Coupling GPS/INS and IMM Radar Tracking Algorithms for Precise Collaborative Ground Vehicle Navigation

by

Joseph M. Selikoff

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

Auburn, Alabama
December 15, 2018

Keywords: GPS, INS, Radar, IMM, Collaborative Navigation

Approved by

David Bevly, Chair, Professor of Mechanical Engineering
John Hung, Professor of Electrical and Computer Engineering
Dan Marghitu, Professor of Mechanical Engineering

Abstract

This thesis describes a method of collaborative ground vehicle navigation utilizing shared radar data to provide observations during periods of GPS degradation. Navigational errors that typically arise from degraded GPS signals can be reduced by providing relative observations between vehicles from an Interacting Multiple Model (IMM) radar tracking filter. Loosely coupled GPS/INS Extended Kalman Filters provide navigation solutions for each vehicle. When a vehicle experiences GPS outages, other vehicles provide external observations from the IMM tracking filter to correct the INS solution and bound error growth during the outage. The IMM tracking filter uses constant velocity, constant acceleration, and constant turn models in combination to generate a tracking solution. An evaluation of the performance of the proposed method is presented using both simulated and experimental data. The IMM tracking algorithm is implemented using range, range-rate, and azimuth data from a Delphi electronically scanning radar. Results show improved navigation performance when utilizing the relative observations during GPS outages. Specifically, the drift of the INS solution is bounded by the external measurements provided by the IMM tracking filter when GPS is unavailable. Results from both simulated and experimental data sets show that the system provides drastic improvements over standalone INS navigation, with up to a 94% decrease in error on position. These results demonstrate that the proposed combination of GPS/INS and Radar IMM algorithms constitute a feasible method of maintaing navigational accuracy during GPS outages.

Acknowledgments

There have been many meaningful and influential people in my life to whom I am grateful, but none more so than my parents Danita and Richard. They have always been supportive of me through my many and varied pursuits, even when it was inconvenient for them. I would not be who I am today if it was not for their consistent involvement in my life. I would also like to say thank you to the many teachers, mentors, and friends who have been pivotal in my development as an engineer and as a person.

My experience at Auburn has been filled with characters. I've got to mention my baja boys Saxon and Zack, who put up with me in the Wiggins basement for years during some of the most stressful times of our lives. They were the reliable anchored buoys in a turbulent sea of newbies. I am lucky I had the chance to build ***race cars*** with those guys. I also must mention the Parsons crew of Ryan and Nick. It is always interesting working with those guys, one way or another. They have also been helpful in the process of finishing this degree in many different ways.

I also want to thank Dr. Bevly, who gave me the opportunity to work in his lab. My time in the GAVLAB has been an amazing learning experience. I have had a chance to work on various interesting projects and learn about topics I would never have been exposed to otherwise. To everyone who I've crossed paths with in the lab, I want to say thanks for the help and entertainment. Whether it was ping pong, or arguments about Star Wars, or playing Minecraft music in the office, or throwing shade on Slack, we always found a way to have a good time.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

As the autonomous ground vehicle industry becomes increasingly present in everyday life, so too does the necessity for accurate and reliable navigation. The operation of autonomous vehicles in certain environments can present difficulties for navigation, especially navigation schemes which are dependent on measurements from the Global Positioning System (GPS). It is common to fuse GPS and Intertial Navigation Systems (INS) into GPS/INS algorithms, but these systems are vulnerable to GPS outages, during which only the INS propagation occurs. In short, a GPS outages or degradations of the GPS measurements can reduce even the most intricate GPS/INS algorithms to simple deduced reckoning. One method of alleviating this problem is collaborative navigation, in which navigation information is shared from user to user.

There are many well-established navigation techniques for use in ground vehicles, all of which generally fall into two categories; GPS dependent and GPS independent methods. GPS dependent systems tend to combine sensors that are stable in the short term but provide measurements at a high update rate. GPS itself has the advantage of being stable in the long term but usually has much lower measuerment rates. These sensors are commonly combined in a stochastic filtering technique such as Kalman Filters (KFs), Extended KFs (EKFs), or Particle Filters (PFs). These filters use high rate, low stability sensors to generate navigation solutions between the lower rate GPS measurement epochs, at which point the solution is corrected. GPS independant systems may use a similar set of sensors as GPS dependant systems, but will

use other methods to navigate locally such as road marking detection and landmark detection. These techniques are inherently tolerant of GPS outages, but are also more computationally intensive and dependent on road infrastructure to be predictable and well maintained. Methods of tracking other vehicles on the road are also important, especially for collision avoidance purposes. Ideally, collaborative navigation systems would combine the information from multiple vehicles that may be using differing navigation methods in order for the total navigational accuracy of the entire network to improve.

Maintaining consistent communication between vehicles is a non-trivial task. As communication links become more prevalent, bandwidth limits become a large concern, especially in high traffic situations where many vehicles may be communicating at similar frequency ranges. Timing and prioritization of received messages is also a challenge. Fault tolerance will also be an important avenue for development. Poor vehicle maintenance or simple degradation of electronic components necessitate robust fault detection and exclusion methods in these systems. Another factor that may inhibit the acceptance of inter-vehicle communications is the willingness of car makers to agree on communication standards. As challenging as it was to create a standardized diagnostic system (currently OBD-II), the inter-operability of manufacturer's vehicle-to-vehicle systems has the possibility of much higher penalties for discrepencies. Protocols for these communication networks are in development, such as Dedicated Short-Range Communications (DSRC), but may be a speedbump in the proliferation of inter-vehicle communication networks. A more sinister concern is the jamming of communication networks. Assuming that, one day, these networks are integral to the safe operation to a large portion of vehicles on the road, the effects of a jamming attack could be quite serious. The details of the vehicle-to-vehicle communication are out of the scope of this thesis. It is assumed that the information is instantaneously available to each vehicle, and that the data is faultless.

