**Graph-Based Relative Path Estimation Using Landmarks for Long Distance Ground Vehicle Following**

by

Dan Pierce

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
December 10, 2022

Keywords: Autonomous Ground Vehicles, Path Following, Differential GPS, Lidar Feature Extraction, SLAM



Approved by

David M. Bevly, Chair, Professor of Mechanical Engineering
Scott M. Martin, Professor of Mechanical Engineering
John Y. Hung, Professor of Electrical and Computer Engineering
Dan B Marghitu, Professor of Mechanical Engineering

Abstract

This dissertation presents a graph-based sensor fusion framework for the localization of automated ground vehicles for leader-follower path duplication. A specific focus is given to scenarios where the following distance between vehicles is high. Localization accuracy is critical for any automated path following task. While high localization accuracy can be achieved using GPS corrections or *a priori* maps, these resources are not available in certain scenarios such as remote areas with limited infrastructure and *a priori* information. In this dissertation, a novel method is proposed for solving this localization problem that does not depend on built infrastructure or *a priori* information. A graph-based framework is used to estimate the path taken by the lead vehicle using relative measurements of differential GPS between vehicles, odometry from onboard sensor measurements, and exchanged landmark observations.

The graph-based estimation framework developed in this dissertation allows for ad-hoc, nonlinear measurements for estimating a near-optimal path solution. Each sensor input provides complementary benefits: differential GPS allows for centimeter-level accuracy, vehicle odometry allows for spanning GPS outages and gaps in landmarks, and landmark observations bound path errors with respect to following distance. The findings of an observability analysis are presented to show failure conditions for certain vehicle and landmark configurations. A simulation study is developed and used to demonstrate the success of the proposed method and characterize the estimator's performance in a number of controlled scenarios. The findings from the simulation study are validated with experimental data from a pair of Class 8 tractor-trailers, each equipped with a GPS receiver, a multi-channel lidar, wheel encoders, an inertial measurement unit, and a Dedicated Short Range Communications (DSRC) radio.

An overview is presented for each method used to generate the measurements used in the graph-based estimator. This includes an overview and error characterization of Time Differenced Carrier Phase (TDCP) and Dynamic-Base RTK (DRTK) used for precise odometry and inter-vehicle relative positions, respectively. An odometry model is provided for determining vehicle motion from wheel speed and yaw rate sensors. Additionally, two unique approaches are presented for detecting road sign and pole-like objects from 3D point cloud data for use as landmark observations.

Results show that the presented method improves performance when compared to existing methods in terms of both accuracy and availability. Compared to many existing map-matching approaches, the presented approach requires a relatively small number of landmarks (~11/km) to achieve the accuracy target regardless of GPS availability. Under nominal conditions, the presented method is successful in meeting the estimation accuracy required for path control with average lateral path position errors of 0.94 cm and path orientation errors of 0.14°. It is also shown that the path errors remain bounded with respect to following distance when a sufficient number of landmarks are present, allowing for large gaps between vehicles.

Acknowledgments

This dissertation would not be possible without the support of those who have been with me along the way. I would like to thank my advisor, Dr. Bevly, for recognizing my potential and providing me with the opportunity to further my education and professional experience in such a fascinating field. I'm also grateful for the friends and colleagues in the GAVLAB; together we found enjoyment in the hard work and shared challenges. In particular, I'd like to acknowledge Scott Martin who served as a mentor throughout my doctoral pursuit.

I would also like to acknowledge FHWA, TARDEC, and FP Innovations for providing the funding that made this work possible. I'm truly grateful for the opportunity to work on state-of-the-art technology and to see my work in action on the trucks. I'd also like to thank the drivers that helped with data collection on the trucks, especially James Johnson and all the miles that we put down together.

Finally, I would like to thank my family and friends for their encouragement throughout the process of writing this dissertation. I am especially grateful for my brother, Will, and sister-in-law, Jackie, who have shown through example how to act as a leader and a Christian in all aspects of life. I'd also like to acknowledge my parents who provided support and shelter as I completed this chapter of my life. It's truly a privilege to have such a strong network of support and I am ever grateful.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AGR Autonomous Ground Resupply

CACC Cooperative Adaptive Cruise Control

CAN Controller Area Network

CCDC Combat Capabilities Development Command

DoF Degrees of Freedom

DOT Department of Transportation

DRTK Dynamic-Base Real Time Kinematic

DSRC Dedicated Short Range Communications

EKF Extended Kalman Filter

FHWA Federal Highway Administration

GNSS Global Navigation Satellite System

GPS Global Positioning System

GVSC Ground Vehicle Systems Center

ICP Iterative Closest Point

ILF Immediate Lead Following

IMU Inertial Measurement Unit

INS Inertial Navigation System

LIDAR Light Detection and Ranging

LoS Line-of-Sight

MPC Model Predictive Control

ODD Operational Design Domain

PCA Principal Component Analysis

RADAR Radio Detection And Ranging

RANSAC Random Sample Consensus

RMS Root Mean Square

RPG Relative Pose Graph

RPV Relative Position Vector

RTK Real-Time Kinematic

SLAM Simultaneous Localization and Mapping

TARDEC Tank Automotive Research, Development and Engineering Center

TDCP Time Differenced Carrier Phase

ULF Ultimate Lead Following

UTM Universal Transverse Mercator

UWB Ultra-Wideband

# Chapter 1

# Introduction and Background

## 1.1 Motivation

Increased autonomy in ground vehicles has many advantages in terms of safety, efficiency, and rider comfort. These advantages are especially realizable in convoy networks in which multiple vehicles act cooperatively to achieve a common goal. In the case of a leader-follower, platoon-based approach, the goal is to lower fuel consumption; vehicles are situated nose-to-tail in order to reduce air drag. Experimental studies at Auburn University have shown combined fuel savings of 6 % to 8 % for Class 8 trucks in highway conditions [5].

Perhaps the most important advantage of autonomous convoys is the potential to reduce highway related incidents. In 2019, 36,096 people died in a motor vehicle crash [6]. The economic impact of motor vehicle crashes was estimated to be $242 billion dollars in 2010 [6]. Truck drivers are often driving 10 hours on a given shift and become fatigued over time, increasing the risk of an accident. A computerized driving system, however, will not become drowsy from long hours or distracted from the task at hand. Autonomous driving systems have the potential to reduce the mental energy required of the human operator, leading to safer driving conditions.

From a military standpoint, autonomous convoys are desirable for both operation flexibility and safety. The U.S. Army Combat Capabilities Development Command's (CCDC) Ground Vehicle Systems Center (GVSC), formerly the Tank Automotive Research, Development and Engineering Center (TARDEC), has been investigating technologies in the field of autonomous path following. The group has investigated methods for reducing the number of soldiers required for ground resupply through the Autonomous Ground Resupply (AGR) program [7]. Resupply missions are vulnerable to attack and can be a strategic target for

military adversaries [8]. The soldiers tasked with manning an autonomous following system will have better situational awareness without the distraction of operating the vehicle. Autonomous following can be highly accurate compared to a human driver and a reduced footprint lowers the odds of any one vehicle triggering an IED or encountering other road-side obstacles.

Autonomous convoys have the potential to transform the transportation sector in the not-so-distant future. Much of the progress made in recent years has focused on partially autonomous applications in which a human driver commands the front-most vehicle. With a human in the loop, many of the complex problems in decision making (ethically and otherwise) are addressed. By framing the problem in this way, the proposed solutions are somewhat familiar to the public, making it easy to answer questions regarding legislation and liability. The scenario where one or more vehicles act to recreate the path of a leading vehicle is referred to as a leader-follower convoy.

There are two main frameworks that can be used to accomplish path following in a leader-follower convoy: 1) Ultimate Lead Following (ULF) and 2) Immediate Lead Following (ILF). For the ILF framework, each vehicle in the convoy acts to follow the vehicle that immediately precedes them in the convoy. ILF is convenient for ad-hoc networks, where vehicles join and exit the convoy freely, and for dynamic environments in which path obstructions could appear at any moment. If each follower was able to perfectly follow its immediate lead, then the overall convoy goal would be satisfied. However, each additional link is subject to errors from both estimation and control that accumulate down the line and potentially cause instability [9]. This problem is referred to as *string stability* in which the response of the entire convoy network is considered.

Alternatively, in the ULF framework, each vehicle references the path taken by the front-most vehicle in the convoy. The string stability problem is not an issue for the ULF framework since errors are not accumulated through vehicle pairs. However, from an implementation standpoint, there are other challenges involved with ULF. In some cases, especially for

larger convoys, communications can be an issue due to radio range or bandwidth limitations. Another challenge in ULF is one of estimating the leader's path, which is the focus of this dissertation. The benefits of ULF provide motivation for studying path following with large separation distances; as the number of vehicles in a convoy increases, so does the following distance of the last vehicle. Large separation between vehicles may also be desired from an operational standpoint (e.g., to reduce the collateral damage if one of the vehicles in the convoy is compromised).

## 1.2 Design Considerations

An autonomous convoy system should be flexible to operate in various environments. Because of this, the system should not rely exclusively on an existing infrastructure. For example, one solution to path duplication is for each vehicle to track lane markings (using computer vision) and to control to the center of the lane, thereby duplicating the path of the first vehicle. However, lane marking detection can be unreliable due to environmental conditions or completely unavailable in off-road or rural scenarios. Another common solution for path duplication is absolute waypoint following where the lead vehicle drops "bread crumbs" for the following vehicles to reference. While it is possible to achieve centimeter level accuracy with differential GPS techniques, this requires a fixed base station to broadcast corrections to the remote vehicles. Base stations may not always be available, for example in cross-country platooning operations through remote areas with limited LTE connectivity, or in military applications where such critical infrastructure could be compromised.

The work presented in this dissertation aims to determine a localization solution for path following that does not rely on built infrastructure such as base stations or mapped landmark databases. In this way, none the measurements used in determining the path of the lead vehicle are absolute (i.e., no measurements of the vehicle's absolute position and no maps of absolute landmark positions are used). Therefore, the problem can be referred to

as a relative path estimation problem where the path solution is relative to the position of the following vehicle.

### 1.3 Problem Statement

The presented work focuses on estimating the relative path of a lead vehicle. The ULF strategy is chosen as the framework for path following in multi-vehicle convoys so that the problem of string stability is avoided. The presented solution considers only the estimation performance of a single vehicle pair and the performance can be applied to any vehicle pair in a ULF framework. The solution should accommodate for any number of vehicles in a ULF network and therefore the path estimation errors should be bounded with respect to increasing following distance as the number of vehicles in the network grows.

The solution to the relative path estimation problem should provide continuous availability. A system with perfect availability means that a path solution (from which a control reference can be calculated) is available throughout the mission. For example, a system that relies solely on GPS and/or radio communication could have compromised availability in the event of signal jamming or obstructions. Similarly, systems that require line-of-sight (LoS) to the lead vehicle would have poor availability in dense forest environments or when navigating around buildings.

Additionally, the solution to the path estimation problem should satisfy accuracy requirements sufficient for autonomous path following control. The accuracy requirements depend on the designated Operational Design Domain (ODD) [10]. The ODD may include highway applications where the vehicle is expected to stay within lane markers or logger roads in forestry applications that are bound by the tree line or runoff drains at the road's shoulder. The accuracy requirement is therefore a function of the lane width and vehicle dimensions. The estimation accuracy requirements in this dissertation were chosen to allow for safe lateral control of a Class 8 tractor trailer to stay within lane lines of a highway defined by US road geometry standards. Following the results of the study conducted in [10],

estimation accuracy requirements were determined to be 7 cm for the lateral position estimate and 0.25° for the yaw angle estimate, both in terms of $67^{\text{th}}$ percentile error. These values will stand as design criteria and allow for an evaluation of the proposed localization solution in meeting these accuracy requirements.

To accomplish the problem of path estimation as described above, it is assumed that each vehicle can make the observations as described in this paragraph. This includes measurements of the relative position vector (RPV) from the following vehicle to the lead vehicle, resolved in a global frame. Also, each vehicle can measure the change in position over time, resolved in a global frame. These two measurements are referred to in this dissertation as inter-vehicle RPV and GPS odometry, respectively, and experimental results will be shown where these values are measured using differential GPS techniques. Each vehicle is also capable of measuring ego-motion (change in position and orientation over time), referred to as body-centric odometry. For the particular application, results will be shown for a body-centric odometry model that uses wheel speeds and yaw rates as measured by onboard sensors. Lastly, each vehicle is able to make observations of landmarks in the environment (measured as the relative position and orientation of a landmark, resolved in the sensor frame). The particular application shown in the experimental results uses landmarks observed by a multi-channel lidar mounted rigidly on the vehicle's chassis.

## 1.4 Prior Art

This section provides an overview of the various methods existing in the literature for accomplishing autonomous following with consideration to the challenges described in the preceding sections. The prior art is first presented in two main categories based on their respective design objectives: short distance and long distance following. Within these two categories, there are two predominant forms of path estimation: some that rely on body-centric measurements and others that rely primarily on GPS-based measurements. The summary of short and long distance following prior art is followed by an overview of

landmark navigation which is a heavily researched field warranting its own section. Each of the presented literature sought to satisfy the given mission objectives and, therefore, have their own benefits and limitations. The review in this section will relate these limitations to the problem at hand in order to motivate the contributions that follow.

### 1.4.1 Short Distance Following

A summary of the prior literature on short distance following is provided in this section. Short distance following approaches are common for single vehicle pairs and for Immediate Lead Following where vehicles are separated by less than a few car lengths. In this scenario, steering the following vehicle directly towards the lead will effectively duplicate the leader's path with an acceptable level of lateral path error. This approach is also known as the Virtual Trailer Link Model [11]. The control law for the Virtual Trailer Link Model acts to drive the relative bearing between the follower and the leader to zero. The bearing can be calculated given the Relative Position Vector (RPV) between the leader and follower, resolved in the body frame of the following vehicle.

#### Body-Centric Relative Positioning

When vehicles are close together, there is an increased possibility for making line-of-sight (LoS) observations of the lead vehicle using perception-based sensors such as cameras, lidars, and radar [12, 13]. A clear LoS between vehicles also allows for time-of-flight ranging with Ultra-Wideband (UWB) radios [14] or even by a physical tethering device that measures range and angle [15]. The measurements provided by these sensors are resolved in the body frame of the vehicle, which is ideal for calculating the control references used in short distance following [16].

Radars are increasingly common on ground vehicles and can handle much of the overhead involved in tracking the lead vehicle. One of the measurements provided by a radar track is azimuth angle, which could be used directly as feedback for the Virtual Trailer Link

Model [11]. The relative position of the lead vehicle is attained by combining the range and angle associated with the radar measurement (referred to as a "track"). Radars typically provide tracks of multiple objects in the field of view and because of this, the correspondence problem must be solved to associate the correct track with the lead vehicle [17].

Lidars also have the potential to observe the inter-vehicle RPV, but require some additional preprocessing. Lidars calculate time-of-flight range measurements of light waves that are reflected from the environment. Each range measurement is associated with a particular azimuth and elevation angle. In effect, the lidar provides a dense cloud of cartesian points of the surrounding environment, also known as a point cloud. Given a point cloud, object recognition strategies are employed to detect the lead vehicle and extract the associated relative position vector [18].

Monocular cameras have also been used for short distance following [19]. The lead vehicle is extracted from a given image using image processing techniques [17]. Some image processing techniques require a fiducial marker on the lead vehicle [20], which helps with target correspondence, while others leverage artificial intelligence approaches [21]. Monocular cameras lack the ability to directly observe depth in an image and so the body frame RPV measurement does not have scale/magnitude. However, the relative bearing measurement is directly observable from an image and is proportional to the horizontal pixel location in an image [22].

Perception-based relative positioning methods do not rely on a communication link with the lead vehicle which can be advantageous in certain scenarios. These sensors can be leveraged in the event of lost inter-vehicle communications and/or degraded GPS reception. However, perception-based methods often suffer from poor availability. The availability is diminished by the LoS requirement and also the dependence on environmental factors such as lighting, reflectivity, and weather [12]. Perception-based methods are less reliable because they depend on the recognition and correspondence of the lead vehicle. A false

correspondence could result in following the wrong vehicle or discontinuities in the control reference.

A body-frame RPV measurement can also be made using an array of Ultra-Wideband (UWB) radios mounted to each vehicle in a leader-follower pair. UWB radios are increasingly common for relative positioning tasks and have been included in the autonomy kit of TARDEC's AGR program [7]. UWB radios use round-trip delay to precisely measure range between radio pairs with centimeter level accuracy. As opposed to perception-based methods, UWB radios do not require object recognition/correspondence and also have the ability to range in certain non-LoS conditions [12].

A single UWB radio pair can observe the scalar distance between vehicles, which is only useful for longitudinal convoy control and not full autonomous following. With two radios on the following vehicle and a single radio on the lead vehicle, the inter-vehicle RPV can be calculated [14]. With two radios on each vehicle, the inter-vehicle RPV along with a relative orientation can be calculated. The accuracy of the resulting measurements are highly dependent on the geometry of the radios.

In [23], a model that relates the motion of the two vehicles is used to overcome the limitations due to poor geometry of UWB radios. The range measurements from UWB are fused with constraints from a kinematic bicycle model using an Extended Kalman Filter (EKF). The EKF inherently accounts for the system geometry and provides complementary estimates of relative position and orientation.

Like the other approaches to relative positioning mentioned in this section, the method presented in [23] will suffer from cut corners and LoS issues when used for autonomous path following. Section 1.4.2 will show how the relative positioning methods mentioned in this section can be used to extend following distance through path estimation. However, these adaptations suffer from odometry drift, especially at large following distances.

### GPS-Based Relative Positioning

The relative position information used for short-distance following can also be calculated from a GPS solution. The information from the lead vehicle's GPS receiver is communicated to the following vehicle using a wireless communication line either directly or relayed through some ground-based infrastructure [12].

The simplest way to calculate the RPV between vehicles is to difference the stand-alone GPS positions. GPS signals suffer from atmospheric effects, which can result in poor positioning accuracy (around 3 meters of error). However, the atmospheric effects are spatially correlated and the differenced positions are accurate to within a meter [25]. This level of accuracy is acceptable for many applications, but to achieve centimeter level accuracy in positions, differential GPS techniques can be used. A technique known as Real-Time Kinematic (RTK) positioning uses a local base station to perform differential GPS for centimeter level measurements of absolute position. The drawback to RTK is the required base station infrastructure, which must be within a 20 km radius of the vehicles [12]. The infrastructure requirement is not ideal for long distance transports or for convoys operating in hostile territory. The same principles of RTK can be applied between vehicles in an approach called Dynamic-Base RTK (DRTK) [25,26]. The result is an RPV between vehicles that is accurate to within centimeters.

Relative positions from a GPS solution are resolved in a global frame (e.g., difference in North-East-Down positions of the vehicle). These GPS-based relative positions must be rotated into the body frame of the follower to calculate control commands. A common source of heading estimate is from GPS/INS or a dual-antenna system [27].

### 1.4.2 Long Distance Following

As described earlier in this chapter, certain scenarios call for relatively large spacing between the lead and following vehicle. The short distance approaches described in the previous section are not suitable for long distances as they tend to cut the corners of the

leader's path [11]. Long distance following is inherently more challenging than short distance following and this section presents existing approaches for accomplishing this goal.

Long distance following is made possible by estimating the trajectory, or path, taken by the lead vehicle, which is typically represented by a set of waypoint positions. Given a set of path waypoints, the following vehicle attempts to duplicate the path. The control strategies used for short distances can be adopted for path duplication. From the set of path waypoints, a "virtual lead vehicle" is declared at some location ahead of the follower on the path and the follower acts to pursue this virtual lead [28].

Other path following control strategies exist that directly reference the follower's lateral path deviation. One approach is based on classical control with cascading feedback loops for lateral path position and path heading error [9]. In steady state conditions, this approach can drive the lateral path error to zero. However, because this approach only considers the current state in time, transient errors due to changing path curvature will degrade the solution. Alternatively, a Model Predictive Control (MPC) method can mitigate transient errors by predicting future states over a range of possible control inputs. The cost function, formed by the prediction's path deviation, is minimized with respect to the control input [29]. The path following accuracy of the MPC comes at the cost of the increased computational demand in minimizing the cost function. However, even for a simple cost function and motion model, the improvements can be significant. Regardless of the control strategy used, the prerequisite is a set of path waypoints relative to the following vehicle and resolved in the follower frame. The methods used in generating this reference path are described in the following sections.

### Absolute Path Generation

Absolute path generation is perhaps the most intuitive approach for the path duplication problem. In this scenario, both vehicles can measure their position in a common coordinate frame (e.g., a Universal Transverse Mercator coordinate system). The positions of the lead

vehicle are saved as it travels and a path is formed for the follower to reference. Because the path is resolved in an absolute frame, a measurement of the following vehicle heading is needed to rotate the path into the follower frame, which is then used to calculate control commands.

Absolute position measurements are most often measured using GPS. A path formed using standalone GPS position accuracy may not be sufficient in the path duplication scenario; the positions could change significantly in time as satellites come in and out of view [25]. These position jumps will result in spikes in the path, which could potentially drive the following vehicle to travel into undesired terrain. When base station infrastructure is available, RTK positioning should instead be used for absolute path following. However, as mentioned before, the infrastructure requirement is undesirable for certain applications.

### GPS-Based Relative Path Generation

In many cases, accurate absolute position information is not available, but the relative positions between vehicles is well known. Path generation is still possible with relative position measurements by making use of vehicle odometry. Odometry is a measure of the change in position and orientation and can be provided by sensors such as wheel encoders [30], inertial devices [27], visual structure-from-motion [31], etc.

The precision of RPV measurements from DRTK have been used for path generation by Travis [25] and expanded on by Martin [26]. The source of odometry in Travis and Martin's work comes from Time Differenced Carrier Phase (TDCP). The TDCP algorithm estimates the change in antenna position from one time step to the next, represented as 3-dimensional RPV, and is capable of achieving millimeter-level accuracy. As its name suggests, the TDCP solution works by differencing carrier phase measurements thereby removing common-mode errors. The odometry measurement from TDCP is resolved in the global frame, as is the RPV measurement from DRTK. The DRTK and TDCP observations are combined with

vector math calculations to form the path relative to the follower. The path is rotated into the follower frame using an external heading estimate for calculating the control reference.

Essentially, the following vehicle is dead reckoned over a certain number of epochs, depending on the spacing between vehicles. As for any dead reckoning solution, the localization uncertainty grows with each epoch due to the accumulation of noisy measurements. Therefore, the path uncertainty is unbounded with respect to increasing following distance. This is not a major concern in most situations since the uncertainty from TDCP grows linearly and at a relatively low rate. Therefore, in ideal scenarios, the DRTK/TDCP method of path generation can be used for very long following distances (less than 0.1 m of error at 500 m distance, traveling at 20 m/s) [25]. To put this value in context, most RF communication methods are limited to a maximum range of about one kilometer [12].

The accuracy requirement for DRTK/TDCP method is sufficient for lane-level accuracy, however, the availability could be compromised due to the strict dependence on GPS and inter-vehicle communications. Periods of DRTK outages result in gaps in the relative path. TDCP outages cause the path to be lost completely. There is also the potential for faulty DRTK and TDCP measurements due to incorrect integer ambiguity resolution, multipath, etc. Faulty DRTK measurements will cause spikes in the path, while a faulty TDCP measurement will bias the path.

To enhance the availability and reliability of this method, Travis and Martin presented a method of closely coupling the DRTK and TDCP algorithms with measurements from an inertial measurement unit (IMU). By coupling the algorithms with IMU measurements, the solution can bridge gaps due to GPS outages [25, 26]. However, a full 6-DoF IMU is not commonly available on most production vehicles. Additionally, the solution will still suffer from unbounded error growth with respect to following distance. In periods of GPS outages, this error growth will be exponential due to the double integration of accelerations and the influence of orientation errors on position [32]. The contributions presented in this

dissertation instead seek to address the issue of GPS outages and unbounded error growth by exchanging landmark observations between vehicles.

### Body-Centric Relative Path Generation

The body frame RPV measurements from perception-based sensors and UWB radios can also be used for path duplication when combined with odometry of the follower [33, 34]. The follower vehicle odometry from visual structure-from-motion [33], vehicle model [34], or otherwise is dead reckoned to generate the relative path.

In many cases, the accuracy of follower frame RPV measurements is relatively poor and the resulting path can be noisy. There are many ways of dealing with the noisy path to allow for a smooth controller response. One approach is to design the control law to act as a low pass filter and smooth out the closed loop response [35]. However, this may restrict the freedom in designing the controller. For example, in the case of a UWB RPV path, the control smoothing approach does not consider the geometry used to form the measurement.

Another method to handle the noisy path is to fit the path with a curve model such as a smoothing b-spline [34] or clothoid fit [36]. A curve fit is beneficial because it will bridge the gap between measurement outages with a reasonably shaped curve. The curve fitting process may also incorporate fault rejections with Random Sample Consensus (RANSAC) [37] or other robust regression techniques [38]. However, the curve fit process does not consider the model used to form the measurements or the model that describes the leader's motion.

Other methods address the problem with Bayesian estimation schemes such as an Extended Kalman Filter [33] or Particle Filter [39]. These methods include a model of the measurements used to form the path (e.g., the geometry used to calculate the RPV from UWB radios). A motion model of the lead vehicle can also be taken into account by this approach. When measurements of lead vehicle odometry are available, this information is used in the prediction step of the estimator. This method was employed in [26] by augmenting

the DRTK relative positioning method with IMU measurements from both the leader and follower vehicle.

Even when the lead vehicle odometry is unavailable, some knowledge of the model can be used to improve the solution. Certain motion constraints of the leader can be imposed, such as non-holonomic, constant velocity, constant rotation rate, as used in [23]. The Bayesian estimation scheme also allows for probabilistic fault rejection at the measurement level and also bridges small gaps in measurement outages in the prediction phase. However, bridging larger gaps of measurement outages proves to be more challenging, especially when the motion of the leader is not well known.

When follower frame odometry methods are used (e.g., INS, vehicle model, visual odometry), the dead reckoning position uncertainty is now influenced by heading errors and the uncertainty grows exponentially [40]. Most odometry sources of this kind have relatively low accuracy (compared to TDCP) which limits the maximum achievable following distance.

### 1.4.3 Landmark Navigation

Existing literature has used landmark navigation for mitigating the error growth involved in dead reckoning. Landmark navigation is the basis of most navigation strategies in the natural world [44]. The intuitive nature of this approach makes landmark navigation attractive in many applications.

Landmarks can be observed using a variety of sensors. Some examples of landmark observations include bearing from a camera, relative positions from radar/lidar, or range measurements to a fixed beacon. There is plenty of freedom as to what can be considered a landmark, but ideally the landmark should be easily detected and uniquely described. The process of detecting and describing landmarks from raw images or point clouds is referred to as feature extraction [45]. Some common examples of landmarks include trees, light posts, fiducial markers, etc.

In some cases, the absolute positions of the landmarks are known and absolute position estimates can be made from the landmark observations, effectively bounding position error [46]. When landmark positions are not known, they can be estimated in a process referred to as Simultaneous Localization and Mapping (SLAM). SLAM has been proven to bound positional error growth when making loop closure (repeated observation of a scene) [47]. Collaborative SLAM approaches have also been considered in the literature, where multiple mobile platforms make observations of the same environment. When two vehicles observe the same scene with a sufficient number of landmarks, their relative positions become observable [48, 49]. This dissertation builds off the concepts of collaborative SLAM and applies them to the leader-follower path estimation problem.

The principles of landmark navigation have been applied to the long distance leader-follower problem to bound path errors with respect to following distance. In [31], the authors use features extracted from a monocular camera as landmark observations. From this, correspondence was made between the scene captured by the lead (a human with a helmet-mounted camera) and that of the follower (a differential drive robot with a 360° camera). The registration process in [31] requires many features (100–200 per frame) for sufficient accuracy and may experience diminished performance in certain sparse environments (e.g., snow conditions or desert environment). Comparatively, the method proposed in this dissertation leverages the accuracy of differential GPS observations and requires only a few landmarks to bound the errors at large following distances.

In [50], the author adapted the DRTK/TDCP path generation approach to include opportunistic landmark observations from lidar sensors on each vehicle. Because of the accuracy in TDCP and DRTK, relatively few landmark correspondences were needed for bounding path errors. Because of this, the solution could be applied in sparse environments or to use only the most distinguishable features in an environment. The work presented in [50] shows that the landmark observations alone could be used for estimating the relative path, enhancing the solution availability. The work presented in this dissertation expands

on this approach with a thorough evaluation and an in-depth study of the observability conditions.

## 1.5 Contributions

The work presented in this dissertation expands on the odometry based path duplication method described in [25, 26] by incorporating landmark registration and exchange between vehicles. The landmark observations effectively bound the error that would otherwise accumulate in long distance following scenarios. Additionally, the use of differential GPS measurements allows for sufficient path accuracy to be attained with a relatively low number of landmarks compared to common map-matching techniques.

A graph-based estimation framework is presented as the method for estimating the relative path. A theoretical and empirical study of the error propagation of the path estimate is presented. The study uses observability analysis to characterize the solution's availability in certain conditions. Additionally, a Monte Carlo evaluation is used to verify the state estimate uncertainty reported by the path estimator.

The proposed method for relative path estimation is demonstrated with experimental data. This application consists of differential GPS observations of odometry and inter-vehicle RPV using the outputs of TDCP and DRTK algorithms, respectively. The body-centric odometry measurements used in the experimental results are calculated from wheel speed and yaw rate sensors. Additionally, a multi-channel lidar provides landmark observations in this application. The landmarks are detected from dense lidar point clouds using a unique feature extraction process.

In summary, this dissertation provides the following contributions:

- A study of the error propagation in existing relative path estimation solutions.

- The introduction of a novel localization approach for relative path following of non-homogeneous networks with large following distances using differential GPS, vehicle odometry, and landmark observations.
- An analysis of the observability conditions of the proposed localization approach.
- A statistical analysis of the proposed algorithm compared to existing solutions using a controlled simulation environment that is verified using experimental data.

## 1.6 Dissertation Overview

The dissertation is structured as follows. Chapter 2 provides background information for path following by presenting the relevant coordinate frames and nomenclature as well as detailing the path generation and path following control techniques mentioned in Section 1.4. This chapter also describes the error characteristics of relative path generation and further motivates the multi-sensor fusion approach that follows. Next, Chapter 3 details the graph-based path estimation approach that was developed for this dissertation. Chapter 4 describes the front-end of relative path estimation, which takes raw sensor data and produces the observations used in the estimator. This includes an in-depth description of the differential GPS algorithms used, the process of turning wheel speeds and rotation rates into vehicle odometry, and the landmark feature extraction process that operates on lidar point cloud data. A simulation study is provided in Chapter 5 that characterizes the statistical behavior of the presented path estimation approach for various controlled scenarios. Experimental data is used to validate the findings of the simulation study. The experimental setup and results are given in Chapter 6. Conclusions and considerations for future work are provided in the final chapter.

# Chapter 2

## Relative Path Following

This chapter introduces the concepts involved in the process of relative path following. This chapter will serve as a reference for the work that follows by introducing the relevant coordinate frames, nomenclature, and variables used in path following. Also, a high-level description of certain path following control strategies is provided for specifying the outputs required by a path estimation approach. Lastly, this chapter provides a derivation of existing path generation approaches for characterizing their shortcomings and motivating the contributions that follow.

The chapter is structured as follows. Section 2.1 introduces the relevant coordinate frames along with the equations used to transform positions and orientations between frames. Section 2.2 follows with a description of the nomenclature, notations and variables used in the process of path estimation and control. Next, Section 2.3 describes three common path duplication control strategies and equations for calculating the path variables used as controller feedback. Section 2.4 details the process of relative path generation as presented in [25,26] that serve as the basis for the contributions of this dissertation. Section 2.4 provides the equations of relative path generation used to calculate path waypoint positions and their associated covariance. From these equations, the error characteristics and other limitations of relative path generation are highlighted for motivating a sensor fusion approach as is presented in the chapters that follow.

## 2.1 Coordinate Frames

### 2.1.1 Coordinate Frame Descriptions

This section describes the coordinate frames involved in path following. For brevity, the frames are described and illustrated in two-dimensions. First, the navigation frame ($n$) is introduced, which is the fixed frame of reference. For the work presented in this dissertation, the navigation frame is aligned with the North and East directions as shown in Figure 2.1. The navigation frame's origin is fixed to the earth at some location in the vicinity of the convoy vehicles. In some cases, the UTM coordinate system is used to define a standardized origin. However, since the path following approaches described here are relative in nature, the origin is arbitrary.

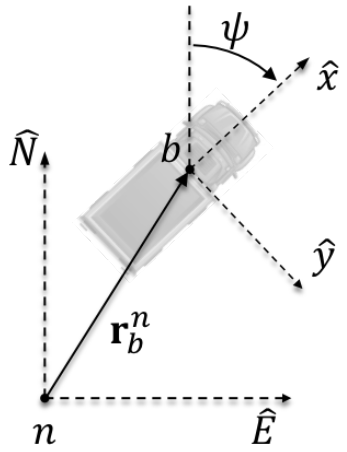


Figure 2.1: Coordinate Frame Diagram

Next, the body frame ($b$) is defined as the coordinate frame attached to a vehicle. The origin of the body frame is located at the vehicle's control point, which is often chosen to be at the vehicle's center of gravity or the center of the front or rear axle. The $x$-axis of the body frame points in the vehicle's forward direction and the $y$-axis points out the passenger side. The heading angle ($\psi$) is the angle from the North axis ($\hat{N}$) to the body frame's $x$-axis as shown in Figure 2.1. When the heading is zero, the $x$-axis is aligned with the North direction and the $y$-axis is aligned with the East direction. The vehicle's position is defined

by a vector pointing from the origin of the navigation frame to the origin of the body frame resolved in the navigation frame ($\boldsymbol{r}_b^n$). The 2D position vector is made up of the north and east components ($\boldsymbol{r}_b^n = [N, E]^T$).

In general, the notation used for position vectors is as follows. The superscript denotes the frame in which the vector is resolved. The subscript denotes the vector's endpoint. In the previous example ($\boldsymbol{r}_b^n$), the subscript $b$ refers to the origin of the body frame and the superscript $n$ refers to the navigation frame. For relative position vectors, the subscript includes additional information that is best described through example: for two points in space, $c$ and $d$, $\boldsymbol{r}_{c|d}^n$ represents the position of $c$ with respect to $d$, resolved in the navigation frame ($n$).

At the vehicle level, each sensor is associated with its own reference frame. Knowledge of the sensor position and orientation on the vehicle is used to transform sensor measurements to a common frame on the vehicle, i.e., the body frame. Figure 2.2 is shown as an example of vehicle-level frames including sensor frames (GPS, radar, lidar) along with other relevant frames such as wheels and trailer links. The sensor frames are assumed to be fixed to the vehicle such that the position and orientation on the vehicle are constant. A sensor's position and orientation on the vehicle, also referred to as extrinsic parameters, must be measured or calibrated in order to transform sensor measurements into a common frame for use in a sensor fusion process. In some cases, the extrinsic parameters are included as states in an estimation scheme for online calibration.

Another coordinate frame that is relevant for path following control is the path tangent frame ($t$) as depicted in Figure 2.3. The origin of the path tangent frame is located on the path and parallel to the origin of the following vehicle's body frame ($f$). The $x$-axis of the path tangent frame lies tangential to the path and the $y$-axis is orthogonal to the $x$-axis (normal to the path) following the right-hand rule. The path tangent frame is used to calculate path deviation variables as described later in Section 2.3.2.

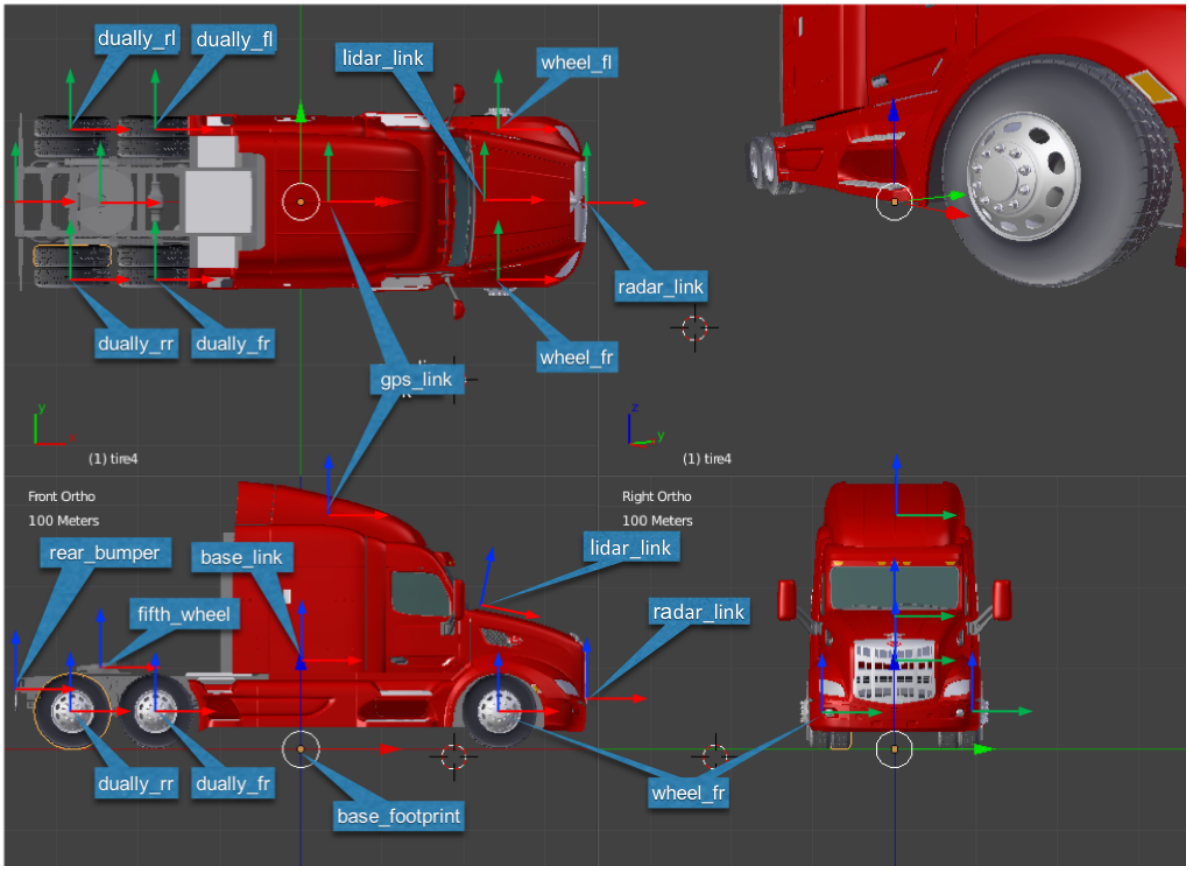


Figure 2.2: Vehicle-Level Reference Frames

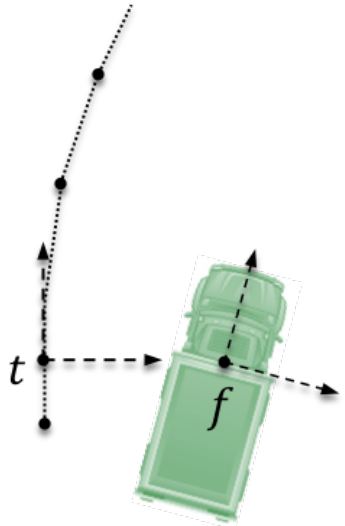


Figure 2.3: Path Tangent Frame Diagram

### 2.1.2 Coordinate Frame Transformations

Coordinate frame transformations are essential to sensor fusion and localization. For positional information, the transform is composed of a translation and rotation. Take for example a sensor that measures the position of a point in space ($p$) with respect to the sensor origin ($s$) resolved in the sensor frame ($s$). The goal is to transform the sensor measurement ($\boldsymbol{r}^s_{p|s}$) into the body frame as $\boldsymbol{r}^b_{p|b}$. The sensor's extrinsic parameters ($\boldsymbol{r}^b_{s|b}$, $\mathbf{C}^b_s$) are used to perform the transformation as shown in the equations below [27].

$$\boldsymbol{r}^b_{p|s} = \mathbf{C}^b_s \boldsymbol{r}^s_{p|s} \tag{2.1}$$

$$\boldsymbol{r}^b_{p|b} = \boldsymbol{r}^b_{p|s} + \boldsymbol{r}^b_{s|b} \tag{2.2}$$

Equation (2.1) shows the pure rotation of the measurement into the body frame using the rotation matrix ($\mathbf{C}^b_s$). For the work presented in this dissertation, rotation matrices are used to perform vector rotations. In general, the notation used for rotation matrices is to have the target frame in the superscript and source frame in the subscript. In other words, the rotation matrix ($\mathbf{C}^d_c$) rotates a vector from the $c$ frame to $d$ frame. The 2D rotation matrix from the body frame ($b$) to the navigation frame ($n$) is a function of the heading angle ($\psi$) as shown in Equation (2.3).

$$\mathbf{C}^n_b(\psi) = \begin{bmatrix} \cos(\psi) & \sin(\psi) \\ -\sin(\psi) & \cos(\psi) \end{bmatrix} \tag{2.3}$$

More information on coordinate frame transformations can be found in Appendix A.

## 2.2 Path Variables and Notations

### 2.2.1 Vehicle Notations

In practice, collaborative navigation involves the tracking of multiple vehicles and therefore multiple body frames. To distinguish the body frame of two vehicles in a convoy, two frames are introduced: the follower frame ($f$) and the leader frame ($\ell$). Both the follower frame and leader frame are stand-ins for the body frame of the respective vehicle.

### 2.2.2 Path Waypoint Indexing

In the context of leader-follower path following in this dissertation, the path is a set of $N$ waypoints, each representing a past position of the lead vehicle. The path is indexed such that the first waypoint ($i = 1$) corresponds to the lead vehicle's current state (its position at time $t_k$). In general, the $i^{th}$ waypoint corresponds to the lead vehicle's state at $i - 1$ epochs in the past (its position at time $t_{k-i+1}$). Figure 2.4 shows an example path resolved in the follower frame. In this diagram, the vector $\boldsymbol{r}^{f}_{p_i|f}$ represents the relative position of the $i^{th}$ waypoint with respect to the follower.

