defense Mapping Agency. Department of Defense World Geodetic System 1984. 1997.

[40] Lu Xiong, Xin Xia, Yishi Lu, Wei Liu, Letian Gao, Shunhui Song, Yanqun Han, and Zhuoping Yu. IMU-Based Automated Vehicle Slip Angle and Attitude Estimation Aided by Vehicle Dynamics. *Sensors*, 19(8):1930, April 2019.

[41] Xiaoji Niu, Sameh Nassar, and Naser El-Sheimy. An Accurate Land-Vehicle MEMS IMU/GPS Navigation System Using 3D Auxiliary Velocity Updates. *Navigation*, 54(3):177–188, September 2007.

[42] G. Dissanayake, S. Sukkarieh, E. Nebot, and H. Durrant-Whyte. The aiding of a low-cost strapdown inertial measurement unit using vehicle model constraints for land vehicle applications. *IEEE Transactions on Robotics and Automation*, 17(5):731–747, October 2001.

[43] Rajesh Rajamani. *Vehicle Dynamics and Control*. Mechanical Engineering Series. Springer US, Boston, MA, 2012.

[44] Jason Kong, Mark Pfeiffer, Georg Schildbach, and Francesco Borrelli. Kinematic and dynamic vehicle models for autonomous driving control design. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 1094–1099, Seoul, South Korea, June 2015. IEEE.

[45] Brandon Dewberry and Alan Petroff. Precison Navigation with AD-HOC Autosurvey using UltraWideBand Two-Way Ranging Network. page 6.

[46] TDSR. UWB Technology: https://tdsr-uwb.com/technology/.

[47] TDSR. P440-Data-Sheet-User-Guide: https://tdsr-uwb.com/documentation/.

[48] Eric Broshears. Ultra-wideband Radio Aided Carrier Phase Ambiguity Resolution in Real-Time Kinematic GPS Relative Positioning.

[49] Pablo Corbalán and Gian Pietro Picco. Concurrent Ranging in Ultra-wideband Radios:Experimental Evidence, Challenges, and Opportunities. page 12.

[50] TimeDomain. PulsON 400 Ranging and Communications. Part Three: Two-Way Time-of-Flight (TW-TOF) Ranging.

[51] Patrap Misra. *Global Positioning System: Signals, Measurements, and Performance*. Ganga-Jamuna Press, 2nd edition.

[52] Pil Hun Choi, Jinsil Lee, and Jiyun Lee. Dilution of Precision (DOP) Based Landmark Exclusion Method for Evaluating Integrity Risk of LiDAR-based Navigation Systems. *Journal of Positioning, Navigation, and Timing*, 9(3):285–292, September 2020.

[53] Young Bum Park, Hyun Cheol Jeon, and Chan Gook Park. Analysis of Geometric Effects on Integrated Inertial/Vision for Lunar Descent Navigation. *Journal of Guidance, Control, and Dynamics*, 39(4):937–943, April 2016.

[54] R.E. Kalman. A New Approach to Linear Filtering and Prediction Problems, 1960.

[55] Leonard A McGee and Stanley F Schmidt. Discovery of the Kalman Filter as a Practical Tool for Aerospace and. page 25.

[56] Robert Stengel. *Optimal Control and Estimation*. Dover Publications.

[57] Ramsey Faragher. Understanding the Basis of the Kalman Filter Via a Simple and Intuitive Derivation [Lecture Notes]. *IEEE Signal Processing Magazine*, 29(5):128–132, September 2012.

[58] Yaakov Bar-Shalom. *Estimation with Applications to Tracking and Navigations: Theory Algorithms*. John Wiley and Sons, 2001.

[59] Nathanael L. Baisa. Derivation of a Constant Velocity Motion Model for Visual Tracking. Technical Report arXiv:2005.00844, arXiv, October 2020. arXiv:2005.00844 [cs].

[60] Jung Min Pak, Pyung Soo Kim, Sung Hyun You, Sang Seol Lee, and Moon Kyou Song. Extended least square unbiased FIR filter for target tracking using the constant velocity motion model. *International Journal of Control, Automation and Systems*, 15(2):947–951, April 2017.

[61] Kevin M. Brink. Multi-agent relative pose estimation: approaches and applications. In *Open Architecture/Open Business Model Net-Centric Systems and Defense Transformation 2018*, volume 10651, page 106510D. International Society for Optics and Photonics, May 2018.

[62] Flavia Causa, Amedeo Rodi Vetrella, Giancarmine Fasano, and Domenico Accardo. Multi-UAV formation geometries for cooperative navigation in GNSS-challenging environments. page 11.

[63] Joel E Huff. Absolute and Relative Navigation of an sUAS Swarm Using Integrated GNSS, Inertial and Range Radios. page 79.

[64] Jake Pryor. Evaluation of Cooperative Navigation Strategies with Maneuvering UAVs.

[65] Shunkai Sun, Jianping Hu, Jie Li, Ruidong Liu, Meng Shu, and Yang Yang. An INS-UWB Based Collision Avoidance System for AGV. *Algorithms*, 12(2):40, February 2019.

[66] Steven van der Helm, Mario Coppola, Kimberly N. McGuire, and Guido C. H. E. de Croon. On-board range-based relative localization for micro air vehicles in indoor leader–follower flight. *Autonomous Robots*, 44(3-4):415–441, March 2020.

[67] Xin Xia, Runsheng Xu, Jiaqi Ma, and You Li. Secure cooperative localization for connected automated vehicles based on consensus estimation. page 14.

[68] Reza Olfati-Saber. Kalman-Consensus Filter : Optimality, stability, and performance. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 7036–7042, Shanghai, China, December 2009. IEEE.

[69] Wei Ren, R.W. Beard, and D.B. Kingston. Multi-agent kalman consensus with relative uncertainty. In *Proceedings of the 2005, American Control Conference, 2005.*, pages 1865–1870, Portland, OR, USA, 2005. IEEE.

[70] Amanda Prorok and Alcherio Martinoli. Accurate indoor localization with ultra-wideband using spatial models and collaboration. *The International Journal of Robotics Research*, 33(4):547–568, April 2014.