Neglecting the issues with the networking itself, the fusion of data from multiple vehicles into a collaborative navigation scheme is also a difficult task. As mentioned previously, standardization between manufacturers is not an easy feat. This means that the navigation sensors,

algorithms, physical configurations, data structure, and even units may differ from manufacturer to manufacturer or even from vehicle to vehicle. Due to this inconsistency between vehicles, the data processing becomes more complicated. Information about the quality of the shared navigation data must be known in order to properly integrate that data into existing navigation algorithms, otherwise the inclusion of lower accuracy data may cause the navigation solutions to be worse than they would be without. Again, this thesis assumes that all of the relevant information about the quality of the shared data is known by each vehicle.

This thesis proposes a solution to the problem of GPS degradation by combining relative measurements between vehicles generated from a radar unit with an implementation of the Interacting Multiple Model (IMM) algorithm in order to produce a collaborative navigation framework. This combined GPS/INS/Radar/IMM (GRIMM) algorithm is shown to be a viable solution to the problem of unreliable GPS measurements in both simulation and experimental tests.

1.2 Prior Work

This thesis develops a collaborative navigation scheme combining GPS/INS and IMM radar tracking systems. Collaborative methods of navigation are common in the literature. In [1], a method of collaborative mapping for use in rescue or law enforcement is developed. Various Simultaneous Localization and Mapping (SLAM) algorithms are combined in a joint network to fuse navigation data from various pedestrian navigation systems to generate maps of the operating area. Collaborative navigation methods for surveying and infrastructure applications were also developed in [2]. Multiple users were collected as a 'neighborhood' and the data was processed in a collaborative positionaing architecture. A multi-platform collaborative navigation scheme is presented in [3], in which methods of networking and accounting for varying dynamics and sensor quality are proposed.

There is considerable work in the literature for the use of multiple-model algorithms such as the IMM in navigation systems. An application fusing multiple models within the IMM framework with GPS and in-vehicle sensors for ground vehicle navigation is described in [4]. A similar application utilizing an ultra-tightly coupled GPS/INS system is described in [5].

Both works aim to improve the navigation of a single vehicle. This thesis differs from those works by utilizing the IMM algorithm with radar measurements to improve the navigation of another vehicle.

Radar tracking methods are also prevalent in the literature. The earliest common uses for radar tracking systems were air traffic control and maritime vessel tracking applications. Examples of these applications can be seen in [6, 7]. A method of tracking multiple aircraft in real-time is also developed in [8]. Radar tracking algorithms making use of multiple models in an IMM architecture gained traction in air traffic control applications, since aircraft undergo several common maneuvers during the landing sequence. Surveys of the use of IMM in tracking applications can be found in [9, 10]. The applications described in these surveys relate to the content of this thesis, but do not include the use of IMM radar tracking in a collaborative ground vehicle application. A collaborative navigation system which utilizes a modified IMM algorithm in order to track and match objects on the road is described in [11]. The information about the tracked landmarks is shared from vehicle to vehicle and used to localize based on the relative position estimates of the landmarks. This thesis differs from that work by using radar measurements from vehicle to vehicle in order to improve the navigation solution. Similar work by [12] uses IMM tracking algorithms to classify and predict the behavior of surrounding vehicles. This work relates closely to this thesis by using an IMM radar tracking algorithm to generate estimates of surrounding vehicles, but does not share that data in order to improve navigation. Kinematic and motion-based models commonly used in IMM applications are developed in [13]. An IMM tracking algorithm is used in an adaptive cruise control (ACC) application in [14], utilizing the estimates of the relative positioning to drive the cruise control system.

1.3 Contributions

Although the utilization of IMM radar tracking algorithms for use in ground vehicle applications exists in the literature, this thesis proposes a new method of collaborative navigation which hinges on the sharing of combined GPS/INS and radar IMM navigation data in order to

maintain an accurate navigation solution even in the absense of GPS data. The major contributions to this field presented in this thesis are:

- Integrating radar measurements within an IMM algorithm to produce relative vehicle to vehicle position and velocity estimates
- Improving navigation solutions of a vehicle with no GPS measurements by sharing the relative state estimates between vehicles
- Testing the performance of the proposed combined system with multiple simulated and experimental scenarios
- Developing a secondary implementation of the GRIMM algorithm for use in lane intrusion prediction and detection

1.4 Thesis Outline

This chapter has served as a basic introduction for the work laid out in this thesis. Chapter 2 discusses the sensors, measurements, and measurement error models used in the GPS/INS navigation filter. First, the process of generating the measurements from GPS is detailed in order to describe the errors intrinsic to the signal. Then the IMU measurement models are discussed, detailing the noise and bias terms in both the accelerometer and gyroscope measurements. The various levels of GPS/INS coupling are then described briefly, and the reasoning for the selection of GPS/INS architecture used in this thesis is given. The Loosely Coupled GPS/INS algorithm is then described in detail, including descriptions of the INS propagation, and the time and measurement updates of the error state EKF.

Next, the IMM algorithm is introduced in Chapter 3. The errors inherent to radar units are described in this chapter, and the IMM algorithm itself is provided in detail. An illustrative example of the IMM algorithm at work is shown, as the operation is not necessarily intuitive. The three models used within the GPS/INS/Radar/IMM (GRIMM) filter are then described, and examples of each model's performance are shown. Then the entire process of propagating

the IMM algorithm is detailed, including descriptions of the time and measurement updates, and a description of the non-linear measurement model necessary for the radar measurements.

The combination of the GPS/INS navigation and IMM tracking systems is described in Chapter 4. The two-vehicle case is described and the algorithm structure is detailed. Some simplifying assumptions are also asserted in this chapter, which are left to be expanded on in future work. Chapter 5 then describes the scenarios developed in simulation and duplicated in experiment which were used to validate the GRIMM algorithm. Results from both simulated and experimental data sets are shown and analyzed to verify the feasibility of the GRIMM system.

Chapter 6 then briefly discusses autonomous platooning aplications of the GRIMM filter, such as lead vehicle tracking and lane intrusion detection. A simulated example of a lane intrusion scenario is shown to demonstrate this application. Finally, conclusions formed throughout the thesis are restated and some possible avenues for future work are provided in Chapter 7.