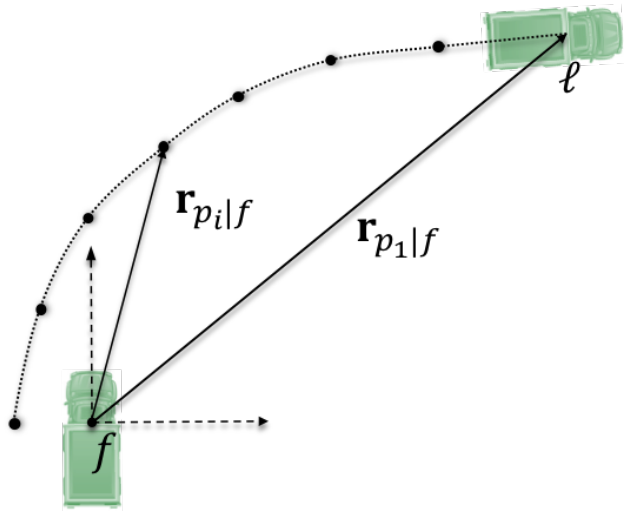


Figure 2.4: Leader-Follower Path Diagram

## 2.3 Path Following Control

Although vehicle control is not performed in this dissertation, an understanding of certain path following control schemes is provided for understanding the required outputs of a path estimator. In its purest form, a path following control scheme acts to determine a series of actuator commands that result in a trajectory that duplicates the path taken by a lead vehicle. In this section, three common path following control strategies are described.

Ground vehicle control is often separated into two components: lateral and longitudinal control [1]. Lateral control strategies, which are relevant to the work presented in this dissertation, determine steer angle commands for regulating the lateral path deviation. Longitudinal control, which actuates engine and braking commands for controlling to a desired following distance, is less critical in this dissertation due to the focus on long distance following scenarios. Therefore, the three control strategies described in this section are all types of lateral controllers.

### 2.3.1 Virtual Lead Following

A common choice for path following control is the Virtual Lead Following method [11]. The goal of Virtual Lead Following is the same as any other path following control schemes: to minimize lateral path deviation. However, Virtual Lead Following works indirectly by mimicking short distance following approaches. The Virtual Lead Following method is a popular choice since it is comparatively stable, easy to implement, and intuitive to how human operators accomplish the same goal.

As its name suggests, the Virtual Lead Following Method chooses a point on the path to serve as a virtual lead (sometimes referred to as a ghost vehicle) as depicted in Figure 2.5. The position of the virtual lead is based on the path waypoint positions, the follower's current position, and a user-defined lookahead distance. When waypoints are sufficiently dense, the virtual lead is chosen by reverse iterating through the path waypoints until the distance from the follower to the waypoint is just outside the lookahead radius. For sparse

paths, the virtual lead position can be interpolated between two path waypoints to improve the continuity of the controller's feedback signal, resulting in a smoother response.

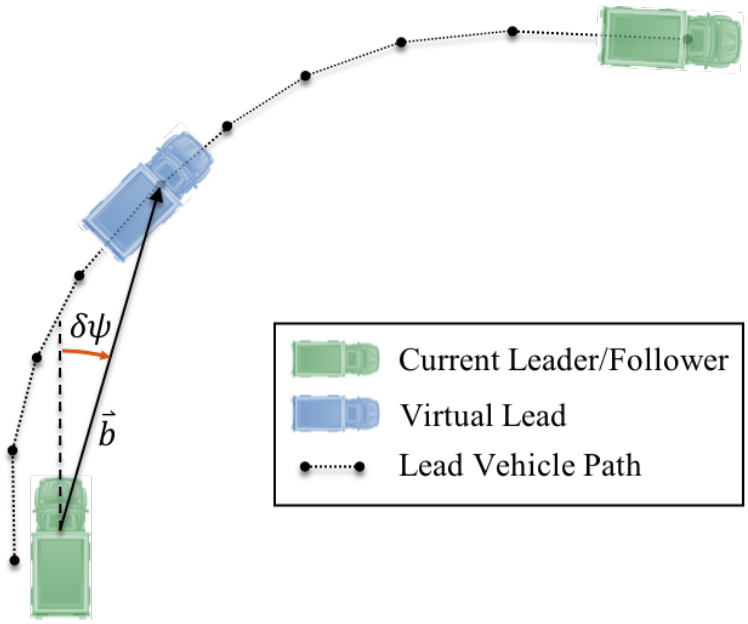


Figure 2.5: Virtual Lead Following Diagram

The follower vehicle is then controlled in order to drive the relative bearing ($\delta\psi$) to zero. The relative bearing from a following vehicle to a virtual lead waypoint can be seen in Figure 2.5. Given the RPV from the follower to the virtual lead, resolved in the body frame of the follower ($\mathbf{b}^f = [b_x, b_y]^T$), the relative bearing is calculated as shown in Equation (2.4).

$$\delta\psi = \text{atan2}(b_y, b_x) \tag{2.4}$$

The choice of lookahead distance will impact the controller's response. The choice of lookahead distance changes the effective damping of the response. A short lookahead distance may result in oscillations about the path while a larger lookahead distance will suffer from cut corners on turns [51]. To allow for consistency in the controller's response at various speeds, the lookahead distance can be designed to scale proportionally with speed. In most

applications the lookahead distance is chosen through a process of hand tuning in order to suit the user's preference.

### 2.3.2 Lateral Control Following

Another path following control scheme is Lateral Control Following, which seeks to minimize path deviation by operating directly on path deviation variables. The path deviation variables include lateral path position ($Y$) and path yaw ($\psi_p$) which are represented visually in Figure 2.6. By regulating the path deviation variables in a control law, the following vehicle will be successful in duplicating the leader's path [9]. The steps for calculating the path deviation variables are described below.

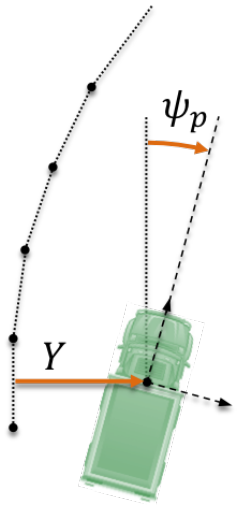


Figure 2.6: Lateral Control Following Diagram

The first step is to determine the waypoint index to use for calculating path deviation variables. The path is first transformed into the following vehicle's body frame resulting in a set of $N$ positions relative to the follower and resolved in the follower frame $\left\{\boldsymbol{r}^f_{p_1|f}, \boldsymbol{r}^f_{p_2|f}, \ldots, \boldsymbol{r}^f_{p_N|f}\right\}$. Then, the path index ($i$) is iterated until reaching the $y$ intercept, at which point the sign of the $x$ component will be different for the $i$ and $i+1$ waypoints. This point on the path will be referred to as the follower intercept index ($i_{int}$).

The path yaw angle ($\psi_p$) is determined from the relative positions of the $i_{int}$ and $i_{int}+1$ waypoints as shown below.

$$\psi_p = -\operatorname{atan2}(y_{i_{int}} - y_{i_{int}+1}, x_{i_{int}} - x_{i_{int}+1}) \tag{2.5}$$

The waypoint position at $i_{int}$ is then rotated into the path tangent frame for determining lateral path position.

$$\boldsymbol{r}^t_{p_{i_{int}}|f} = \mathbf{C}^t_f \boldsymbol{r}^f_{p_{i_{int}}|f} \tag{2.6}$$

In this equation, $\mathbf{C}^t_f$ is the rotation matrix from the follower frame to the path tangent frame that is formed using path yaw. The lateral path position ($Y$) is then taken as the negative of the $y$ component of $\boldsymbol{r}^t_{p_{i_{int}}|f}$.

### 2.3.3 Model Predictive Control Following

Continuing with the trend of increasing complexity, a Model Predictive Control approach is perhaps the most involved. As its name suggests, this method uses a model of the vehicle that relates control action inputs (e.g., steer angle) to vehicle motion. For a set of potential future steer angles, a trajectory is predicted by propagating the vehicle's state according to the model. The predicted trajectory is compared to the waypoints and, using a cost function as decided by the designer, a cost is calculated to quantify how well the trajectory follows the waypoints. The control inputs are determined by minimizing the cost function with respect to steer angle over a certain prediction horizon. The cost function could also include qualitative metrics (e.g., limit jerk for rider comfort), collision avoidance, safety factors and/or physical limitations of the vehicle [29, 52].

The details are much more involved, but a simplified example is shown in Figure 2.7 for a visual representation. This simplified example only predicts constant steer angles over the horizon (resulting in circular paths), but in practice higher order functions of steer angle

can be used for more complex trajectory predictions. This figure shows the "minimum cost trajectory" in orange, which is the trajectory that most closely tracks the path waypoints.

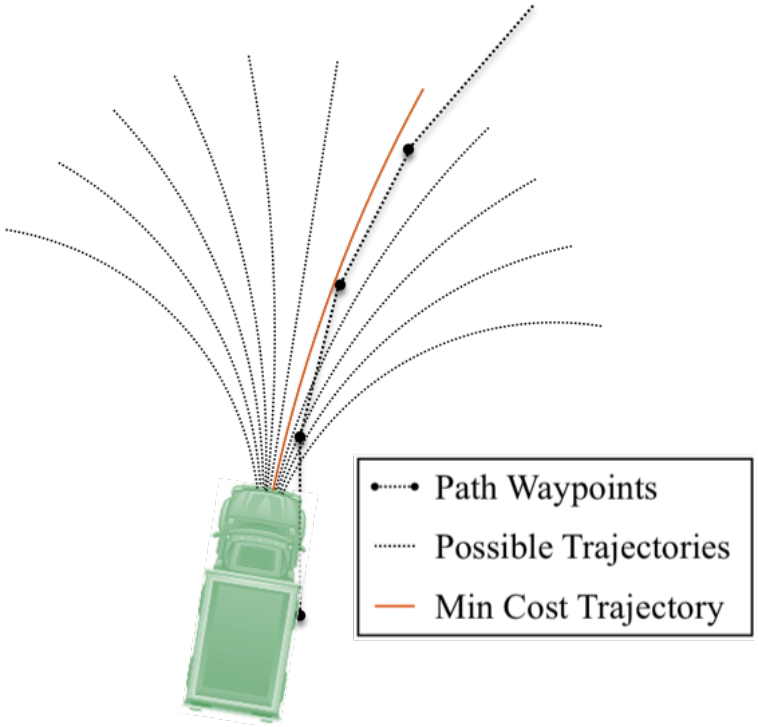


Figure 2.7: Model Predictive Control Following Diagram

Note that in order to calculate cost, say as the average path deviation, a rotation must be known from the vehicle's body frame to the frame in which the waypoints are resolved. For a set of waypoints that are resolved in the navigation frame, the vehicle heading is used to make this rotation. It may also be advantageous to estimate higher order states, such as lateral acceleration, if the designer seeks to optimize for safety and rider comfort.

### 2.3.4 Path Variables Required for Path Following

All three path following control methods listed here require the path to be resolved in the follower frame ($f$) for calculating variables for feedback control. If the estimated path is resolved in the navigation frame, then the heading of the follower is required to rotate the

path into the appropriate frame. For the relative path estimator presented in Chapter 3, vehicle heading is included in the state vector for estimation.

Additionally, the control reference must be provided at a sufficiently high rate to allow for safe controlled following. The estimator presented in this dissertation provides path estimates at 2 Hz (the GPS sample rate), which is too low relative to the bandwidth of ground vehicles [1]. In practice, an odometry solution (e.g., vehicle model, INS) can be used to both track the vehicle's heading and propagate the path at a higher sample rate.

The path estimation rate of 2 Hz allows for sufficient density of path waypoints for the given objective. The level of waypoint density required for accurate control is determined by a) the target accuracy of a path following controller and b) the expected dynamics of the lead vehicle. If the application calls for increased density of path waypoints, the GPS sample rate may be adjusted accordingly. However, in any situation, a large enough gap in the path caused by missing waypoints would prove challenging for path following control. For the presented work, the density requirement is said to be fulfilled when no waypoints are dropped.

In conclusion, path following control requires the provided path estimate to be

(a) resolved in the body frame of the following vehicle.

(b) provided at a high sample rate.

(c) sufficiently dense in order to recreate the path taken by the lead vehicle.

An overview of relative path generation is given in the next section with these high-level design requirements in mind.

## 2.4 Relative Path Generation

This section provides an overview of the relative path generation approach presented in [25, 26] to serve as the foundation for the contributions made in this dissertation. The relative path generation approach in [25,26] assumes the following observations are available:

following vehicle odometry ($\Delta \boldsymbol{r}_f^n$) and inter-vehicle RPV ($\boldsymbol{r}_{\ell|f}^n$). The odometry measurement is an RPV that represents the change in position of a vehicle between consecutive time epochs. The inter-vehicle RPV is the position of the lead vehicle with respect to the follower at a given time epoch. In this case, both the odometry and inter-vehicle RPV measurements are resolved in the navigation frame. Figure 2.8 shows these vector observations.

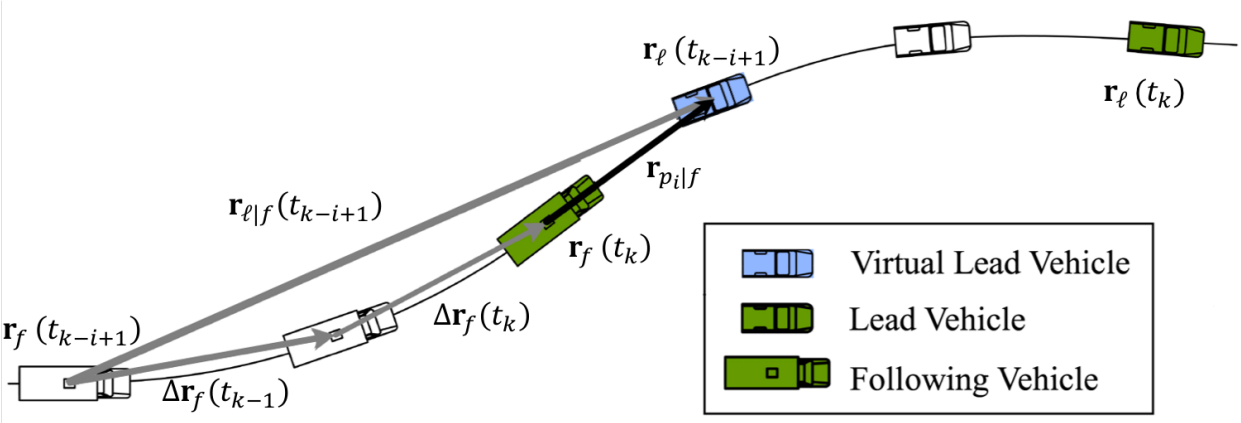


Figure 2.8: Relative Position Vector Observations

### 2.4.1 Relative Path Generation Equations

Given this set of observations, the relative path can be calculated. The solution to calculating the relative path is given in Equation (2.7).

$$\boldsymbol{r}_{p_i|f} = \boldsymbol{r}_{\ell|f}(t_{k-i+1}) - \sum_{j=1}^{i-1} \Delta \boldsymbol{r}_f(t_{k-j+1}) \tag{2.7}$$

This is the approach presented in [25, 26] for generating relative paths using differential carrier phase GPS techniques. In [25, 26], the following vehicle odometry is estimated using the Time Differenced Carrier Phase (TDCP) approach and the inter-vehicle RPV from a Dynamic-Base RTK (DRTK) solution.

In a similar fashion, the relative path could be calculated using lead vehicle odometry or some combination thereof. In fact, there are $i$ different solutions (not independent) for

the $i^{th}$ path element and so the problem is over-defined. An over-defined problem such as this lends itself well to estimation, where the optimal estimate is some weighted combination of all solutions.

### 2.4.2 Relative Path Generation Errors

In practice, the measurements used in generating the path are not perfect. Each RPV and odometry measurement contains noise, which results in errors in the resulting path. This section describes the way in which the measurement errors map to path errors.

The noisy measurements $(\mathbf{z}_{rpv}, \mathbf{z}_{\Delta})$ are modeled as having additive noise $(\boldsymbol{\nu}_{rpv}, \boldsymbol{\nu}_{\Delta})$ as shown in Equations (2.8) and (2.9).

$$\mathbf{z}_{rpv}(t_k) = \boldsymbol{r}_{\ell/f}(t_k) + \boldsymbol{\nu}_{rpv}(t_k) \tag{2.8}$$

$$\mathbf{z}_{\Delta}(t_k) = \Delta\boldsymbol{r}_f(t_k) + \boldsymbol{\nu}_{\Delta}(t_k) \tag{2.9}$$

The measurement noise is modeled as zero-mean and Gaussian with a known covariance matrix $(\mathbf{R})$. For the analysis presented in this section, it is also assumed that the noise is stationary such that $\mathbf{R}(t-\tau) = \mathbf{R}(t) = \mathbf{R}$. The covariance is defined using the expectation operator as shown in Equations (2.10) and (2.11).

$$\mathbf{R}_{rpv} = E\left[(\boldsymbol{\nu}_{rpv})(\boldsymbol{\nu}_{rpv})^T\right] \tag{2.10}$$

$$\mathbf{R}_{\Delta} = E\left[(\boldsymbol{\nu}_{\Delta})(\boldsymbol{\nu}_{\Delta})^T\right] \tag{2.11}$$

For the path generation formula given in Equation (2.7), the measurement noise maps to path errors $(\boldsymbol{\nu}_{p_i})$ according to Equation (2.12).

$$\boldsymbol{\nu}_{p_i}(t_k) = \boldsymbol{\nu}_{rpv}(t_{k-i+1}) + \sum_{j=1}^{i-1} \boldsymbol{\nu}_{\Delta}(t_{k-j+1}) \tag{2.12}$$

In this equation, $i$ is the path index with $i = 1$ corresponding to the leader's current position. The covariance of the $i^{th}$ path element ($\mathbf{R}_{p_i} = E\left[(\boldsymbol{\nu}_{p_i})(\boldsymbol{\nu}_{p_i})^T\right]$) can be solved analytically using Equation (2.13).

$$\mathbf{R}_{p_i} = \mathbf{R}_{rpv} + \sum_{j=1}^{i-1} \mathbf{R}_{\Delta} = \mathbf{R}_{rpv} + (i-1)(\mathbf{R}_{\Delta}) \tag{2.13}$$

Equation (2.13) shows the linear relation of path variance to path index. In practice, the path is maintained to a finite length ($N$) based on the following distance. The $N^{th}$ waypoint will have the highest amount of expected error $\left(\mathbf{R}_{p_N} = \mathbf{R}_{rpv} + (N-1)(\mathbf{R}_{\Delta})\right)$. Therefore, the path errors are expected to increase linearly with following distance.

### 2.4.3 Limitations of Relative Path Generation

Relative path generation as it is described in this section has its limitations. First, the method assumes that measurements are always available at a consistent sample rate. In practice, measurements may be corrupt or unavailable at times. For example, in the case where GPS provides the observations, measurements may be dropped due to a lack of satellites or corrupted by multipath acting on the carrier signal.

The measurement errors map directly to errors in the generated path. Each inter-vehicle RPV measurement that is dropped will result in a gap in the path (missing path waypoints). Gaps in the path may cause the following vehicle to cut corners in turns or lose a control reference entirely for large enough gaps. Further, a faulty inter-vehicle RPV measurement will cause a spike in the path. Each odometry measurement that is dropped will result in total loss of the path since the path is formed as a cumulative sum of odometry measurements. For the same reason, a faulty odometry measurement will cause a bias in the path since errors are propagated in the path generation equations. Even in the ideal case with full availability, the path errors are unbounded with respect to following distance, as shown in the previous section.

To address the shortcomings of relative path generation, additional information can be incorporated and fused into the solution. One option is to use odometry information of the lead vehicle. An estimation scheme that optimally fuses the leader and follower odometry along with the inter-vehicle RPV would reduce the rate at which path error grows as a function of following distance. However, even with an improved error growth rate, the error would still be unbounded.

Additional improvements to availability and accuracy can be realized by incorporating body-centric odometry and landmark observations. With sufficient landmark coverage, gaps in GPS availability can be spanned for estimating the relative path. Further, the body-centric odometry measurements allow for spanning gaps between landmarks and in smoothing the fused solution. The use of landmark observations has the added benefit of bounding the error growth with respect to following distance, allowing for long distance path estimation with accuracy sufficient for path following control. The fusion approaches mentioned here are detailed in the next chapter.

## 2.5 Conclusion

This chapter presented the problem of relative path following from a high level to motivate the contributions that follow. The first two sections serve as reference for the coordinate frames, variables, and nomenclature that is used throughout the document. Next, Section 2.3 provided high-level descriptions of common control techniques for path following. This understanding of path following control algorithms provided clarity on the required outputs of a path estimator. Finally, a method of relative path generation that uses vehicle odometry and inter-vehicle RPV was summarized in Section 2.4. The section revealed how the relative path generation approach exhibits unbounded errors with respect to following distance and motivated the use of sensor fusion which is the subject of the next chapter.

# Chapter 3

## Graph-Based Relative Path Estimation

In this chapter, a novel approach to relative path estimation is presented. Relative path generation, as described in the previous chapter, exhibits linear error growth with following distance and is dependent on continuous GPS coverage. The estimation-based approach presented in this chapter is designed for improving the availability and accuracy of a leader-follower system. These improved characteristics are the result of fusing information from additional sensors generally available in automated vehicles.

A graph-based estimation scheme is used as the framework for fusing the measurements provided by the various sensors. These measurements include the navigation frame odometry (also referred to as GPS odometry) and inter-vehicle RPV as used in relative path generation. Additionally, body-centric odometry measurements are used to improve availability and observability conditions of the graph-based estimation. Landmark observations are exchanged between vehicles and used as observations in the graph-based estimator. The positions of the landmarks are unknown and are included in the state vector for estimation. The landmark observations are used to improve the availability of the path estimate and to bound the error growth with respect to following distance.

The material in this chapter is presented for generic relative observations. Unless used to serve as an example, the specific sensors used to generate the observations are not mentioned. The material is presented in this way to allow for a wide range of future implementations. This chapter is followed by Chapter 4 where specifics are provided on the sensors and methods used for generating measurements from the experimental data.

This chapter starts in Section 3.1 by discussing the considerations made in choosing the graph-based estimation framework. Section 3.2 follows with a derivation and the underlying

equations of the graph-based path estimation scheme in its most basic form. Then, Section 3.3 describes the computational demand of the estimation scheme and the steps taken to reduce this demand by considering the topology of the graph. Section 3.4 demonstrates how errors manifest in graph-based path estimation through specific examples. Section 3.5 expands on the graph-based path estimation scheme with modifications that allow for processing nonlinear measurements. The graph-based estimator is modified further in Section 3.6 by augmenting the state vector to allow for landmark observations. An overview of the different measurement models used in the estimation is given in Section 3.7. Finally, Section 3.8 discusses the observability conditions of the nonlinear graph-based path estimator and provides a method for evaluating the observability using graph-theoretic properties.

## 3.1 Estimation Framework Considerations

There are various estimation frameworks that could be employed to accomplish the same goal of fusing multiple sensor measurements for estimating the path taken by the lead vehicle. The chosen framework should have the following attributes:

- Flexibility for ad-hoc measurements
- Scalability with increasing following distance
- Ability to process nonlinear measurements

Of the various frameworks, two stand out for initial consideration: a Kalman filter framework and a graph-based information framework. A Kalman-based framework is considered due to its computational efficiency and ease of implementation. On the other hand, a graph-based approach is considered due to the connected nature of the measurements and vehicle states along with other numerical advantages [53].

The fusion approach should allow for measurements to be incorporated in an ad-hoc fashion since the availability of each sensor may vary. This is where a graph-based estimation scheme is preferable for reasons that will become clear in later sections. The Kalman

filter is flexible in which information is used in the measurement update phase, however, it requires continuously available motion measurements for the time update step, which is not necessarily the case with GPS odometry.

Another consideration in choosing the estimation framework is the scalability with increasing following distance. For a Kalman-based approach, state estimates would need to be maintained for the entirety of the path. This includes estimates of all path waypoints and landmarks from the leader to the follower. The state dimension would eventually become intractably large and exhaust processor and memory resources. Conversely, a graph-based approach is flexible to which states are included in the state vector. By this, only path waypoints in the vicinity of the following vehicle would need to be estimated.

The chosen estimation framework must also be accepting of nonlinear measurements since orientation estimates are required to transform observations to a common frame. The extended Kalman filter handles nonlinear measurements by a first order Taylor series expansion [27]. In contrast, the graph-based approach handles nonlinearities by iterating on a nonlinear least squares problem using Gauss-Newton reduction. In other words, the graph-based estimator solves the *full SLAM problem*, which better addresses the nonlinearities in the system [54]. One of the main drawbacks to solving the *full SLAM problem* is the demand on the processor. However, through graph reduction and intelligent state selection, this processor demand can be reduced. After these considerations, a graph-based approach was chosen and will be described in detail in the following sections.

## 3.2 Graph-Based Relative Path Estimation

Graph-based estimation is a method of forming the least squares problem in a way that takes advantage of the graphical nature of the problem at hand. In general, a graph is made up of a set of nodes (also known as vertices), a set of edges, and a relation that associates each edge with two vertices, called endpoints [55]. For the case of path estimation as presented here, the graph is a Relative Pose Graph (RPG) where the nodes represent

vehicle poses in time and the edges represent relative measurements that relate two poses. Each edge is weighted based on the accuracy of a measurement. The optimization process then solves the least squares problem to determine the optimal state estimates. Conceptually, the optimization step acts like a mechanical spring-mass-damper system. In this mechanical analogy, high confidence measurements are given a greater spring stiffness compared to the other links in the system and the equilibrium positions of the masses correspond to the optimal estimate [47].

### 3.2.1 Graph Representations

This section will introduce the graph-based estimation approach with a simple example. Consider the fusion problem first introduced in Section 2.4 in which the following measurements are given: a) GPS odometry of the following vehicle, b) GPS odometry of the lead vehicle, and c) inter-vehicle RPV measurements. It was shown in Section 2.4 that there are multiple ways to calculate the relative path positions by using some combination of leader and follower odometry. Intuition says that if all available information is utilized then the resulting estimate would be more accurate, on average, than any directly calculated result. Figure 3.1 shows the pose graph representation of this example represented in cartesian space.

The graph is made up of two subgraphs: the follower subgraph ($F$) and leader subgraph ($L$) as defined in Equations (3.1) and (3.2).

$$\{f_1, f_2, \ldots, f_{N-1}, f_N\} \in V(F) \tag{3.1}$$

$$\{\ell_1, \ell_2, \ldots, \ell_{N-1}, \ell_N\} \in V(L) \tag{3.2}$$

The two subgraphs are connected by inter-vehicle RPV measurement edges. The odometry edges connect vertices at consecutive time epochs within their corresponding subgraph. The

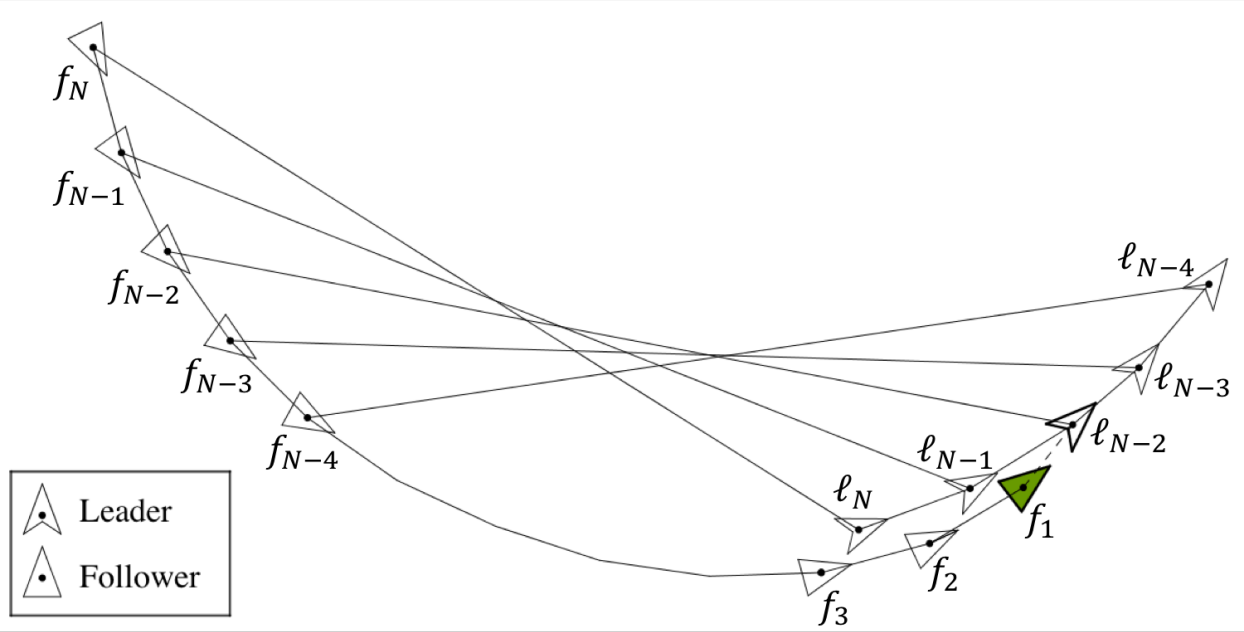


Figure 3.1: Graph-Based Path Estimation: Cartesian Graph

graph shown in Figure 3.1 is displayed in cartesian space, but the graph can also be organized by subgraph as shown in Figure 3.2.

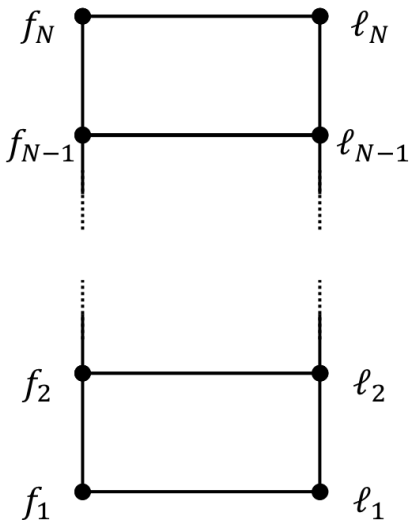


Figure 3.2: Graph-Based Path Estimation: Graphical Representation

### 3.2.2 Graph Optimization

The underlying mathematics behind estimating the state variables is referred to as graph optimization. Continuing with the previous example, the state variables are shown in vector form in Equation (3.3). For this simplified, linear example, only the 2D positions are to be estimated.

$$\mathbf{x} = \left[\begin{array}{c} \mathbf{x}_F \\ \hline \mathbf{x}_L \end{array}\right] = \left[\begin{array}{c} \mathbf{r}_f(t_k) \\ \mathbf{r}_f(t_{k-1}) \\ \vdots \\ \mathbf{r}_f(t_{k-N+1}) \\ \hline \mathbf{r}_\ell(t_k) \\ \mathbf{r}_\ell(t_{k-1}) \\ \vdots \\ \mathbf{r}_\ell(t_{k-N+1}) \end{array}\right] \tag{3.3}$$

The state vector is of size $n_x \times 1$ where $n_x$ is the total number of state variables to be estimated. For this specific example, this equates to $n_x = (2)(N)(n)$ where $N$ is the path length and $n$ is the state dimension at each vertex.

The estimated value of $\mathbf{x}$ is notated as $\hat{\mathbf{x}}$. Graph optimization seeks to determine the state estimate ($\hat{\mathbf{x}}$) along with the expected accuracy in the form of a covariance matrix. The state estimate covariance matrix ($\mathbf{P}$) is defined in Equation (3.4).

$$\mathbf{P} = E\left[(\mathbf{x} - \hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})^T\right] \tag{3.4}$$

The information theoretic approach consists of an information matrix ($\boldsymbol{\Omega} \in \mathbb{R}^{n_x \times n_x}$) and an information vector ($\boldsymbol{\xi} \in \mathbb{R}^{n_x \times 1}$). The rows of both $\boldsymbol{\Omega}$ and $\boldsymbol{\xi}$ correspond to the state vector. The columns of $\boldsymbol{\Omega}$ also correspond to the state vector. For each graph optimization, the information matrix and vector are preallocated to the proper dimension and initialized with all zeros.

Each measurement is then accounted for individually by adding information to the appropriate rows and columns of $\mathbf{\Omega}$ and $\boldsymbol{\xi}$. The measurement ($\mathbf{z}_{uv}$) is modeled with additive noise ($\boldsymbol{\nu}_{uv}$) and has an associated measurement matrix ($\mathbf{H}_{uv}$) as given in Equation (3.5).

$$\mathbf{z}_{uv} = \mathbf{H}_{uv} \begin{bmatrix} \mathbf{x}_v \\ \mathbf{x}_u \end{bmatrix} + \boldsymbol{\nu}_{uv} \tag{3.5}$$

The $u$ and $v$ subscripts denote the endpoint vertices of the relative measurement $\mathbf{z}_{uv}$. For this example, the RPV and odometry measurements are linear and share the same measurement matrix

$$\mathbf{H}_{uv} = [\mathbf{I}_{2\times 2}, -\mathbf{I}_{2\times 2}] \tag{3.6}$$

where $\mathbf{I}_{2\times 2}$ is a $2 \times 2$ identity matrix. The noise is modeled as zero-mean Gaussian with covariance matrix ($\mathbf{R}_{uv}$) as defined in Equation (3.7).

$$\mathbf{R}_{uv} = E\left[(\boldsymbol{\nu}_{uv})(\boldsymbol{\nu}_{uv})^T\right] \tag{3.7}$$

The rows and columns of $\mathbf{\Omega}$ and $\boldsymbol{\xi}$ corresponding with endpoints $u$ and $v$ are summed with the measurement's information (negative log likelihood) as shown in Equations (3.8) and (3.9).

$$\mathbf{\Omega}_{[uv]} \mathrel{+}= \mathbf{H}_{uv}^T \mathbf{R}_{uv}^{-1} \mathbf{H}_{uv} \tag{3.8}$$

$$\boldsymbol{\xi}_{[uv]} \mathrel{+}= \mathbf{H}_{uv}^T \mathbf{R}_{uv}^{-1} \mathbf{z}_{uv} \tag{3.9}$$

In these equations, $\mathbf{\Omega}_{[uv]}$ and $\boldsymbol{\xi}_{[uv]}$ represent the submatrix and subvector of $\mathbf{\Omega}$ and $\boldsymbol{\xi}$ corresponding to nodes $u$ and $v$. The "+=" operator indicates the addition of the right-hand side of the equation to $\mathbf{\Omega}_{[uv]}$ and $\boldsymbol{\xi}_{[uv]}$ in each equation.

Since all measurements are relative, an anchor node is chosen to serve as the origin of the estimated states. The convenient choice for the anchor node is the current position of

the follower. This node ($f_1$) is anchored by adding information to the rows/columns of the anchor node in the information matrix as shown in Equation (3.10) [56].

$$\mathbf{\Omega}_{[f_1]} \mathrel{+}= \mathbf{I}_{n\times n} \tag{3.10}$$

After accounting for all measurements and constraints in $\mathbf{\Omega}$ and $\boldsymbol{\xi}$, the result is a least squares problem. The state estimate ($\hat{\mathbf{x}}$) is determined by solving Equation (3.11).

$$\mathbf{\Omega}\hat{\mathbf{x}} = \boldsymbol{\xi} \tag{3.11}$$

The resulting estimate for the anchor states will be zero and all other state estimates will be relative to that point. The covariance of the state estimate is given by the inverse of the information matrix as shown below.

$$\mathbf{P} = \mathbf{\Omega}^{-1} \tag{3.12}$$

### 3.2.3 Observability Considerations for the Linear Case

The graph-based representation allows for the observability of the system to be analyzed using graph-theoretic properties. Here, the estimation problem is said to be observable if the information matrix is invertible. For the linear case, a node ($v_i$) is said to be observable if the graph contains a path from the anchor node ($v_a$) to node $v_i$.

In practice, the graph is used to find all $v_a$-connected nodes of the system. All other nodes are unobservable and their inclusion in the problem makes the information matrix degenerate and unable to be inverted. The rows and columns in the information matrix corresponding to unobservable nodes are removed and the states are not included in the estimation.

### 3.2.4 Path Length Determination

One aspect of graph-based path estimation that has not been discussed up to this point is the determination of the path length. The path length ($N$) is the number of waypoints in the path which is a design consideration when implementing the path estimation algorithm. One choice is to allow for $N$ to vary based on the current following distance such that the last waypoint is the follower intercept index ($N = i_{int}$). In this way, the waypoint positions at $N - 1$ and $N$ will be directly in front of and behind the follower's current position, respectively. A path of this size will be sufficient for calculating the path deviation variables in most situations.

However, in light of the observability conditions presented in Section 3.2.3, it may be advantageous to allow the path to extend beyond the follower intercept index ($N > i_{int}$). For example, in some situations the inter-vehicle RPV may not be available for a duration of time. With a larger path size, it would be possible to form the path on odometry measurements, effectively dead reckoning the path solution from the most recent inter-vehicle RPV.

Certain implementations may choose to set $N$ to a fixed value. This could be preferable in situations where the machine's memory must be allocated statically. With a set value for $N$, static memory can be allocated for the vectors and matrices involved in graph optimization. This would cut down on the processor demand of dynamically allocating memory for varying path sizes, resulting in a more computationally efficient implementation.

The method used for path length determination in this dissertation is based on a user-defined distance, referred to as the path tail length ($d_{tail}$). The path tail length determines the along-path distance from the waypoint position at $N$ and the waypoint position at the follower intercept index, $i_{int}$. With the path size dictated by a measure of distance, some intuition can be used in setting the path tail length. This could be based on the drift rate of the odometry system to ensure that a dead reckoned path will be within accuracy requirements. It may also be chosen based on the maximum range of the perception sensors in the case of landmark navigation which will be introduced in later sections.

## 3.3 Considerations for Computational Load

The graph-based path estimator described in the previous section exhibits an increasingly high computational demand as following distance, and thereby path length, increases. The matrix inversion step is the main contributor to the computational demand. As path length increases, so does the size of the state vector. Each additional waypoint adds $2 \times n$ states to be estimated, where $n$ is the state dimension of a node. The computational complexity would grow on the order of $O(n_x^2 \log n_x)$ where $n_x$ is the dimension of the information matrix ($n_x = 2 \times n \times N$) [57]. The computational load would quickly become infeasible as following distance increases. However, by making adjustments to the graph topology and taking advantage of graph reduction techniques, the computational complexity can be reduced to grow linearly with path length.

### 3.3.1 Measurement Edge Topology Selection

The computational complexity of the problem can be reduced by excluding certain measurement edges from the optimization. The inter-vehicle RPV measurements are a strong candidate for removal since their elements lie in the off-diagonals of the information matrix to be inverted. The inter-vehicle RPV measurements at the start and end of the path have the greatest influence on the path's accuracy since they connect critical nodes (waypoint positions used to calculate path deviation variables) to the anchor node (the latest follower pose). The inclusion of the intermediate RPV measurements has a marginal impact on the path accuracy.

This characteristic is demonstrated by a simulation study that compares the accuracies resulting from using a different number of inter-vehicle RPV edges in the estimation. The measurements were generated using the simulation environment described in Chapter 5. The number of RPV measurements ($N_{RPV}$) was increased from 1 to $N$ and the positional accuracy of the virtual lead waypoint was recorded. Note that when $N_{RPV} = 1$, the problem becomes equivalent to that of relative path generation as described in Section 2.4. Also,

when $N_{RPV} = N$, the full set of RPV edges is used and this result serves as the benchmark accuracy for the test. Figure 3.3 shows the accuracy of the virtual lead waypoint plotted against $N_{RPV}$. These results show a diminishing return on accuracy as $N_{RPV}$ increases past 5, at which point the difference is only 0.2 % from the benchmark.

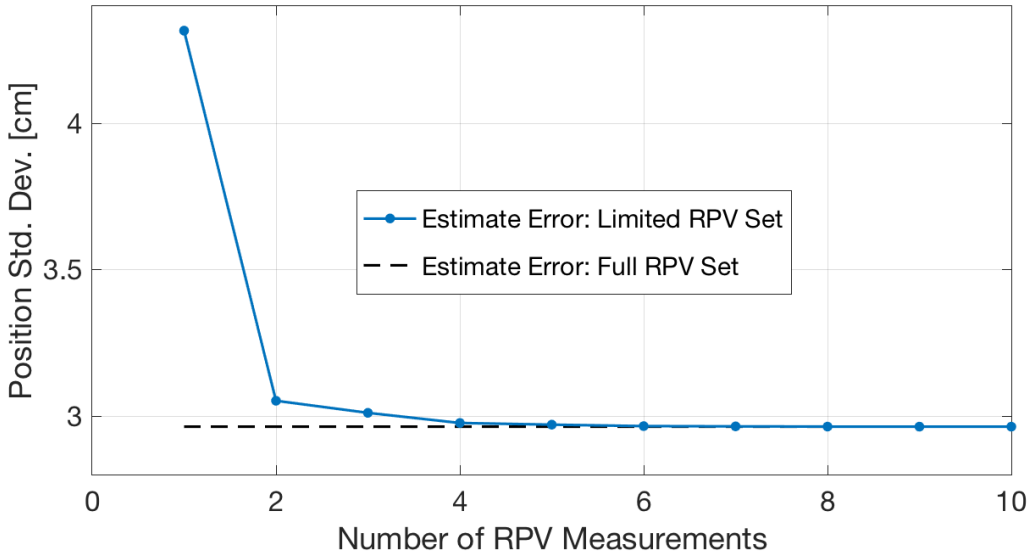


Figure 3.3: Impact of Number of RPV Measurements on Virtual Lead Position Error

By using only RPV measurements at the path boundaries, the intermediate odometry measurements can be combined and used as a single edge in the graph. The result is an estimation problem with a finite dimension and therefore a computational complexity that is not exponentially related to path length. The added step of summing intermediate odometry measurements can be accomplished with complexity that is linear with path length. From this, the computational complexity is reduced from $O(N^2 \log N)$ to $O(N)$ with negligible impact on the resulting path's accuracy, making it a feasible solution for long distance following.

The adjacency matrix is helpful for visualizing the process and impact of RPV edge removal and the combination of intermediate odometry edges. The adjacency matrix is a square matrix where the rows and columns correspond to nodes in the graph. The adjacency

matrix is populated with 0's and 1's where a 1 indicates two adjacent nodes (connected by a measurement edge). In Figure 3.4, the adjacency matrix is shown for each step in the removal and reduction process.