Chapter 2

GPS/INS Navigation

GPS/INS navigation systems fuse data from GPS and Inertial Measurement Unit (IMU) sensors to generate a navigation solution that is more accurate than what either could generate alone. There are multiple formulations of the GPS/INS algorithm that are commonly used. In general, measurements from the GPS receiver and IMU are combined through some type of Bayesian filtering technique to develop the solution. While there are many possible Bayesian techniques to use, they all require some knowledge of the measurement errors. The GPS and INS errors are described in the following sections.

2.1 GPS Errors

In order to define the errors within the measurements provided by the GPS receiver, the calculations used to generate these measurements must be discussed. Position measurements provided by the GPS receiver are computed from the pseudorange measurements between the user antenna and multiple satellites. This pseudorange measurement can be described as

$$\rho^k(t) = r^k(t, t-\tau) + c[\delta t_u(t) - \delta t^k(t-\tau)] + I^k(t) + T^k(t) + \epsilon_\rho^k(t) \tag{2.1}$$

where ρ is the pseudorange, $r(t, t-\tau)$ is the actual range between the user at time t and the satellite at transmission time $(t-\tau)$, c is the speed of light, δt_u and δt are the user and satellite clock offsets, I and T are the ionosphere and troposphere atmospheric delays, and ϵ_ρ accounts for random measurement errors. The superscript k denotes the satellite index. Corrections can be applied to the pseudorange measurement described in Equation (2.1) with information

included in the ephemeris data sent by the satellite. The corrected pseudorange equation is shown below.

$$\rho_c^k = r^k + c * \delta t_u + \tilde{\epsilon}_\rho^k \tag{2.2}$$

where $\tilde{\epsilon}_\rho^k$ represents the combined residual errors in the model, which is usually modelled as white noise.

The range from the user to the satellite can be described as:

$$r^k = \sqrt{(x^k - x_u)^2 + (y^k - y_u)^2 + (z^k - z_u)^2} = \|\boldsymbol{X}^k - \boldsymbol{X}_u\| \tag{2.3}$$

where x^k, y^k, z^k and x_u, y_u, z_u are the satellite and user coordinates in ECEF, also written as $\boldsymbol{X}^k$ and $\boldsymbol{X}_u$. The standard GPS solution combines Equation (3.28) and Equation (2.2) as shown below.

$$\rho_c^k = \|\boldsymbol{X}^k - \boldsymbol{X}_u\| + b + \tilde{\epsilon}_\rho^k \tag{2.4}$$

where b is the clock bias term $c * \delta t_u$.

To solve Equation (2.4) for $\boldsymbol{X}_u$, an iterative error-state least squares approach is used, normally called the Newton-Raphson method. Initial estimates of $\boldsymbol{X}_u$ and b are selected, and the operative equation is defined as

$$\boldsymbol{\delta\rho} = \boldsymbol{G} \begin{bmatrix} \boldsymbol{\delta X}_u \\ \delta b \end{bmatrix} + \tilde{\boldsymbol{\epsilon}}_{\boldsymbol{\rho}} \tag{2.5}$$

where $\boldsymbol{\delta\rho}$, $\boldsymbol{\delta X}_u$, and δb are the pseudorange, user position, and user clock bias errors, and $\boldsymbol{G}$ is the geometry matrix, which is defined by the unit vectors pointing from the user to each satellite. Equation (2.5) is then rearranged into least squares format as shown in Equation (2.6).

$$\begin{bmatrix} \boldsymbol{\delta X}_u \\ \delta b \end{bmatrix} = (\boldsymbol{G}^T\boldsymbol{G})^{-1}\boldsymbol{G}^T\boldsymbol{\delta\rho} \tag{2.6}$$

The errors in user position and clock bias are added to the previous estimates, which are then used to initialize the next iteration of the algorithm. This process is repeated until the corrections become acceptably small.

It can be seen from Equations (2.5) and (2.6) that the white noise described by $\tilde{\boldsymbol{\epsilon}}_{\boldsymbol{\rho}}$ is included in the estimation process, resulting in a position measurement with additive white noise. In most consumer grade GPS receivers, the position accuracy can be expected to have a standard deviation of approximately 2-3 m in each axis [15].

Velocity measurements provided by the GPS receiver are computed from the pseudorange rate measurements between the user antenna and multiple satellites. The pseudorange rate is directly related to a doppler shift in the carrier frequency of the GPS signal, which is found within the tracking system. This pseudorange rate is given by

$$\dot{\rho}^k = \dot{r}^k + (\dot{b} - \dot{b}^k) + \dot{I}^k + \dot{T}^k \tag{2.7}$$

where $\dot{\rho}$ is the pseudorange rate, $\dot{r}^k$ is the actual range rate between the user and the satellite, $\dot{b}$ is the clock bias rate, and $\dot{I}$ and $\dot{T}$ are the rate of change of the ionosphere and troposphere errors. Simplifying this expression gives

$$\dot{\rho}^k = (\boldsymbol{V}^k - \boldsymbol{V}_u) * \boldsymbol{U}^k + \dot{b} + \epsilon_{\dot{\phi}}^k \tag{2.8}$$

where $\boldsymbol{V}$ and $\boldsymbol{V}_u$ are the satellite and user velocity, $\boldsymbol{U}$ is the unit vector from the user to the satellite, and $\epsilon_{\dot{\phi}}^k$ is the combined error due to ionosphere, troposphere, and modelling errors. Equation (2.8) is solved for $\boldsymbol{V}_u$ and $\dot{b}$ in the same fashion as the position estimates.

Similar to the position measurements, it can be seen that the white noise described by $\epsilon_{\dot{\phi}}^k$ propagates through the solution and ends up in the final velocity measurements. The expected standard deviation on these measurements are between 3-5 cm/s. This thesis assumes additive white noise on both position and velocity measurements from GPS.