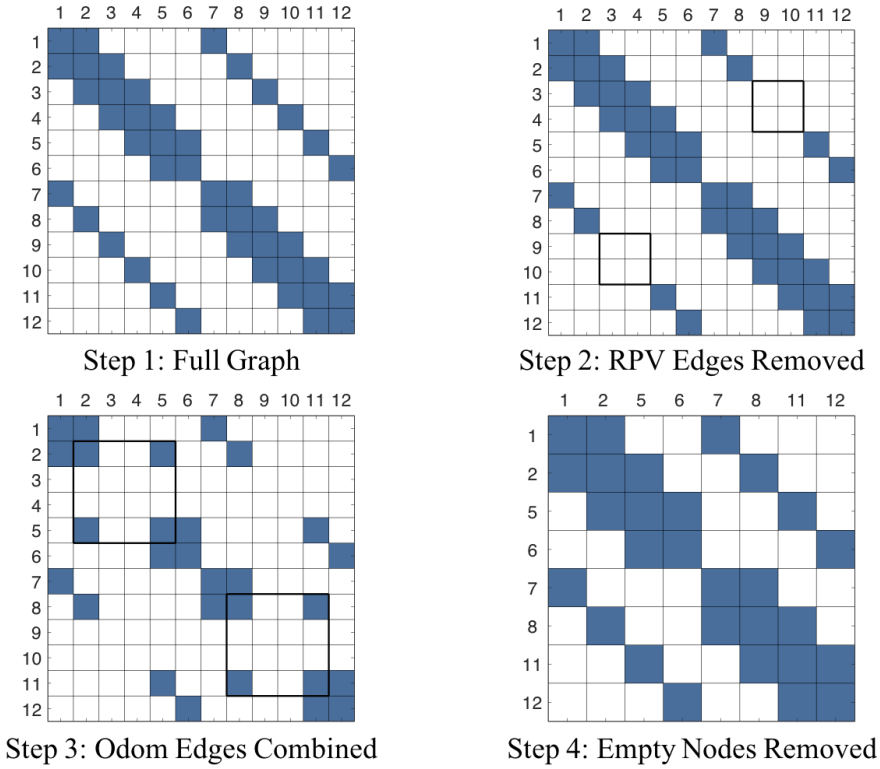


Figure 3.4: Adjacency Matrices in RPV Removal and Graph Reduction Process

In this simplified example, the path size ($N$) is 6 and the desired number of RPV edges ($N_{RPV}$) is 4. The adjacency matrix in Step 1 is of the full graph with all available RPV edges. The RPV edges can be seen as a band running along the off-diagonals of the adjacency matrix. The next step removes 2 of the RPV edges, leaving the desired number of RPV edges (4). In Step 3, the intermediate odometry edges are combined, leaving the cells of intermediate nodes (3, 4, 9 and 10) empty. Since the rows and columns of the intermediate nodes are all zero, they can be removed from the graph. The result is a graph of finite size, as shown in Step 4, that can be solved in a predictable amount of time.

Informally, the intermediate odometry measurements can be combined by cumulatively summing consecutive measurements. The covariance of the combined odometry edge would also be a cumulative sum of individual odometry covariance matrices. However, when landmark nodes are included in the graph, this step is not as straight-forward. Combining intermediate nodes is accomplished more generally through the process of graph reduction, allowing for the reduction of intermediate vehicle and landmark nodes alike.

### 3.3.2 Graph Reduction

It was shown in [47] that a relative position graph can be reduced significantly by taking advantage of the graph's topology. Graph reduction is the process of removing nodes one at a time while maintaining the information provided by the node's incident edges. Take the simple example shown in Figure 3.5 consisting of three nodes: $u$, $v$ and $w$. In this example, node $w$ is reduced from the graph and the information from $uw$ and $wv$ are collapsed into a single edge, $uv$.

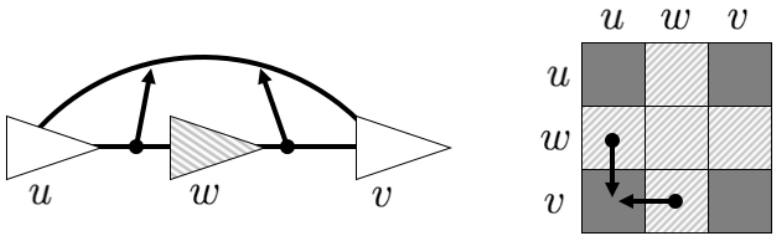


Figure 3.5: Node Reduction Example

Mathematically, the general procedure of node reduction is as follows. Let $\tau(w)$ be the set of nodes adjacent to and including node $w$. For the example shown in Figure 3.5, $\tau(w) = \{w, u, v\}$. To reduce node $w$ from the graph, the following operations are performed:

$$\boldsymbol{\Omega}_{[\tau(w),\tau(w)]} \mathrel{-}= \boldsymbol{\Omega}_{[\tau(w),w]} \boldsymbol{\Omega}^{-1}_{[w,w]} \boldsymbol{\Omega}_{[w,\tau(w)]} \tag{3.13}$$

$$\boldsymbol{\xi}_{[\tau(w)]} \mathrel{-}= \boldsymbol{\Omega}_{[\tau(w),w]} \boldsymbol{\Omega}^{-1}_{[w,w]} \boldsymbol{\xi}_{[w]} \tag{3.14}$$

The "−=" operator indicates the subtraction of the right-hand side of the equation from the left-hand side. After this operation, the rows and columns of $\boldsymbol{\xi}$ and $\boldsymbol{\Omega}$ associated with node $w$ will be zero. This is shown visually in the example in Figure 3.5 by the block elements that are shaded out. Node $w$ can then be removed from the optimization without loss of information [47].

The process of graph reduction is more relevant when additional measurements are included in the estimation. The use of landmark observations introduces new nodes representing landmark positions, which will be described in detail in later sections. By use of graph reduction, all intermediate landmark and vehicle nodes that bridge inter-vehicle RPV measurements can be combined into a single edge while maintaining their information and contribution to the final estimate. By this, the landmark observations can bridge gaps in odometry measurements. One lost odometry measurement would make summing intermediate odometry impossible, but by including landmark nodes and using graph reduction, these limitations are removed.

### 3.4 Relative Path Estimation Errors

The path variance is calculated from the graph estimate covariance matrix as follows. First, the covariance matrix is partitioned by the leader and follower subgraphs. The partitioned submatrices are the covariance of the estimated leader states ($\mathbf{P}_L$), the covariance of the estimated follower states ($\mathbf{P}_L$), and the cross-covariance of the leader and follower states ($\mathbf{P}_{FL}, \mathbf{P}_{LF}$).

$$\mathbf{P} = \boldsymbol{\Omega}^{-1} = \begin{bmatrix} \mathbf{P}_F & \mathbf{P}_{FL} \\ \mathbf{P}_{LF} & \mathbf{P}_L \end{bmatrix} \tag{3.15}$$

The covariance of the path estimate is therefore $\mathbf{P}_L$. From this, the covariance of the vehicle pose estimate at path index $i$ is calculated as

$$\mathbf{P}_i = \mathbf{P}_L \left[ (n(i-1)+1), \ldots, n(i) \right] \left[ (n(i-1)+1), \ldots, n(i) \right] \tag{3.16}$$

where $n$ is the state dimension of the each node ($n = 2$ for the 2D, linear case). The position estimate variance is taken as the sum of the diagonal elements of $\mathbf{P}_i$.

Figure 3.6 shows the variance from a graph-based path estimate compared to a path calculated from only follower odometry and inter-vehicle RPV measurements as described in Section 2.4. The path variance in this figure is plotted by path index. This example shows that by incorporating lead vehicle odometry into the path estimate, the error growth rate is reduced, allowing for larger paths and longer following distances. However, the error is still unbounded with respect to path index (proportional to following distance). To bound error growth, Section 3.6 will show how landmark observations can be included in the graph-based estimation framework.

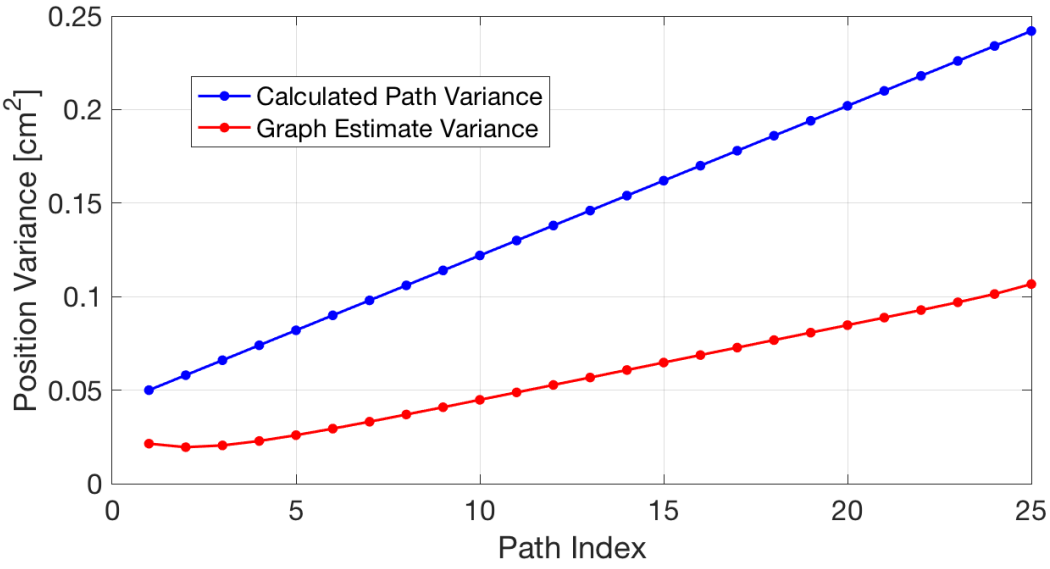


Figure 3.6: Graph-Based Estimation Example: Position Variance vs. Path Index

## 3.5 Nonlinear Graph-Based Estimation

The landmark observations are resolved in the vehicle's body frame as opposed to the odometry and RPV measurements considered so far which are resolved in the navigation

frame. Because of this, the vehicle orientation is required to transform landmark observations into a common frame. Assuming external knowledge of vehicle orientation is not available, the orientation is included in the state vector for estimation, resulting in a nonlinear estimation problem. In this case, each node of the graph represents a vehicle pose consisting of a 2D position ($\mathbf{r}$) and heading ($\psi$). For some graph node $u$, the associated vehicle pose is as shown below.

$$\mathbf{x}_u = \begin{bmatrix} \mathbf{r}_u \\ \psi_u \end{bmatrix} \tag{3.17}$$

A nonlinear measurement ($\mathbf{z}_{uv}$) relating two nodes, $u$ and $v$, is modeled as a function of the states ($\mathbf{x}_u$ and $\mathbf{x}_v$, respectively) and noise ($\boldsymbol{\nu}_{uv}$) as shown in the equation below.

$$\mathbf{z}_{uv} = \mathbf{h}_{uv}(\mathbf{x}_u, \mathbf{x}_v, \boldsymbol{\nu}_{uv}) \tag{3.18}$$

The noise is modeled as zero-mean Gaussian with covariance matrix ($\mathbf{R}_{uv}$).

Nonlinearities in graph-based estimation are handled by an iterative process [47]. First, the state estimate ($\hat{\mathbf{x}}$) is initialized with approximate values. The graph optimization process is then performed with the following modifications. For the nonlinear estimator, the measurement matrix ($\mathbf{H}_{uv}$) is replaced with the measurement Jacobian matrix evaluated at the latest state estimate as defined below.

$$\mathbf{H}_{uv} = \begin{bmatrix} \frac{\partial \mathbf{h}_{uv}}{\partial \mathbf{x}_v} & \frac{\partial \mathbf{h}_{uv}}{\partial \mathbf{x}_u} \end{bmatrix} \Bigg|_{(\hat{\mathbf{x}}_u, \hat{\mathbf{x}}_v, \mathbf{0})} \tag{3.19}$$

The information matrix and vector are updated using equations similar to those of the linear case, shown previously in Equations (3.8) and (3.9). In the case of the nonlinear estimator,

the measurement innovation is used as shown below.

$$\mathbf{\Omega}_{uv} \mathrel{+}= \mathbf{H}_{uv}^T \mathbf{R}_{uv}^{-1} \mathbf{H}_{uv} \tag{3.20}$$

$$\boldsymbol{\xi}_{uv} \mathrel{+}= \mathbf{H}_{uv}^T \mathbf{R}_{uv}^{-1} (\mathbf{z}_{uv} - \hat{\mathbf{z}}_{uv}) \tag{3.21}$$

In Equation (3.21), $\hat{\mathbf{z}}_{uv}$ is calculated from Equation (3.18) using the latest estimate of the state as seen in Equation (3.22).

$$\hat{\mathbf{z}}_{uv} = \mathbf{h}_{uv}(\hat{\mathbf{x}}_u, \hat{\mathbf{x}}_v, \mathbf{0}) \tag{3.22}$$

The information matrix and vector are then used to determine state estimate corrections ($\delta\hat{\mathbf{x}}$) by solving the equation below.

$$\mathbf{\Omega}\delta\hat{\mathbf{x}} = \boldsymbol{\xi} \tag{3.23}$$

The corrections are applied to the state estimate ($\hat{\mathbf{x}} \mathrel{+}= \delta\hat{\mathbf{x}}$) for each iteration. The process is repeated with the latest estimate until convergence is reached, indicated by a sufficiently small $\delta\hat{\mathbf{x}}$.

### 3.6 Augmenting the Graph-Based Estimator for Landmark Observations

Continuing with the estimation framework described in Section 3.2.2, landmark observations are incorporated by expanding the graph to include landmark nodes as indicated by stars in Figure 3.7. The landmark observations are represented in the graph as edges connecting vehicle nodes to landmark nodes. The landmark nodes are grouped in a new subgraph ($M$) as defined in Equation (3.24) where $N_M$ is the total number of landmarks.

$$\{m_1, m_2, \ldots, m_{N_M-1}, m_{N_M}\} \in V(M) \tag{3.24}$$

A graphical representation is also given in Figure 3.8 organized by subgraph.

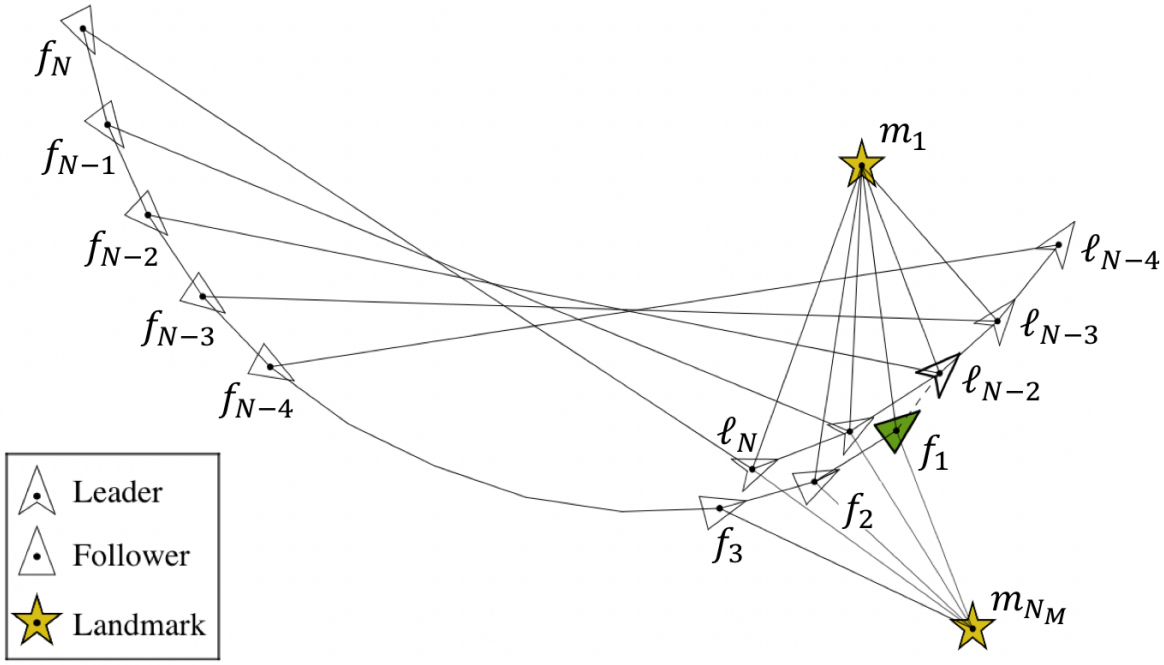


Figure 3.7: Graph-Based Path Estimation: Cartesian Graph With Landmark Nodes

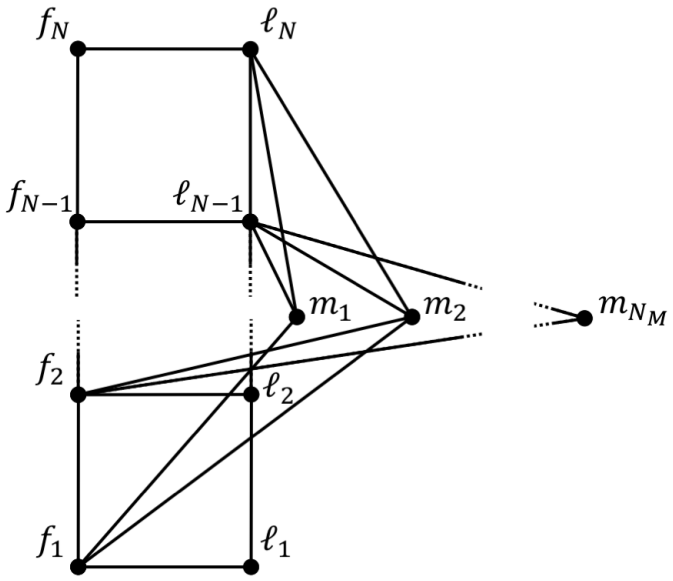


Figure 3.8: Graph-Based Path Estimation: Graphical Representation With Landmark Nodes

The landmark positions are unknown and must be estimated along with the vehicle states. From this, the estimation problem becomes a form of Simultaneous Localization

And Mapping (SLAM) as introduced in Section 1.4. Certain landmarks exhibit a distinct orientation that is detectable from the feature extraction process. While the landmark's orientation is not relevant to the ultimate goal of path following, it is included in the state vector to improve observability conditions as shown in Section 3.8. The full state vector, augmented with landmark states, is given in Equation (3.25).

$$\mathbf{x} = \left[\begin{array}{c} \mathbf{x}_F \\ \hline \mathbf{x}_L \\ \hline \mathbf{x}_M \end{array}\right] = \left[\begin{array}{c} \mathbf{x}_f(t_k) \\ \mathbf{x}_f(t_{k-1}) \\ \vdots \\ \mathbf{x}_f(t_{k-N+1}) \\ \hline \mathbf{x}_\ell(t_k) \\ \mathbf{x}_\ell(t_{k-1}) \\ \vdots \\ \mathbf{x}_\ell(t_{k-N+1}) \\ \hline \mathbf{x}_{m_1} \\ \mathbf{x}_{m_2} \\ \vdots \\ \mathbf{x}_{m_{N_M}} \end{array}\right] \tag{3.25}$$

## 3.7 Measurement Models

### 3.7.1 Differential GPS Measurement Models

There are two observation types provided by differential GPS: odometry and inter-vehicle RPV, both resolved in the navigation frame. The measurement models for these observations consist of differencing the position states of two nodes in the graph. The differential GPS measurement models are inherently linear as introduced in Section 3.2.2. Both types follow the same model as provided in Equation (3.26).

$$\begin{aligned}\mathbf{h}_{uv}(\mathbf{x}_u, \mathbf{x}_v, \boldsymbol{\nu}_{uv}) &= \begin{bmatrix} \mathbf{I}_{2\times 2} & \mathbf{0}_{2\times 1} & -\mathbf{I}_{2\times 2} & \mathbf{0}_{2\times 1} \end{bmatrix} \begin{bmatrix} \mathbf{x}_v \\ \mathbf{x}_u \end{bmatrix} + \boldsymbol{\nu}_{uv} \\ &= \mathbf{r}_v - \mathbf{r}_u + \boldsymbol{\nu}_{uv}\end{aligned} \tag{3.26}$$

The GPS odometry and inter-vehicle RPV measurements differ in which nodes $(u, v)$ they apply. For inter-vehicle RPV, $u$ belongs to the follower subgraph ($F$) and $v$ belongs to the leader subgraph ($L$), both nodes being associated with the same instance in time. For odometry measurement, $u$ and $v$ belong to the same subgraph (either $L$ or $F$) with $v$ being associated with a particular time ($t_i$) and $u$ associated with the previous time ($t_{i-1}$).

Because the differential GPS measurements are linear, the associated measurement matrix ($\mathbf{H}_{uv}$) is not a function of the states. The measurement matrix is simply the matrix that pre-multiplies the states in Equation (3.26). For completeness, the measurement matrix for differential GPS observations is given below in Equation (3.27).

$$\mathbf{H}_{uv} = \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \end{bmatrix} \tag{3.27}$$

The differential GPS observations are considered degenerate, meaning that $\mathbf{x}_u$ and $\mathbf{x}_v$ are not fully observable from these measurements alone. This is given by the columns of zeros in the measurement matrix which indicates that no information of the vehicle's heading is provided. However, by combining with other measurements, degenerate or otherwise, it's possible to achieve full observability of the states.

The noise acting on the differential GPS observations is additive, as shown in Equation (3.26). This noise is modeled as zero-mean Gaussian. The noise acting on inter-vehicle RPV measurements ($\boldsymbol{\nu}_{uv} = \boldsymbol{\nu}_{rpv}$) has a covariance matrix ($\mathbf{R}_{rpv}$). Similarly, the noise acting on GPS odometry measurements ($\boldsymbol{\nu}_{uv} = \boldsymbol{\nu}_{\Delta}$) has a covariance matrix ($\mathbf{R}_{\Delta}$). The covariance matrices for inter-vehicle RPV and GPS odometry measurements are defined in Equations

(3.28) and (3.29), respectively.

$$\mathbf{R}_{rpv} = E\left[\boldsymbol{\nu}_{rpv}\boldsymbol{\nu}_{rpv}^T\right] \tag{3.28}$$

$$\mathbf{R}_{\Delta} = E\left[\boldsymbol{\nu}_{\Delta}\boldsymbol{\nu}_{\Delta}^T\right] \tag{3.29}$$

### 3.7.2 Body-Centric Odometry Model

The body-centric odometry model is similar to the GPS odometry model in that it operates on two nodes within the same subgraph ($L$ or $F$) at consecutive times ($t_i$ and $t_{i-1}$). Body-centric odometry differs from GPS odometry in that the RPV is body-centric and provides an additional measurement of the change in heading between the two nodes. The body-centric odometry measurement relates to the states by differencing the position states and rotating into the body frame of the previous pose at $t_{i-1}$ along with an additional term for the change in heading. The model resulting from this calculation is given in Equation (3.30).

$$\mathbf{h}_{uv}(\mathbf{x}_u, \mathbf{x}_v, \boldsymbol{\nu}_{uv}) = \begin{bmatrix} \mathbf{C}_n^{b_u}\left(\mathbf{r}_v - \mathbf{r}_u\right) \\ \psi_v - \psi_u \end{bmatrix} + \boldsymbol{\nu}_{uv} \tag{3.30}$$

The body-centric odometry measurements are nonlinear and therefore their Jacobian matrix is a function of the states. For simplicity, the measurement Jacobian matrix is broken down into two parts as

$$\mathbf{H}_{uv} = \begin{bmatrix} \frac{\partial \mathbf{h}_{uv}}{\partial \mathbf{x}_v} & \frac{\partial \mathbf{h}_{uv}}{\partial \mathbf{x}_u} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_v & \mathbf{H}_u \end{bmatrix} \tag{3.31}$$

with $\mathbf{H}_u$ and $\mathbf{H}_v$ defined below.

$$\mathbf{H}_v = \begin{bmatrix} \mathbf{C}_n^{b_u} & \mathbf{0}_{2\times 1} \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{3.32}$$

$$\mathbf{H}_u = \begin{bmatrix} -\mathbf{C}_n^{b_u} & \delta\mathbf{C}_n^{b_u}(\mathbf{r}_v - \mathbf{r}_u) \\ \mathbf{0}_{1\times 2} & -1 \end{bmatrix} \tag{3.33}$$

In Equations (3.30), (3.32), and (3.33), the matrix $\mathbf{C}_n^{b_u}$ represents the rotation matrix from the navigation frame to the body frame of $u$.

$$\mathbf{C}_n^{b_u} = \begin{bmatrix} \cos(\psi_u) & \sin(\psi_u) \\ -\sin(\psi_u) & \cos(\psi_u) \end{bmatrix} \tag{3.34}$$

Also, $\delta\mathbf{C}_n^{b_u}$ is the partial derivative of the rotation matrix with respect to $\psi_u$ as shown below.

$$\delta\mathbf{C}_n^{b_u} = \frac{\partial\mathbf{C}_n^{b_u}}{\partial\psi_u} = \begin{bmatrix} -\sin(\psi_u) & \cos(\psi_u) \\ -\cos(\psi_u) & -\sin(\psi_u) \end{bmatrix} \tag{3.35}$$

The body-centric odometry measurements are non-degenerate since the full pose $\mathbf{x}_v$ can be determined from $\mathbf{x}_u$ (and vice versa) with a body-centric odometry measurement. In practice, body-centric odometry measurements have a high degree of availability and their inclusion in the Relative Pose Graph significantly increases observability of the full estimation problem.

Equation (3.31) shows that the noise acting on body-centric odometry ($\boldsymbol{\nu}_{uv} = \boldsymbol{\nu}_{odom}$) is additive. This noise is modeled as zero-mean Gaussian with a covariance matrix ($\mathbf{R}_{odom}$) as defined in Equation (3.36).

$$\mathbf{R}_{odom} = E\left[\boldsymbol{\nu}_{odom}\boldsymbol{\nu}_{odom}^T\right] \tag{3.36}$$

### 3.7.3 Landmark Observation Model

The landmark observations are modeled similarly to that of the body-centric odometry measurements in that a body-centric RPV and a relative orientation is measured. For landmarks with a distinct orientation (e.g., from that of a road sign), the same model provided

in Equation (3.30) applies. In this model, the node $u$ represents the ego-vehicle from which the observation was made while the node $v$ represents the observed landmark.

For landmarks that lack a distinct orientation (e.g., from a cylindrical, pole-like landmark), the measurement is made up solely of the RPV resolved in the ego-vehicle's body frame. This model is realized by removing the last row of the measurement model ($\mathbf{h}_{uv}$) and the measurement Jacobian matrix ($\mathbf{H}_{uv}$) of Equations (3.30) and (3.31). The result is a measurement Jacobian matrix with all zeros in the third column corresponding to the landmark's heading state. Because of this, landmark observations that lack a distinct orientation are degenerate. The information is still valuable in path estimation, however, it will require multiple commonly observed landmarks to provide full observability between two vehicle poses. The observability conditions are studied in greater detail in Section 3.8.

As shown in Equation (3.31), the noise acting on landmark measurements ($\boldsymbol{\nu}_{uv} = \boldsymbol{\nu}_{LM}$) is additive. This noise is again modeled as zero-mean Gaussian with a covariance matrix ($\mathbf{R}_{LM}$) as defined in Equation (3.37).

$$\mathbf{R}_{LM} = E\left[\boldsymbol{\nu}_{LM}\boldsymbol{\nu}_{LM}^T\right] \tag{3.37}$$

## 3.8 Observability Considerations for the Nonlinear Case

In Section 3.2.3, the conditions for a valid path to be observable were described for the linear case. This section describes the observability conditions for the nonlinear case, which are slightly more involved. The nonlinear observability is first described through a number of graph topology examples. Next, an algorithm is presented for evaluating the nonlinear observability and ensuring that the estimation problem is non-degenerate.

### 3.8.1 Case Study for Nonlinear Observability

First, consider a Relative Position Graph with body-centric odometry and landmark observations that are non-degenerate (i.e., include both relative position and relative orientation). In this case, the observable graph nodes can be determined using the same method presented in Section 3.2.3: a node ($v_i$) is said to be observable if the graph contains a path from the anchor node ($v_a$) to node $v_i$. When all nodes in the graph satisfy this condition, the information matrix ($\mathbf{\Omega}$) will be invertible.

The complications arise when considering degenerate edges in the graph such as those provided by pole-like landmark observations that lack a distinct orientation. A degenerate edge is one that, by itself, does not contain enough information to resolve the full pose state of one of the incident nodes from the other. These complications are demonstrated using the example topologies given below.

**Example 1: Two vehicle nodes, a body-centric odometry edge, and a GPS odometry edge (observable)**

In this first example, the graph contains two nodes ($f_1$ and $f_2$) representing a single vehicle at two instances in time. The nodes are connected by a body-centric odometry edge and a GPS odometry edge. Only the position states of $f_1$ are anchored in this example and the yaw state of $f_1$ is left to be estimated in the graph optimization. The body-centric odometry edge contains information on the relative pose between vehicles with respect to $f_2$ and is resolved in the $f_2$ body frame. In contrast, the GPS odometry edge is resolved in the navigation frame. With this topology, there is only one possible solution that satisfies the constraints provided by the two measurement edges. This is expressed mathematically as a full rank information matrix. The rank of the information matrix in this example is 6, which is equal to the dimension of the information matrix (3 states for the 2D pose of both $f_1$ and $f_2$) and satisfies the full rank requirement for observability.

**Example 2: Two vehicle nodes, one degenerate landmark node (unobservable)**

Next, consider a topology with two vehicle nodes (an anchor node at $f_1$ and a path node $\ell_1$) and one landmark node ($m_1$) lacking a distinct orientation ("degenerate landmark"). Both vehicle nodes are connected to $m_1$ through a degenerate measurement edge. In this example, since no navigation frame observations are used, the yaw state of $f_1$ is anchored and the resulting estimate will be resolved in the body frame of $f_1$. Given this topology, an infinite number of solutions exist for the pose of $\ell_1$, forming a circle centered around the landmark position. This is expressed mathematically as rank deficiency in the information matrix. The rank of the information matrix formed with this topology is 7, which is less than the value of 8 required for full rank (3 states for both $f_1$ and $\ell_1$ plus 2 states for the position of $m_1$).

**Example 3: Two vehicle nodes, one degenerate landmark node, and an inter-vehicle RPV edge (observable)**

Next, consider the topology in the first example, but with an inter-vehicle RPV edge (resolved in the navigation frame) between $f_1$ and $\ell_1$. Since a navigation frame measurement is included in the graph, only the position states of $f_1$ are anchored and the yaw state of $f_1$ is left to be estimated. The information gained by adding this edge makes $\ell_1$ fully observable (position and orientation) despite the landmark observations lacking an orientation component. This topology also allows for full yaw observability including that of the position-anchored node, $f_1$. The rank of the information matrix from this set of measurements is 8, which is full rank.

**Example 4: Two vehicle nodes, one degenerate landmark node, and a body-centric odometry edge (observable)**

In this example, two vehicle nodes observe a degenerate landmark and are connected by a body-centric odometry edge. In this case, the yaw state of $f_1$ is anchored since no

navigation frame information is present in the graph. All nodes are fully observable relative to a reference frame defined by the pose of $f_1$. This is related to the information matrix having a full rank of 8.

**Example 5: Two vehicle nodes, two degenerate landmark nodes (observable)**

This example considers two vehicle nodes ($f_1$ and $\ell_1$) that both observe two degenerate landmark nodes ($m_1$ and $m_2$). Assume in this example that $f_1$ is anchored in both position and yaw since none of the measurement information is resolved in the navigation frame. As opposed to Example 2 in which there are infinite solutions for $\ell_1$, the addition of a second degenerate landmark node in this example results in only one possible solution for the state of $\ell_1$. The rank of the information matrix in this case is 10 which satisfies full rank (3 states for both $f_1$ and $\ell_1$, 2 states for both $m_1$ and $m_2$).

### 3.8.2 Observability Evaluation

Motivated by the findings in the previous examples, a method was developed that ensures sufficient conditions for observability of the nodes in the graph. For this, a few graph-theoretic properties are introduced. For some graph $G$, a *path* is a subgraph of $G$ that connects two nodes within the graph. A *connected graph* is a graph for which a path can be drawn between any two nodes in the graph. A graph is said to be *$k$-edge-connected* if it remains connected when less than $k$ edges are removed [55].

For the Relative Position Graph ($G$) described in this chapter, observability can be ensured if a graph $G_v = L + F$ is at least 2-edge-connected. That is to say, a subgraph of $G$ made up of only vehicle nodes ($G_v \subseteq G$) is observable if it is 2-edge-connected. This condition is not necessary for observability (as shown in Example 3), but is sufficient for ensuring observability and can be used in practice to declare nodes as valid for use in graph optimization. This condition will ensure that any node in the graph will have at least two independent connections to the anchor node in which case the node will be fully observable.

To increase the efficiency of an algorithm that evaluates observability, an assumption is made that body-centric odometry is continuously available. This is a reasonable assumption since the onboard wheel speed and yaw rate measurements are critical for vehicle safety measures and are designed with high standards on availability. With full availability of body-centric odometry, the two vehicle subgraphs have full internal connectivity by non-degenerate edges. All poses within a vehicle subgraph can be calculated from any other node in the same subgraph. Since the follower subgraph contains an anchoring node, all other nodes in $F$ can be labeled as valid. For validating the leader subgraph, a 2-edge-connectivity must exist between $F$ and $L$.

The observability check is performed as follows. Given a graph $G$ representing the Relative Pose Graph, $F \subseteq G$ and $L \subseteq G$ representing the vehicle subgraphs with $f_1 \in V(F)$ as an anchor node, the $k$-edge-connectivity between $f_1$ and $L$ is to be determined. On initialization, set the connectivity $k = 0$ and an intermediate graph $G = G'$ for use in the operation. For each $\ell_i \in V(L)$, find a path $P_i \subseteq G'$ that connects $\ell_i$ to $f_1$ ($V(P_i) = \{f_1, \ldots, \ell_i\}$). If the path exists, then increment $k = k + 1$ and remove all edges of $P_i$ from $G'$ ($G' = G' - E(P_i)$). Set $i = i + 1$ and continue to the next iteration. After iterating through all node combinations, the resulting value of $k$ is the edge-connectivity of the graph. To perform the observability check, it isn't necessary to know the true $k$-edge-connectivity, but whether $G$ is least 2-edge-connected. Therefore, the algorithm can exit the operation once the minimum connectivity number (2) is reached.

The assumption of continuous body-centric odometry is not entirely necessary and may be restrictive in some applications. The validity test can be adapted to account for any gaps in body-centric odometry. In this case, a number of subgraphs can be partitioned, each with full internal connectivity. Then, the 2-edge-connectivity with the anchoring node can be evaluated for each subgraph separately.

## 3.9 Conclusion

This chapter introduced a novel method for relative path estimation using a graph-based estimation scheme. The key contribution of the proposed path estimation approach is to bound error with respect to following distance while leveraging the accuracy provided by differential GPS measurements. This attribute of bounded errors is realized by exchanging landmark observations between vehicles. A secondary contribution of the proposed method is the improved availability provided by landmark observations and body-centric odometry compared to solutions relying solely on GPS. These attributes will be demonstrated through an evaluation carried out with both simulated and experimental data in Chapters 5 and 6, respectively. Before this evaluation can take place, details on the specific embodiment must be provided. For this, Chapter 4 will detail the sensors and methods used to generate measurements from the experimental data and provide a characterization of the resulting measurement errors.

# Chapter 4

# Relative Path Estimation Front-End

The previous chapter detailed the proposed graph-based estimation approach using generic measurements in the form of relative position vectors and relative orientation. In the robotics community, the process of generating these measurements from raw sensor data is referred to as the SLAM Front-End. By describing the SLAM Front-End separately, the core estimator can be replicated with different measurement sources by referring exclusively to Chapter 3. For this chapter, consideration is given to the sensors used for this particular application. The sensors discussed in this chapter are based on the experimental setup which is described further in Chapter 6.

The sensor data used in this dissertation can be split into three categories: raw GPS observables, onboard vehicle measurements, and lidar data. The raw GPS observables of carrier phase and pseudorange are used for generating measurements of inter-vehicle RPV and vehicle odometry, both resolved in the navigation frame. The onboard vehicle measurements are provided over the Controller Area Network (CAN) bus and are commonly available on most production vehicles. This includes measurements of wheel speed and yaw rate which are used to generate body-centric vehicle odometry. Lastly, this chapter details how data from a multi-channel lidar is processed for generating landmark observations, a process referred to as feature extraction.

## 4.1 Front-End Process Overview

A block diagram is shown in Figure 4.1 summarizing the Relative Path Estimation Front-End. This diagram shows a number of vehicle blocks up to Vehicle $i$, each of which performs the front-end process locally. The diagram implies an Ultimate Lead Following scheme

given that Vehicle 1 (the convoy's ultimate lead) transmits to all subsequent vehicles. An Immediate Lead Following scheme would look the same except each Vehicle $i$ would transmit to its immediate follower at $i + 1$.

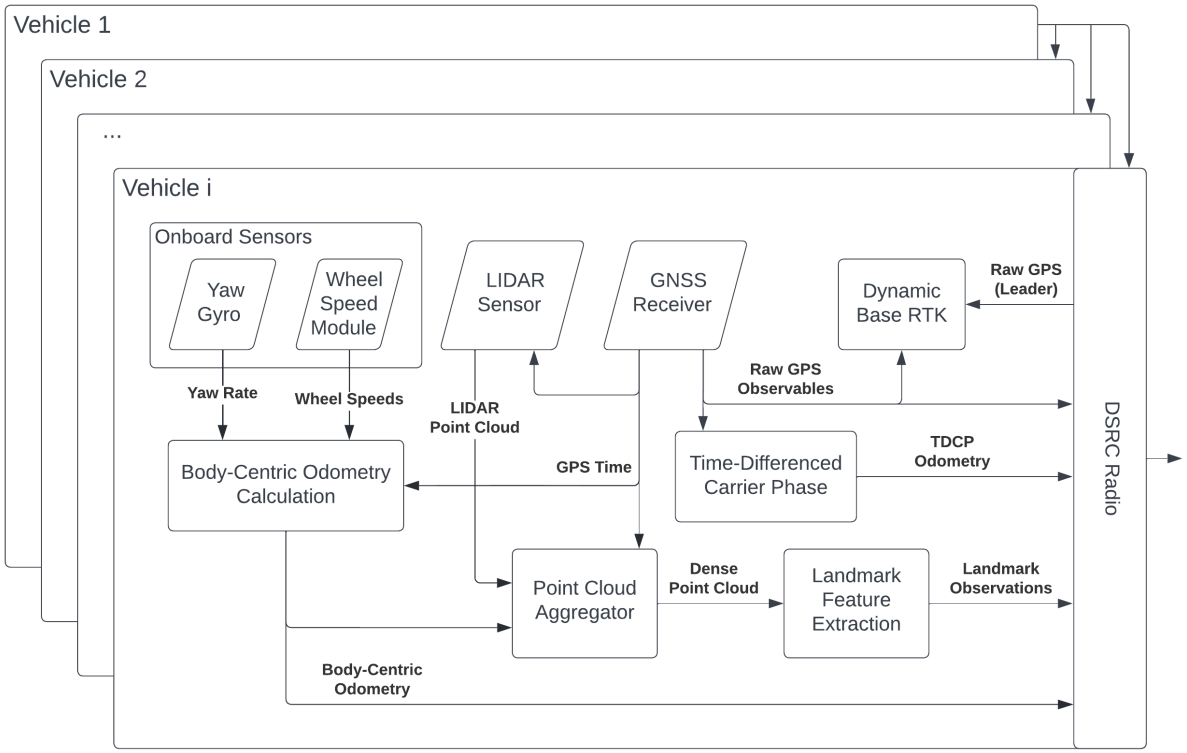


Figure 4.1: Relative Path Estimation Front-End Block Diagram

The block diagram shows the four sensor modules as parallelogram-shaped containers. The blocks that follow are front-end algorithms. The yaw rates and wheel speeds feed into Body-Centric Odometry Calculation, which outputs odometry synchronized with GPS time. The Point Cloud Aggregator block takes in point clouds at the rate of the lidar and combines consecutive data for outputting a dense point cloud at each GPS time epoch. The Point Cloud Aggregator accepts body-centric odometry in order to compensate for vehicle motion when combining sequential lidar measurements. The dense point cloud data is then used in the Landmark Feature Extraction block for detecting the landmark observations. Raw GPS observables are output from the GPS receiver and used for differential GPS algorithms.

The raw GPS observables, along with the processed observations, are transmitted via DSRC radio for use in a following vehicle's path estimation algorithm.

## 4.2 Differential GPS Algorithms

The differential GPS methods detailed in this section are used as measurements in the path estimator. Differential techniques provide precise relative position information by mitigating the impact of common mode errors in the GPS measurements. Two algorithms will be described in this section: Time Differenced Carrier Phase Odometry (TDCP) and Dynamic Base RTK (DRTK). These two algorithms result in precise estimates of vehicle odometry and inter-vehicle RPV which are then used as measurements in path estimation.

### 4.2.1 GPS Observations

The GPS receiver provides code and carrier based range measurements for each satellite being tracked. The measurements provided by receiver $A$ of satellite $j$ are modeled as shown in the equations below [58].

$$\rho_A^j = r_A^j + c(\delta t_A - \delta t^j) + \lambda(T_A^j + I_A^j) + M_{\rho,A}^j + \epsilon_{\rho,A}^j \tag{4.1}$$

$$\phi_A^j = r_A^j + c(\delta t_A - \delta t^j) + \lambda(T_A^j - I_A^j + N_A^j) + M_{\phi,A}^j + \epsilon_{\phi,A}^j \tag{4.2}$$

In these equations, the true range from receiver $A$ to satellite $j$ is represented as $r_A^j$, which is the magnitude of the relative position vector ($r_A^j = \|\boldsymbol{r}_A^j\|$). Both the pseudorange and carrier phase measurements are corrupted by atmospheric delays due to the ionosphere ($I_A^j$) and troposphere ($T_A^j$). The atmospheric errors are multiplied by the signal wavelength constant ($\lambda$). The receiver and satellite clock deviations from GPS time ($\delta t_A$ and $\delta t^j$, respectively) also produce biases in the measured range from receiver to satellite. The clock deviations are multiplied by the speed of light constant ($c$). The measurements also involve random noise due to thermal effects on the receiver; the random error in the pseudorange

and carrier phase measurements are represented by $\epsilon^j_{\rho,A}$ and $\epsilon^j_{\phi,A}$, respectively. The terms $M^j_{\rho,A}$ and $M^j_{\phi,A}$ represent errors due to multipath of the signal. Errors due to multipath are abrupt and therefore difficult to model, however, they can be detected through the receiver's internal operations. For the presented work, measurements with multipath are treated as outliers and rejected. Compared to the pseudorange model, the carrier phase measurement has an additional bias term $N^j_A$ which represents the carrier phase ambiguity. The carrier phase ambiguity is an integer number of cycles that is unknown to the receiver.

### 4.2.2 Dynamic Base RTK

The Dynamic Base RTK (DRTK) algorithm is used to provide high precision estimates of the relative position vector between two receivers. The DRTK algorithm is a 3 step process consisting of 1) estimating approximate values of carrier phase ambiguity, 2) fixing the ambiguity estimates as integer values and 3) estimating the RPV using the fixed integer range measurements.

#### Single Differenced Observations

It is useful to define the single differenced pseudorange and carrier phase observables as a precursor to the DRTK algorithm. The single differenced measurements are computed by subtracting the pseudorange and carrier phase from two GPS receivers, $A$ and $B$. The benefit of subtracting the measurements is that the atmosphere induced delays and the satellite clock errors are highly correlated in the measurements from two vehicles in relatively close proximity. Models of the single differenced measurements are given in Equations (4.3) and (4.4).

$$\rho^j_{AB} = \rho^j_B - \rho^j_A = (r^j_B - r^j_A) + c\delta t_{AB} + \epsilon^j_{\Delta\rho} \tag{4.3}$$

$$\phi^j_{AB} = \phi^j_B - \phi^j_A = (r^j_B - r^j_A) + c\delta t_{AB} + \lambda N^j_{AB} + \epsilon^j_{\Delta\phi} \tag{4.4}$$

The atmospheric and satellite clock errors are canceled out by differencing the measurements. The remaining terms are combined as $\delta t_{AB} = \delta t_B - \delta t_A$ and $N^j_{AB} = N^j_B - N^j_A$.

With knowledge of the system geometry, the scalar ranges can be decomposed into their 3D components using

$$r^j_i = \mathbf{G}^j_i \boldsymbol{r}^j_i \tag{4.5}$$

where $\mathbf{G}^j_i$ is the geometry matrix which contains the unit vectors from receiver $i$ to satellite $j$ as shown in Equation (4.6) [25].

$$\mathbf{G}^j_i = \begin{bmatrix} \frac{r^j_{i,x}}{\|\boldsymbol{r}^j_i\|} & \frac{r^j_{i,y}}{\|\boldsymbol{r}^j_i\|} & \frac{r^j_{i,z}}{\|\boldsymbol{r}^j_i\|} \end{bmatrix} \tag{4.6}$$

From Equation (4.6) and defining $\boldsymbol{r}_{AB} = \boldsymbol{r}_B - \boldsymbol{r}_A$, the single differenced observations can be written in matrix form as shown in Equation (4.7).

$$\begin{bmatrix} \rho^j_{AB} \\ \phi^j_{AB} \end{bmatrix} = \begin{bmatrix} \mathbf{G}^j & 1 \\ \mathbf{G}^j & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{r}_{AB} \\ c\delta t_{AB} \end{bmatrix} + \begin{bmatrix} 0 \\ \lambda \end{bmatrix} N^j_{AB} + \begin{bmatrix} \epsilon^j_{\Delta\rho} \\ \epsilon^j_{\Delta\phi} \end{bmatrix} \tag{4.7}$$

Equation (4.7) assumes that the line-of-sight unit vector from receiver to satellite is approximately equal for receivers $A$ and $B$. This is reasonable given the proximity of the receivers relative to the distance to the satellites. For the remainder of the derivation, the geometry matrix is assumed to have known values.

### Ambiguity Estimation

The carrier phase ambiguities are estimated using a Kalman filter with measurements as described in the previous section. The state vector ($\mathbf{x}$) is defined as

$$\mathbf{x} = \begin{bmatrix} N^{1,L_1}_{AB} & \dots & N^{m,L_1}_{AB} & N^{1,L_2}_{AB} & \dots & N^{m,L_2}_{AB} \end{bmatrix}^T \tag{4.8}$$

and contains two ambiguities (one on GPS L1 and one on GPS L2) for each satellite that is being tracked by the receivers on both vehicles. For $m$ satellites, the state vector has dimension of $2m \times 1$.

To isolate the ambiguities from the line of sight range, Equation (4.7) is pre-multiplied by the null space of the geometry matrix ($\mathbf{L}^j$) as

$$\mathbf{L}^j \begin{bmatrix} \rho^j_{AB} \\ \phi^j_{AB} \end{bmatrix} = \mathbf{L}^j \begin{bmatrix} 0 \\ \lambda \end{bmatrix} N^j_{AB} + \begin{bmatrix} \epsilon^j_{\Delta\rho} \\ \epsilon^j_{\Delta\phi} \end{bmatrix} \tag{4.9}$$

where the null space of the geometry matrix is the matrix that satisfies Equation (4.10).

$$\mathbf{L}^j \begin{bmatrix} \mathbf{G}^j & 1 \\ \mathbf{G}^j & 1 \end{bmatrix} = \mathbf{0}. \tag{4.10}$$

With the use of the null space operation, the range components in Equation (4.7) are removed in Equation (4.9). The measurement model now fits the form $\mathbf{z} = \mathbf{H}\mathbf{x} + \boldsymbol{\epsilon}$ according to the equations below.

$$\mathbf{z} = \mathbf{L} \begin{bmatrix} \boldsymbol{\rho}_{AB} \\ \boldsymbol{\phi}_{AB} \end{bmatrix} \tag{4.11}$$

$$\mathbf{H} = \mathbf{L} \begin{bmatrix} \mathbf{0} \\ \lambda\mathbf{I} \end{bmatrix} \tag{4.12}$$

The measurement update of the Kalman filter is then performed as shown in Equations (4.13–4.15).

$$\mathbf{K} = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H}\mathbf{P}_k^- \mathbf{H}^T + \mathbf{R})^{-1} \tag{4.13}$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}(\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}_k^-) \tag{4.14}$$

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}_k^- \tag{4.15}$$

In practice, the measurement noise matrix ($\mathbf{R}$) is calculated based on the carrier-to-noise density ratio of the received signal as shown in [25].

Since the carrier ambiguities are constant values, the Kalman filter time update is trivial. The state transition matrix is an identity matrix. A small non-zero process noise matrix is used to prevent the filter from ignoring new measurements after initial convergence.

### Ambiguity Fixing

The state estimates resulting from the Kalman filter are float-valued carrier phase ambiguities. Knowing that the true ambiguities are integers, these estimates can be refined in a process called integer ambiguity fixing. The carrier phase ambiguity integer fixing is performed using the LAMBDA method developed in [59].

Recall that the carrier phase ambiguities estimated in the Kalman filter are the single differenced ambiguities (i.e., the difference between the carrier phase ambiguity of receiver $A$ and $B$). Also note that the single differenced measurement models show that the measurements are dependent on the receiver clock error. The Kalman filter estimates are therefore dependent on receiver clock stability. To remove the dependency on the receiver clock error, the double differenced ambiguity estimate is computed prior to using the LAMBDA method to determine the fixed integer values. The vector of double differenced ambiguities is calculated by multiplying the single differenced estimates by a linear transform where one estimate

is selected as the base measurement and is subtracted from all other measurements. An example of the transform (with the second measurement selected as the base measurement) is shown in Equation (4.16).

$$\mathbf{N}_{AB}^{\nabla\Delta} = \mathbf{C}_{\nabla\Delta}\mathbf{N}_{AB} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \mathbf{N}_{AB} \tag{4.16}$$

In this equation, $\mathbf{C}_{\nabla\Delta}$ is the matrix which transforms the ambiguities to the double differenced space.

The LAMBDA method determines the statistically optimal set of integer values based on the float-valued estimates ($\hat{\mathbf{x}}$) and associated covariance matrix ($\mathbf{P}$) as provided by the Kalman filter. First, the state covariance matrix is transformed to the double differenced space using $\mathbf{C}_{\nabla\Delta}$ as shown in Equation (4.17).

$$\mathbf{P}_{\nabla\Delta} = \mathbf{C}_{\nabla\Delta}\mathbf{P}\mathbf{C}_{\nabla\Delta}^{T} \tag{4.17}$$

Multiple candidate sets of integer values are produced by the LAMBDA method. The ratio test described in [59] is used for deciding to select or reject the optimal integer set.

**Relative Position Vector Estimation**

The double differenced carrier phase measurements ($\boldsymbol{\phi}_{AB}^{\nabla\Delta} = \mathbf{C}_{\nabla\Delta}\boldsymbol{\phi}_{AB}$) are used to estimate the RPV between receivers ($\mathbf{r}_{AB}$) after accounting for phase ambiguity. A least squares procedure is used to produce the RPV estimate as shown in Equation (4.18).

$$\hat{\mathbf{r}}_{AB} = \left[\mathbf{G}_{\nabla\Delta}^{T}\mathbf{R}_{\nabla\Delta\phi}^{-1}\mathbf{G}_{\nabla\Delta}\right]^{-1}\mathbf{G}_{\nabla\Delta}^{T}\mathbf{R}_{\nabla\Delta\phi}^{-1}(\boldsymbol{\phi}_{AB}^{\nabla\Delta} - \boldsymbol{\lambda}\mathbf{N}^{\nabla\Delta}) \tag{4.18}$$

Assuming that the integer ambiguities were resolved to their true values, the only noise acting on the RPV estimate is the random noise of the carrier phase measurements. The covariance of the RPV estimate resulting from the least squares estimate procedure is calculated as shown in Equation (4.19)

$$\mathbf{P}_{\boldsymbol{r}_{AB}} = (\mathbf{G}_{\nabla\Delta}^{T}\mathbf{R}_{\nabla\Delta\phi}^{-1}\mathbf{G}_{\nabla\Delta})^{-1} \tag{4.19}$$

The RPV estimate and associated covariance are output from the DRTK algorithm for use in the higher-level sensor fusion.

### 4.2.3 Time Differenced Carrier Phase Odometry

The Time Differenced Carrier Phase (TDCP) method is used for providing high precision estimates of vehicle odometry as the change in position of a single GPS receiver in time. This procedure takes advantage of the spatial correlation of atmospheric errors in the carrier phase measurement between consecutive measurement epochs.

Using the model provided in Equation (4.2), the change in the carrier phase measurement for a single receiver from time $t_{k-1}$ to time $t_k$ is given in Equation (4.20).

$$\begin{aligned}\boldsymbol{\phi}_{A,k}^{j} - \boldsymbol{\phi}_{A,k-1}^{j} = r_{A,k}^{j} + c(\delta t_{A,k} - \delta t_{k}^{j}) + \lambda(T_{A,k}^{j} + I_{A,k}^{j} + N_{A,k}^{j}) + \epsilon_{\phi,A,k}^{j} \quad \dots \\ - r_{A,k-1}^{j} - c(\delta t_{A,k-1} - \delta t_{k-1}^{j}) - \lambda(T_{A,k-1}^{j} + I_{A,k-1}^{j} + N_{A,k-1}^{j}) + \epsilon_{\phi,A,k-1}^{j}\end{aligned} \tag{4.20}$$

For short time periods, the satellite clock error along with the ionospheric and tropospheric delays are nearly constant. Also, assuming the receiver maintains lock on the phase of the received signal, the carrier ambiguity is constant. Therefore, those errors are removed by subtraction and the time differenced carrier phase measurement model is reduced as shown

in Equation (4.21).

$$\begin{aligned}\boldsymbol{\phi}^j_{A,k} - \boldsymbol{\phi}^j_{A,k-1} = \Delta\boldsymbol{\phi}^j_{A,k|k-1} &= (r^j_{A,k} - r^j_{A,k-1}) + c(\delta t_{A,k} - \delta t_{A,k-1}) + \epsilon^j_{\phi,A,k|k-1} \\ &= (r^j_{A,k} - r^j_{A,k-1}) + c\delta t_{A,k|k-1} + \epsilon^j_{\phi,A,k|k-1}\end{aligned} \tag{4.21}$$

Using Equation (4.5), the line of sight ranges ($r^j_{A,k}$ and $r^j_{A,k-1}$) in Equation (4.21) are decomposed into the line of sight unit vectors and the RPV from receiver $A$ to satellite $j$ ($\boldsymbol{r}^j_A$) as shown in Equation (4.22).

$$\Delta\boldsymbol{\phi}^j_{A,k|k-1} = \mathbf{G}^j_k\boldsymbol{r}^j_{A,k} - \mathbf{G}^j_{k-1}\boldsymbol{r}^j_{A,k-1} + c\delta t_{A,k|k-1} + \epsilon^j_{\phi,A,k|k-1} \tag{4.22}$$

Assuming the unit vectors do not change significantly over the time interval ($\mathbf{G}^j_k \approx \mathbf{G}^j_{k-1}$), Equation (4.22) can be written as shown in Equation (4.23).

$$\begin{aligned}\Delta\boldsymbol{\phi}^j_{A,k|k-1} &= \mathbf{G}^j_k(\boldsymbol{r}^j_{A,k} - \boldsymbol{r}^j_{A,k-1}) + c\delta t_{A,k|k-1} + \epsilon^j_{\phi,A,k|k-1} \\ &= \mathbf{G}^j_k(\boldsymbol{r}_{j,k} - \boldsymbol{r}_{j,k-1}) - \mathbf{G}^j_k(\boldsymbol{r}_{A,k} - \boldsymbol{r}_{A,k-1}) + c\delta t_{A,k|k-1} + \epsilon^j_{\phi,A,k|k-1} \\ &= \mathbf{G}^j_k\boldsymbol{r}_{j,k|k-1} - \mathbf{G}^j_k\boldsymbol{r}_{A,k|k-1} + c\delta t_{A,k|k-1} + \epsilon^j_{\phi,A,k|k-1}\end{aligned} \tag{4.23}$$

In this equation, $\boldsymbol{r}_j$ is the global position of satellite $j$ which is known from the decoded ephemeris data. Equation (4.23) can then be arranged as

$$\begin{aligned}\Delta\boldsymbol{\phi}^j_{A,k|k-1} - \mathbf{G}^j_k\boldsymbol{r}_{j,k|k-1} &= -\mathbf{G}^j_k\boldsymbol{r}_{A,k|k-1} + c\delta t_{A,k|k-1} + \epsilon^j_{\phi,A,k|k-1} \\ &= \begin{bmatrix} -\mathbf{G}^j_k & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{r}_{A,k|k-1} \\ c\delta t_{A,k|k-1} \end{bmatrix} + \epsilon^j_{\phi,A,k|k-1}\end{aligned} \tag{4.24}$$

which fits the form $\mathbf{z} = \mathbf{H}\mathbf{x} + \boldsymbol{\epsilon}$ with

$$\mathbf{z} = \begin{bmatrix} \Delta\phi^1_{A,k|k-1} - \mathbf{G}^1_k \boldsymbol{r}_{1,k|k-1} \\ \vdots \\ \Delta\phi^m_{A,k|k-1} - \mathbf{G}^m_k \boldsymbol{r}_{m,k|k-1} \end{bmatrix} \tag{4.25}$$

$$\mathbf{H} = \begin{bmatrix} -\mathbf{G}^1_k & 1 \\ \vdots & \\ -\mathbf{G}^m_k & 1 \end{bmatrix} \tag{4.26}$$

$$\mathbf{x} = \begin{bmatrix} \boldsymbol{r}_{A,k|k-1} \\ c\delta t_{A,k|k-1} \end{bmatrix} \tag{4.27}$$

Least squares can then be used to estimate the state vector ($\mathbf{x}$) in the above equations. The estimated change in position along with the associated covariance matrix are output from the TDCP algorithm for use in the higher-level sensor fusion.

### 4.2.4 Error Characterization

Experimental data was collected for characterizing the errors of the TDCP and DRTK outputs. Two Novatel receivers, described further in Chapter 6, collected raw GPS observables from antennas spaced approximately 50 m apart. For this characterization data, the two antennas remained stationary and data was collected over a two minute duration. This duration allows for a constant geometry and availability of GPS satellites from which a clear multi-variate distribution would manifest. The covariance reported from the differential GPS algorithms are compared to that of the sample distribution by visually inspecting the $3\sigma$ covariance ellipse over a 2D scatter plot [53].

The results are shown in Figure 4.2 for DRTK outputs and Figure 4.3 for the TDCP outputs. In these figures, the samples have the bias removed by subtracting the mean of the data and the covariance ellipses are plotted about the origin. These figures show that the reported and empirical covariance ellipses share resemblance in both magnitude

and orientation. The magnitude of the ellipse speaks to the accuracy of the output, where 99.7 % of the error samples should fall within the $3\sigma$ ellipse [60]. The ellipse also shows the cross correlation, indicated by the skewed orientation of the ellipse. By using the full covariance matrix in a higher-level estimator, the fusion will properly weight the direction of measurement corrections.

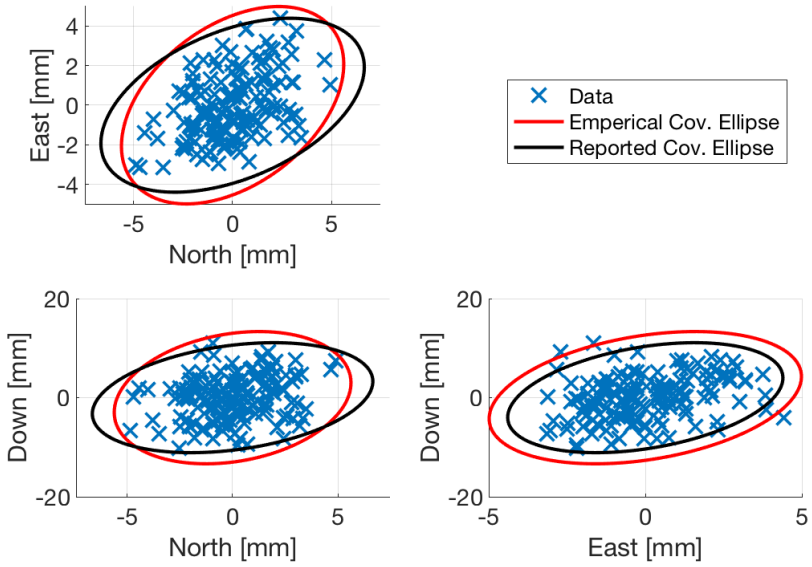


Figure 4.2: DRTK Covariance Analysis

## 4.3 Body-Centric Odometry Using Onboard Sensors

Body-centric odometry is a measurement of the change in a vehicle's position and heading between two instances in time. This measurement is derived from integrating a vehicle motion model over a set time interval. There are many different ways of deriving body-centric vehicle odometry, e.g., from an inertial navigation system [61], visual odometry from cameras [62], or iterative closest point (ICP) from lidar data [63], just to name a few. The method used in this dissertation uses wheel speeds and yaw rate for deriving body-centric vehicle odometry. Wheel speed and yaw rate measurements were chosen for this application given that they are commonly available as signals on the CAN bus of production vehicles.

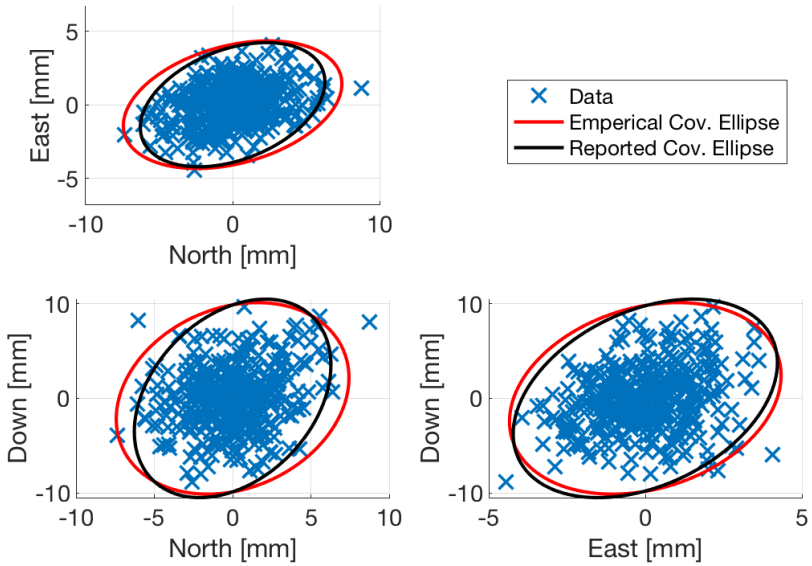


Figure 4.3: TDCP Covariance Analysis

### 4.3.1 Onboard Sensor Measurements

#### Wheel Speeds

Wheel speeds are measured by encoders which measure the number of discrete ticks in a time period. The number of ticks are converted to an angular distance using the known value of ticks per revolution of the encoder. Numerically differentiating the angular distance over the elapsed time results in a measure of the wheel's angular rate. An embedded system performs these lower level calculations at a high frequency to reduce errors due to the approximation of angular rate.

Assuming zero longitudinal slip, the wheel's linear speed can be calculated from the tire radius. The onboard system uses a nominal tire radius to convert the values to linear speed before reporting to the CAN bus. The tire radius, however, is subject to change due to varying tire pressure or varying temperatures which influence tire pressure. The change in radius will result in a scaling error in the resulting wheel speed. To compensate for this error, a calibration routine is performed at the start of each run and intermittently

throughout a run. Using the reference GPS-INS system as a speed reference, a maximum likelihood estimator is employed to determine the best scaling value to apply to the wheel speed measurements [53].

The wheel speed error distributions for both before and after compensating for wheel radius scaling are shown in Figure 4.4. The distribution shows a characteristic bell curve of a Gaussian distribution. These errors can then be characterized by a standard deviation value that will be used to model the confidence assigned to body-centric odometry measurements. The standard deviation of speed errors taken empirically over the data set is 7.8 cm/s. Note that some of the error in Figure 4.4 is due to the reference system, which reported a standard deviation of approximately 4.3 cm/s throughout the run. After accounting for the reported accuracy of the reference system, the value of the wheel speed standard deviation ($\sigma_{ws}$) was taken to be 6.5 cm/s.

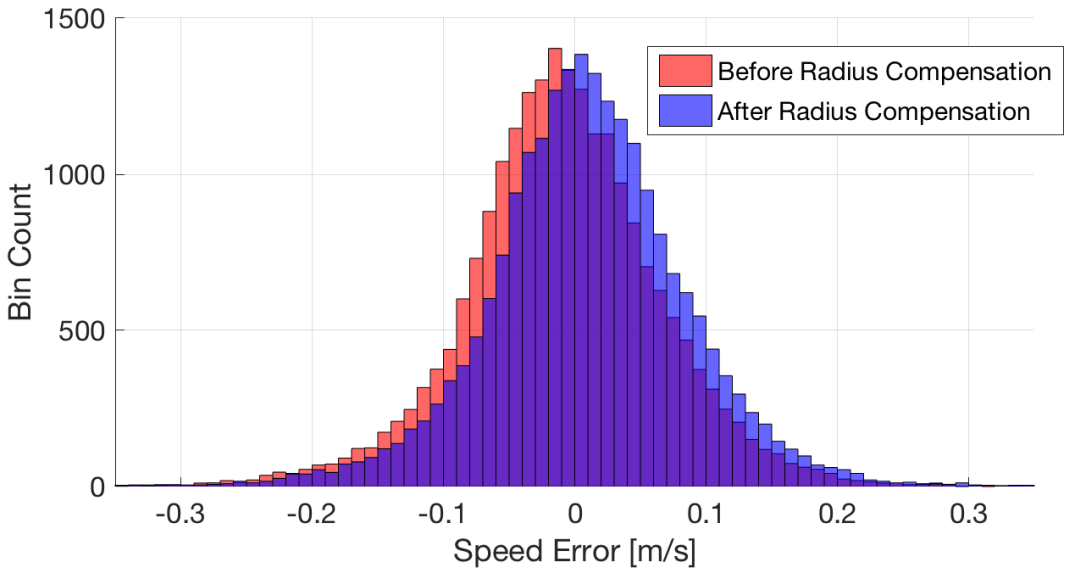


Figure 4.4: Wheel Speed Error Distribution Before and After Radius Compensation

### Yaw Rate

The yaw rate is measured by a gyroscope that is mounted to the vehicle's chassis and measures rotation about the $z$-axis. The errors in gyroscope measurements are predominantly due to white noise and biases caused by thermal effects on the sensor. Other errors include misalignment errors and scaling errors, which are largely deterministic and can be addressed in a factory calibration routine [27].

The error model for yaw rate ($\dot{\psi}$) measured by the gyroscope is

$$\omega = \dot{\psi} + b_\omega + \nu_\omega \tag{4.28}$$

where $\dot{\psi}$ is the true yaw rate, $b_\omega$ is the bias, and $\nu_\omega$ is the additive noise [32]. The additive noise is modeled as zero-mean Gaussian with standard deviation ($\sigma_\omega$) as defined in the equation below.

$$\sigma_\omega^2 = E\left[\nu_\omega^2\right] \tag{4.29}$$

The bias term is stochastic in nature and will vary ("walk") throughout a run. There are many different ways to estimate and account for the bias. One option would be to estimate the yaw rate bias online by appending the state vector of the graph-based estimator given in Equation (3.17). This, however, is outside the scope of the presented work and would require extensive modifications to the estimator. Other methods exist, such as estimating the bias online using a GPS/INS system or a vehicle model with steer angle inputs [30]. These methods were not used for the sake of brevity and for keeping the solution as platform independent as possible.

For the results shown in Chapter 6, a bias value is estimated prior to each data collection using a 60 second static initialization period. With the vehicle stationary, the true yaw rate is zero, and the bias value is taken as the average of the yaw rate readings over the stationary interval. The resulting bias value is then removed from the yaw rate measurements over the entire run. In this way, the dynamics of the yaw rate bias are neglected ($\dot{b}_\omega = 0$). It's

assumed that any errors caused by a walking bias term are small compared to other error sources in the body-centric odometry model.

Another source of error in yaw rate measurement is the breakdown of the planar motion assumption. The planar motion assumption implies that the $z$-axis of the body frame is aligned with the $D$-axis of the navigation frame. However, this is not always the case in the real world due to roll and pitch angles caused by vehicle dynamics and the bank and grade angles of the road. Any unaccounted for misalignment between the $z$ and $D$ axis will bias the true yaw rate about the $D$ axis. To demonstrate the magnitude of these misalignment errors, Equation (A.6) of the appendix is used to calculate yaw rate errors over a range of roll angles at three different yaw rates. The resulting yaw rate errors can be seen in Figure 4.5.

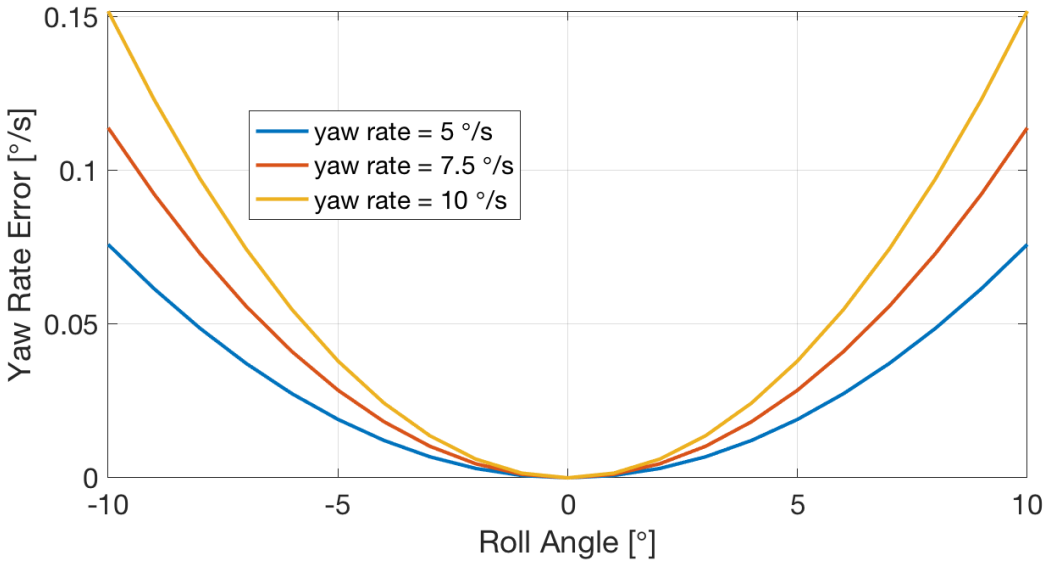


Figure 4.5: Yaw Rate Errors Due To Roll Angle With Planar Motion Assumption

For context, consider the non-planar errors present in the NCAT data set, whose results are shown in Section 6.2.2. In this data set, the test vehicles experience roll angles of ~8° at yaw rates of ~7.5° through the banked turns. The yaw rate measurements in this case will be biased by 0.07 °/s due to the breakdown of the 2D planar motion assumption. This

value is taken into account when choosing the noise level assigned to yaw rate measurements. Future work may consider using a 3D motion model for body-centric odometry to properly compensate for the biasing effect of the non-planar yaw rate errors.

The value of $\sigma_\omega$ used in the results that follow was chosen to be $0.2\,^\circ$/s. This value was chosen from empirical analysis of the system as presented in Section 4.3.3. The value is meant to encompass any uncertainty in the bias compensation as well as breakdowns in the planar motion assumption.

### 4.3.2 Body-Centric Odometry Calculation

The vehicle provides four independent wheel speed measurements at each wheel: front left ($\omega_{FL}$), front right ($\omega_{FR}$), rear left ($\omega_{RL}$), and rear right ($\omega_{RR}$). These are measurements of the wheel's angular rate in radians/second and, given the wheel radius ($R_\mathrm{w}$), the linear speed is calculated as $\mathrm{w}_{FL} = R_\mathrm{w}\omega_{FL}$, $\mathrm{w}_{FR} = R_\mathrm{w}\omega_{FR}$, $\mathrm{w}_{RL} = R_\mathrm{w}\omega_{RL}$ and $\mathrm{w}_{RR} = R_\mathrm{w}\omega_{RR}$. These values can be seen in the four tire diagram shown in Figure 4.6.

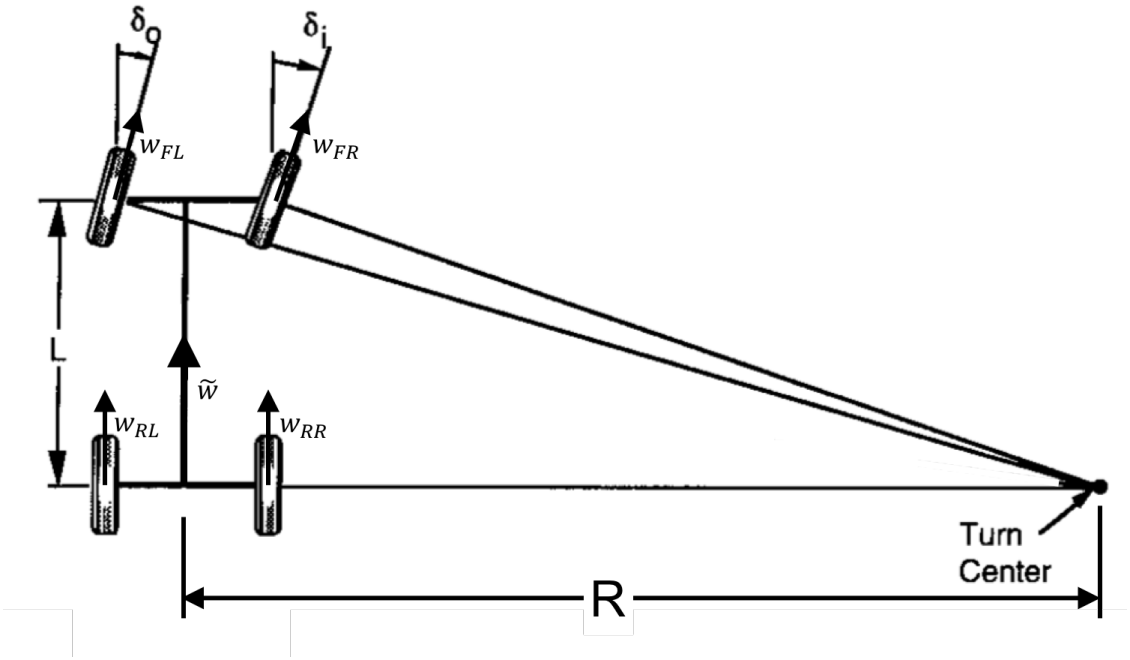


Figure 4.6: Four Tire Diagram [1]

Figure 4.6 also shows the instantaneous center of rotation for the vehicle assuming no lateral slip at the tires. By nature of the geometry, the inside wheels have a lower speed than the outside wheels. Also, the vehicle is designed with Ackermann steering geometry so that the front tires can travel in concentric circular patterns [1].

To simplify the problem, the bicycle model is used, which collapses the model to a single wheel on the front and rear axle. The equivalent wheel speed for each axle is taken as the average speed of the left and right wheels. Assuming no lateral slip, the front and rear tires of the bicycle model trace the same path and travel at the same speed. Also assuming no lateral slip, the velocity of the vehicle at the center of the rear axle ($\mathbf{v}_c^b$) will have zero lateral component [1]. From this, a velocity vector is formed as $\mathbf{v}_c^b = [\,\bar{\mathrm{w}}\,, 0\,]^T$ where $\bar{\mathrm{w}}$ is taken from averaging all four wheel speeds ($\bar{\mathrm{w}} = (\mathrm{w}_{FL} + \mathrm{w}_{FR} + \mathrm{w}_{RL} + \mathrm{w}_{RR})/4$.

An assumption of zero longitudinal slip is made when using wheel speeds for vehicle velocity. This is a reasonable assumption when force is not being applied by the tires. In fact, the force is proportional to wheel slip and one cannot exist without the other [1]. This relationship is considered when choosing which wheels are used for velocity calculation. When the vehicle is coasting (no torque is applied from the drive/braking system), all wheels are used. In all cases, the non-drive wheels (the front two wheels for the rear-wheel drive vehicle) are used. During braking events, all wheel speeds are used and the effects of slip are present, which results in a source of error in the velocity measurements. This was a design decision which emphasizes availability of the body-centric odometry over any biasing effects from slip during these times.

Using a rigid body assumption on the vehicle chassis, the velocity at the control point ($\mathbf{v}_p^b$) can be calculated as a function of the velocity at the rear center axle and rotation rate. The equation governing the motion at the control point is given below in Equation (4.30) [65].

$$\mathbf{v}_p^b = \mathbf{v}_c^b + \boldsymbol{\Omega} \times \boldsymbol{r}_{p|c}^b \tag{4.30}$$

Neglecting the rotation about the $x$ and $y$ axes allows for $\mathbf{\Omega} = \left[\, 0\,,\, 0\,,\, \dot{\psi} \,\right]^T$. Also, assuming that the control point is located on the vehicle's center line at distance $b$ ahead of the rear axle leads to $\boldsymbol{r}^b_{p|c} = [\, b\,, 0\,, 0\,]^T$. The velocity at the control point can be formed as shown in Equation (4.31).

$$\mathbf{v}^b_p = \begin{bmatrix} \bar{\mathrm{w}} \\ b\,\dot{\psi} \end{bmatrix} \tag{4.31}$$

Next, a state vector ($\mathbf{x}_{VM}$) is defined for use in propagating the vehicle motion between measurement intervals. The state vector, given in Equation (4.32), includes the 2D position $(x_{rel}, y_{rel})$ and yaw ($\psi_{rel}$) of the vehicle at the control point with respect to the reference frame defined by the vehicle's initial pose.

$$\mathbf{x}_{VM} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_{rel} \\ y_{rel} \\ \psi_{rel} \end{bmatrix} \tag{4.32}$$

The motion model ($\mathbf{f}_{VM}$) describes the rate change of the state vector ($\dot{\mathbf{x}}_{VM}$) as a function of the current state, model inputs ($\mathbf{u}$) and model disturbances ($\boldsymbol{\nu}$).

$$\dot{\mathbf{x}}_{VM} = \mathbf{f}_{VM}\left(\mathbf{x}_{VM}, \mathbf{u}, \boldsymbol{\nu}\right) \tag{4.33}$$

For the presented vehicle model, the motion model is given below.

$$\mathbf{f}_{VM}\left(\mathbf{x}_{VM}, \mathbf{u}, \boldsymbol{\nu}\right) = \begin{bmatrix} \mathbf{C}^o_b(\psi_{rel})\mathbf{v}^b_p \\ \dot{\psi} \end{bmatrix} \tag{4.34}$$

In the above equation, $\mathbf{C}_b^o(\psi_{rel})$ is the rotation matrix from the vehicle frame to the odometry reference frame as shown below.

$$\mathbf{C}_b^o = \begin{bmatrix} \cos(\psi_{rel}) & \sin(\psi_{rel}) \\ -\sin(\psi_{rel}) & \cos(\psi_{rel}) \end{bmatrix} \tag{4.35}$$

The input vector $\mathbf{u}$ consists of wheel speed and yaw rate as $\mathbf{u} = [\, u_1 \,, u_2 \,]^T = \left[\, \bar{\mathrm{w}} \,, \dot{\psi} \,\right]^T$.

The disturbances are modeled as having three separate components: the noise acting on wheel speed ($\nu_{\mathrm{w}}$), the noise acting on yaw rate ($\nu_{\dot{\psi}}$), and the zero sideslip modeling error ($\nu_{v_y}$) such that $\boldsymbol{\nu} = [\, \nu_1 \,, \nu_2 \,, \nu_3 \,]^T = \left[\, \nu_{\mathrm{w}} \,, \nu_{\dot{\psi}} \,, \nu_{v_y} \,\right]^T$. Each disturbance acts on the model in a different way. The wheel speeds and yaw rate disturbances are additive to their corresponding measurements. The zero sideslip disturbance is included in the model in the $y$ component of $\mathbf{v}_c^b$.

The disturbance acting on wheel speed and yaw rates are modeled as zero-mean Gaussian noise as characterized in the previous section. The zero sideslip modeling error term is meant to capture additional uncertainties in the model. In real-world conditions, this disturbance is correlated in time (i.e., the slip will bias in one direction through the entirety of a turn). However, for use in the estimation scheme, this disturbance is treated as white noise without temporal correlation. In practice, this value serves as a tuning parameter based on the expected dynamics in a run.

The disturbances have a covariance matrix ($\mathbf{Q}_c$) as defined in Equation (4.36).

$$\mathbf{Q}_c = E\left[\boldsymbol{\nu}\boldsymbol{\nu}^T\right] = \begin{bmatrix} \sigma_{ws}^2 & 0 & 0 \\ 0 & \sigma_{\dot{\psi}}^2 & 0 \\ 0 & 0 & \sigma_{v_y}^2 \end{bmatrix} \tag{4.36}$$

The disturbances are modeled as independent and uncorrelated, resulting in zeros in the off-diagonals of $\mathbf{Q}_c$. Plugging in from previous equations, the motion model can be shown

as a function of the states, inputs, and disturbances as shown below.

$$\mathbf{f}_{VM}\left(\mathbf{x}_{VM}, \mathbf{u}, \boldsymbol{\nu}\right) = \begin{bmatrix} \cos(x_3)(u_1+\nu_1) - \sin(x_3)(\nu_3 + b(u_2+\nu_2)) \\ \sin(x_3)(u_1+\nu_1) + \cos(x_3)(\nu_3 + b(u_2+\nu_2)) \\ u_2 + \nu_2 \end{bmatrix} \tag{4.37}$$

The covariance of the body-centric odometry is an important factor for use as a measurement in the higher-level sensor fusion. For calculating body-centric odometry covariance, the covariance of the propagated state is also maintained throughout the propagation process. $\mathbf{x}_{VM}$ is defined as the true vehicle pose with respect to the odometry frame. Accordingly, $\hat{\mathbf{x}}_{VM}$ is the value of $\mathbf{x}_{VM}$ attained through propagation of the motion model with all its uncertainties. The error in the propagated state ($\mathbf{e}_{VM} = \mathbf{x}_{VM} - \hat{\mathbf{x}}_{VM}$) has a covariance matrix ($\mathbf{P}_{VM}$) as defined in Equation (4.38).

$$\mathbf{P}_{VM} = E\left[(\mathbf{e}_{VM})(\mathbf{e}_{VM})^T\right] \tag{4.38}$$

The state covariance matrix evolves in time according to the Lyapunov equation as shown in Equation (4.39) [53].

$$\dot{\mathbf{P}}_{VM} = \mathbf{A}_c\mathbf{P}_{VM} + \mathbf{P}_{VM}\mathbf{A}_c^T + \mathbf{D}_c\mathbf{Q}_\mathbf{c}\mathbf{D}_c^T \tag{4.39}$$

The matrices ($\mathbf{A}_c$) and ($\mathbf{D}_c$) are the Jacobian matrices of the motion model as defined in Equations (4.40) and (4.41), respectively.

$$\mathbf{A}_c = \left.\frac{\partial \mathbf{f}_{VM}}{\partial \mathbf{x}_{VM}}\right|_{(\mathbf{x}_{VM}, \mathbf{u}, \boldsymbol{\nu})} = \begin{bmatrix} 0 & 0 & -\sin(x_3)(u_1+\nu_1) - \cos(x_3)(b(u_2+\nu_2)+\nu_3) \\ 0 & 0 & \cos(x_3)(u_1+\nu_1) - \sin(x_3)(b(u_2+\nu_2)-\nu_3) \\ 0 & 0 & 0 \end{bmatrix} \tag{4.40}$$

$$\mathbf{D}_c = \left.\frac{\partial \mathbf{f}_{VM}}{\partial \boldsymbol{\nu}}\right|_{(\mathbf{x}_{VM},\mathbf{u},\boldsymbol{\nu})} = \begin{bmatrix} \cos(x_3) & -b\sin(x_3) \\ \sin(x_3) & b\cos(x_3) \\ 0 & 1 \end{bmatrix} \tag{4.41}$$

In practice, the motion model is integrated between specific time intervals to produce a body-centric odometry measurement. At the start of each interval, the state vector is reset to all zeros, thereby defining the odometry reference frame. On initialization, the error is known to be zero and the initial state covariance matrix is set to all zeros. The models given in Equation (4.37) and Equation (4.39) are used to propagate $\hat{\mathbf{x}}_{VM}$ and $\mathbf{P}_{VM}$ throughout the interval, all while updating $\dot{\psi}$ and $\bar{\mathrm{w}}$ as new measurements are obtained. At the end of the interval, the final state value and covariance are extracted for use as a measurement in the graph-based path estimator.

### 4.3.3 Error Characterization

Experimental data was used to evaluate the errors involved in body-centric odometry calculation. The data was taken from a typical highway driving scenario with the experimental platform described later in Chapter 6. The onboard measurements of wheel speeds and yaw rates are sampled at 50 Hz while body-centric odometry is calculated and output at a rate of 2 Hz. The outputs are compared to ground truth provided by a GPS-INS system.

The resulting body-centric odometry errors are shown in Figure 4.7 along with a $3\sigma$ bound taken from the reported covariance. The $3\sigma$ bound envelops a majority of the corresponding errors, which gives credibility to the variance reported by the odometry module. The RMS of the errors, taken over the length of the data set, are 2.2 cm for $\Delta x$, 1.9 cm for $\Delta y$, and 0.018° for $\Delta\psi$.