2.2 INS Errors

Inertial Measurement Units (IMUs) generally provide measurements of acceleration and rotational rate. These measurements can generally be modelled as

$$\tilde{a}_b = b_a + (I + M_a)a_b + \nu_a \tag{2.9}$$

$$\tilde{\omega}_b = b_g + (I + M_g)\omega_b + G_g a_b + \nu_g \tag{2.10}$$

where $\tilde{a}_b$ and $\tilde{\omega}_b$ are the measured acceleration and angular rate and a_b and ω_b are the true values of acceleration and angular rate [16]. M_a and M_g represent misalignment between the axes, and G_g accounts for the gyroscope sensitivity to acceleration, or g-sensitivity. ν_a and ν_g are the white noise terms, and b_a and b_g are the accelerometer and gyroscope bias terms.

When generating IMU models to be used in automotive applications, it is common to simplify the representation of the measurements to the following expressions.

$$\tilde{a}_b = a_b + b_a + \nu_a \tag{2.11}$$

$$\tilde{\omega}_b = \omega_b + b_g + \nu_g \tag{2.12}$$

Comparing Equations (2.9 - 2.10) to Equations (2.11 - 2.12) shows that the misalignment and g-sensitivity terms have been neglected, leaving just the true values in addition to the bias and Gaussian noise terms. The misalignment errors can be assumed constant and 'absorbed' in the bias term. The g-sensitivity term, however, is not constant. Because of the limited capabilities of ground vehicles, namely the tires, accelerations seen in normal driving conditions are relatively low ($<$ 0.5 g). Therefore, the effects of acceleration on the gyroscope measurements are negligible in this application.

The bias terms for both accelerometer and gyroscope measurements can be modelled as constant, as some time independent process, or as a time dependent process. This thesis takes the latter approach, modelling the biases as first order Markov processes, where the bias is

expressed as [17]

$$\dot{b} = -\frac{1}{\tau}b + \eta \tag{2.13}$$

where $\dot{b}$ is the time derivative of the bias, b is the bias itself, τ is the time constant of the process, and η is the white noise that drives the process. Equation (2.13) can then be integrated to generate the bias values. An example of a first order Markov process is shown in Figure 2.1.

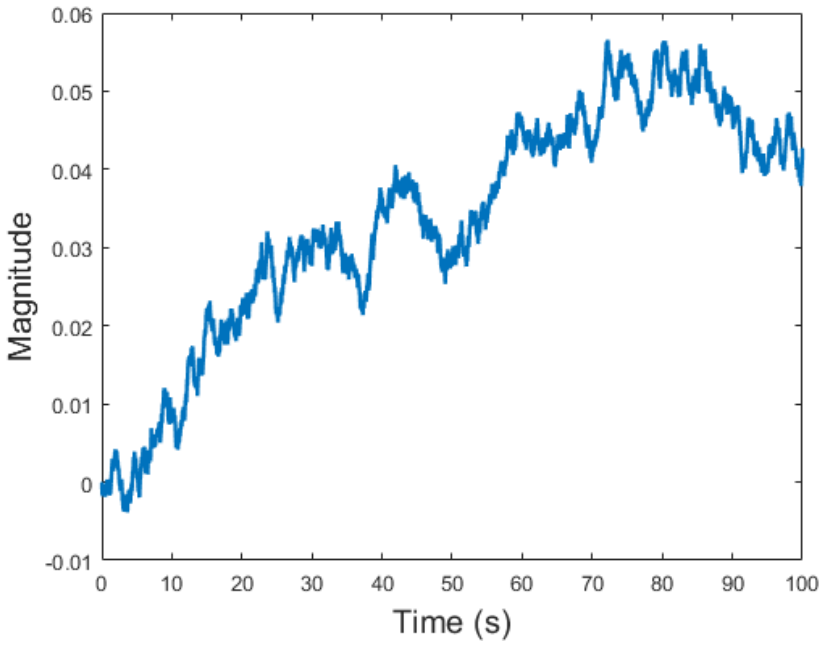

Figure 2.1: Example of a 1^{st} Order Markov Process ($\tau = 150$, $\eta = \boldsymbol{N}(0, 0.1)$)

2.3 GPS/INS Architectures

The level of coupling between GPS and INS is dependent on the corrections to the INS, the types of GPS measurements used, and how the navigation solution is fed back into the GPS receiver software [16]. There are four common levels of GPS/INS integration: Uncoupled, Loosely Coupled, Tightly Coupled, and Deeply Coupled.

Uncoupled GPS/INS systems are rather simple. The INS is used to generate navigation solutions using only the IMU measurements. When GPS measurements are available, these measurements are used to directly reset the state of the INS estimate, which is then propagated forward with IMU measurements until the next GPS measurement is available. This method is

computationally simple and requires the least information about the quality of the GPS measurements.

Loosely Coupled GPS/INS systems takes a cascaded approach to the integration. The INS solutions are computed independently, and are then combined in the integration algorithm with the GPS position and velocity measurements. The Loosely Coupled GPS/INS system is more demanding computationally than the uncoupled system, but only requires the common measurements, where the higher levels of coupling require 'internal' measurements from the receiver. This method also requires knowledge of the quality of the GPS measurements to perform properly.

Closely and Tightly Coupled GPS/INS systems similary integrate the GPS/INS systems in a cascaded structure, but utilize the pseudorange, pseudorange-rate, or other receiver internal measurements to update the integration algorithm. This method requires access to these internal measurements, which may not always be possible on consumer level receiver units.

Deeply Coupled GPS/INS systems operate differently to the previous integration levels in that they involve aiding the actual receiver funtionality with the solutions from the integration algorithm. Deeply Coupled systems use in-phase and quadrature data from the tracking system as measurements and generate NCO commands to enhance the accuracy of the code and carrier references.

In this thesis, the Loosely Coupled GPS/INS system is used exclusively, for two reasons. Firstly, and most importantly, the work in this thesis is applied to ground vehicle applications where low cost sensors are used. In low cost GPS receivers, access to internal operations and measurements, which are necessary to operate the higher integration level systems, are not as common. Secondly, the main contribution of this work is the framework of the GRIMM algorithm. Different navigation algorithms can be implemented within the GRIMM framework, but the purpose of this thesis is to generate baseline performance parameters of the GRIMM algorithm itself.