The errors from Figure 4.7 are shown as an error probability distribution in Figure 4.8. Also shown in Figure 4.8 is the Gaussian bell curve associated with the model's reported covariance. The empirical distribution doesn't exactly fit the Gaussian bell curve which indicates that there are non-Gaussian errors acting on the model. The non-Gaussian shape

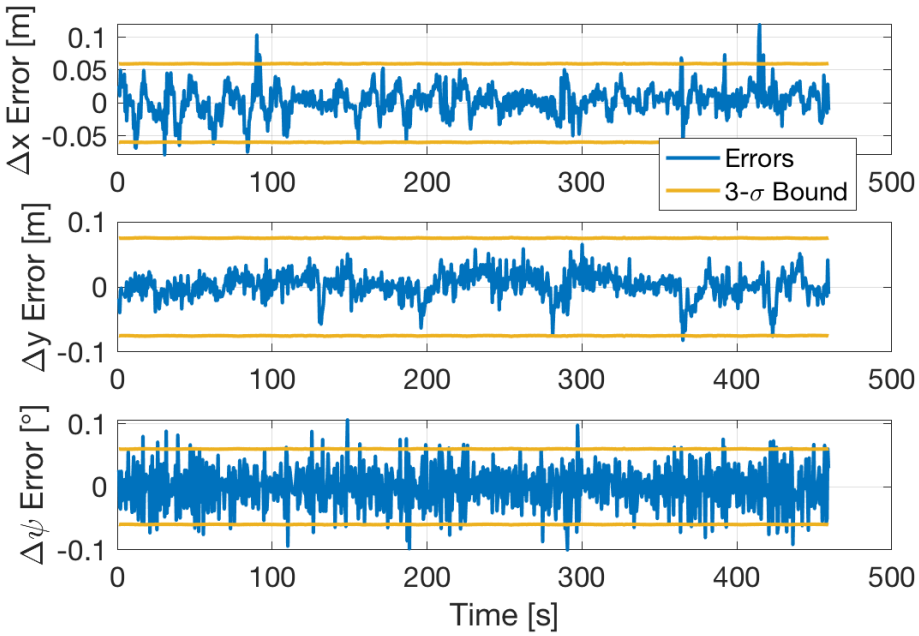


Figure 4.7: Body-Centric Odometry Errors vs. Time

is a result of the unmodeled errors described previously: wheel slip, breakdowns in the 2D planar assumption, and yaw rate bias. However, the important result of this plot is that the bell curve encompasses most of the empirical error distribution so that the measurements are weighted appropriately in the graph-based path estimator.

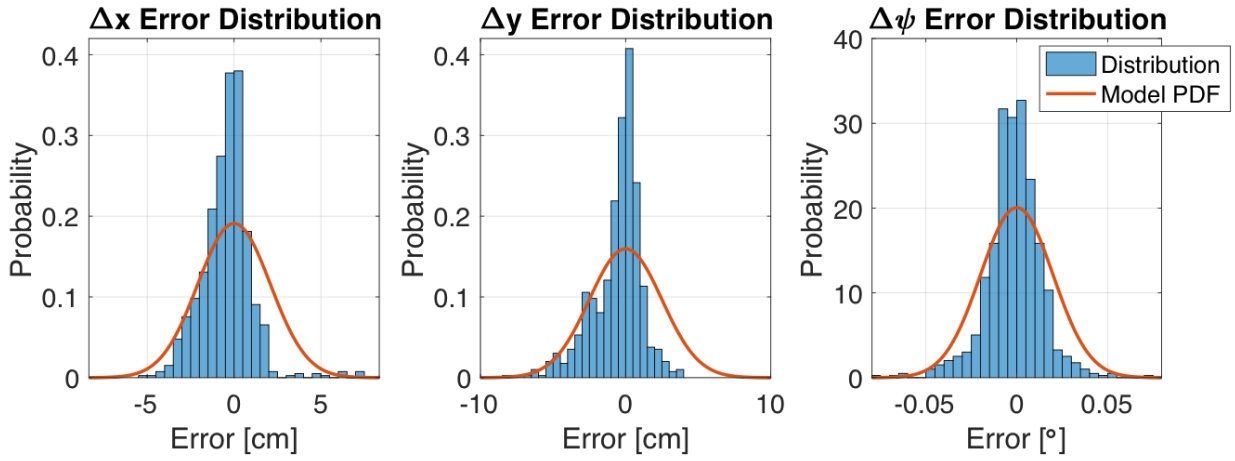


Figure 4.8: Body-Centric Odometry Error Distributions

## 4.4 Landmark Feature Extraction

The estimator described in Chapter 3 was presented with generic landmark observations resolved in the vehicle's body frame. Many different perception-based sensors are capable of producing such an observation (e.g., radar, lidar, camera). The process of producing a landmark observation from raw perception data is referred to as feature extraction. Feature extraction is non-trivial and has been the subject of a plethora of research in recent decades. Many commercial automotive groups have entire divisions dedicated solely on processing perception data. The approaches for feature extraction are often tailored to the particular type of sensor.

This section will focus on feature extraction from lidar data, which is the sensor used in experimental validation in this dissertation. First, a general overview of feature extraction is provided and followed by an overview of the lidar sensor's operation. The remaining sections will describe the particular methodologies employed in this work. This includes a section on point cloud pre-processing followed by sections on road sign and pole-like feature detection.

### 4.4.1 Feature Extraction Overview

The process of feature extraction can be broken down into two main components: detection and description. The detection step takes raw sensor data, such as a camera image or a point cloud, and selects a set of distinct points that stand out from the scene. By extracting features, the dimensionality of a set of data is significantly reduced, which alleviates the demand on bandwidth and processor resources. For example, the 3D point cloud shown in Figure 4.9 is made up of thousands of unsorted 3D points which have little context on their own. A feature extraction algorithm may detect the cylindrical object in the foreground (a tree) in addition to a number of other corners and edges not easily perceived by the human eye. The criteria for what should be considered a feature is arbitrary so long as the same feature can be repeatedly detected in a scene. In the case of place detection or collaborative navigation, the feature should be detectable regardless of vantage point.

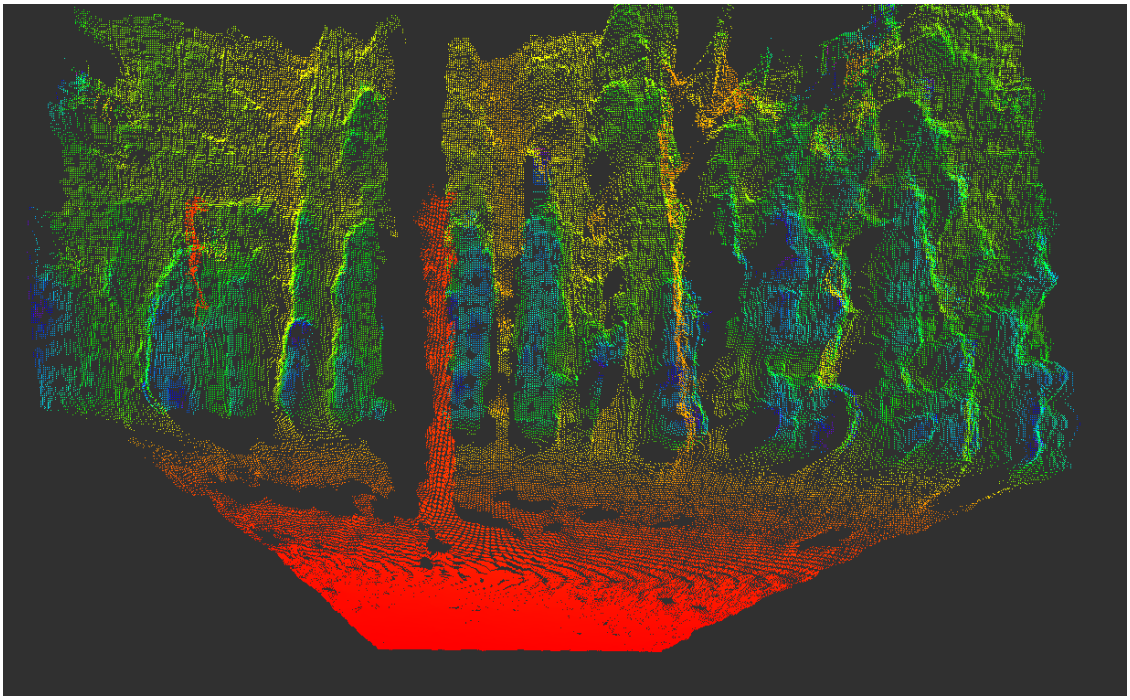

Figure 4.9: Point Cloud Data of a Cylindrical Landmark Environment

The description component of feature extraction is an important aspect for data association. For each feature detected in a scene, the raw perception data is used to describe that feature in the form of a descriptor [66]. Descriptors often look at the geometry of neighboring points/pixels along with other available data such as intensity and color [67]. The descriptors will be used to associate features from one scene to another. A successful descriptor will be capable of unambiguously identifying a feature regardless of vantage point. In the example shown in Figure 4.9, one may choose to describe the tree feature by the radius of the trunk, which should be consistent across different vantage points. However, this choice of descriptor will likely break down as more trees of similar size are detected. The descriptor could be improved further by considering the geometry of neighboring points.

### 4.4.2 Lidar Overview

Lidars calculate time-of-flight range measurements from light packets that are reflected from the environment. Each range measurement is associated with a particular azimuth and

elevation angle. In effect, the lidar provides a dense cloud of points in cartesian space of the surrounding environment, also known as a point cloud. Each point is also associated with an intensity reading. The intensity reading is influenced by the incident angle of the reflected surface along with the surface material [68].

A single-channel lidar, also referred to as planar or 2D lidar, rotates about a single axis and measures range and azimuth angle at regular intervals. In many cases, a single-channel lidar can be used to provide 3D information of the environment by pitching the sensor frame with respect to the vehicle's direction of travel. Due to the scanning nature of lidar sensors, the data is commonly described in terms of polar coordinates.

The Velodyne VLP-16 was chosen as the lidar for experimental validation in this dissertation. The VLP-16 consists of 16 channels that rotate about the same axis and are equally spaced in elevation angle increments of 2°. Each channel scans with a 360° field-of-view with range measurements spaced approximately 0.2° in azimuth. The reflecting mirrors rotate at 600 RPM, which results in a full 360° point cloud every tenth of a second and nearly 30,000 points per cloud.

Each point within the point cloud is subject to errors. The errors in the point's range measurement is predominantly due to random noise. In [68], controlled experiments performed on a VLP-16 found this error to have a standard deviation of around 7.5 mm at nominal range. The errors in the azimuth angle are small relative to the resolution of the angular steps. The error model used in the detection process is approximated as Gaussian with the $2\sigma$ value equal to half the azimuth step size.

The VLP-16 was designed for use on mobile robotic platforms and timing information becomes increasingly important at higher speeds. The VLP-16 accepts external GPS timing information which aids in synchronization for sensor fusion and in disciplining the rotating motor. Additionally, when a GPS time fix is provided, the Velodyne will stamp individual points with unique times. The vehicle motion can be used in combination with the precise timing of each point to prevent warping of the point cloud at higher speeds.

### 4.4.3 Point Cloud Pre-Processing

One major concern when using a scanning lidar at highway speeds is the point cloud "warping" effect due to motion of the vehicle. Given that it takes ~0.1 s to form a full rotation scan, even typical speeds will cause significant warping errors between the first and last points measured in the rotation. At a moderate highway speed of 20 m/s (~45 mph), the maximum range error for an individual point would be 2 m, which is well outside acceptable accuracy.

To address this issue, the vehicle's ego-motion is used. Ego-motion refers to the velocity and angular rates of a body, resolved in the body frame of which the values are observed. Given a set of timestamped points measured throughout a lidar's rotation, the ego-motion is used to transform the points to be with respect to the lidar's position at a particular time [69]. With all the points in a cloud transformed into the same respective frame, the point cloud warping effect is mitigated.

The ego-motion compensation method is used to accumulate points over multiple full rotations of the lidar scanner. Accumulating points over larger durations increases the point cloud density compared to the density provided by a single lidar rotation. Increased point cloud density provides a more detailed snapshot of the surrounding environment which aids in the feature extraction process. For this dissertation, 0.5 s was used as the duration for accumulating the lidar points. The dense point cloud is generated from the accumulated points on each GPS epoch (sampled at 2 Hz) resulting in time synchronization between GPS and landmark observations.

There are tradeoffs when choosing the duration used to accumulate points. Errors in the ego-motion will accumulate throughout the duration and result in errors in the accumulated point cloud. Body-centric odometry, as described in Section 4.3, is used for the ego-motion compensation in this work. Based on the characterization shown in Section 4.3.3, this will result in $1\sigma$ errors of 2.2 cm and 1.9 cm due to longitudinal and lateral drift, respectively.

Additionally, errors due to yaw drift will accumulate over the duration. For a landmark at the maximum expected range of 100 m, the $1\sigma$ error in the points position will be 5.6 cm.

These values are considerably high compared to the lidar's point range accuracy, but they are worst case values: the errors between two points sampled at the start and end of an accumulation period of a landmark at maximum range. In practice, this error source did not significantly diminish the ability of feature extraction to detect landmarks in the point cloud. Additionally, point cloud accumulation uncertainty is taken into account in the landmark observation covariance model given later in Section 4.4.4.

### 4.4.4 Road Sign Feature Extraction

The first feature extraction method employed in this dissertation seeks to detect road sign features in the environment. Road signs make excellent features for many reasons. First, road signs are easily distinguished from the surrounding environment due to their flatness and distinct shape. Further, the road sign's smooth, metal composition and tendency to be oriented towards traffic results in relatively high intensity readings. The intensity readings can be used as an additional qualifier in the detection process [68]. Another benefit of using road signs is that they are typically spaced in a way that provides suitable geometry for the navigation solution and eases landmark correspondence.

The dense point cloud described in the previous section is used to perform sign feature extraction. Figure 4.10 shows an example of a point cloud with a road sign in the scene along with an image of the corresponding sign. The background of the visualization to the left shows the 3D mesh of the vehicle on which the lidar is mounted. In the foreground are the points returned from the road sign. The color of each point in the cloud is mapped to the intensity reading. The road sign points stand out due to their high intensity readings and planar geometry. This section will describe the process for automatically detecting road sign features with these characteristics in mind.

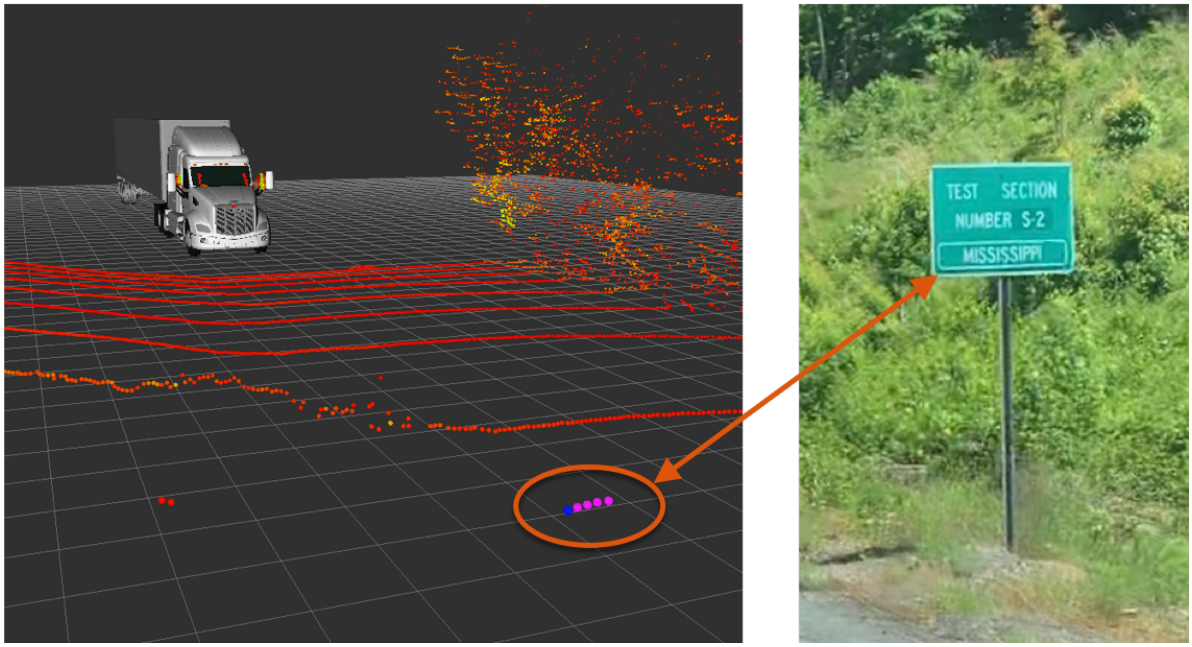


Figure 4.10: Raw Point Cloud of Road Sign Points in RViz [2] and Corresponding Image

For road sign detection, the cloud is first filtered based on a minimum intensity reading. By considering only high intensity points, objects in the natural environment are largely excluded. A clustering algorithm is then employed to group points based on their Euclidean distance [70] with the following parameters:

- Minimum number of points: 5
- Cluster tolerance (minimum distance from one cluster to the next): 0.55 m

Each cluster is then treated as a candidate for a detected road sign. The clusters are conditioned by their size, measured as the largest distance between two points in the cluster. Only clusters below a certain width (1.8 m) are considered for road sign detection.

For the remaining candidate clusters, a least squares planar fit is applied for estimating the object's orientation. The quality of the fit, taken as the root mean squared (RMS) of the residuals, is used for conditioning [53]. Only clusters with a low enough RMS value (3 cm) are considered as road sign candidates. Conditioning by RMS removes objects like telephone poles or neighboring vehicles, which do not fit a clean, flat plane. The planar fit

also provides the road sign's relative orientation, which is saved as a unit vector of the plane's normal direction. The sign's relative position measurement is taken from the centroid of all points in the cluster.

A simplified, 2D example of the planar fit is shown in Figure 4.11. The line segment in the figure corresponds to the sign's orientation estimate resulting from the fit. The width of the line segment corresponds to the sign's approximated width. The line-of-sight to the sensor is also shown, which comprises the relative position of the landmark observation.

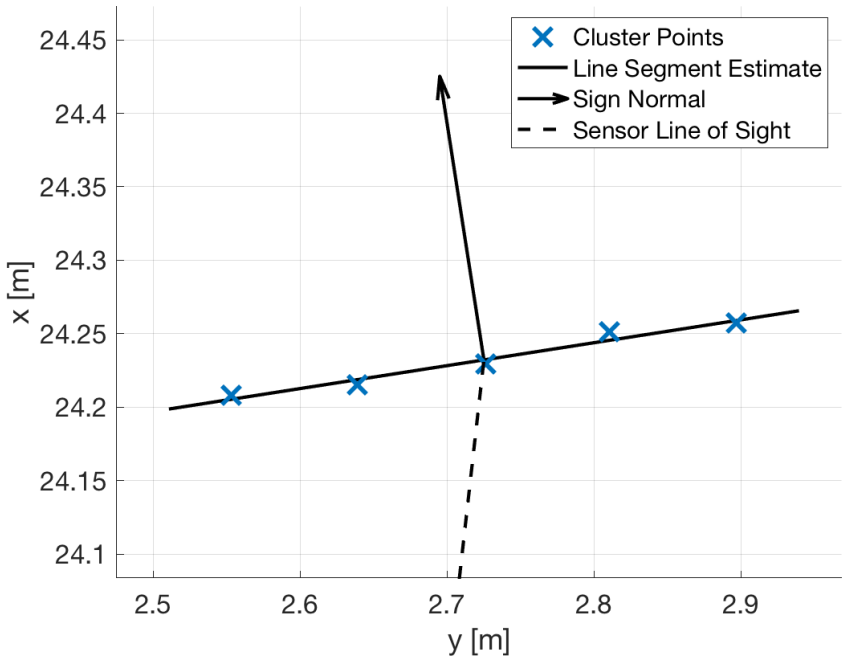


Figure 4.11: Road Sign Landmark Line Fit Example

### Road Sign Landmark Covariance Model

Next, a covariance is assigned to each detected road sign. This includes a covariance of the relative position ($\tilde{x}$ and $\tilde{y}$) and orientation ($\tilde{\beta}$) observations. The relative position covariance is a function of the number of points ($N_p$), the detected orientation of the sign, the step size of the lidar's azimuth angles ($\Delta_\alpha$), and the standard deviation of range measurements for a single point ($\sigma_\rho$).

The errors in the detected position manifest primarily in the along-sign direction (tangential to the sign) due to the limited azimuth angle resolution. At a distance of 60 m for example, the points are separated by ~0.2 m and the sign could fall anywhere within +/- 0.1 m in its tangential direction. The true error distribution in the sign's tangential direction is uniform with a width equal to the distance between points ($\Delta_t$), which is approximated using the azimuth step size ($\Delta_\alpha$) and range measurement ($\tilde{r} = \sqrt{\tilde{x}^2 + \tilde{y}^2}$) as $\Delta_t = \tilde{r}\Delta_\alpha$. This error is approximated as a Gaussian distribution by setting the $2\sigma$ value equal to half the width of the uniform distribution as

$$\sigma_t = \frac{1}{4}(\Delta_t) \tag{4.42}$$

where $\sigma_t$ is the standard deviation of the approximated Gaussian distribution in the tangential direction. This error contribution is rotated into the lidar frame using the sign's relative orientation measurement ($\tilde{\beta}$) as

$$\mathbf{R}_t = \mathbf{C}(\tilde{\beta}) \begin{bmatrix} 0 & 0 \\ 0 & \sigma_t^2 \end{bmatrix} \mathbf{C}(\tilde{\beta})^T \tag{4.43}$$

where $\mathbf{C}(\angle)$ is the 2D rotation matrix operator.

Another error source in the detected relative position is errors in the range measurement of each point. The point range error is treated as zero-mean Gaussian with a standard deviation $\sigma_\rho = 7.5mm$. As the number of points on a sign increases, the position uncertainty in the lidar's radial direction drops. This decrease in uncertainty is modeled as the averaging of $N_p$ random variables. Therefore, the equation used to calculate the standard deviation in the radial direction ($\sigma_r$) is given below.

$$\sigma_r = \frac{\sigma_\rho}{\sqrt{N_p}} \tag{4.44}$$

This error contribution is rotated into the lidar frame using the relative bearing to the sign ($\tilde{\alpha} = \text{atan2}(\tilde{y}, \tilde{x})$) as shown in Equation (4.45).

$$\mathbf{R}_r = \mathbf{C}(\tilde{\alpha}) \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & 0 \end{bmatrix} \mathbf{C}(\tilde{\alpha})^T \tag{4.45}$$

Next, the contributions from the radial and tangential error sources are combined by summing the two covariance matrices. The resulting covariance matrix is resolved in the lidar's frame and is used as the relative position measurement covariance, $\mathbf{R}_{LM,pos}$.

$$\mathbf{R}_{LM,pos} = \mathbf{R}_t + \mathbf{R}_r \tag{4.46}$$

The covariance assigned to the detected sign orientation comes from the covariance of the least squares estimate ($\mathbf{R}_{LM,yaw}$) as shown in Appendix C. The orientation measurement is assumed to be uncorrelated to the relative position measurement. From this, the $3 \times 3$ covariance matrix for the landmark observation ($\mathbf{R}_{LM}$) is formed as shown in Equation (4.47).

$$\mathbf{R}_{LM} = \begin{bmatrix} \mathbf{R}_{LM,pos} & \mathbf{0}_{2\times 1} \\ \mathbf{0}_{1\times 2} & \mathbf{R}_{LM,yaw} \end{bmatrix} \tag{4.47}$$

The covariance model presented in this section provides a representative shape of the error ellipse as is demonstrated in Section 4.4.6. The magnitude of the covariance model is inflated in practice to account for uncertainties introduced in the point cloud accumulation step. The inflation is applied as an additive term as

$$\mathbf{R}_{LM,inf} = \mathbf{R}_{LM} + \mathbf{R}_{inf} \tag{4.48}$$

where $\mathbf{R}_{inf}$ is a diagonal matrix of inflation values as shown below.

$$\mathbf{R}_{inf} = \begin{bmatrix} \sigma^2_{x,inf} & 0 & 0 \\ 0 & \sigma^2_{y,inf} & 0 \\ 0 & 0 & \sigma^2_{yaw,inf} \end{bmatrix} \tag{4.49}$$

The values of $\sigma_{x,inf}$, $\sigma_{y,inf}$, and $\sigma_{yaw,inf}$ were chosen to be 1 cm, 2.5 cm, and 1.5° respectively. The inflated covariance is applied to the covariance model for both the simulation and experimental results in the chapters that follow.

### 4.4.5 Pole Feature Extraction

Another landmark type used in this dissertation is vertical, pole-like objects. Figure 4.12 shows an example of two poles detected from an accumulated point cloud as described previously in Section 4.4.3. In this figure, the poles are depicted as cylindrical markers that fit the scattered points that make up the pole object. This figure also shows the vehicle body frame and sensor frame from which the poles were detected.

Pole-like landmark types are ideal for detection with mobile lidar data and navigation. For one, these objects are commonly found in many environments: utility poles are often placed alongside major highways, support structures are commonly available in urban areas, and trees can be used for navigation in forestry applications. Another reason for using pole-like landmarks is their distinct geometric properties that help distinguish these objects in a lidar point cloud. Since these objects are vertical, the associated points in the cloud are stacked and often fairly isolated from neighboring points. With these distinct geometric properties in mind, a detection algorithm is employed for automatically classifying pole-like objects from lidar data and determining their center positions.

One important distinction between pole-like landmarks and road sign landmarks is the lack of a dominant orientation in the yaw angle. The poles of interest are cylindrical and vertical and therefore appear similar to a lidar from all vantage points. These objects, while

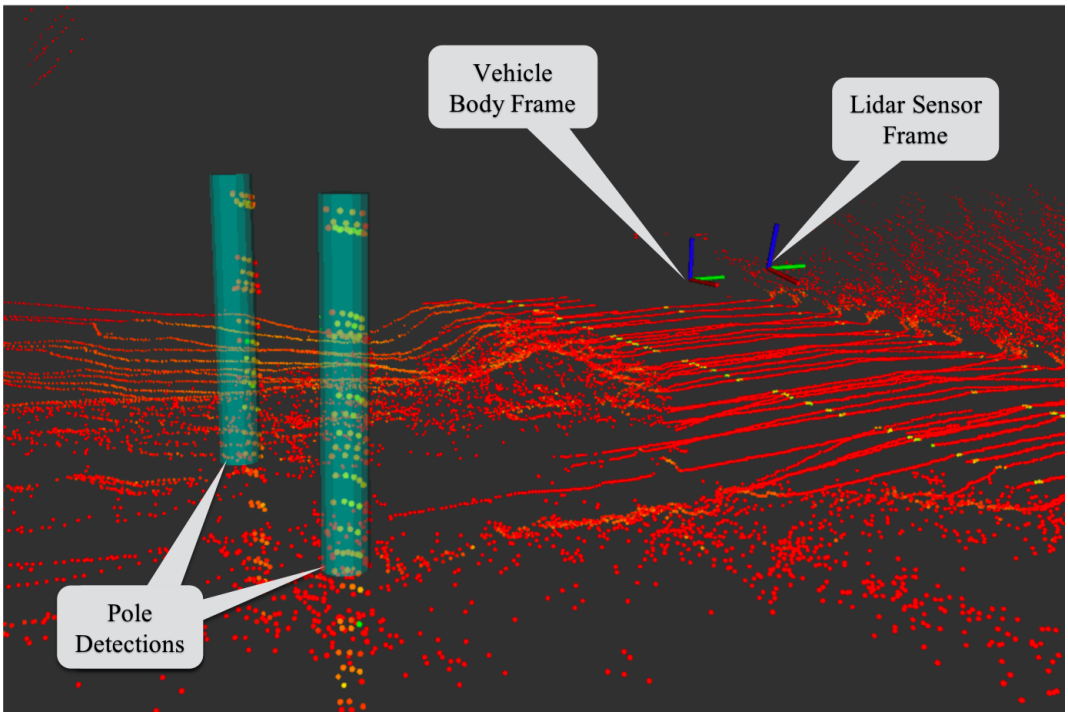


Figure 4.12: Pole Detection in Dense Point Cloud in RViz [2]

lacking a dominant yaw orientation, are still beneficial for localization by observing their relative position. In terms of a Relative Position Graph, these observations are considered degenerate as described in Section 3.8. Fortunately, the graph-based estimation approach is poised for handling degenerate observations.

The algorithm used for the detection of pole-like objects is adapted from the method employed by Lehtomäki in [71] with an additional step to refine the landmark position. The algorithm in [71] consists of segmenting and classifying points as belonging to a pole feature. The landmark position is then refined by estimating the pole's center based on the cluster of points. Last, an error model is used to assign the covariance of the extracted landmark observation.

### Pole-Like Object Point Classification and Segmentation

The classification and segmentation step seeks to classify points corresponding to pole-like objects and to group them for later refinement. A typical clustering algorithm will not

work with the full point cloud since pole-like features are adjoined by ground points, power lines, tree canopies, etc. So instead, the algorithm works incrementally on slices of the point cloud [71].

The full point cloud is split up according to the ring/channel by which they were measured. For each ring, cartesian clustering is performed to group points within 0.3 m of each other. The clustering is performed in the 2D space to account for any gaps in the point cloud due to shadowing or other occlusions. The clusters are tagged with the following geometric properties: 2D centroid position, centroid height, and width (taken as the largest distance between two points in the cluster). Each cluster is then conditioned to be considered as part of a pole-like object using the following criteria:

- The width of the cluster is less than the max expected pole diameter (0.6 m)
- The cluster contains sufficient number of points (5)

The next step is to group clusters along the vertical dimension. The clusters in the lowest ring are used as seeds in the region growing algorithm and are saved as candidate pole clusters. The clusters from the next ring are then considered. Each cluster in the ring is compared to each candidate pole cluster in search of a match. A match is assigned if the following conditions are met:

- The clusters are sufficiently vertical: the tilt angle between centroids is less than the max expected pole tilt angle (15°)
- The clusters are within a similar height: the difference in height is less than 1.5 m
- The clusters are similar width: the percent difference in width is less than 100 %

The algorithm then steps through each ring and grows the regions for candidate pole clusters.

**Principal Component Analysis for Pole Direction Vector**

Principal Component Analysis (PCA) is then performed on each candidate pole cluster and used for further refinement of the extracted landmark position. The set of points within a cluster are used to calculate the eigenvalues and eigenvectors. The eigenvector associated with the largest eigenvalue is taken as the pole's direction vector [72]. Together, the eigenvectors form a rotation matrix from the sensor frame to the eigenspace, which is defined by the cluster's principal axes.

The covariance matrix of the pole cluster ($\mathbf{\Sigma}$) is calculated as

$$\mathbf{\Sigma} = \sum_{i=1}^{N_c} (\mathbf{p}_i - \bar{\mathbf{p}})(\mathbf{p}_i - \bar{\mathbf{p}})^T \tag{4.50}$$

where $N_c$ is the number of points in the cluster and $\bar{\mathbf{p}}$ is the cluster's centroid position, calculated as shown below.

$$\bar{\mathbf{p}} = \frac{1}{N_c} \sum_{i=1}^{N_c} \mathbf{p}_i \tag{4.51}$$

By definition, $\mathbf{\Sigma}$ is symmetric, positive, and semi-definite and therefore can be decomposed and written as

$$\mathbf{\Sigma} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \end{bmatrix}^T \tag{4.52}$$

where $\mathbf{e}_1$, $\mathbf{e}_2$ and $\mathbf{e}_3$ are the eigenvectors with corresponding eigenvalues $\lambda_1$, $\lambda_2$ and $\lambda_3$. The eigen decomposition is performed such that the eigenvalues are in ascending order ($\lambda_1 < \lambda_2 < \lambda_3$).

The result of PCA is twofold. For one, the object's shape can be inferred from the value of the eigenvalues. For elongated clusters (such as that of a cylinder), the third eigenvalue will be much greater than the other two ($\lambda_3 \gg \lambda_2 \approx \lambda_1$). The PCA also provides a rotation matrix which can be used to transform all points to the eigenspace. The rotation matrix is formed from the eigenvectors as $\mathbf{C}^e = [\, \mathbf{e}_1 \,, \mathbf{e}_2 \,, \mathbf{e}_3 \,]$.

The direction vector is used to calculate the tilt angle (the angle between the direction vector and the vertical axis). Additionally, the height of each cluster is taken as the difference in height between the highest and lowest points in eigenspace. These characteristics are used for further conditioning of the candidate sets with the following criteria:

- The height must exceed the minimum expected pole height (3 m)
- The tilt angle is less than the max expected pole tilt angle (15°)

**Hough Transform for Pole Center Position Determination**

At this point, the algorithm has found a set of clusters, each containing points of individual pole landmarks. The landmark observation (relative position) must then be extracted from the set of points for use in localization. A straight-forward approach would be to take the landmark position as the geometric centroid of all points in the cluster ($\bar{\mathbf{p}}$), however, this leads to bias in the extracted position. Due to the lidar's perspective, the points only make up a small arc of the full cylinder and the centroid is biased in the direction of the lidar compared to the true center of the pole.

To extract a landmark observation that is rotation invariant, the position of the cylinder's center should be used instead. The center position is estimated from the arc created by the lidar points using the Hough Transform [73]. The Hough Transform is a technique popularized in the computer vision field for estimating the parameters of geometric patterns in unorganized data. A common example is the use of the Hough Transform to detect lines within a pixelated image, however, this technique can be used with any parameterized shape in different types of data.

In this case, the Hough Transform is used to fit a circle from a set of scattered points and estimate the circle's 2D center position and radius. The Hough Transform is well suited for this case for a number of reasons. For one, the Hough Transform operates over a defined search space which limits possible values of center positions and radius. This is preferable compared to other methods, such as least squares, which can suffer from numerical issues or

result in infeasible values of radius [74]. Moreover, the Hough Transform is a voting-based approach and is therefore less susceptible to outliers (knots on trees, signs attached to poles, etc). Details on the Hough Transform Circle Fit operation can be found in Appendix D.

The first step in estimating the pole's center position is to rotate the 3D points into the cluster's eigenspace using the rotation matrix provided by PCA ($\mathbf{C}^e$). In this way, the $z$-axis ($z'$) aligns with the length of the pole, the $y$-axis ($y'$) aligns with the exposed face of the pole, and the $x$-axis ($x'$) follows orthogonally. The x and y components of the rotated points form a semicircle as shown in Figure 4.13. The 2D data is used to perform the circle fit for determining the center position $(a, b)$ and radius $(R)$.

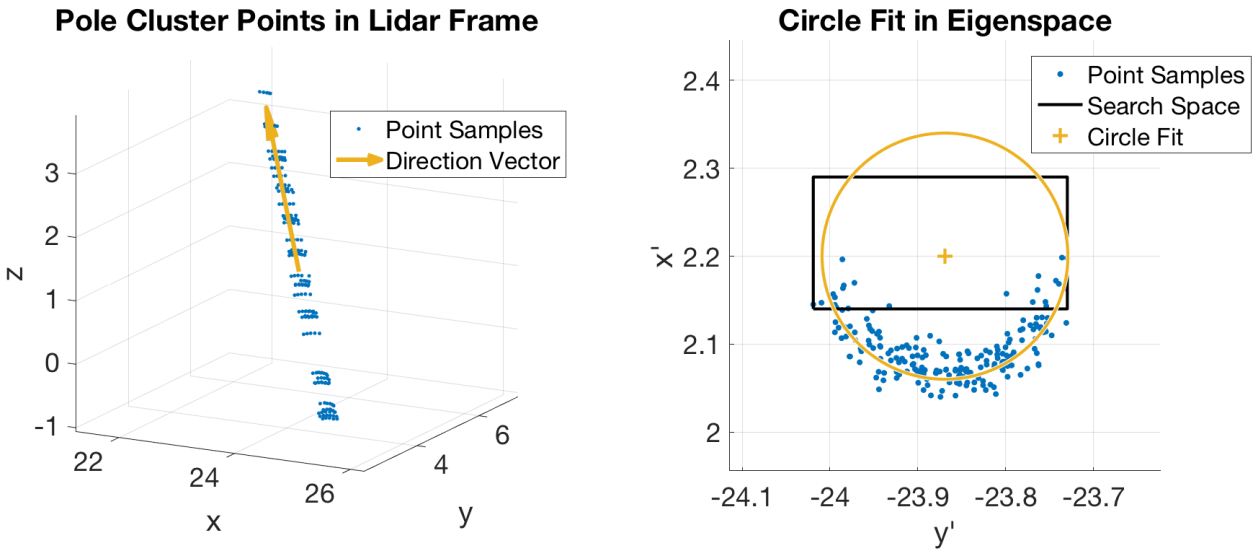


Figure 4.13: PCA Hough Transform Circle Fit Example

The next step is to determine the search space of possible values of circle parameters. The minimum and maximum expected radius is determined from some knowledge of the environment. For the case of utility pole and tree detection, the minimum and maximum radius were chosen as 0.1 and 0.3 m, respectively. The search space of the center position is chosen based on intuition of lidar operation: the points returned from a lidar will always lie somewhere between the sensor and the pole's true center. The search space for $a$ is taken from the left and right-most points. The search space for $b$ is taken from the closest point

plus the minimum expected radius to the closest point plus the maximum expected radius. The square shown in Figure 4.13 represents the center position search space for the points shown.

The Hough Transform Circle Fit, as described in Appendix D, is then performed using the $x'$ and $y'$ components of the $N_p$ points with a search space as described above. The $z$ component of the cylinder's center position ($\hat{z}'_c$) is taken as the midpoint between the max and min $z'$ values. The center position estimate ($\mathbf{p}_c^e = \left[\, \hat{a}\,, \hat{b}\,, \hat{z}'_c \,\right]^T$) is rotated back into the original frame using the rotation matrix as shown below.

$$\mathbf{p}_c^s = (\mathbf{C}^e)^T \mathbf{p}_c^e \tag{4.53}$$

**Pole Landmark Covariance Model**

The covariance assigned to a pole feature is used to weight the observation in the higher-level sensor fusion. The model for calculating the covariance of the estimated circle fit parameters comes from the geometry of the sample points used in the fit. It was shown in [75] that the covariance of the circle fit parameters is not a function of the radius, but of the number of points and their spatial distribution. First, $\mathbf{x}_{fit}$ is declared as a vector of the true circle parameters ($\mathbf{x}_{fit} = [\, a\,, b\,, R\,]^T$) and $\hat{\mathbf{x}}_{fit}$ as a vector of the estimated values. The circle fit covariance ($\mathbf{R}_{fit}$) is a $3 \times 3$ matrix as defined in the equation below.

$$\mathbf{R}_{fit} = E\left[ \left(\mathbf{x}_{fit} - \hat{\mathbf{x}}_{fit}\right) \left(\mathbf{x}_{fit} - \hat{\mathbf{x}}_{fit}\right)^T \right] \tag{4.54}$$

As presented in [75], the covariance matrix is a function of the direction angle of each point ($\alpha_i$) which is the angle from the fit center to the $i^{th}$ sample point.

$$\alpha_i = \text{atan2}\left(y'_i - b, x'_i - a\right) \tag{4.55}$$

The covariance matrix is shown in Equation (4.56) as a function of the $\alpha_i$ angles, the number of points ($N_p$) and a radial noise component ($\sigma_r$).

$$\mathbf{R}_{fit} = \sigma_r^2 \begin{bmatrix} \sum_{i=1}^{N_p} \cos^2 \alpha_i & \sum_{i=1}^{N_p} \sin \alpha_i \cos \alpha_i & \sum_{i=1}^{N_p} \cos \alpha_i \\ \sum_{i=1}^{N_p} \sin \alpha_i \cos \alpha_i & \sum_{i=1}^{N_p} \sin^2 \alpha_i & \sum_{i=1}^{N_p} \sin \alpha_i \\ \sum_{i=1}^{N_p} \cos \alpha_i & \sum_{i=1}^{N_p} \sin \alpha_i & N_p \end{bmatrix}^{-1} \tag{4.56}$$

The term $\sigma_r$ in this equation is the standard deviation of points in the circle's radial direction. Due to the nature of lidar returns, it is most often the case that points make up an arc that faces the lidar sensor. Because of this, the radial direction of the circle fit is similar to the radial direction of the lidar. In practice, the standard deviation of lidar point range measurements ($\sigma_\rho$) is used in place of $\sigma_r$.

The covariance of the estimated fit center ($\mathbf{R}_{ab}$) is taken as the first $2 \times 2$ submatrix of $\mathbf{R}_{fit}$. This covariance matrix, resolved in the eigenspace, is then rotated to the lidar frame as shown below.

$$\mathbf{R}_{LM} = (\mathbf{C}^e)^T \mathbf{R}_{ab} \mathbf{C}^e \tag{4.57}$$

The rotated covariance matrix is then assigned to the detected landmark position for use in higher-level sensor fusion.

### 4.4.6 Error Characterization

A stationary test was set up to characterize the road sign landmark measurements and to evaluate the error model used to approximate the error covariance. The sign in this example was positioned approximately 25 m from the lidar with a width of just under half a meter. For each scan of the lidar, 5 or 6 points were reflected from the sign. Over 1000 sign observations were extracted over the test duration. This sampling of sign observations was used to determine the empirical error distribution. Figure 4.14 shows each sample as a point in the lidar's cartesian space. The figure also shows $3\sigma$ error ellipses for the empirical

and modeled covariances. The error ellipses match sufficiently in terms of magnitude and the orientation of the dominant axes.

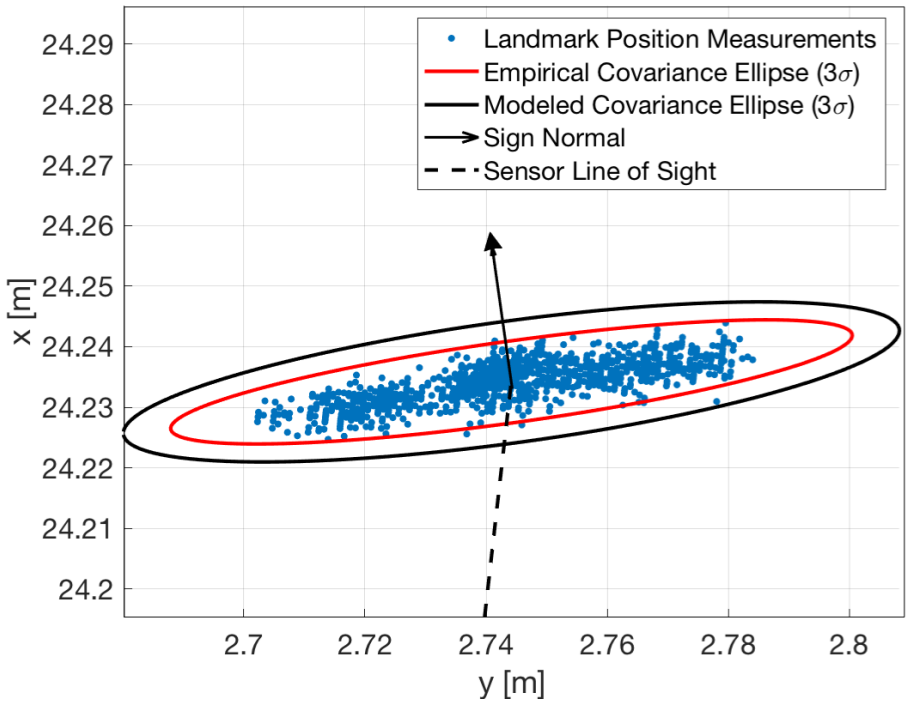


Figure 4.14: Sign Landmark Measurement Covariance Analysis

### 4.4.7 Landmark Correspondence

At this stage of pre-processing, the landmark observations of relative position and orientation are available along with an associated covariance, all resolved in the lidar sensor frame. The last step of landmark feature extraction is correspondence. In the correspondence step, the landmark observations are associated with specific landmarks. This is also known as the data association problem and is critical for accurate navigation. A false correspondence (where a landmark observation is assigned to the wrong landmark) could be detrimental to navigation performance [54].

Solving the data association problem is not an emphasis of this dissertation for the following reasons. First, the performance of a landmark correspondence algorithm would warrant an in-depth evaluation and would distract from the key contributions of the presented work. Additionally, the landmark density required for meeting the accuracy requirements is relatively low for the proposed graph-based path estimation approach (~50 m between landmarks) as is demonstrated in Section 5.5. For this reason, a lesser number of landmarks can be detected and the spacing between them can be relatively large. For the landmark detection approach detailed in Section 4.4.4, the landmarks will be spaced by at least 0.55 m as dictated by the cluster tolerance parameter.

Future work may consider building the landmark correspondence step into the path estimation algorithm. Some adaptations to the graph-based estimation approach allows for correspondence to be built into the estimation [47]. However, these adaptations dramatically increase the number of unknowns and therefore the size of the graph to be optimized. Additional information can be used to improve the performance of landmark correspondence. For example, the descriptors assigned to landmarks can be compared using a descriptor distance function, which is a measure of the similarity of two landmarks in terms of their shape and other distinctive attributes [67].

For the results presented in this dissertation, the data association problem is solved *a priori* (i.e., prior to being used in the graph-based path estimation). For the simulation study in Chapter 5, the landmark observations are assigned a landmark ID within the simulation. As for experimental data (the results of which are shown in Chapter 6), landmark correspondence is performed using a GPS/INS ground truth as described below.

Given the absolute pose of a vehicle and the lidar extrinsics, landmark observations are transformed into the navigation frame using the equations provided in Equations (2.1) and (2.2). The transformation results in a measurement of a landmark's global pose ($\mathbf{x}^n$) along with an associated covariance matrix ($\mathbf{P}^n$). The Mahalanobis distance ($\chi$) is then calculated for comparing two landmark observations as shown below [76].

$$\chi^2 = (\mathbf{x}_1^n - \mathbf{x}_2^n)^T (\mathbf{P}_1^n + \mathbf{P}_2^n)^{-1} (\mathbf{x}_1^n - \mathbf{x}_2^n) \tag{4.58}$$

The Mahalanobis distance is a powerful metric for correspondence in that it appropriately weights the positional and rotational distance between two landmarks, resulting in a single, scalar value of distance. Each new landmark observation is compared to all previously identified landmarks. If the Mahalanobis distance is below a certain threshold, then a correspondence is made. If the Mahalanobis distance is above a certain threshold, then a new landmark is added to the set of previously identified landmarks. All other observations with Mahalanobis distance between the two thresholds are treated as outliers and discarded from further use.

## 4.5 Conclusion

This chapter provided specifics on the measurements used for graph-based path estimation in this dissertation. This includes the differential GPS algorithms (TDCP and DRTK) that provide GPS odometry and inter-vehicle RPV measurements. Additionally, the model used to produce body-centric odometry from wheel speeds and yaw rates was presented. Lastly, details were provided on the process of feature extraction for producing landmark observations from raw point cloud data.

This chapter also included a characterization of the errors associated with the generated measurements. For each measurement type, a theoretical model is used to produce the measurement covariance that weights the measurements within the graph-based path estimator. The validity of these error models was shown by comparing the reported covariance to the sample error distributions. This understanding of the measurement's characteristics and error models allows for the measurements to be simulated in a way that emulates experimental data. By emulating experimental data, the findings of a simulation study will be applicable

to real-world applications. The details of this simulation study and the associated results are given next in Chapter 5.

# Chapter 5

# Simulation Study

A simulation environment is used to conduct a statistical evaluation of the proposed path estimation approach. The evaluation is performed by comparing the performance of the full graph-based path estimator as presented in Chapter 3 to three other solution types. In total, four different path solution types are used for comparison:

- "Single RPV Calculation" - Path generation from a single inter-vehicle RPV and following vehicle odometry as described in Section 2.4 (relative path generation).
- "GPS-Only Fusion" - Graph-based path estimation using body-centric odometry and differential GPS observations (i.e., inter-vehicle RPV and odometry of both vehicles).
- "Landmark-Only Fusion" - Graph-based path estimation using body-centric odometry of both vehicles and exchanged landmark observations (GPS denied).
- "Full Fusion" - Graph-based path estimation using all available observation sets.

Each solution type uses a different set of observations. This is summarized in tabular form in Table 5.1.

The path variables described in Section 2.2 are calculated for each solution and used for evaluation. This includes, primarily, the lateral path position and path yaw. Both of these variables are influenced by the path's positional accuracy along with the accuracy in follower yaw estimate. Unless specified, the estimated yaw is used for path variable calculation. However, yaw is not observable in the case of Single RPV Calculation. For the results shown in which Single RPV Calculation is compared, the true value of follower yaw is used for all solution types.

Table 5.1: Measurement Sets by Path Solution Type

| | Body-Centric Odom (Follower) | Body-Centric Odom (Leader) | GPS Odom (Follower) | GPS Odom (Leader) | Landmark Obs. (Follower) | Landmark Obs. (Leader) | Inter-Vehicle RPV |
|---|---|---|---|---|---|---|---|
| Single RPV Calculation | | | ✓ | | | | ✓ |
| GPS-Only Fusion | ✓ | ✓ | ✓ | ✓ | | | ✓ |
| Landmark-Only Fusion | ✓ | ✓ | | | ✓ | ✓ | |
| Full Fusion | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

With simulated data, a scenario can be repeated any number of times, each with measurements corrupted by randomly sampled noise according to the respective error models. The expected performance of an estimator can then be assessed using statistical properties such as the mean and Root Mean Square (RMS) of the empirical errors. For an unbiased estimate with proper covariance estimation, the RMS of the empirical errors should match the reported standard deviation taken as the square root of the estimated state variance [53]. This method is known as a Monte Carlo evaluation and is used to assess the different path solutions.

## 5.1 Simulation Environment

The simulation environment was developed and implemented using MATLAB R2018a. The user is able to adjust the parameters for testing under a variety of conditions. These parameters include trajectory type, speed, following distance, landmark density, and noise

parameters. The simulation environment also allows for GPS outages to be simulated in specific regions along the trajectory.

From the specified parameters, a trajectory is generated for each vehicle and landmarks are placed along the path. The measurements are calculated from the trajectory and landmark positions and random noise is added according to their respective error models described in the previous chapter. The simulation is purely kinematic, so effects of wheel slip and other unmodeled errors are not present. The simulation environment is described in further detail in the following sections.

### 5.1.1 Trajectory Generation

The trajectories are generated using piece-wise sections of straight-aways and constant radius turns to emulate a built highway environment. Trajectory generation is randomized to allow for a variation of the system geometry. An example trajectory is provided in Figure 5.1 where the leader, follower, and landmark positions are shown from a bird's eye view. The details for how the trajectories are generated is given in the following paragraphs.

The section type (straight-away or turn) is randomly assigned with a 50 % probability for each type. Straight-away sections vary in length from 100 m to 500 m with the length being sampled from a uniform distribution within this range. The radius of curvature for the turn sections is sampled from a uniform distribution of possible highway curvatures according to the Federal Highway Administration (FHWA) U.S. Department of Transportation (DOT) design standards [77]. The length of the turns are assigned according to the arc angle which is sampled uniformly from 10° to 45° for each turn. The direction of the turn, left or right, is randomly assigned with a 50 % probability.

The trajectories are simulated with a kinematic motion model with inputs of speed and steer angle in order to achieve the generated curvature profile. The kinematic model assumes zero sideslip such that the vehicle heading is tangential to the trajectory. A constant speed

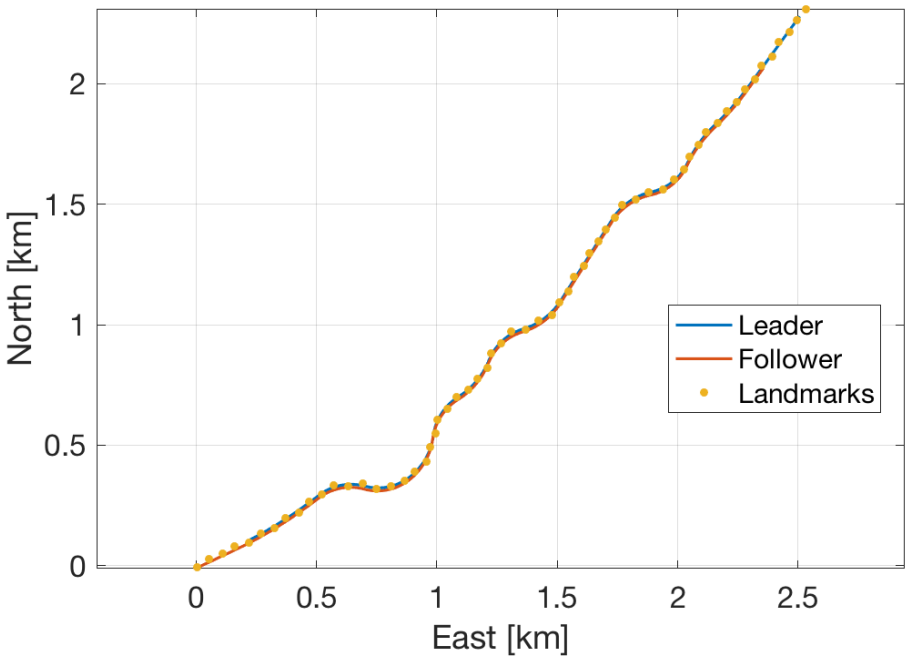


Figure 5.1: Example of a Simulated Randomized Trajectory

of 20 m/s (~45 mph) is used for the simulations. The result is a trajectory consisting of 2D positions and headings over time.

The trajectory is assigned to the lead vehicle and the following vehicle is simulated along the same path at a specified following distance and lateral path offset. Landmarks are placed alongside the trajectory at 10 m to either side of the route and evenly spaced according to the landmark density. Unless specified, a landmark density of 20 landmarks per kilometer is used which results in a spacing of 50 m between landmarks. The landmark orientation is assigned randomly according to a Gaussian distribution with zero mean and a standard deviation of 10°.

### 5.1.2 Measurement Simulation

The simulated trajectory provides the true position and orientation of the leader, follower, and all landmarks in the scene. From this, the measurements are simulated with additive noise according to their respective measurement models presented in Chapter 3.

The inter-vehicle RPV measurements are calculated from differencing leader and follower positions in the navigation frame for a given time epoch. The GPS odometry measurements are calculated from differencing vehicle positions at consecutive time epochs for both the leader and follower. Body-centric odometry measurements are calculated by rotating the navigation frame odometry into the body frame plus a change in yaw from differencing consecutive vehicle heading values. The landmark observations consist of the RPV from vehicle to landmark, resolved in the body frame, plus the relative orientation calculated from differencing the landmark and vehicle headings.

The noise acting on all measurements is stationary except for the landmark observations which are influenced by geometry. To emulate the real-world errors impacting landmark observations, the noise varies with the position and orientation of a sign relative to the vehicle. With lidar observations, the error is influenced by the number of points returned from a given landmark. The hypothetical number of points is a function of the lidar's azimuth step size (0.2°), sign width (4 ft), and relative pose in the lidar frame. The sign geometry and number of points are processed through the error model described previously in Section 4.4.4 with a point range standard deviation of 0.75 cm. A maximum range of 120 m is imposed on landmark observations which roughly corresponds to 3 lidar points for a nominal sign orientation. Additionally, the field-of-view is limited to ± 90° to emulate a lidar mounted on a vehicle's front bumper. A summary of the noise values and parameters used in the simulation is given in Table 5.2. These values were chosen based on the characterization of experimental data as described in Chapter 4.

## 5.2 Monte Carlo Evaluation

The results of a Monte Carlo evaluation are provided here for verifying the outputs of the estimator and comparing performance between different path solution types. In this evaluation, a single, randomized trajectory with a following distance of 250 m is used for all Monte Carlo trials. The Monte Carlo loop is iterated for a total of 2500 trials.

Table 5.2: Simulation Parameters

| Parameter | Symbol | Value |
|---|---|---|
| Body-Centric Odometry Longitudinal Measurement Noise | $\sigma_{odom,x}$ | 2 cm |
| Body-Centric Odometry Lateral Measurement Noise | $\sigma_{odom,y}$ | 2.5 cm |
| Body-Centric Odometry Relative Yaw Measurement Noise | $\sigma_{odom,\psi}$ | 0.02° |
| GPS Odometry North/East Measurement Noise | $\sigma_{\Delta}$ | 0.76 cm |
| Inter-Vehicle RPV North/East Measurement Noise | $\sigma_{rpv}$ | 1.15 cm |
| Lidar Point Range Measurement Noise | $\sigma_{\rho}$ | 0.75 cm |
| Lidar Azimuth Step Size | $\Delta_{\alpha}$ | 0.2° |
| Landmark Density | $D_{\ell}$ | 20/km |

The number of Monte Carlo trials was chosen based on the 5 % confidence intervals of the test statistics. The largest of the 5 % confidence interval over all path position error statistics is less than 1.5 cm after 2500 trials. Additionally, 2500 trials results in a 5 % confidence interval of less than 2 mm for the lateral position error statistic. This level of the confidence intervals allows for a statistically significant comparison to be made between the different solution types [53].

Within each Monte Carlo iteration, true sensor measurements are corrupted by randomly sampled noise according to the error models. The simulated measurements are used for path estimation and the resulting errors are recorded. By repeating the simulation for each Monte Carlo trial, an error distribution is provided from which statistics can be calculated empirically. For an unbiased estimator, the RMS Error taken across all Monte Carlo trials should match the standard deviation resulting from the estimator's reported covariance.

The plot included in Figure 5.2 shows the RMS error along with the reported standard deviation for the path position estimates. These errors are shown for each of the four path solution types listed previously. In each case, the RMS Error and standard deviation are approximately the same, which gives validity to the estimator's reported uncertainty and unbiased estimation errors.

The errors resulting from Single RPV Calculation, shown in blue in Figure 5.2, are similar to those shown previously in Chapter 3. In this case, the error is smallest at the

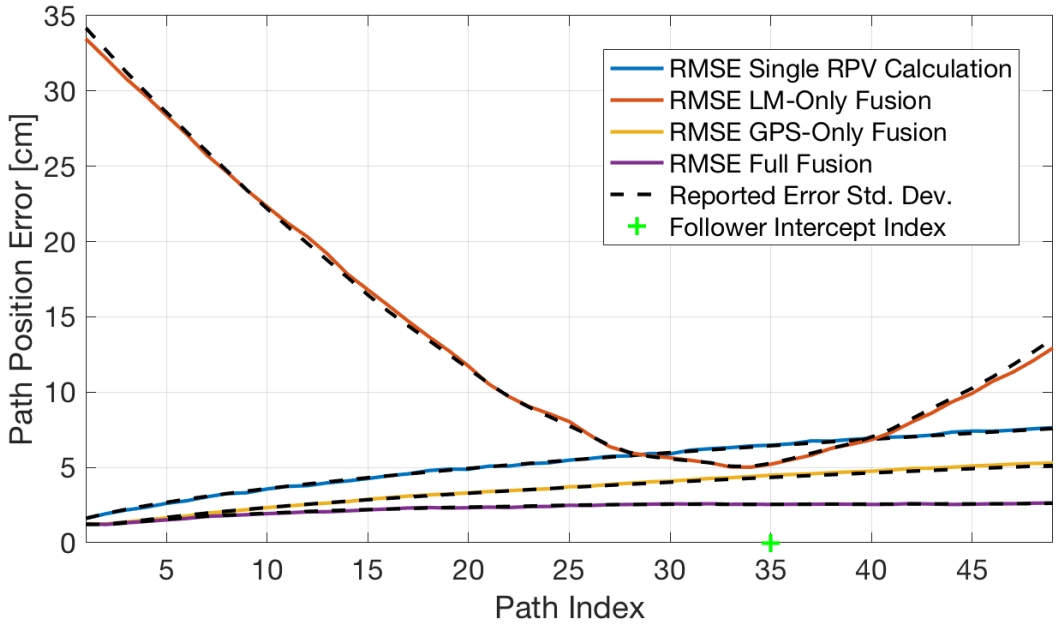


Figure 5.2: Monte Carlo Evaluation: Path Waypoint Position Errors

last waypoint (path index = 1) since this value is only influenced by the inter-vehicle RPV accuracy. Each preceding waypoint is the result of a cumulative sum of odometry measurements. The variance for each waypoint is linear with path index which causes the standard deviation to take the square root shape as shown in the figure.

The errors resulting from GPS-Only Fusion are shown in yellow in Figure 5.2. The GPS-Only Fusion result takes on a similar shape to that of Single RPV Calculation, but the error is reduced for all waypoints in the path. The largest improvement from one case to the next is present at the last waypoint (path index = $N = 49$), where the error is ~2/3 that of the Single RPV Calculation result.

For the case of Landmark-Only Fusion, shown in red in Figure 5.2, the error takes on an upwardly convex shape when plotted against path index. The smallest errors for Landmark-Only Fusion occur at a path index of 34, near the follower intercept index ($i_{int} = 35$). This is due to the waypoint's vicinity to the following vehicle's latest position (anchor node) from which the relative position errors are calculated. In graph-theoretic terms, the position error

increases with the number of edges between the anchor node ($f_1$) and a given waypoint. Since the waypoint node at path index = 1 ($\ell_1$) has the most edges between it and $f_1$ for Landmark-Only Fusion, this waypoint also has the highest errors. Conversely, all other solution types have only one edge separating $f_1$ and $\ell_1$ (the inter-vehicle RPV) and so the error is the lowest at path index of 1 for these solution types.

The position error results for the Full Fusion case, which incorporates all available measurements, is shown in purple in Figure 5.2. This result demonstrates a complementary fusion when compared to all other solution types. The position errors for Full Fusion are below that of any other solution type across all path indices.

## 5.3 Error as a Function of Following Distance

The previous section considered a single following distance for characterizing and comparing the performance of the different path estimation approaches. In this section, the error growth as a function of following distance is studied. The key objective of this section is to show how the error growth characteristics of the proposed algorithm are in fulfillment of the objectives stated in Section 1.3. Specifically, the simulation study will show that path estimation errors are bounded with respect to following distance when landmark observations are incorporated.

This test consists of two nested loops: the outer loop sweeps through a number of following distances while the inner loop simulates the measurements used for path estimation. On each iteration, the errors in the lateral path position and path yaw estimates are recorded. The RMS of the errors is then taken across all iterations of the inner loop and recorded for each following distance.

The results are shown in Figure 5.3 where lateral path position errors are plotted against following distance. As expected, the errors for the Single RPV Calculation grow unbounded with respect to following distance and exceed the accuracy target of 7 cm for following distances of 1 km or greater. The GPS-Only Fusion result shows a similar trend, but with a

reduced error growth rate and exceeds the accuracy target for following distances of 2 km or greater.

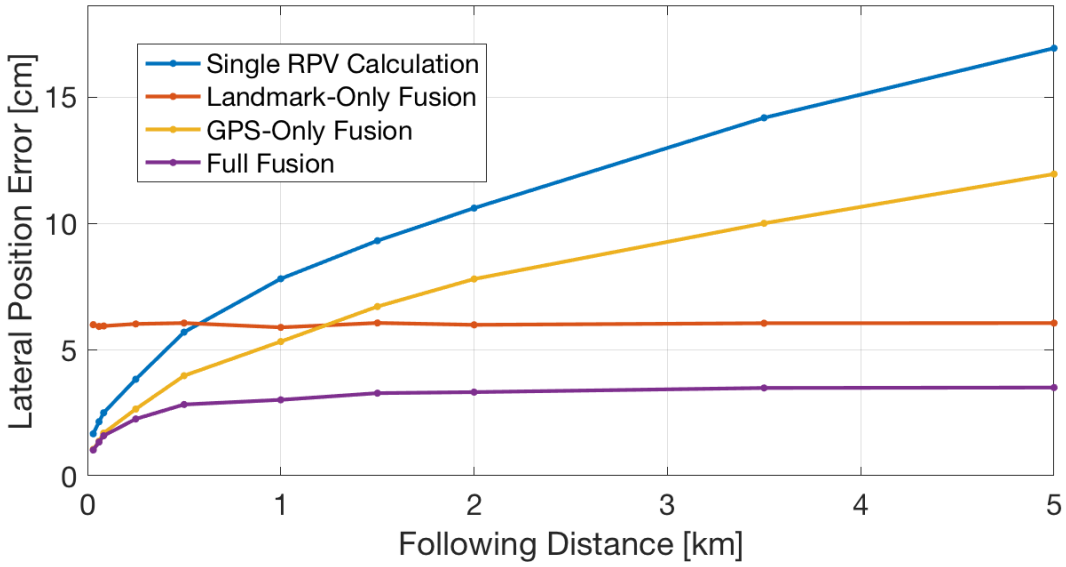


Figure 5.3: Lateral Position Errors vs. Following Distance

The errors associated with Landmark-Only Fusion in Figure 5.3 are effectively flat over all following distances with an average error of 6.01 cm. The Landmark-Only Fusion solution is not subject to accumulating odometry errors and, therefore, there is no relation between the error and the following distance. At low following distances, the errors from Landmark-Only Fusion are relatively high compared to all other solution types. However, as following distance increases, Landmark-Only Fusion solution type eventually out performs the Single RPV Calculation and GPS-Only Fusion solution.

In Figure 5.3, the Full Fusion solution type shows results that are a complementary blend of the GPS-Only Fusion and Landmark-Only Fusion result. For all following distances, the Full Fusion errors are lower than that of any other solution type. At low following distances, the inter-vehicle RPV and odometry are effectively weighted higher than the landmark observations and their contributions have a great deal of influence on the path

estimate. Conversely, as following distance increases, the landmark observations are weighted higher and the error asymptotically approaches the errors from Landmark-Only Fusion.

As for path yaw errors, each of the fusion-based solutions showed path yaw errors that were effectively flat over all following distances. The resulting path yaw RMS errors for Landmark-Only Fusion, GPS-Only Fusion, and Full Fusion are 0.20°, 2.01°, and 0.11°, respectively. Path yaw errors are influenced by two factors: the follower yaw uncertainty and the uncertainty of the path tangent angle. Of these two factors, the main contributor to path yaw error is the uncertainty in the follower yaw estimate. The errors contributed from path tangent angles (taken from arc tangent of path positions) are an order of magnitude lower than those from follower yaw. Errors in the follower yaw estimate are not influenced by following distance as long as GPS odometry or landmark observations are in the follower's vicinity. From the examples presented in Section 3.8.1, it was shown that two vehicle nodes connected by both GPS odometry and body-centric odometry edges will have yaw observability. As opposed to the position states, yaw observability is gained from nodes and edges in the vicinity of the follower node. Because of this, follower yaw errors do not compound with increasing path size and the path yaw estimate is not significantly influenced by following distance.

**Influence of Measurement Accuracy on Error as a Function of Following Distance**

The shape of Figure 5.3 is dictated by the accuracy of the measurement inputs. To demonstrate how the error as a function of following distance is influenced by measurement accuracy, the results from three additional analyses are provided. The three analyses include simulated path estimation results with a) increased inter-vehicle RPV measurement uncertainty, b) increased GPS odometry measurement uncertainty, and c) increased landmark observation uncertainty. These three variations were chosen because they best capture the

path estimation error trends for varying levels of measurement accuracy. For these simulations, the noise level used to simulate the measurements is increased and the corresponding measurement covariance used in the graph-based estimator is adjusted accordingly.

### Increased GPS Odometry Measurement Uncertainty

In this analysis, the uncertainty of GPS odometry is increased by doubling the measurement standard deviation ($\sigma_\Delta$) from 0.76 cm to 1.52 cm. The GPS odometry measurement is only used in three of the four solution types, so the Landmark-Only Fusion errors will be unchanged from those shown in Figure 5.3. The RMS lateral path position errors from this analysis for the four different path solution types are plotted against following distance in Figure 5.4.

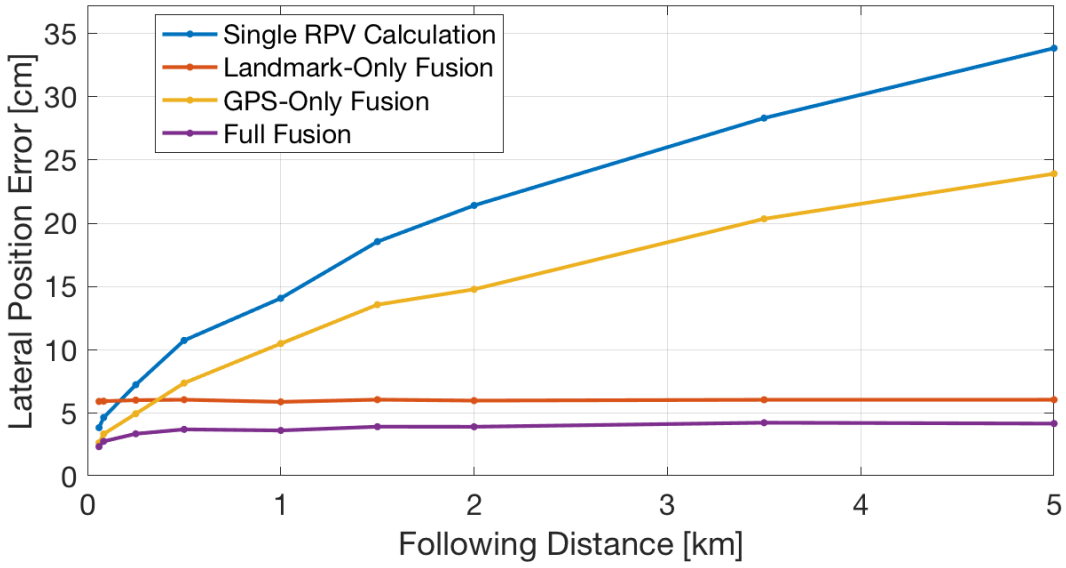


Figure 5.4: Lateral Position Error vs. Following Distance With Degraded GPS Odometry

As expected, the Single RPV Calculation and GPS-Only Fusion solution types show an increased error growth rate with following distance when the accuracy of GPS odometry is reduced. This result matches the trend shown previously in Figure 3.6 where following distance is proportional to path index. The maximum error for Single RPV Calculation and

GPS-Only Fusion solution types are at the largest following distance of 5 km. Compared to the analysis with nominal noise values, the maximum error for Single RPV Calculation and GPS-Only Fusion increased from 16.94 cm and 11.97 cm to 33.83 cm and 23.92 cm, respectively.

The Full Fusion solution type is influenced by the decreased accuracy of GPS odometry measurements, but not to a detrimental degree. The Full Fusion errors are still bounded by the errors of Landmark-Only Fusion, which are not influenced by GPS odometry. In the case of decreased GPS odometry, the Full Fusion errors approach the Landmark-Only Fusion bound more quickly with following distance. Stated another way, the Full Fusion solution type will rely more heavily on landmark observations at lower following distances when the uncertainty of odometry measurements is increased.

**Increased Inter-Vehicle RPV Measurement Uncertainty**

For this analysis, the inter-vehicle RPV accuracy is degraded by increasing the measurement standard deviation ($\sigma_{rpv}$) from 1.15 cm to 5.75 cm. In the same way, the inter-vehicle RPV measurement is only used in three of the four solution types, so the Landmark-Only Fusion errors will be unchanged from those shown in Figure 5.3. The RMS lateral path position errors from this analysis for the four different path solution types are plotted against following distance in Figure 5.5.

In this analysis, the error growth rates with following distance from Single RPV Calculation and GPS-Only Fusion solution types are not influenced by the decrease in inter-vehicle RPV measurement accuracy. Instead, the errors with respect to following distance are shifted upward for these two solution types. Since the inter-vehicle RPV errors do not accumulate along the path, a decrease in their measurement accuracy will impact all path waypoint indices equally. Changes in the inter-vehicle RPV accuracy will effectively shift the $y$-intercept of the error vs. following distance plot while keeping the same shape and error growth rate.

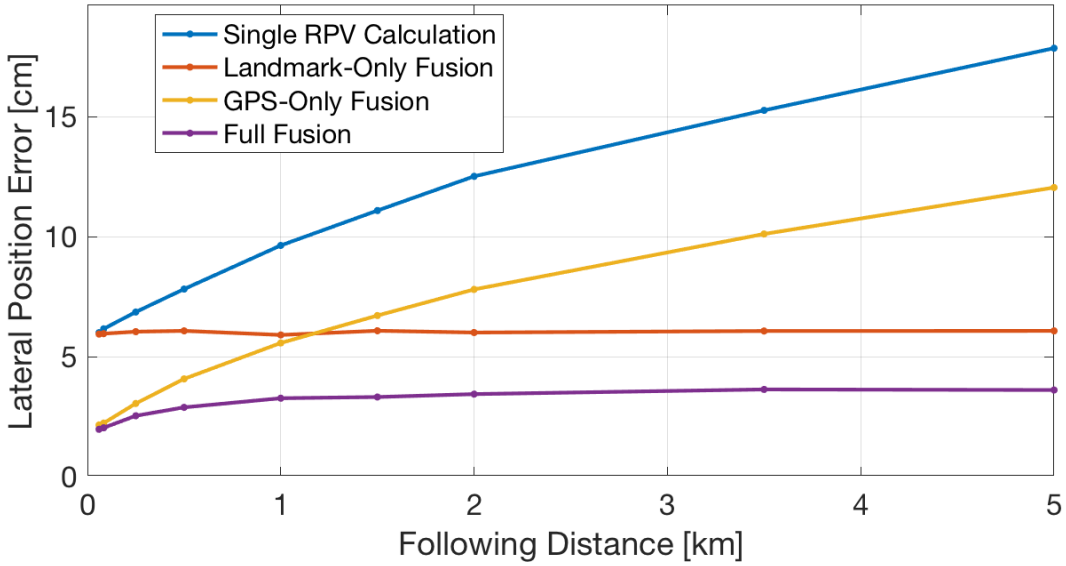


Figure 5.5: Lateral Position Error vs. Following Distance With Degraded Inter-Vehicle RPV

Interestingly, the $y$-intercept of the GPS-Only Fusion solution type in Figure 5.5 does not shift as much as that of Single RPV Calculation due to the increased uncertainty of inter-vehicle RPV measurements. As stated previously in Section 5.3, the $y$-intercept of the Single RPV Calculation solution type is equal to the inter-vehicle RPV accuracy (5.75 cm for this analysis). Comparatively, the GPS-Only Fusion is a graph-based solution and the connectivity with other graph nodes allows for the odometry to improve the errors even at low following distances. In the same way, the Full Fusion solution type is also improved by odometry accuracy at lower following distances. As with the previous analyses, the Full Fusion errors are bounded by all other solution types and is bounded with respect to following distance.

### Increased Landmark Observation Uncertainty

Finally, the analysis in this section studies the impact of increased landmark observation uncertainty on the lateral path errors as a function of following distance. There are three main parameters that influence the landmark observation's impact on path errors: the

landmark measurement accuracy and the number of landmarks (landmark density). Higher landmark density results in more landmark observations, which decreases the path errors on average.

For lidar landmark observations as described in Section 4.4, the measurement accuracy is influenced by the lidar's azimuth step size ($\Delta_\alpha$) and point range standard deviation ($\sigma_\rho$). The point range standard deviation impacts the magnitude of the landmark observation errors. The lidar's azimuth resolution not only impacts the magnitude of the landmark observation errors, but the number of points returned by a sign landmark. Since the sign landmarks are conditioned by a minimum number of points of 5 (as described in Section 4.4.4), the azimuth resolution also impacts the effective maximum perception range. By increasing the azimuth step size used in the simulation, the average number of landmarks will decrease, resulting in increased uncertainty in the path solution.

For this analysis, the landmark measurement accuracy is decreased by adjusting the parameters used in the lidar landmark simulation model described in Section 5.1.2. Specifically, the azimuth step size ($\Delta_\alpha$) and point range standard deviation ($\sigma_\rho$) are increased from 0.2° and 0.75 cm to 0.3° and 7.5 cm, respectively. The landmark density ($D_\ell$) is kept at 20/km. The RMS lateral path position errors from this analysis for the four different path solution types are plotted against following distance in Figure 5.6.

It can be seen in Figure 5.6 that the landmark observation accuracy changes the level of the Landmark-Only Fusion errors. The Landmark-Only Fusion solution type for this analysis exhibits an average error of 13.25 cm as compared to the 6.01 cm resulting from the analysis performed with nominal noise values. As with all other analyses of lateral position error as a function of following distance, the Full Fusion solution type remains bounded by all other solution types. Because the Landmark-Only Fusion errors are above the 7 cm accuracy target in this case, the Full Fusion solution cannot be guaranteed to be within tolerance for all following distances. However, because the Full Fusion solution is greatly improved by the inter-vehicle RPV and GPS odometry measurements at lower following distances, the Full

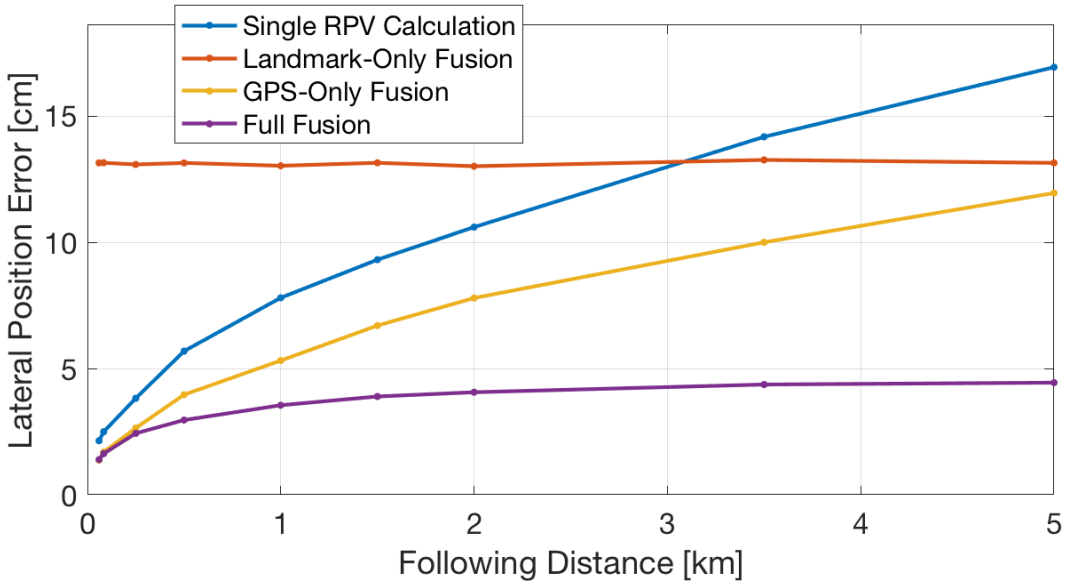


Figure 5.6: Lateral Position Error vs. Following Distance With Degraded Landmarks

Fusion solution remains within tolerance for all following distances used in this analysis (up to 5 km).

## 5.4 Availability in the Presence of GPS Outages

Another advantage to using landmark observations in the path estimator is the increased availability in the presence of GPS outages. In most applications, landmark and GPS observations have complementary availability. Environments with poor GPS reception are likely to be feature rich and provide ample landmarks to use for navigation (e.g., forests and urban environments). Conversely, the sparse environments with little to no landmarks will likely be benign environments for GPS (e.g., deserts and open plains). This section focuses on the path estimation performance through GPS outages and the benefits that landmark observations provide.

To demonstrate the effect of GPS outages on path estimation, a simulation was performed with a regionally-defined GPS outage. The GPS outage region was set to a size of 60 by 60 meters in which the inter-vehicle RPV and GPS odometry for a vehicle become

unavailable. The GPS outage region and the impact on graph topology can be seen in Figure 5.7 which shows the Relative Position Graph in cartesian space along with the GPS outage region.

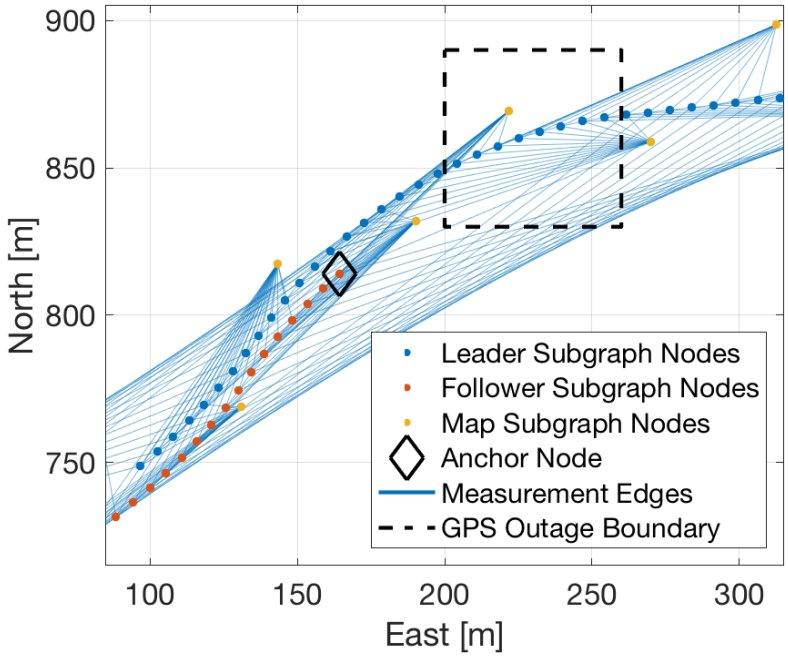


Figure 5.7: Cartesian Graph View with GPS Outage

The results from a Monte Carlo evaluation of the GPS outage scenario are given in Figure 5.8. For this analysis, the Single RPV Calculation solution is unable to resolve the positions of the waypoints within the GPS outage which explains the missing data points in the plot. Conversely, the fusion-based path solutions, which use body-centric odometry, are able to estimate all path waypoints despite the GPS outage. The Landmark-Only Fusion errors are independent of GPS availability and are identical to the errors shown in Figure 5.2. The Full Fusion solution is also able to estimate the entire path while maintaining accuracies that exceed all other solutions. Interestingly, the accuracies of the Full Fusion solution also outperform the Landmark-Only Fusion during the GPS outage. This is due to the graph connectivity: the correlation of the state estimates is propagated through the landmark observations, even in sections where GPS is unavailable.

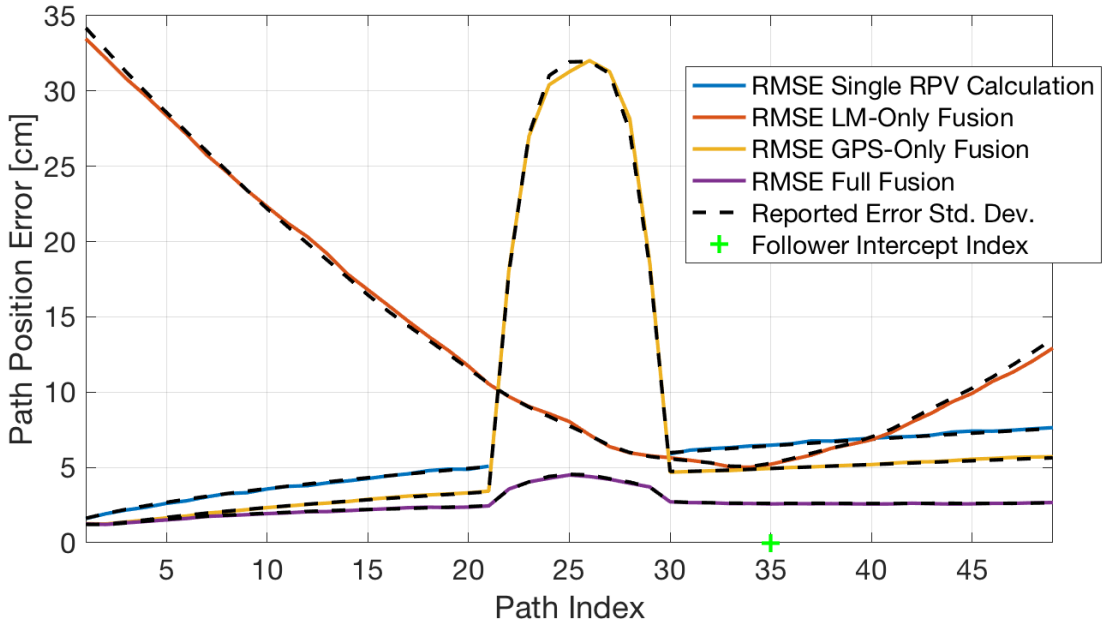


Figure 5.8: Monte Carlo Evaluation: Path Waypoint Position Errors in GPS Outage Scenario

## 5.5 Performance as a Function of Number of Landmarks

In this test, the number of landmarks is used as the test variable for evaluating path estimation performance. This test provides insight into what environments would be conducive to path estimation as implemented in this dissertation. For this test, only the Landmark-Only Fusion solution is evaluated. The results from the previous sections showed that the Full Fusion errors remain bounded by that of Landmark-Only Fusion. Therefore, the landmark only fusion can be considered as an upper bound on Full Fusion performance and the results in this test can be applied accordingly for any given following distance.

The number of landmarks in a test is controlled by a user-defined landmark density. The landmark density is the average number of landmarks per kilometer. For each landmark density, a Monte Carlo evaluation is performed on Landmark-Only Fusion path estimation and the resulting RMS error is recorded. The results can be seen in Figure 5.9 along with the accuracy targets of 7 cm and 0.25°.