2.4 Loosely Coupled GPS/INS Navigation

As mentioned in the previous section, the Loosely Coupled GPS/INS algorithm combines the IMU and GPS measurements in a three-stage cascaded approach. A block diagram showing the design of this algorithm can be seen in Figure 2.2. The first stage of the algorithm is the INS propagation. The IMU measurements alone are used to generate an estimate of the vehicle state. The second stage is the propagation of an error state Extended Kalman Filter (EKF), which takes in the error between the INS solution and the GPS measurements to generate corrections for the INS solution. Finally, the third stage is the correction of the INS solution. The error state generated from the EKF is used to correct the INS solution to provide the navigation solution, and the estimates of accelerometer and gyroscope bias are given to the INS in order to improve its solution.

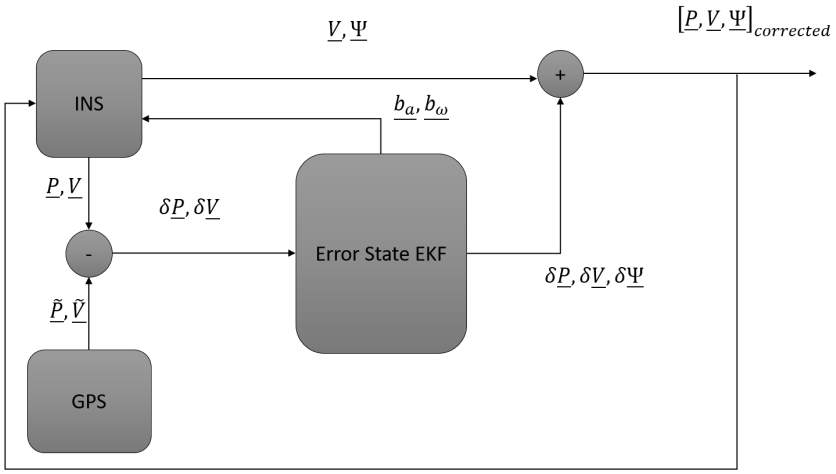

Figure 2.2: Loosely Coupled GPS/INS Block Diagram

The state of the error state EKF is given as

$$\delta\underline{X} = \begin{bmatrix} \delta\underline{P} & \delta\underline{V} & \delta\underline{\Psi} & \underline{b}_a & \underline{b}_\omega \end{bmatrix}^T \tag{2.14}$$

and the full state of the Loosely Coupled GPS/INS filter is given as

$$\underline{X} = \begin{bmatrix} \underline{P} & \underline{V} & \underline{\Psi} \end{bmatrix}^T \tag{2.15}$$

where $\underline{P}$ is the estimated position, $\underline{V}$ is the estimated velocity, $\underline{\Psi}$ is the estimated attitude, and $\underline{b}_a$ and $\underline{b}_\omega$ are the estimated accelerometer and gyroscope biases.

2.4.1 INS Propagation

The INS takes in accelerometer and gyroscope measurements and uses the bias estimates from the EKF to correct them as shown in Equations (2.16 - 2.17).

$$\underline{a}_{INS} = \underline{a}_{IMU} - \underline{b}_a \tag{2.16}$$

$$\underline{\omega}_{INS} = \underline{\omega}_{IMU} - \underline{b}_\omega \tag{2.17}$$

where $\underline{a}_{INS}$ and $\underline{\omega}_{INS}$ are the corrected accelerometer and gyroscope measurements, $\underline{a}_{IMU}$ and $\underline{\omega}_{IMU}$ are the raw accelerometer and gyroscope measurements, and $\underline{b}_f$ and $\underline{b}_\omega$ are the estimated accelerometer and gyroscope biases from the EKF. Comparing Equations (2.16 - 2.17) to Equations (2.11 - 2.12) shows that while the estimates of the IMU biases approach the true bias values, the corrected accelerometer and gyroscope measurements approach the true accelerations and angular rates with added white noise.

The corrected gyroscope measurements are then used to propagate the rotation matrix that represents the missalignment between the coordinate frame used for navigation, East-North-Up (ENU), and the body frame of the vehicle. This rotation matrix is initialized with a 3-2-1 Euler rotation, and is propagated in the process described below. More information on these coordinate frames can be found in Appendix A. The propagation of the rotation matrix occurs in two forms based on the magnitude of the gyroscope measurements. When that magnitude is above some threshold (much smaller than the rotation rate of the Earth) the Rodriguez formula shown in Equation (2.18) is used, otherwise a simple Euler integration is performed as shown in Equation (2.19),

$$\boldsymbol{C}_{b+}^{b-} = \boldsymbol{I}_3 + \frac{\sin|\underline{\alpha}|}{|\underline{\alpha}|}[\underline{\boldsymbol{\alpha}}\wedge] + \frac{1-\cos|\underline{\alpha}|}{|\alpha|^2}[\underline{\boldsymbol{\alpha}}\wedge]^2 \tag{2.18}$$

$$\boldsymbol{C}_{b+}^{b-} = \boldsymbol{I}_3 + [\underline{\boldsymbol{\alpha}}\wedge] \tag{2.19}$$

where

$$\underline{\alpha} = \underline{\omega}_{INS}\Delta t \tag{2.20}$$

$\boldsymbol{C}_{b+}^{b-}$ represents the rotation from the previous time step to the current time step, $\underline{\alpha}$ is the angle increment over the time step, and $[\underline{\boldsymbol{\alpha}}\wedge]$ is the skew symmetric form of the angle increment, which is given as

$$[\underline{\boldsymbol{\alpha}}\wedge] = \begin{bmatrix} 0 & -\underline{\alpha}_Z & \underline{\alpha}_Y \\ \underline{\alpha}_Z & 0 & -\underline{\alpha}_X \\ -\underline{\alpha}_Y & \underline{\alpha}_X & 0 \end{bmatrix} \tag{2.21}$$

The new rotation from the body frame to the navigation frame can then be given as

$$\boldsymbol{C}_b^n(+) = \boldsymbol{C}_b^n(-)\boldsymbol{C}_{b+}^{b-} \tag{2.22}$$

where $(-)$ and $(+)$ mark the rotation before and after the update. Note that in higher accuracy systems it is common to include the rotation of the Earth in this update [16], but because of the quality of the sensors used in this thesis, this term is neglected.