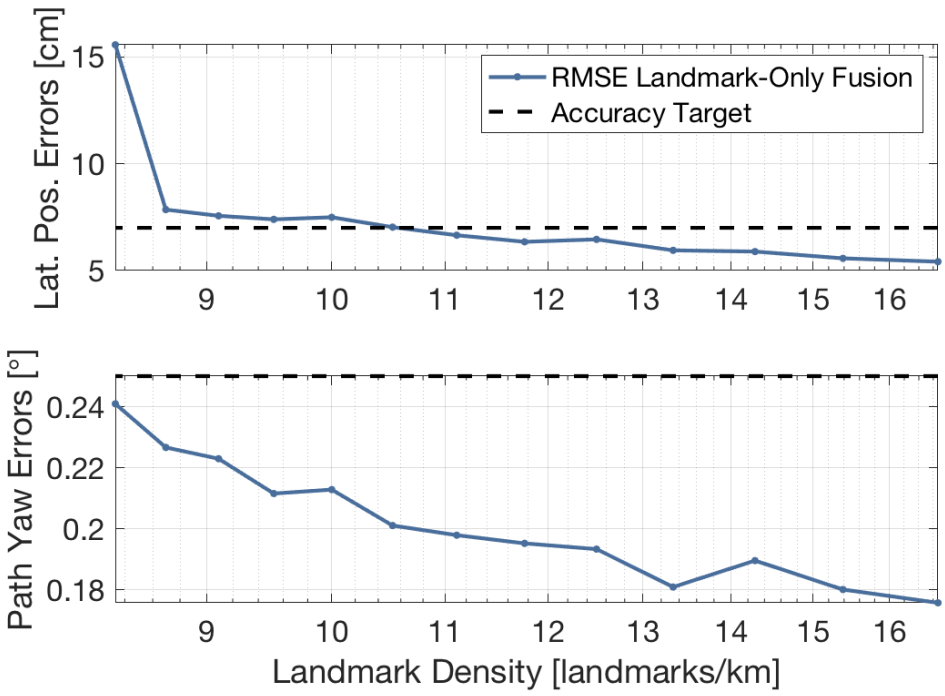


Figure 5.9: Landmark-Only Fusion Path Deviation Errors vs. Landmark Density

As expected, the error is lower for higher landmark densities. The accuracy target is achieved for all landmark densities greater than 11 landmarks per kilometer. At 11 landmarks/km, the average spacing between landmarks in the run is around 90 m. At this landmark spacing, with a simulated max perception range of 120 m, each vehicle observes (on average) 1 to 2 landmarks at any given time. This result can be related to the observability analysis presented in Section 3.8 which showed that two common landmarks allows for observability of the lead vehicle pose. At landmark densities less than 11 landmarks/km, the system is still observable, but will rely more heavily on body-centric odometry to span gaps of fewer than two common landmark observations.

Since the Landmark-Only Fusion result is taken as an upper bound on Full Fusion performance, conclusions can be drawn from the results of this test. This test result shows that sufficient path accuracies can be achieved with the Full Fusion path estimation over all following distances, regardless of GPS availability, in environments with at least 11 landmarks per kilometer. This conclusion can be used when evaluating certain environments for

long distance following using the Full Fusion path estimation scheme as described in this dissertation.

## 5.6 Conclusion

This chapter focused on the simulations used to evaluate the performance of the graph-based path estimator presented in Chapter 3. The simulation environment allows for control over certain conditions such as system geometry, error characteristics, and measurement availability. Measurements were simulated in this environment using noise levels as characterized in Chapter 4.

The simulation environment was used to evaluate path estimation performance using a Monte Carlo evaluation. The Monte Carlo evaluation in Section 5.2 demonstrated the efficiency of the graph-based path estimator's reported covariance through comparison with empirical RMS errors. These results also showed the complementary nature of fusing landmark observations with the differential GPS observations.

Next, results were presented from a test that evaluated the key contribution of this dissertation: bounded errors with respect to following distance. The errors of the Full Fusion solution were lower than all other solutions for all following distances up to 5 km. Given that the Full Fusion errors were bounded by all other solutions and the fact that the Landmark-Only Fusion errors were unchanged by following distance, the Full Fusion solution was shown to be in fulfillment of the bounded error design requirement.

Section 5.4 demonstrated the secondary contribution of this dissertation: improved availability in the presence of GPS outages. A Monte Carlo evaluation was used to demonstrate this behavior. The results over all path indices showed how landmark and body-centric odometry observations helped span GPS outages and improve path availability.

Lastly, an analysis of the performance as a function of landmark density was provided to relate the estimator's performance to this environmental condition. This test concluded

that an environment with at least 11 landmarks per kilometer (on average) allows for target accuracies to be achieved.

The findings in this chapter underscore the key contributions and show the statistical validity of the estimator given perfectly modeled errors. To underscore these findings, the path estimator should be demonstrated in the presence of real-world conditions. For this, an analysis is performed using experimental data and the results are tied to those shown in this chapter. This experimental validation is provided next in Chapter 6.

# Chapter 6

# Experimental Validation

Experimental data was collected and post-processed to validate the findings of the simulation study. The key objective of this chapter is to show how the path estimation algorithms perform with real-world data and to tie these results to those provided in the simulation study. As opposed to the Monte Carlo simulation study presented in Chapter 5, the experimental results are not meant to provide statistical significance. Since only a finite amount of experimental data is available, their results are anecdotal and only depict the performance for the scenarios tested. By linking the experimental data to the simulation environment, the expounded results of the simulation study are grounded to real-world application.

This chapter starts in Section 6.1 by describing the experimental setup used to collect data. Next, in Section 6.2, experimental results are shown from data collected in two scenarios: one on a public highway and another on a closed-course, oval track. The resulting path estimates and estimation errors from the different solution types are characterized. Additionally, results are provided for a test that emulates a following distance by shifting the timestamps of the following vehicle's data.

## 6.1 Experimental Setup

The motivation for the presented work is to allow Level 2 autonomy in Class 8 tractor-trailer convoy networks [78]. Two 2015 Peterbilt 579's, shown in Figure 6.1, were used as the test platforms with one serving as the lead and the other as a follower vehicle. For the data shown in this chapter, each truck towed an empty 53 foot box trailer and the cabs were outfitted with the sensor components characterized in Chapter 4. Both trucks are capable

of full actuation of the steering and drivetrain systems, making them ideal candidates for automated convoying.


Figure 6.1: Tractor-Trailers on Logging Roads

The sensor suite is identical for both trucks in terms of the sensor models and mounting locations. The trucks are both outfitted with a Novatel FlexPak6 GNSS receiver which records raw GNSS observables (pseudorange, carrier phase, Doppler) from the antenna mounted above the cab. A Velodyne VLP-16 lidar is mounted along the centerline of the hood, just in front of the windshield, as shown in Figure 6.2. The lidars are disciplined with GPS time from an external GARMIN receiver, allowing for accurate time synchronization using time-of-validity stamps. Onboard vehicle measurements are received over the CAN bus via J1939 protocol, in particular, the wheel speeds used for calculating body-centric odometry. Additionally, a Memsense IMU (MS-IMU3020) is mounted near the cab's center-of-gravity and measures the yaw rates used for calculating body-centric odometry.


Figure 6.2: VLP-16 Lidar Mounted to Hood of Peterbilt Cab

The trucks are each outfitted with a Dedicated Short Range Communications (DSRC) radio allowing for networked communication between vehicles. In practice, the DSRC radios pass sensor information to following vehicles which then fuse the data for path estimation. For the results shown in this chapter, the data was collected for each vehicle individually and combined in post-process. In this way, the raw lidar data could be studied closely for developing the feature extraction algorithm. In the future, feature extraction would take place onboard each vehicle before passing the data as landmark observations over DSRC radio as illustrated previously in Figure 4.1.

The raw sensor data is processed for generating the measurements used in the graph-based relative path estimator presented in Chapter 4. The GPS observables are processed for inter-vehicle RPV and GPS odometry measurements using the DRTK and TDCP algorithms as described in Section 4.2. Yaw rate and wheel speed measurements are used

to generate body-centric odometry measurements using the model presented previously in Section 4.3. Additionally, the landmark observations are taken from lidar data using the feature extraction algorithm on dense point clouds as described in Section 4.4.

The ground truth is provided by an RTK positioning system with corrections from a nearby base station. The RTK positions are calculated using Novatel GrafNav software and are globally accurate to sub-centimeter level. A ground truth path is calculated by differencing RTK positions between the leader and follower vehicles.

## 6.2 Experimental Results

This section provides the results of path estimation using experimental data. The Full Fusion estimator is evaluated against the accuracy requirements detailed in Section 1.3. The results are also shown as a comparison of the four path solution types outlined in Table 5.1. Two data sets are used in the evaluation: one collected on a public highway and another on a closed-course track.

For both data sets, the following distance was controlled autonomously using the Co-operative Adaptive Cruise Control (CACC) system described in [79]. The steer angles were controlled manually by the operator of the following truck in order to trace an approximate path of the leader. By-wire control of the steering angle is possible for these trucks, but the results presented in this dissertation focus solely on estimation errors. When the ground truth reference is available, the errors in lateral path position and path yaw are used to assess the path estimation accuracy.

### 6.2.1 Highway Data Results

The first scenario takes place on Highway 280 in Lee County, Alabama. The highway data consists of speeds from 45 to 65 mph and a platoon following distance of 300 ft as regulated by the CACC system. This scenario is valuable since it involves challenges of operating in a real-world environment: realistic road sign geometry, shadowing and false

detections from neighboring traffic, variable speeds, degraded GPS due to overhead foliage, and more.

This section of Highway 280 provides a variation of landmark types that are detected in the lidar feature extraction process. This includes road signs, trees, power line poles, and support columns of roadside buildings. Some sections along the route provide ample number of landmarks while other sections are totally void of any landmarks, allowing the behavior to be evaluated for both cases. On average, the highway route contains approximately 18 landmarks per kilometer. A satellite view of the highway trajectory is shown in Figure 6.3 along with the pole and sign landmark positions.

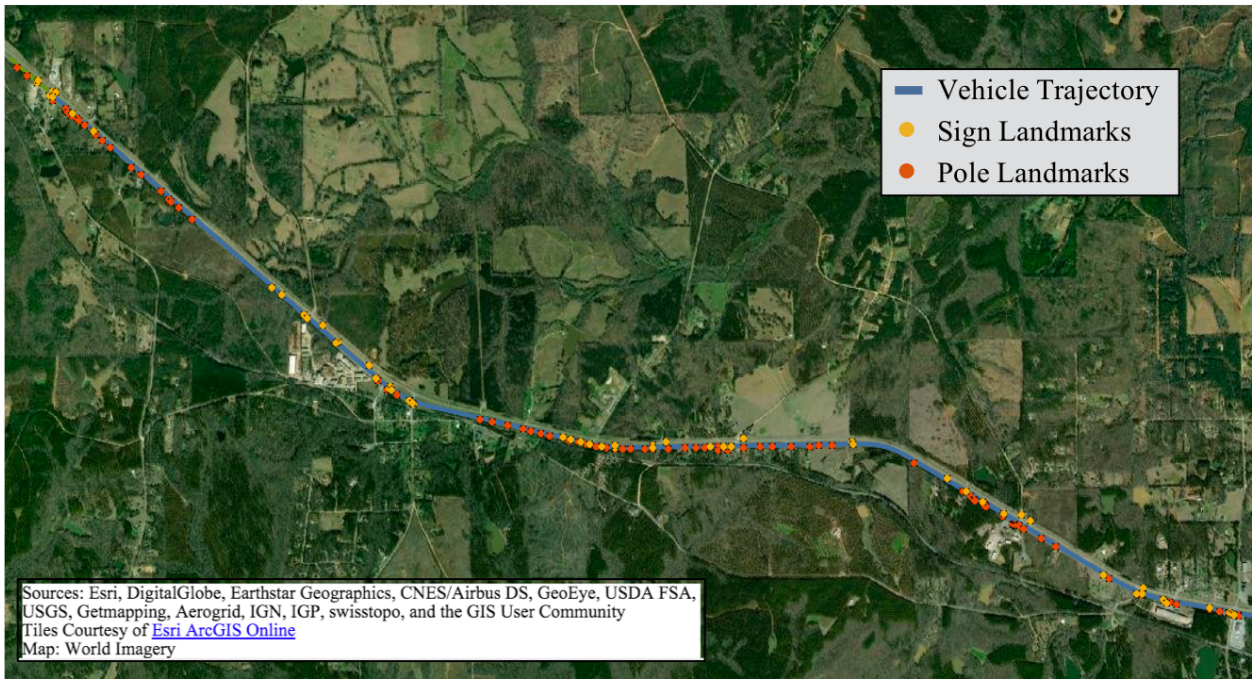


Figure 6.3: Highway Test: Satellite View of Trajectory and Landmark Positions

The availability of the critical observations varies over the run. In this context, the critical observations include the landmark and inter-vehicle RPV observations that relate the leader and follower poses. One of these two observation types is required for path observability. The availability of these observations is shown in Figure 6.4. The first subplot shows the number of landmarks that are common between the lead vehicle and the follower's

current position plotted against time. The second subplot shows the status of the DRTK fix plotted against time. When DRTK is not fixed, the inter-vehicle RPV is not included in path estimation.

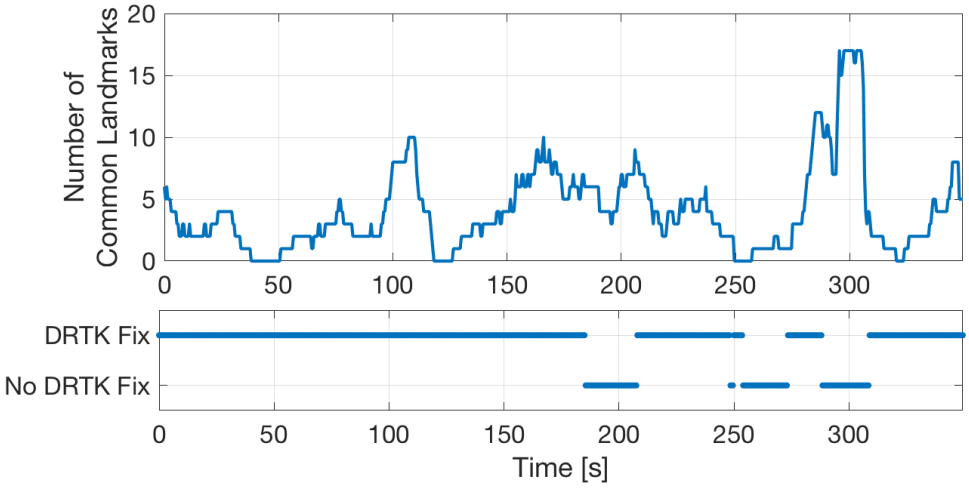


Figure 6.4: Highway Test: Landmark and DRTK Availability

The plots in Figure 6.4 show some interesting conditions for evaluating the estimator. First, there are two sections where no common landmarks are observed for a substantial amount of time. These two sections start at 38 and 118 seconds into the run and last 12.5 and 8 seconds, respectively. Another section of interest can be seen at 254 seconds into the run. In this region, there are no common landmarks and also no DRTK fix over a duration of 3.5 seconds. The path estimation performance through these sections of interest will be highlighted in the results that follow.

Ground truth is not available for this dataset; however, the results still provide insight on the estimator's performance in terms of availability and qualitative metrics. The estimated lateral path position is plotted vs. time as shown in Figure 6.5. The typical lane width of a U.S. highway is also included in the plot for reference of scale. It is worth noting that the values shown in Figure 6.5 are not estimation errors, but rather the estimated lateral path position as controlled by the human driver. This figure also shows the reported standard

deviation of the estimated lateral position, which stays below the accuracy target of 7 cm throughout the duration of the test.

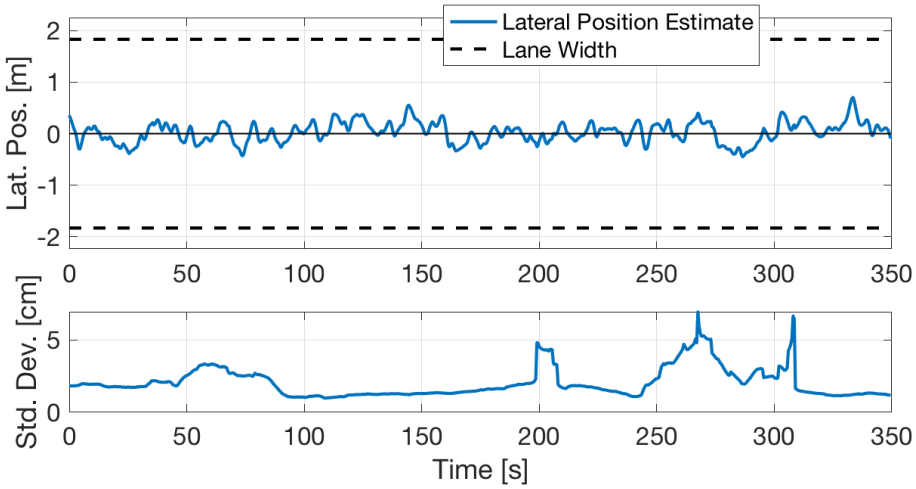


Figure 6.5: Highway Test: Full Fusion Lateral Path Position Estimate

The estimated value of path yaw is shown in Figure 6.6. In this figure, the path yaw estimate fluctuates about zero which is the result of the human driver tracking the lead vehicle's path. This figure also shows the standard deviation associated with the path yaw estimate, which stays below the accuracy target of 0.25° for a majority of the run.

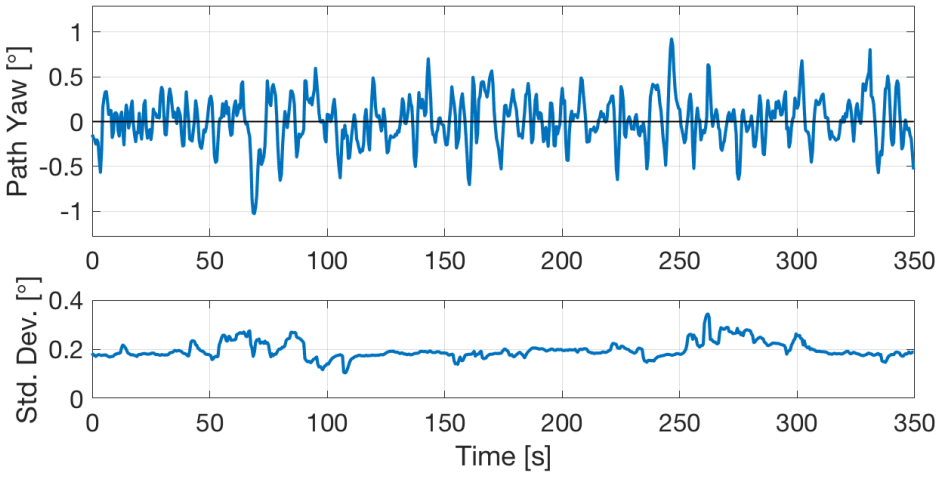


Figure 6.6: Highway Test: Full Fusion Path Yaw Estimate

A comparison of lateral position estimates for the different path solution types is given in Figure 6.7. This figure also shows the availability of lateral position estimates over time in the last subplot. Note that the colors used in the availability plot correspond with the rest of the figure to serve as a legend. The lateral position estimates shown in this figure use the true value of follower heading to rotate positions into the path tangent frame. The use of a common truth heading in lateral position calculation is to isolate the position performance from yaw estimation performance such that a comparison can be made with Single RPV Calculation, which does not have yaw observability.

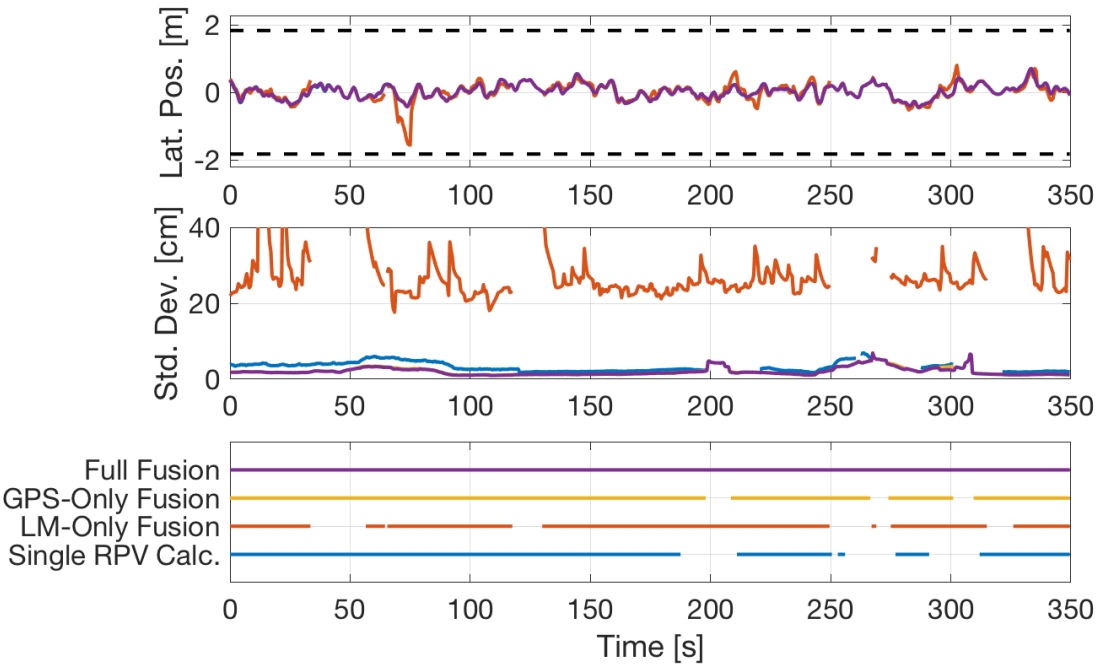


Figure 6.7: Highway Test: Lateral Path Position Estimates and Availability Comparison

The Full Fusion solution has full availability throughout the run despite the loss of both inter-vehicle RPV and landmarks at $t = 254s$. This is the result of a sufficiently large path tail length (250 m) that allows for dead reckoning of the path through landmark and DRTK outages. However, this path tail length is not large enough to allow for full availability of GPS-Only Fusion and Landmark-Only Fusion solutions. The Landmark-Only Fusion result eventually loses a path estimate due to the large gaps in landmarks present throughout the

run. The same can be said for the GPS-Only Fusion solution which loses a path estimate in three different instances throughout the run. The Single RPV Calculation has a strict dependence on GPS coverage and loses a path estimate wherever a DRTK fix is lost. The GPS-Only Fusion and Single RPV Calculation errors are virtually the same as Full Fusion when the solutions are available. This is a result of the accuracy of the differential GPS observations and their influence on the fused path estimate at the relatively low following distance of the run (~90 m).

In a similar way, a comparison of path yaw estimates for the different path solution types is given in Figure 6.8. For the Full Fusion solution type, the path yaw estimate is available throughout the run. Again, the GPS-Only Fusion and Landmark-Only Fusion solutions lose the path estimate in certain sections of GPS and landmarks outages, respectively. The Single RPV Calculation solution has no yaw observability and does not appear in this plot.

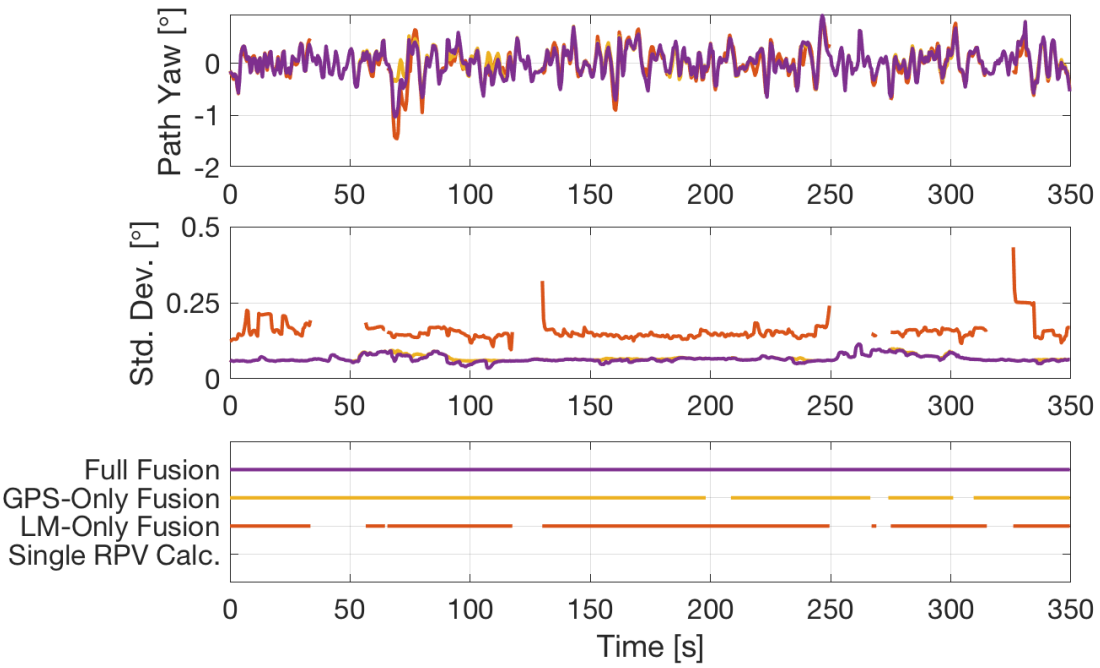


Figure 6.8: Highway Test: Path Yaw Estimates and Availability Comparison

### 6.2.2 Track Data Results

The next experimental scenario takes place on the test track at Auburn University's NCAT asphalt testing facility shown in Figure 6.9. As opposed to the highway scenario, the test track is a closed course, allowing for a more controlled test environment. The track is a 1.7 mile loop with straight sections of nearly half a mile and 8° banks on each of the two turns. In this scenario, the trucks travel at a near-constant 50 mph (22.4 m/s) and following distance of 200 ft (61 m). The trucks collected data for a total of 8 laps around the track allowing for roughly 18 minutes of data.


Figure 6.9: NCAT Track Test: Satellite View of Trajectory and Landmark Positions

The track is surrounded by a number of typical highway signs which can be seen in Figure 6.10. There are road signs every 200 ft along the track's outer perimeter, each 4 ft in width. Other signs include turn, yield, and stop signs which were all used as navigational landmarks when detected. Landmarks are detected at distances of up to 120 m (~400 ft). The landmark spacing and max detection distance allow for anywhere from 0 to 3 landmarks

to be observed at a given time. On average, the track provides close to 20 landmarks per kilometer.

Figure 6.10: Cab View of Lidar and Road Signs at NCAT Test Track

Ground truth data is available for this scenario and is used to calculate the path estimation errors for evaluation. Figure 6.11 shows the errors in lateral path position and the reported $3\sigma$ confidence level vs. time. The lateral position errors have a mean of 0.26 cm indicating that the estimate is not significantly biased. Additionally, the errors stay well beneath the accuracy target of 7 cm ($1\sigma$, $67^{th}$ percentile) with a maximum error of 2.68 cm and RMS error of 0.94 cm. The $3\sigma$ confidence level reported by the estimator mostly envelops the errors (save a few instances) which gives credence to the covariance reported by the graph-based estimator.

The errors in path yaw estimate are shown in the second subplot of Figure 6.11. The resulting path yaw errors are virtually unbiased with a mean of 0.01°. The path yaw errors have an RMS of 0.14° taken over the duration of the run which satisfies the accuracy requirement of 0.25° ($1\sigma$, $67^{th}$ percentile).

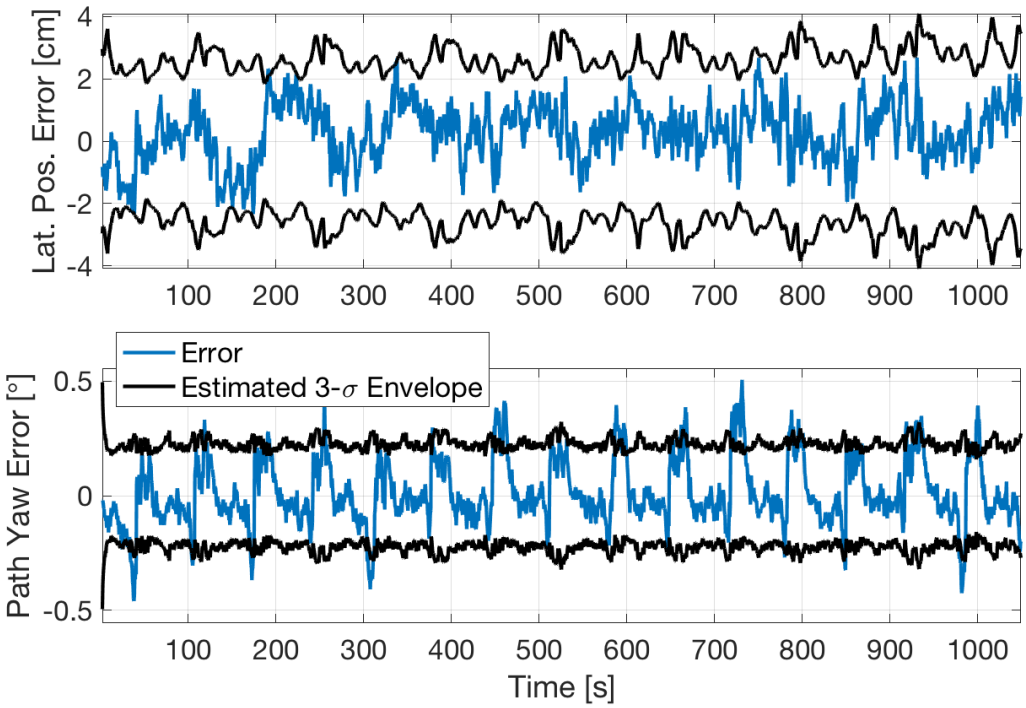


Figure 6.11: NCAT Track Test: Full Fusion Path Deviation Errors

Note that the path yaw errors in Figure 6.11 show a periodic trend with spikes occurring every 70 seconds and a maximum value of 0.51°. Periodic errors of this type suggest that certain observations violate the zero-mean Gaussian assumption. The periodicity in this plot aligns with the turn sections on the track. Given the correlation with the turns, the errors are likely caused by the unmodeled effects from the track's bank angle, which violate the 2D planar assumption. The body-centric odometry is the most susceptible to errors with a banked vehicle since the true yaw rate (rotation about the vertical axis) does not align with body yaw rate. The reported covariance during the turns is overly optimistic, as indicated by the spikes exceeding the $3\sigma$ confidence level. In practice, when the reported covariance is artificially low, an autonomous control system may not be able to raise alerts and the controller would be allowed to operate in potentially hazardous situations. Future work may consider addressing this issue by estimating vehicle roll and pitch angles online and compensating the body-centric odometry model accordingly.

Next, the lateral path position errors of the Full Fusion estimator are compared to the solution types listed in Table 5.1. Figure 6.12 shows the lateral position errors vs. time along with their associated standard deviation for all four solution types. The errors from Landmark-Only Fusion show the same periodic trend as seen in Figure 6.11, correlating with the banked turn sections. This periodic trend is not as apparent in the lateral position errors from Full Fusion but is very pronounced in the Landmark-Only Fusion result. This is a result of how heavily the differential GPS measurements are weighted in the graph-based estimation. The accuracy of the differential GPS measurements suppress the body-centric odometry contributions, including any unmodeled errors such as the breakdown of the 2D planar assumption during the banked turns.

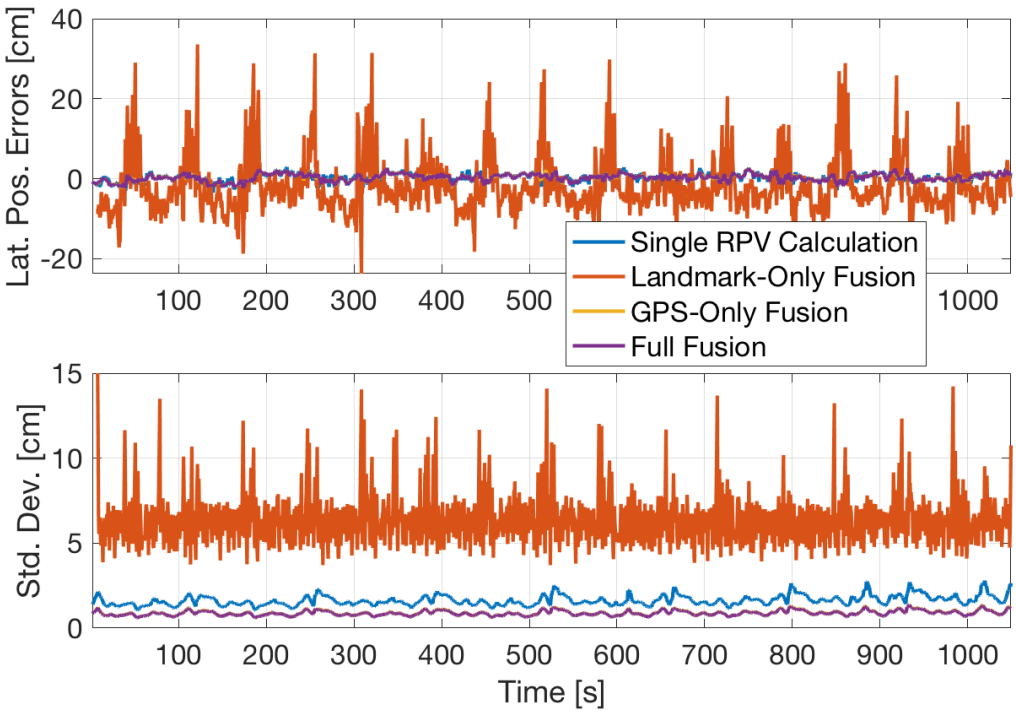


Figure 6.12: NCAT Track Test: Lateral Path Position Error Comparison

At the relatively low following distance of 200 ft (61 m), the lateral position errors from GPS-Only Fusion are effectively the same as Full Fusion. Both the errors and standard deviations of GPS-Only Fusion are behind the Full Fusion results and are not visible in

the plots in Figure 6.12. This similarity also speaks to how differential GPS measurements are weighted much higher compared to body-centric odometry and landmark observations. However, the differential GPS measurements carry less weight as the following distance increases, as demonstrated in Section 6.2.4.

In Figure 6.13, the path yaw errors are plotted against time along with their associated standard deviation. This plot only includes errors from the three fusion-based results since Single RPV Calculation does not provide yaw observability. As opposed to the results shown in Figure 6.12, the Full Fusion solution shows noticeable improvement over GPS-Only Fusion. This demonstrates that the accuracy of differential GPS measurements do not correlate as highly to the yaw estimates as they do for lateral position. The graph-based estimator appropriately weights the observations, resulting in a complementary fusion of the different sensor information.

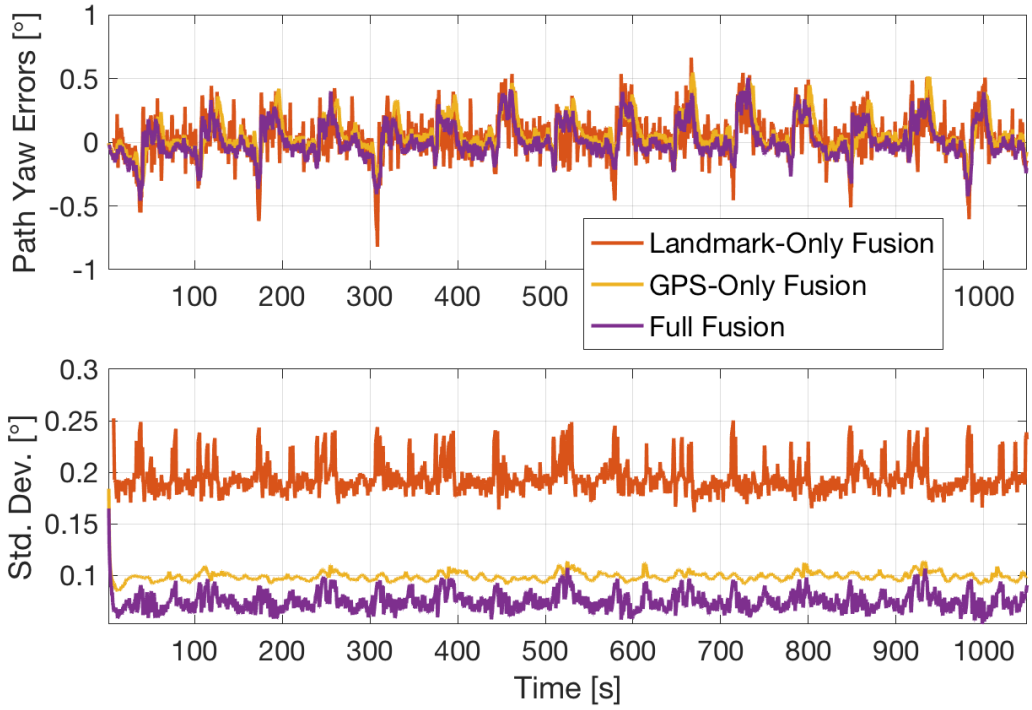


Figure 6.13: NCAT Track Test: Path Yaw Error Comparison

### 6.2.3 GPS Outage Test Results

Next, the track data is used to evaluate path deviation estimation errors through a GPS outage. Similar to the test conducted in Section 5.4, a GPS outage region is specified and any DRTK and TDCP measurements that are recorded while within the region are not used in path estimation. The outage region was chosen to be 150 m in length and a path tail length of 100 m was used in the path estimator. The path tail length was set slightly less than the size of the outage in order to demonstrate the influence that path tail length has on availability. A bird's eye view of the track trajectory, landmark positions, and GPS outage region is shown in Figure 6.14.

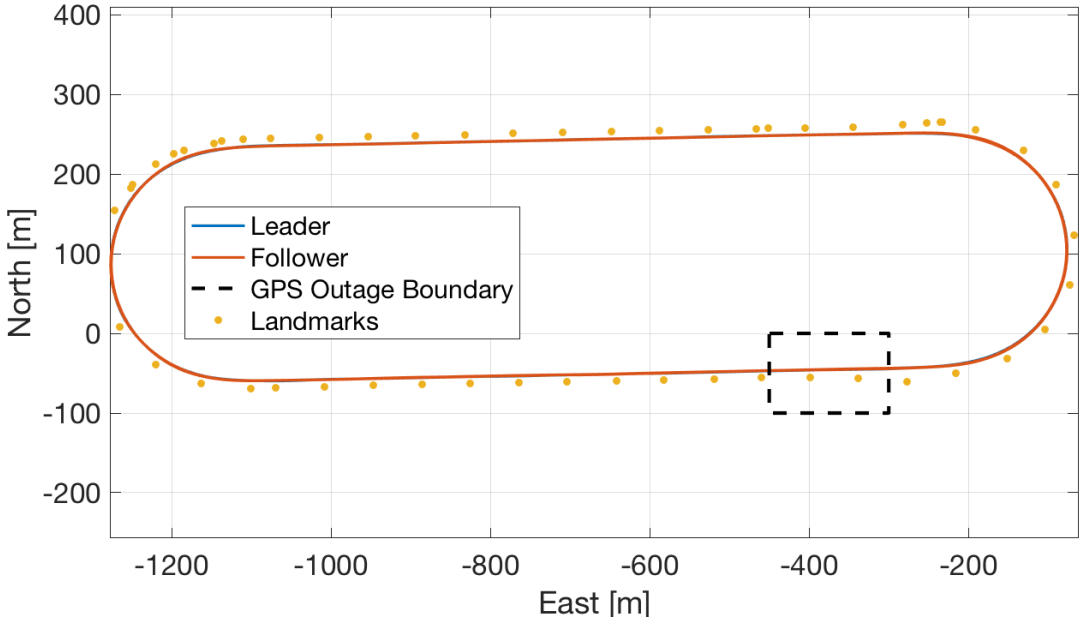


Figure 6.14: GPS Outage Test: Trajectory and GPS Outage Region

The track data allows the performance through the outage region to be studied over multiple, repeated passes. The errors from all the passes are plotted against the East position in order to overlap the results from all the passes. A comparison is made between the path solutions outlined in Table 5.1 in terms of the path deviation errors (lateral position and path yaw), standard deviations, and availability.

Figure 6.15 shows the lateral path position results for the GPS outage test. The Full Fusion solution shows the lowest amount of error throughout the run. During the GPS outage, the Full Fusion errors slowly increase to match the magnitude of the Landmark-Only Fusion errors. The errors of the Landmark-Only Fusion solution are not impacted by the GPS outage and are below the 7 cm accuracy target throughout. On the other hand, the GPS-Only Fusion quickly drifts outside the accuracy target after entering the GPS outage region. In the outage region, GPS-Only Fusion relies solely on body-centric odometry to dead reckon and the resulting path errors grow indefinitely.

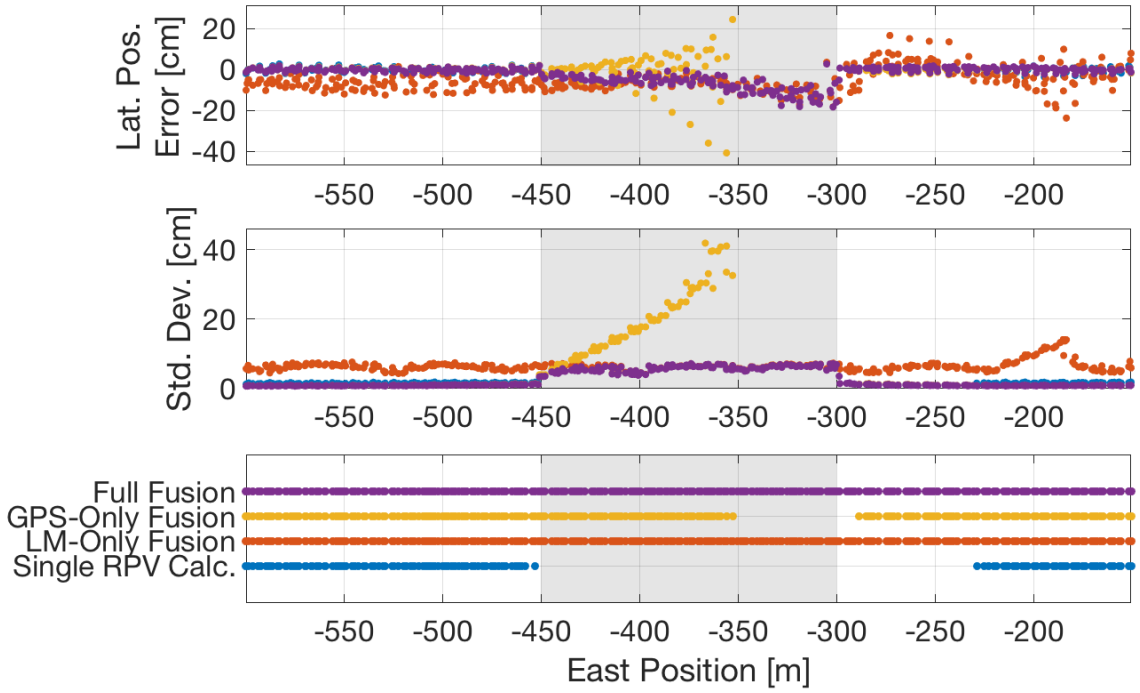


Figure 6.15: GPS Outage Test: Lateral Path Position Error Comparison

The Single RPV Calculation solution maintains acceptable lateral position accuracy, but suffers in terms of availability. The path solution is lost for Single RPV Calculation as soon as the follower reaches the outage region. Since this solution relies exclusively on GPS availability, it is not able to dead reckon through the outage. The Single RPV Calculation eventually regains a path solution, but only after the follower is more than 61 m outside of

the outage region. Since the Single RPV Calculation uses only follower odometry, the path does not become available until the follower travels the full path length outside of the outage region, in this case 61 m.

The GPS-Only Fusion improves on the availability of Single RPV Calculation. The graph-based estimator effectively dead reckons the body-centric odometry for GPS-Only Fusion up until the path tail length is exhausted (at greater than 100 m into the outage region). The GPS-Only Fusion regains a path solution faster than with Single RPV Calculation due to the inclusion of lead vehicle odometry in the solution.