The accelerometer measurements are then rotated into the navigation frame as shown in Equation (2.23).

$$\underline{f}_n = \boldsymbol{C}_b^n(+)\underline{a}_{INS} - \underline{g} \tag{2.23}$$

where

$$\underline{g} = \begin{bmatrix} 0 \\ 0 \\ 9.81 \end{bmatrix} \tag{2.24}$$

$\underline{f}_n$ is the specific force in the navigation frame and $\underline{g}$ is the gravity vector. The value of the gravity vector can be made a function of latitude and altitude, but is considered constant in this work due to the vehicles operating in a local area. The specific force in the navigation frame is then numerically integrated to generate the velocity and position values from the INS.

The attitude of the vehicle is represented in Euler angles in this thesis. These angles can be backed out from the rotation matrix using the following expressions

$$\phi = \arctan\left(\frac{\boldsymbol{C}^n_{b,(3,2)}}{\boldsymbol{C}^n_{b,(3,3)}}\right) \tag{2.25}$$

$$\theta = -\arcsin(\boldsymbol{C}^n_{b,(3,1)}) \tag{2.26}$$

$$\psi = \arctan\left(\frac{\boldsymbol{C}^n_{b,(2,1)}}{\boldsymbol{C}^n_{b,(1,1)}}\right) \tag{2.27}$$

where ϕ is the roll angle, θ is the pitch angle, and ψ is the yaw angle. The estimates of position, velocity, and attitude are then stored to be corrected later by the EKF as shown in Equation (2.28)

$$\underline{X}_{INS} = \begin{bmatrix} \underline{P}_{INS} & \underline{V}_{INS} & \underline{\Psi}_{INS} \end{bmatrix}^T \tag{2.28}$$

where $\underline{X}_{INS}$ is the stored INS state, $\underline{P}_{INS}$ is the INS position solution, $\underline{V}_{INS}$ is the INS velocity solution, and $\underline{\Psi}_{INS}$ is the INS attitude solution. Note that the IMU biases are not contained within the INS state. Since these values are maintained solely by the error state EKF, they must be appended to the corrected INS state to generate the full state. The necessary estimates are differenced with the corresponding external measurements from GPS when they are available. These differences, or errors, are then used to propagate the error states in the EKF.

2.4.2 Loosely Coupled GPS/INS Time Update

The Loosely Coupled GPS/INS filter takes in the INS and error information in order to generate the navigation solution. The INS information is used to propagate the error states, and the error information is used in the measurement update to correct the error states. The non-linear equations of motion used to propagate the error states are given below.

$$\delta\dot{\underline{\Psi}} = \boldsymbol{C}^n_b \underline{b}_g \tag{2.29}$$

$$\delta\dot{\underline{V}} = -[\underline{\boldsymbol{f}}_{\boldsymbol{n}} \wedge]\delta\underline{\Psi} + \boldsymbol{C}^n_b \underline{b}_a \tag{2.30}$$

$$\delta\dot{\underline{P}} = \delta\underline{V} \tag{2.31}$$

In order to more easily perform the time update operations, the Jacobian of Equations (2.29 - 2.31), along with Equation (2.13) is computed as

$$\boldsymbol{J} = \begin{bmatrix} \varnothing_3 & \boldsymbol{I}_3 & \varnothing_3 & \varnothing_3 & \varnothing_3 \\ \varnothing_3 & \varnothing_3 & -[\underline{\boldsymbol{f}}_{\boldsymbol{n}} \wedge] & \boldsymbol{C}_b^n & \varnothing_3 \\ \varnothing_3 & \varnothing_3 & \varnothing_3 & \varnothing_3 & \boldsymbol{C}_b^n \\ \varnothing_3 & \varnothing_3 & \varnothing_3 & \frac{-1}{\tau_a}\boldsymbol{I}_3 & \varnothing_3 \\ \varnothing_3 & \varnothing_3 & \varnothing_3 & \varnothing_3 & \frac{-1}{\tau_g}\boldsymbol{I}_3 \end{bmatrix} \tag{2.32}$$

where τ_a and τ_g are the time constants of the accelerometer and gyroscope biases which can be found in Table 2.1. Note that the biases are represented and propagated in the body frame. The state is then propagated as shown below

$$\delta\hat{\underline{X}}^- = \boldsymbol{J}\delta\hat{\underline{X}} \tag{2.33}$$

$$\delta\hat{\underline{X}}_{k+1}^- = \delta\hat{\underline{X}}_k^- + \delta\hat{\underline{X}}^-\Delta t \tag{2.34}$$

where $\delta\hat{\underline{X}}$ is the time derivative of the error state, the superscript $^-$ denotes the values prior to the measurement update, and Δt is the time step. The time derivative of the error state is computed using Equation (2.33), and then is numerically integrated as shown in Equation (2.34) in order to generate the prior error state estimate. The state covariance is propagated as shown below.