The path yaw estimation results for the GPS outage test are shown in Figure 6.16. This figure shows that when GPS is available, all fusion-based solution types have path yaw errors within the accuracy target of 0.25°. During the GPS outage, the yaw errors for GPS-Only Fusion and Full Fusion solution types are elevated while the Landmark-Only Fusion solution is not affected.

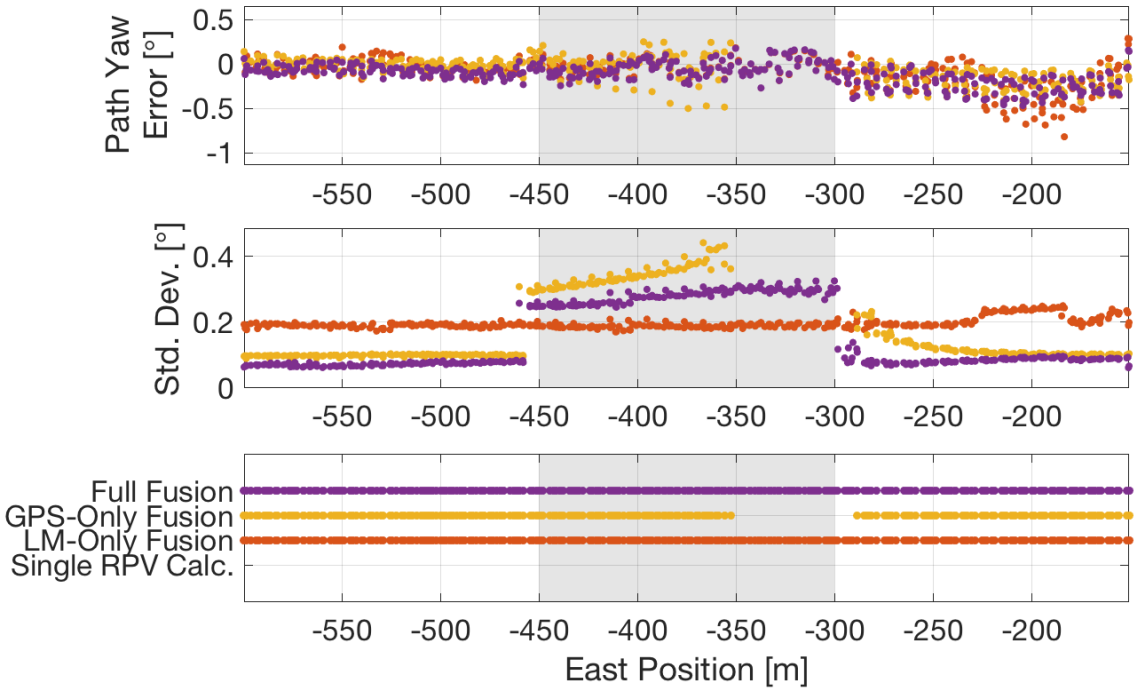


Figure 6.16: GPS Outage Test: Path Yaw Error Comparison

Interestingly, Figure 6.16 shows that the path yaw standard deviations reported by Full Fusion exceed those of Landmark-Only Fusion result during the GPS outage. Since the Full Fusion estimator is resolved in the navigation frame, it must estimate a global yaw estimate (true heading with respect to the North axis). Additional uncertainty is introduced to the yaw estimation when it must be related to the global frame. On the other hand, all of the observations used by Landmark-Only Fusion are body-centric and, by anchoring the follower yaw state, the resulting path estimate is resolved in the follower frame. Therefore, the uncertainty in the path yaw estimate for Landmark-Only Fusion is influenced only by the uncertainty of the path tangent angle, taken as the arc tangent of path positions.

### 6.2.4 Variable Following Distance Analysis

Finally, the performance of the Full Fusion path estimate is compared to other path solutions by examining the errors as a function of following distance. Namely, the four solutions previously summarized in Table 5.1 will be compared. The results from Single RPV Calculation, as presented in prior literature, will serve as the benchmark performance. Because the main benefits of the Full Fusion solution are seen at higher following distances, the performance comparison is made over variable following distances up to 2 km.

In this analysis, the following distance is set artificially to allow for variations over the test variable. This is accomplished by manipulating the timestamps of the recorded data based on a user-defined following distance. Specifically, the following vehicle's data is shifted in time: the landmark observations, the odometry measurements, and the truth positions. After shifting the data's timestamps, the inter-vehicle RPV measurements are no longer valid. Therefore, for the analysis performed in this section, the inter-vehicle RPV measurements were simulated using differenced RTK positions with artificial noise following the error model described in Section 5.1.2.

The results for this analysis are plotted in Figure 6.17 that shows lateral path error plotted against following distance. This plot shows the RMS of the lateral path position

error taken over the entire run for each following distance. The Full Fusion path estimate errors are bounded by all other path solution types at each following distance. At the smallest following distance (100 m), the Full Fusion error is virtually the same as the GPS-Only Fusion error, both of which outperform the Single RPV Calculation result. As following distance increases, the Full Fusion estimator relies more on the landmark observations. At a following distance of 1 km, the errors from Single RPV Calculation surpass the 7 cm, $1\sigma$ accuracy target.

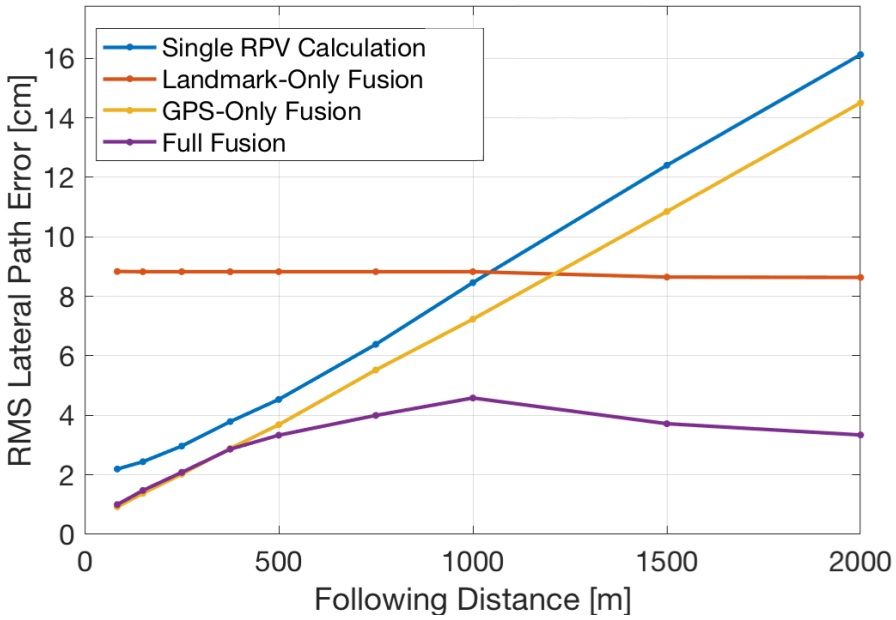


Figure 6.17: Lateral Path Position Errors for Varying Following Distance

The RMS lateral path errors for Landmark-Only Fusion are also shown in Figure 6.17. As expected, the Landmark-Only Fusion errors do not depend on following distance given that the errors in the plot are virtually flat. The findings from the simulation study in Section 5.3 conclude that the Full Fusion errors will be bounded by those of the Landmark-Only Fusion estimator. With a Landmark-Only Fusion error bound of ~8.5 cm, the Full Fusion estimate is not guaranteed to stay within acceptable accuracy targets at large enough following distances. However, the contributions from the GPS observations are enough to

keep the Full Fusion result within acceptable accuracy for all of the following distances used in this analysis.

## 6.3 Conclusion

This chapter presented an evaluation of path estimation performance using experimental data. The experimental evaluation provides tangible results similar to what would be experienced in a real-world application. Additionally, the simulation findings are given more validity by tying together simulated and experimental results.

First, a test was conducted on a public highway with variable levels of landmark and differential GPS availability. The results from this test demonstrated the improved availability of the Full Fusion path estimate compared to other path solutions. Solutions that relied heavily on inter-vehicle RPV (DRTK), such as Single RPV Calculation and GPS-Only Fusion, would eventually lose a path solution over long enough DRTK outages. Similarly, the Landmark-Only Fusion estimate would lose a path solution through large enough regions without landmarks. The body-centric odometry in the fusion-based solutions allowed these solutions to bridge small gaps in landmark and DRTK outages, so long as the gap was less than the path tail length which was set to 250 m.

Data collected on a closed course track provided a larger set of data with a ground truth reference for error calculations. The results in this test showed that the RMS errors from the Full Fusion solution, taken over the duration of the run, were below the accuracy requirement of 7 cm in lateral position and 0.25° in path yaw. This test also demonstrated how violations of the 2D planar assumption impact the final result. It was shown that when the vehicle traversed the 8° banks on the turns, the path yaw errors would spike while the reported standard deviation was virtually unchanged. Future implementations should take this behavior into consideration if the test environment is expected to have bank angles or road grade.

Also in this chapter, the proposed Full Fusion estimator was compared to other solutions in an analysis that emulated a following distance by manipulating timestamps of the collected data. This test, which evaluated the errors as a function of following distance, showed similar characteristics to the equivalent test performed using simulated data. The key contribution of this dissertation was demonstrated in this analysis by showing the Full Fusion path estimate errors were bounded with respect to following distance.

# Chapter 7

# Conclusions and Future Work

## 7.1 Conclusions

This dissertation described the challenges of relative path following at long distances and proposed a graph-based sensor fusion approach overcoming these challenges. A thorough evaluation of the state-of-the-art in leader-following methodologies was first presented to motivate the contributions that followed. These contributions included a proposed solution to path estimation that bounds error with respect to following distance using a graph-based estimator for a complementary fusion of differential GPS, body-centric odometry, and perception-based landmark observations. The observability conditions for the proposed solution were evaluated and a method was presented for ensuring sufficient observability for a given graph topology. An evaluation of the proposed solution was performed with both simulated and experimental data. The simulation environment allowed for a statistical evaluation and for controlled scenarios while experimental data provided further validation of the simulation results.

The literature review, summarized in Chapter 1, showed the potential for improvements in the field of path following for unmanned ground vehicles. Many of the path following solutions that exist in literature rely on a built infrastructure for absolute position accuracy, however, this reliance on infrastructure limits the scalability of these approaches. Other infrastructure-less path following approaches exist in prior literature, each with shortcomings that make them unsuitable for the design requirements stated in Chapter 1. To overcome the dependency on infrastructure, previous work used relative measurements between vehicles along with ego-motion to estimate relative paths, but these approaches are subject to increasing path errors as following distance increases. In the realm of landmark-based

navigation, collaborative SLAM approaches allow for path accuracies that are independent of following distance, but require a high number of distinct landmarks to reach acceptable accuracy limits. The solution presented in this dissertation combined the attributes of high precision differential GPS with SLAM techniques to achieve acceptable accuracies with relatively few landmark observations.

A graph-based estimation approach was chosen to accomplish the complementary sensor fusion of available measurement sets. The differential GPS observations of vehicle odometry (by way of TDCP) and inter-vehicle RPV (by way of DRTK) are accurate to centimeter level, which allows for lane-level accuracy of the estimated path and is sufficient for autonomous path following. The incorporation of body-centric odometry increases solution availability by improving observability conditions and spanning gaps caused by GPS and landmark outages. Landmark observations successfully bound the path error for any following distance by comparing observations exchanged between the leader and follower vehicles. The estimator was designed such that the processing requirement grows linearly with following distance which makes it feasible for application with onboard processors.

A simulation environment was developed to evaluate the graph-based estimator in Chapter 5. The statistical soundness of the estimator was evaluated through a Monte Carlo evaluation. The analysis showed that the covariance reported by the graph-based estimator matches empirical values taken over many runs with randomized data. By performing the simulation at varied following distances, the results demonstrate how the proposed estimator was successful in satisfying the design requirement of bounded error growth. The proposed Full Fusion estimator was compared to other relative path solution types. Single RPV Calculation was selected as the benchmark solution as it was presented in prior literature. The Single RPV Calculation result showed a linear increase in variance with following distance, which allowed for acceptable accuracy up to 1 km. The GPS-Only Fusion solution improved on Single RPV Calculation in the error growth rate, allowing for acceptable accuracy up to 1.5 km. The Landmark-Only Fusion result showed a virtually flat error curve, demonstrating

independence with respect to following distance, but did not satisfy accuracy requirements. The Full Fusion result demonstrated a complimentary combination of results and showed improved errors over all other solutions at all following distances.

The simulation environment was also used to evaluate the path estimation performance under different conditions. For one, a simulation test was used to show the relation between path accuracy and number of landmarks. It was shown that the Full Fusion estimator will meet accuracy requirements for all following distances when at least two common landmarks are observed by both vehicles for any given time. The simulation environment also demonstrated how the Full Fusion solution maintained availability throughout short periods of GPS outages. The path error will grow larger through GPS outages, but will remain bounded by the accuracy of the Landmark-Only Fusion solution.

In Chapter 6, results from experimental data were used to demonstrate the viability of the proposed solution from an application standpoint. The experimental results were also used to give credibility to the simulation environment and validate findings in the simulation study. The experimental data was used to replicate the following distance test from the simulation study in Chapter 5 and the same trends were shown to exist. Also, results from a GPS outage test were shown to demonstrate the impact on the error growth and availability of the different path solution types.

The proposed graph-based estimation approach may be preferred to existing approaches for the reasons discussed in this dissertation. However, it's worth mentioning that this approach may not be ideal in some cases due to the complexity of its implementation. For short distance following with consistent GPS availability, the Single RPV Calculation approach (aka GPS-based relative path generation) would be acceptable when coupled with a method for accurately estimating follower heading. Other scenarios might allow for dense, distinct, repeatable landmarks such as those available in an urban setting. In this case, it might be preferable to implement a map-matching approach in which the follower localizes on a

map generated by the lead vehicle. However, a map-matching approach would show diminished performance in sparse environments or in rural/forestry applications where landmark correspondence becomes challenging.

Despite the upfront implementation cost of the graph-based approach, the framework was shown to have other desirable traits. For one, the intuitive nature of the graph makes it easier to inspect visually; any loss of path accuracy can be diagnosed quickly by examining the graph's topology. Additionally, the framework is fairly simple to augment. Any new measurements the user would like to incorporate can simply be added as edges in the graph. These augmentations are reserved for future contributions that seek to expand upon the work presented in this dissertation.

## 7.2 Future Work

The work presented in this dissertation shows great potential for a graph-based path estimation approach that future work may choose to expand upon. As mentioned above, future implementations may seek to augment the graph with additional sensor measurements. A situation may call for improved GPS-denied performance which could be accomplished using inter-vehicle RPV measurements provided by perception-based sensors or UWB ranging. Other sources of odometry may also be used: odometry from lidar ICP or visual odometry from camera might be preferred to increase platform independence.

Future work may also seek to address the correlation between TDCP and DRTK observations. This correlation was neglected for the work presented in this dissertation in order to generalize the observation models used in the graph-based path estimator. However, neglecting this correlation causes the covariance reported by the estimator to be overly optimistic. One approach to addressing this correlation would be to use carrier phase measurements as edges in the graph (a closely coupled approach) as shown in [41] for a single vehicle. This would add to the complexity of the solution but benefit in the availability of an integer fix even in scenarios with less than four common satellites.

The graph-based solution to path estimation would also be well-suited for multi-vehicle convoy networks of three or more vehicles. In the presented work, an Ultimate Lead strategy was proposed as a way to implement the graph-based solution in multi-vehicle convoys. To expand on this, a centralized approach could be used that factors in information from all vehicles. A centralized approach would improve the solution in terms of availability and accuracy at the cost of computational complexity and communication bandwidth. This could be accomplished by adding subgraphs to the graph, one for each vehicle in the convoy. Inter-vehicle RPV measurements could then be used from each pair of vehicles. This approach would also increase the availability of the solution in the case of lost communications from any single vehicle.

In summary, a graph-based estimation approach is well suited for a variety of future applications. The findings presented in this dissertation show a great deal of potential for future variations on this approach. Future applications may differ in the design requirements or the available measurement sets, but the groundwork established in this dissertation demonstrates the potential for any future implementations that follow.

Bibliography

[1] T. D. Gillespie, *Fundamentals of Vehicle Dynamics*. Society of Automotive Engineers, Inc., 1992.

[2] D. Hershberger, D. Gossow, and J. Faust, "RViz, 3D visualization tool for ROS," May 2018. [Online]. Available: http://wiki.ros.org/rviz

[3] A. Marjovi, M. Vasic, J. Lemaitre, and A. Martinoli, "Distributed graph-based convoy control for networked intelligent vehicles," in *2015 IEEE Intelligent Vehicles Symposium*. IEEE, 2015, pp. 138–143.

[4] J. Ryan, "A fully integrated sensor fusion method combining a single antenna GPS unit with electronic stability control sensors," Master's thesis, Auburn University, 2011.

[5] E. Stegner, J. Ward, J. Siefert, M. Hoffman, and D. M. Bevly, "Experimental fuel consumption results from a heterogeneous four-truck platoon," American Center for Mobility, Tech. Rep., 2021.

[6] N. H. T. S. Administration, "2019 data: Summary of motor vehicle crashes," NHTSA's National Center for Statistics and Analysis, 1200 New Jersey Avenue SE Washington, DC 20590, Tech. Rep. DOT HS 813 209, Nov. 2021.

[7] A. Schneider, W. Becker, J. Howard, A. Cichosz, B. Theisen, and D. Conger, "Autonomous ground resupply autonomy kit," in *Proceedings of the 2018 Ground Vehicle Systems Engineering and Technology Symposium*, 2018.

[8] B. Theisen, "Autonomous mobility appliqué system (AMAS) JCTD," Army Tank Automotive Research Development and Engineering Center, Warren MI, Tech. Rep., 2011.

[9] S. Geiger, "Laterally string stable control at large following distances using DRTK and TDCP," Master's thesis, Auburn University, 2018.

[10] T. G. R. Reid, S. E. Houts, R. Cammarata, G. Mills, S. Agarwal, A. Vora, and G. Pandey, "Localization requirements for autonomous vehicles," *CoRR*, vol. abs/1906.01061, 2019.

[11] T. C. Ng, J. I. Guzman, and M. D. Adams, "Autonomous vehicle-following systems: A virtual trailer link model," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005, pp. 3057–3062.

[12] F. de Ponte Müller, "Survey on ranging sensors and cooperative techniques for relative positioning of vehicles," *Sensors*, vol. 17, no. 2, p. 271, 2017.

[13] J. Lin, Z. Miao, H. Zhong, W. Peng, Y. Wang, and R. Fierro, "Adaptive image-based leader–follower formation control of mobile robots with visibility constraints," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 7, pp. 6010–6019, 2020.

[14] P. Richardson, W. Xiang, and D. Shan, "An outdoor UWB tracking system to improve safety of semi-autonomous vehicle operations," *International Journal of Ultra Wideband Communications and Systems*, vol. 1, no. 3, pp. 209–221, 2010.

[15] Autonomous Solutions, Inc., "ASI guideline for autonomous convoy: Safe, simple, reliable," May 2017. [Online]. Available: https://asirobots.com/asi-guideline-autonomous-convoy-safe-simple-reliable/

[16] Z. Yang, S. Zhu, C. Chen, G. Feng, and X. Guan, "Leader-follower formation control of nonholonomic mobile robots with bearing-only measurements," *Journal of the Franklin Institute*, vol. 357, no. 3, pp. 1628–1643, 2020.

[17] S. Sivaraman and M. M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1773–1795, 2013.

[18] C. Chen, H. Wang, N. T. Chew, J. Ibanez-Guzmán, S. Jian, and C. C. Wah, "Target-tracking and path planning for vehicle following in jungle environment," in *ICARCV 2004 8th Control, Automation, Robotics and Vision Conference*, vol. 1. IEEE, 2004, pp. 455–460.

[19] H. G. Nguyen, G. Kogut, R. Barua, A. Burmeister, N. Pezeshkian, D. Powell, N. Farrington, M. Wimmer, B. Cicchetto, C. Heng *et al.*, "A segway RMP-based robotic transport system," in *Mobile Robots XVII*, vol. 5609. International Society for Optics and Photonics, 2004, pp. 244–256.

[20] S. Kato, S. Tsugawa, K. Tokuda, T. Matsui, and H. Fujii, "Vehicle control algorithms for cooperative driving with automated vehicles and intervehicle communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 3, pp. 155–161, 2002.

[21] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018.

[22] L. Zhang, T. Ahamed, Y. Zhang, P. Gao, and T. Takigawa, "Vision-based leader vehicle trajectory tracking for multiple agricultural vehicles," *Sensors*, vol. 16, no. 4, p. 578, 2016.

[23] B. Jones, K. Thompson, D. Pierce, S. Martin, and D. Bevly, "Ground-vehicle relative position estimation with UWB ranges and a vehicle dynamics model," in *2nd Modeling, Estimation and Control Conference (MECC 2022)*, 2022.

[24] M. Luo, H. Fang, Y. Li, Y. Bai, J. Chen, and Y. Wei, "Distributed robust fault estimation using relative measurements for leader–follower multiagent systems," *IEEE Transactions on Cybernetics*, vol. 51, no. 9, pp. 4707–4715, 2019.

[25] W. Travis, "Path duplication using GPS carrier based relative position for automated ground vehicle convoys," Ph.D. dissertation, Auburn University, 2010.

[26] S. Martin, "Closely coupled GPS/INS relative positioning for automated vehicle convoys," Master's thesis, Auburn University, 2011.

[27] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, 2nd ed. Artech House, 2013.

[28] W. Travis, S. M. Martin, D. W. Hodo, and D. M. Bevly, "Non-line-of-sight automated vehicle following using a dynamic base RTK system," *Navigation*, vol. 58, no. 3, pp. 241–255, 2011.

[29] B. P. Gerkey and K. Konolige, "Planning and control in unstructured terrain," in *ICRA Workshop on Path Planning on Costmaps*, 2008.

[30] D. C. Salmon and D. M. Bevly, "An experimental exploration of low-cost solutions for precision ground vehicle navigation," Master's thesis, Auburn University, 2015.

[31] O. Naroditsky, Z. Zhu, A. Das, S. Samarasekera, T. Oskiper, and R. Kumar, "Videotrek: A vision system for a tag-along robot," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 1101–1108.

[32] W. Flenniken, "Modeling inertial measurement units and analyzing the effect of their errors in navigation applications," Ph.D. dissertation, Auburn University, 2005.

[33] H. K. Goi, T. D. Barfoot, B. A. Francis, and J. L. Giesbrecht, "Vision-based vehicle trajectory following with constant time delay," in *Field and Service Robotics*. Springer, 2010, pp. 137–147.

[34] M. A. Sheta, R. S. Meer, V. S. Chinvar, S. Vadivelu, and M. E. Flinders, "Path accumulation and ego-motion compensation for target path following and situational awareness," in *21st International Conference on System Theory, Control and Computing (ICSTCC)*. IEEE, 2017, pp. 126–133.

[35] J. L. Giesbrecht, H. K. Goi, T. D. Barfoot, and B. A. Francis, "A vision-based robotic follower vehicle," in *Unmanned Systems Technology XI*, vol. 7332. International Society for Optics and Photonics, 2009, p. 73321O.

[36] D. Fassbender, B. C. Heinrich, T. Luettel, and H.-J. Wuensche, "An optimization approach to trajectory generation for autonomous vehicle following," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 3675–3680.

[37] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[38] H. J. Motulsky and R. E. Brown, "Detecting outliers when fitting data with nonlinear regression–a new method based on robust nonlinear regression and the false discovery rate," *BMC bioinformatics*, vol. 7, no. 1, pp. 1–20, 2006.

[39] M. Shan, Y. Zou, M. Guan, C. Wen, and C.-L. Ng, "A leader-following approach based on probabilistic trajectory estimation and virtual train model," in *IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 1–6.

[40] J. Wall, "A study of the effects of stochastic inertial sensor errors in dead-reckoning navigation," Ph.D. dissertation, Auburn University, 2007.

[41] T. Suzuki, "Robust vehicle positioning in multipath environments based on graph optimization," in *Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021)*, 2021, pp. 4223–4233.

[42] R. Sharma, R. W. Beard, C. N. Taylor, and S. Quebe, "Graph-based observability analysis of bearing-only cooperative localization," *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 522–529, 2011.

[43] P. Smith, J. Ward, J. D. Pierce, D. Bevly, and R. Daily, "Experimental results and analysis of a longitudinal controlled cooperative adaptive cruise control (CACC) truck platoon," in *Dynamic Systems and Control Conference*. American Society of Mechanical Engineers, 2019.

[44] T. Collett, "Insect navigation en route to the goal: multiple strategies for the use of landmarks," *Journal of Experimental Biology*, vol. 199, no. 1, pp. 227–235, 1996.

[45] G. D. Tipaldi and K. O. Arras, "Flirt-interest regions for 2D range data," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2010, pp. 3616–3622.

[46] H. Kloeden, D. Schwarz, E. M. Biebl, and R. H. Rasshofer, "Vehicle localization using cooperative RF-based landmarks," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, 2011, pp. 387–392.

[47] S. Thrun and M. Montemerlo, "The graph SLAM algorithm with applications to large-scale mapping of urban structures," *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 403–429, 2006.

[48] R. Dubé, A. Gawel, H. Sommer, J. Nieto, R. Siegwart, and C. Cadena, "An online multi-robot SLAM system for 3D lidars," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1004–1011.

[49] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 3, pp. 194–220, 2017.

[50] D. Pierce, S. M. Martin, and D. M. Bevly, "Opportunistic landmark registration for long distance relative path following," in *Proceedings of the 30th International Technical Meeting of The Satellite Division of the Institute of Navigation*, 2015.

[51] R. C. Coulter, "Implementation of the pure pursuit path tracking algorithm," The Robotics Institute at Carnegie Mellon University, Tech. Rep., 1992.

[52] A. Eick, "A nonlinear model predictive control algorithm for an unmanned ground vehicle on variable terrain," Master's thesis, Auburn University, 2016.

[53] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[54] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, 2006.

[55] D. B. West, *Introduction to Graph Theory*. Prentice-Hall, 2001, vol. 2.

[56] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.

[57] A. Tveit, "On the complexity of matrix inversion," *Mathematical Note*, p. 1, 2003.

[58] P. K. Enge, "The global positioning system: Signals, measurements, and performance," *International Journal of Wireless Information Networks*, vol. 1, no. 2, pp. 83–105, 1994.

[59] P. Teunissen, "The least-squares ambiguity decorrelation adjustment: A method for fast GPS integer ambiguity estimation," *Journal of Geodesy*, vol. 70, pp. 65–82, 1995.

[60] J. L. Crassidis and J. L. Junkins, *Optimal estimation of dynamic systems*. Chapman and Hall/CRC, 2004.

[61] J. L. Crassidis, "Sigma-point Kalman filtering for integrated GPS and inertial navigation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 2, pp. 750–756, 2006.

[62] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE, 2004, pp. I–I.

[63] M. L. Tazir, T. Gokhool, P. Checchin, L. Malaterre, and L. Trassoudaine, "CICP: Cluster iterative closest point for sparse–dense point cloud registration," *Robotics and Autonomous Systems*, vol. 108, pp. 66–86, 2018.

[64] J. Beuchert, M. Camurri, and M. Fallon, "Factor graph fusion of raw GNSS sensing with IMU and lidar for precise robot localization without a base station," *arXiv preprint arXiv:2209.14649*, 2022.

[65] J. Ginsberg, *Engineering dynamics*. Cambridge University Press, 2008, vol. 10.

[66] X.-F. Hana, J. S. Jin, J. Xie, M.-J. Wang, and W. Jiang, "A comprehensive review of 3D point cloud descriptors," *arXiv preprint arXiv:1802.02297*, vol. 2, 2018.

[67] F. Tombari, S. Salti, and L. Di Stefano, "A combined texture-shape descriptor for enhanced 3D feature matching," in *18th IEEE International Conference on Image Processing*. IEEE, 2011, pp. 809–812.

[68] J. R. Kidd, "Performance evaluation of the velodyne VLP-16 system for surface feature surveying," Ph.D. dissertation, University of New Hampshire, 2017.

[69] T. Foote, "tf: The transform library," in *IEEE Conference on Technologies for Practical Robot Applications (TePRA)*. IEEE, 2013, pp. 1–6.

[70] R. B. Rusu, "Semantic 3D object maps for everyday manipulation in human living environments," Ph.D. dissertation, Computer Science department, Technische Universitaet Muenchen, Germany, Oct. 2009.

[71] M. Lehtomäki, A. Jaakkola, J. Hyyppä, A. Kukko, and H. Kaartinen, "Detection of vertical pole-like objects in a road environment using vehicle-based laser scanning data," *Remote Sensing*, vol. 2, no. 3, pp. 641–664, 2010.

[72] J. Gim, C. Ahn, and H. Peng, "Landmark attribute analysis for a high-precision landmark-based local positioning system," *IEEE Access*, vol. 9, pp. 18 061–18 071, 2021.

[73] T. Rabbani and F. Van Den Heuvel, "Efficient Hough transform for automatic detection of cylinders in point clouds," *ISPRS WG III/3, III/4*, vol. 3, pp. 60–65, 2005.

[74] N. Chernov and C. Lesort, "Least squares fitting of circles," *Journal of Mathematical Imaging and Vision*, vol. 23, no. 3, pp. 239–252, 2005.

[75] Q. Ji and Y. Xie, "Randomised Hough transform with error propagation for line and circle detection," *Pattern Analysis & Applications*, vol. 6, no. 1, pp. 55–64, 2003.

[76] S. T. Pfister, S. I. Roumeliotis, and J. W. Burdick, "Weighted line fitting algorithms for mobile robot map building and efficient data representation," in *2003 IEEE International Conference on Robotics and Automation*, vol. 1. IEEE, 2003, pp. 1304–1311.

[77] American Association of State Highway and Transportation Officials, "A policy on design standards — interstate system," 2016. [Online]. Available: https://www.dot.state.al.us/publications/Design/pdf/DesignStandardsInterstateSystem.pdf

[78] "J3016: Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles," SAE International, Standard, Apr. 2021.

[79] J. Ward, P. Smith, D. Pierce, D. Bevly, P. Richardson, S. Lakshmanan, A. Argyris, B. Smyth, C. Adam, and S. Heim, "Cooperative adaptive cruise control (CACC) in controlled and real-world environments: Testing and results," 2019.

[80] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," *Matrix*, vol. 58, no. 15-16, pp. 1–35, 2006.

[81] C. Kimme, D. Ballard, and J. Sklansky, "Finding circles by an array of accumulators," *Communications of the ACM*, vol. 18, no. 2, pp. 120–122, 1975.

Appendices

# Appendix A

## Rotations and Coordinate Frames

The work presented in this dissertation uses a North-East-Down (NED) convention. In order to track an object within this frame, a rotation is made from the coordinate frame attached to the object (body frame) to the NED frame (navigation frame). The body frame aligns with the navigation frame at zero roll, pitch, and yaw. When the two frames align, the North, East, Down directions correspond with the body's X, Y, and Z, respectively. A rotation matrix is used to rotate vectors from the body frame to the navigation frame ($\mathbf{C}_b^n$). These equations are shown in Equation (A.1) where a vector ($\mathbf{r}^b$) is rotated into the navigation frame.

$$\mathbf{r}^n = \mathbf{C}_b^n \mathbf{r}^b \tag{A.1}$$

To perform this rotation in three dimensions, a series of three individual rotations are made. These three rotations are performed in the specific order shown in Equation (A.2) to produce the 3D rotation matrix. The rotation matrix from the navigation frame to the body frame ($\mathbf{C}_n^b$) is found simply by taking the inverse of the rotation matrix from the body frame to the navigation frame. Since these matrices are symmetric, the inverse of that matrix is the same as the transpose [65]. The rotation matrices are a function of the roll ($\phi$), pitch ($\theta$), and yaw ($\psi$) angles as shown in the equations below.

$$\mathbf{C}_b^n = \left(\mathbf{C}_n^b\right)^{-1} = \left(\mathbf{C}_\phi \mathbf{C}_\theta \mathbf{C}_\psi\right)^{-1} \tag{A.2}$$

$$\mathbf{C}_{\psi} = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.3}$$

$$\mathbf{C}_{\theta} = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \tag{A.4}$$

$$\mathbf{C}_{\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \tag{A.5}$$

Angular rate vectors are transformed from the body frame ($\boldsymbol{\omega}^b$) to the navigation frame ($\boldsymbol{\omega}^n$) using the mechanization matrix ($\mathbf{M}$) as shown below.

$$\boldsymbol{\omega}^n = \mathbf{M}\boldsymbol{\omega}^b \tag{A.6}$$

The navigation frame angular rate can also be thought of as the time-derivative of the Euler angles ($\boldsymbol{\omega}^n = [\dot{\phi}, \dot{\theta}, \dot{\psi}]^T$). The matrix $\mathbf{M}$ is a function of the roll and pitch values as shown in the equation below [80].

$$\mathbf{M} = \frac{1}{\cos(\theta)} \begin{bmatrix} 1 & \sin(\phi)\sin(\theta) & \cos(\phi)\sin(\theta) \\ 0 & \cos(\phi)\cos(\theta) & -\sin(\phi)\sin(\theta) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \tag{A.7}$$

# Appendix B

## Least Squares Estimation

The least squares algorithm is used to fit a set of noisy observations with a model. In other words, a set of noisy measurements is used to determine the most likely value of a system's state. Along with a state estimate, the least squares solution provides the covariance associated with that estimate. For a nonlinear system, an iterative process must be performed, updating the estimate on each iteration. When information of the measurement accuracy is available, a weighting can be assigned to each individual measurement and accounted for in the weighted least squares approach [27].

Consider the linear equation

$$\delta\mathbf{z}^- = \mathbf{H}\delta\mathbf{x} + \delta\mathbf{z}^+ \tag{B.1}$$

where $\delta\mathbf{z}^-$ is the measurement innovation and $\delta\mathbf{z}^+$ is the measurement residual as defined in the equations below.

$$\delta\mathbf{z}^- = \mathbf{z} - \mathbf{h}\left(\hat{\mathbf{x}}^-\right) \tag{B.2}$$

$$\delta\mathbf{z}^+ = \mathbf{z} - \mathbf{h}\left(\hat{\mathbf{x}}^+\right) \tag{B.3}$$

Also in Equation (B.1), $\delta\mathbf{x}$ is the state vector innovation as defined below.

$$\delta\mathbf{x} = \hat{\mathbf{x}}^+ - \hat{\mathbf{x}}^- \tag{B.4}$$

The matrix $\mathbf{H}$ is the measurement Jacobian matrix, which is defined in the equation below.

$$\mathbf{H} = \frac{\partial\mathbf{h}}{\partial\mathbf{x}} \tag{B.5}$$

The least squares estimation operates by minimizing the measurement residuals with respect to the state vector ($\mathbf{x}$). From this, the following relation is made.

$$\frac{\partial}{\partial \delta \mathbf{x}} \left( \delta \mathbf{z}^{+^T} \delta \mathbf{z}^{+} \right) = 0 \tag{B.6}$$

After performing substitutions and additional matrix algebra and simplification, the resulting solution for $\delta \mathbf{x}$ is given in the equation below.

$$\delta \mathbf{x} = \left( \mathbf{H}^T \mathbf{H} \right)^{-1} \mathbf{H}^T \delta \mathbf{z}^{-} \tag{B.7}$$

By applying the above equation, an optimal estimate of the states is made given the provided measurements.

**Iterated Least Squares**

In the case that the measurement model ($\mathbf{h}$) is completely independent of the states, the solution to Equation (B.7) provides an optimal estimate of $\delta \mathbf{x}$. In the case of a nonlinear problem, the measurement model and Jacobian matrix will be a function of the state values. In this case, *a priori* estimates of the state are used to approximate the measurement matrix. By performing least squares in an iterative fashion and setting $\hat{\mathbf{x}}^{-}$ to $\hat{\mathbf{x}}^{+}$ after each iteration, a state estimate can be achieved. The iteration process can be discontinued when the value of $\delta \mathbf{x}$ converges below some predefined threshold.

**Weighted Least Squares**

In many cases of state estimation, the measurements provided have varying accuracy. In this case, it is sensible to provide a higher weighting to the more accurate measurements. Similar to the previous least squares problem, the residuals are to be minimized, but in this

case a weighting matrix ($\mathbf{W}$) is applied as shown below.

$$\frac{\partial}{\partial \delta \mathbf{x}} \left( \delta \mathbf{z}^{+^T} \mathbf{W} \delta \mathbf{z}^{+} \right) = 0 \tag{B.8}$$

After performing substitutions and additional matrix algebra and simplification, the following equation results.

$$\delta \mathbf{x} = \left( \mathbf{H}^T \mathbf{W} \mathbf{H} \right)^{-1} \mathbf{H}^T \mathbf{W} \delta \mathbf{z}^{-} \tag{B.9}$$

For many applications, the accuracy of a measurement is provided as a covariance matrix ($\mathbf{R_z}$). The weighting matrix used in Equation (B.9) is simply the inverse of the covariance matrix ($\mathbf{W} = \mathbf{R_z^{-1}}$) . With this, the inverse of the covariance matrix can replace the weighting matrix in the above equation.

**Covariance of Least Squares Residuals**

Since the measurements used in least squares are not perfect, there will inevitably be residuals after the estimation process. The residuals, shown in Equation (B.3), have an associated covariance. The covariance of the residuals is calculated using Equation (B.10).

$$\mathbf{R}_{\delta \mathbf{z}}^{+} = \mathbf{R_z} - \mathbf{H} \left( \mathbf{H}^T \mathbf{R_z^{-1}} \mathbf{H} \right)^{-1} \mathbf{H}^T \tag{B.10}$$

Equation (B.10) is essentially the covariance of the measurement minus the covariance of the state estimate mapped into the measurement domain. The covariance of the state estimate ($\mathbf{P}_{\hat{\mathbf{x}}}^{+}$) is calculated using Equation (B.11). This covariance provides information on the accuracy of the estimation and the correlation between state estimates.

$$\mathbf{P}_{\hat{\mathbf{x}}}^{+} = \left( \mathbf{H}^T \mathbf{R_z^{-1}} \mathbf{H} \right)^{-1} \tag{B.11}$$

# Appendix C

## Least Squares Planar Fit

Least Squares Planar Fit aims to estimate the coefficients of a plane that optimally fits a set of $m$ points in 3 dimensional cartesian space ($\{x_i, y_i, z_i\}_{i=1}^{m}$). The plane is modeled with the $x$ dimension as the dependent variable and $y$ and $z$ as the independent variables. This model follows the form

$$x = a_0 y + a_1 z + b \tag{C.1}$$

where $a_0$, $a_1$, and $b$ are scalar coefficients to be estimated. The error is to be minimized along the $x$ dimension following the error function below.

$$E(a_0, a_1, b) = \sum_{i=1}^{m} \left[(a_0 y_i + a_1 z_i + b) - x_i\right]^2 \tag{C.2}$$

The model is arranged as matrices to fit the $\mathbf{z} = \mathbf{H}\mathbf{x}$ form expected for least squares estimation. The matrix form of the model is

$$\mathbf{x} = \begin{bmatrix} a_0 \\ a_1 \\ \cdots \\ b \end{bmatrix} \tag{C.3}$$

$$\mathbf{z} = \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_m \end{bmatrix} \tag{C.4}$$

$$\mathbf{H} = \begin{bmatrix} y_1 & z_1 & 1 \\ y_2 & z_2 & 1 \\ & \cdots & \\ y_m & z_m & 1 \end{bmatrix} \tag{C.5}$$

where $\mathbf{x}$ is the state vector, $\mathbf{z}$ is the measurement vector and $\mathbf{H}$ is the measurement matrix. The value of $\mathbf{x}$ that minimizes Equation (C.1) ($\hat{\mathbf{x}}$) is estimated using least squares as shown in Section B. The least squares estimation also produces a covariance matrix for the planar coefficients ($\mathbf{P}_{\hat{\mathbf{x}}}$).

The angle of the plane measured about the $z$-axis ($\beta$) is calculated from the planar coefficient $a_0$ using the equation below.

$$\beta = -\arctan(a_0) \tag{C.6}$$

The error in the estimate of $\beta$ has a variance $v_\beta$ which can be approximated from the variance of $\hat{a}_0$ ($v_{a_0}$) which is taken as the first diagonal element in $\mathbf{P}_{\hat{\mathbf{x}}}$. This transformation from $v_{a_0}$ to $v_\beta$ is done using a First Order Taylor Series Linearization of Equation (C.6), resulting in the relation given below.

$$v_\beta = \frac{v_{a_0}}{(\hat{a}_0^2 + 1)^2} \tag{C.7}$$

# Appendix D

## Hough Transform Circle Fit

The Hough Transform Circle Fit seeks to estimate the 2D position center $(x_c, y_c)$ and radius $(r)$ of a circle that best fits a set of $m$ points in 2 dimensional cartesian space $(\{x_i, y_i\}_{i=1}^{m})$. The Hough Transform fit technique is a voting based scheme that requires a 3 dimensional matrix accumulator $(\mathbf{M})$ for estimating the three parameters of a circle [81]. The radius is limited to a search space of size $N_r$ that is determined by a user defined minimum radius $(r_{min})$, maximum radius $(r_{max})$, and step size $(r_{step})$ as $\{r_{min}, r_{min} + r_{step}, \ldots, r_{max}\}$. Similarly, the values of $x_c/y_c$ have corresponding search space of size $N_{x_c}/N_{y_c}$ defined by minimum values $(x_{c,min}/y_{c,min})$, maximum values $(x_{c,max}/y_{c,max})$ and step $(x_{c,step}/y_{c,step})$. The accumulator for a given search space will be an integer-valued matrix of size $N_{x_c} \times N_{y_c} \times (N_r - 1)$ and is initialized with all zeros at the start of the operation.

Let $\mathbf{p}_i = [x_i, y_i]$ denote a sample in the provided set of points. Each $\mathbf{p}_i$ will be evaluated within the search space for scoring the accumulator. For each combination of $x_{c,ii}$ and $y_{c,jj}$ in the search space, a circle of a specific radius exists $(r_{ii,jj})$ that passes through $\mathbf{p}_i$. This radius is calculated using the formula for a circle as shown in the equation below.

$$r_{ii,jj} = \sqrt{(x_i - x_{c,ii})^2 + (y_i - y_{c,jj})^2} \tag{D.1}$$

The value of $r_{ii,jj}$ is compared to the bin edges of the radius search space. If $r_{ii,jj}$ lies within a radius search bin at $kk$, the accumulator will be incremented as $\mathbf{M}_{ii,jj,kk}$ += 1. After looping through all $m$ points, the circle parameters are taken as the $x_c$, $y_c$, and $r$ values associated with the accumulator bin with the highest count.