$$\boldsymbol{P}^- = (\boldsymbol{I}_{15} + \boldsymbol{J}\Delta t)\boldsymbol{P}(\boldsymbol{I}_{15} + \boldsymbol{J}\Delta t)^T + \boldsymbol{B}_\nu \boldsymbol{Q}\boldsymbol{B}_\nu^T \tag{2.35}$$

$$\boldsymbol{Q} = \begin{bmatrix} \sigma_v^2\boldsymbol{I}_3 & \varnothing_3 & \varnothing_3 & \varnothing_3 & \varnothing_3 \\ \varnothing_3 & \sigma_a^2\boldsymbol{I}_3 & \varnothing_3 & \varnothing_3 & \varnothing_3 \\ \varnothing_3 & \varnothing_3 & \sigma_\omega^2\boldsymbol{I}_3 & \varnothing_3 & \varnothing_3 \\ \varnothing_3 & \varnothing_3 & \varnothing_3 & \sigma_{b_a}^2\boldsymbol{I}_3 & \varnothing_3 \\ \varnothing_3 & \varnothing_3 & \varnothing_3 & \varnothing_3 & \sigma_{b_g}^2\boldsymbol{I}_3 \end{bmatrix} \tag{2.36}$$

$$\boldsymbol{B}_\nu = \begin{bmatrix} \boldsymbol{C}_b^n & \varnothing_3 & \varnothing_3 & \varnothing_3 & \varnothing_3 \\ \varnothing_3 & \boldsymbol{C}_b^n & \varnothing_3 & \varnothing_3 & \varnothing_3 \\ \varnothing_3 & \varnothing_3 & \boldsymbol{C}_b^n & \varnothing_3 & \varnothing_3 \\ \varnothing_3 & \varnothing_3 & \varnothing_3 & \boldsymbol{I}_3 & \varnothing_3 \\ \varnothing_3 & \varnothing_3 & \varnothing_3 & \varnothing_3 & \boldsymbol{I}_3 \end{bmatrix} \tag{2.37}$$

where $\boldsymbol{P}$ is the state covariance, σ_v, σ_a, σ_ω, σ_{b_a}, and σ_{b_g} are the 1σ standard deviation values of the white noise on the state estimation error respectively, $\boldsymbol{Q}$ is the process covariance, and $\boldsymbol{B}_\nu$ is the matrix which maps the process covariance into the state domain. The values within the process covariance matrix were hand-tuned, and can be found in Table 2.1.

Table 2.1: Loosely Coupled GPS/INS Filter Values

σ_v	3.16e-4	m/s
σ_a	3.18e-4	m/s^2
σ_ω	3.16e-2	rad/s
σ_{b_a}	0.01	m/s^2
σ_{b_g}	0.01	rad/s
τ_a	120	s^{-1}
τ_g	60	s^{-1}

Once the execution of the time update is complte, there are two paths forward. If there are no external measurements available from GPS, then the INS solution computed previously is then corrected by the following equations.

$$\underline{X}_{corr} = \underline{X}_{INS} + \delta\underline{X}^-_{(1:9)} \tag{2.38}$$

$$\underline{X} = \begin{bmatrix} \underline{X}_{corr} \\ \delta\underline{X}^-_{(10:15)} \end{bmatrix} \tag{2.39}$$

Then, the components of the error state that are not bias terms are reset to zero. If external measurements are available, the measurement update is performed as described in the next section.

2.4.3 Loosely Coupled GPS/INS Measurement Update

When the measurement update is performed, the error information generated from the differencing of the INS state and the corresponding external measurements is used to correct the error state from the time update. This error information is expressed as

$$\delta \underline{Y} = \underline{Y} - \boldsymbol{H}\underline{X} \tag{2.40}$$

$$\underline{Y} = \begin{bmatrix} \underline{P}_{GPS} \\ \underline{V}_{GPS} \end{bmatrix} \tag{2.41}$$

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{I}_3 & \varnothing_3 & \varnothing_3 & \varnothing_3 & \varnothing_3 \\ \varnothing_3 & \boldsymbol{I}_3 & \varnothing_3 & \varnothing_3 & \varnothing_3 \end{bmatrix} \tag{2.42}$$

where $\delta \underline{Y}$ is the measurement error, Y is the set of GPS measurements themselves, and $\boldsymbol{H}$ is the measurement mapping matrix.

The error state is then corrected as shown in Equation (2.43) [18]

$$\delta \underline{X}^+ = \delta \underline{X}^- + \boldsymbol{L}(\delta \underline{Y} - \boldsymbol{H}\delta \underline{X}) \tag{2.43}$$

$$\boldsymbol{L} = \boldsymbol{P}^- \boldsymbol{H}^T (\boldsymbol{H}\boldsymbol{P}^- \boldsymbol{H}^T + \boldsymbol{R})^{-1} \tag{2.44}$$

$$\boldsymbol{R} = \begin{bmatrix} \sigma^2_{P_{GPS}} \boldsymbol{I}_3 & \varnothing_3 \\ \varnothing_3 & \sigma^2_{V_{GPS}} \boldsymbol{I}_3 \end{bmatrix} \tag{2.45}$$

where $\boldsymbol{L}$ is the Kalman gain, R is the measurement covariance matrix, and σ_{PGPS} and $\sigma_{V_{GPS}}$ are the 1σ standard deviation values of the white noise on the GPS measurements respectively. The state covariance is then propagated using the Kalman gain as given in Equation (2.46).

$$\boldsymbol{P}^+ = (\boldsymbol{I}_{15} - \boldsymbol{L}\boldsymbol{H})\boldsymbol{P}^- \tag{2.46}$$

Once the measurement update has been executed, the full state is corrected with the error states in a similar fashion to the process described by Equations (2.38 - 2.39), shown below.

$$\underline{X}_{corr} = \underline{X}_{INS} + \delta\underline{X}^{+}_{(1:9)} \tag{2.47}$$

$$\underline{X} = \begin{bmatrix} \underline{X}_{corr} \\ \delta\underline{X}^{+}_{(10:15)} \end{bmatrix} \tag{2.48}$$

Then, just like the process described for the correction in the time update, the components of the error state that are not bias terms are reset to zero. This completes the description of the operation of the standard Loosely Coupled GPS/INS Navigation algorithm used in this thesis.

Chapter 3

Interacting Multiple Model Radar Tracking

The Interacting Multiple Model (IMM) algorithm is a type of Gaussian sum filter that mixes multiple estimates based on different models using a mixing probability matrix to modify the weights of each model [19]. This mixing probability is calculated based on the previous probability, the likelihoods of each model, and a state switching matrix describing the probability of the true system transitioning between component filter models. Often, the IMM algorithm is used to combine multiple estimation filters, such as KFs, EKFs, and PFs. Because the mixing occurs to the state estimates themselves, the filters can be different forms as long as the state is the same. A block diagram of the operation of a two-model IMM is shown in Figure 3.1. The first operation of the IMM algorithm is the interaction of the states corresponding to the component models. The residuals from each model are then used to generate model likelihoods within the model probability update. Finally, the states are mixed in order to generate the mixed state estimate.

IMM filters are generally used when a system exhibits multiple dynamic modes that would be difficult or inefficient to estimate with a single filter, or when there are multiple estimation algorithms that perform better in different operating conditions. Different etsimation algorithms can be run in parallel and mixed, resulting in a better estimate than the filters produce individually. IMMs are often used in radar tracking applications i.e. Air Traffic Control (ATC), air-space monitoring, etc. where the tracked targets can be described by different models corresponding to different common maneuvers [20]. When tracking passenger aircraft, for example, a kinematic KF can be developed for constant bearing flight, and an EKF can be developed for a non-linear turning maneuver. These two filters can be combined with the IMM algorithm,

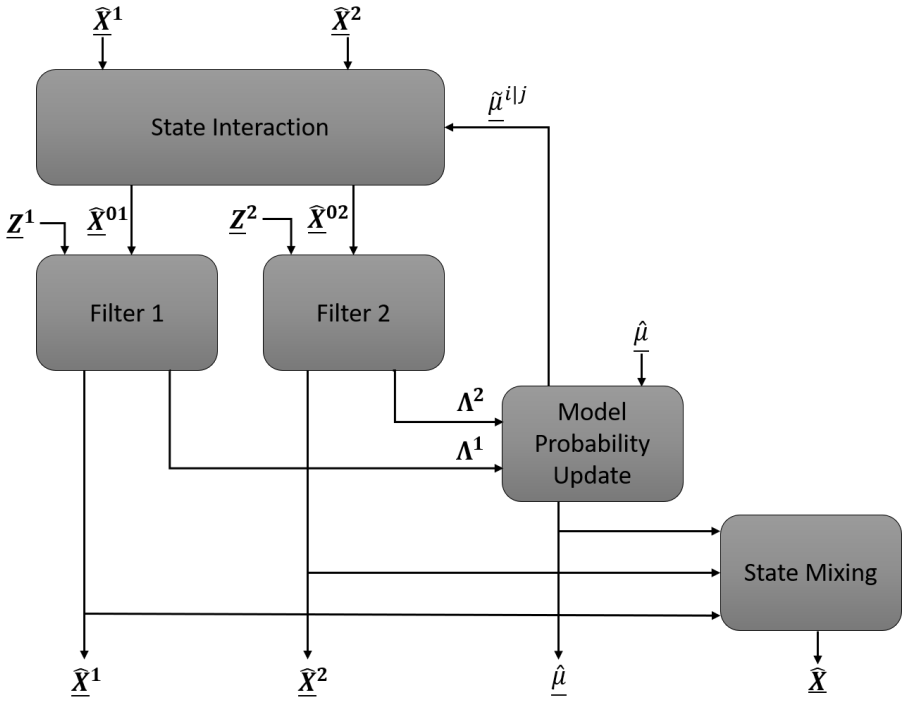

Figure 3.1: IMM Block Diagram

and any transitions the tracked aircraft makes between the flight modes will be reflected in the mixing probabilities, which in turn will change the weighting of the models on the final state estimate. This method allows for specialized filters designed for specific situtations, which can improve the overall output of the algorithm.

This thesis uses three component models within the IMM framework to generate the tracking solution: the Constant Velocity (CV) model, the Constant Acceleration (CA) model, and the Constant Turn (CT) model. All three of these models are used in an EKF estimation scheme, and utilize measurements from a consumer grade radar unit. The description of the CV, CA, and CT models, along with the properties of the radar measurements used, can be found in the following sections.

3.1 Radar Errors

Radar units generally provide measurements of range and range rate. These measurements are derived in a similar fashion to those of GPS. All forms of radar measurements depend on the process of bouncing electromagnetic (EM) waves off of some target object and processing

the returned signal. The Radar Range Equation (RRE) provides a closed form method of approximating the expected signal-to-noise ratio (SNR) of the reflected EM signal. Because the accuracy of the measurements is so heavily dependent on signal quality, the RRE is commonly used to generate high-level predictions of radar system performance [21]. The tracking form of the RRE is shown in Equation (3.1).

$$SNR = \frac{P_{avg} A_e G \sigma}{(4\pi)^2 R^4 k * T_o * PRF * L_s * F} \tag{3.1}$$

The definitions for the values within Equation (3.1) can be found in Table 3.1.

Table 3.1: Definitions of Variables within Radar Range Equation

Variable	**Definition**	**Units**
P_{avg}	Average Transmitted Power	W
A_e	Effective Antenna Area	m
G	Antenna Gain	N/A
σ	Radar Cross-Section	m
R	Target Range	m
k	Boltzmann's Constant	$\frac{m^2 kg}{s^2 K}$
T_o	Reference Temperature	K
PRF	Pulse Repetition Frequency	$1/s$
F	Noise Figure	N/A
L_s	Total System Losses	N/A

Range measurements are computed by measuring the elapsed time between the radiation and return of an EM signal which reflects off of some target object. This elapsed time is then multiplied by the speed of light (the speed at which the EM waves travel) to generate a range measurement. The lower bound of the theoretical covariance of range measurements from a simple pulsed radar is given as

$$\sigma_R^2 >= \frac{c^2 \tau}{8SNR * B_{IF}} \tag{3.2}$$

where σ_R is the standard deviation of the range error, c is the speed of light, τ is the length of the pulse, and B_{IF} is the bandwidth of the intermediate frequency (IF) filter.

Range rate measurements are generated by measuring the frequency of the returned EM signal and comparing it to the transmitted signal frequency. The change in frequency, or Doppler shift, of the signal is found using digital Discrete Fourier Transform (DFT) techniques

which isolate the frequencies of the return signal. The Doppler shift can then be used to directly find the relative velocity between the radar unit and the target. The expression that defines the lower bound of the theoretical covariance of Doppler measurements is given in Equation (3.3)

$$\sigma_f^2 >= \frac{3f_s^2}{\pi^2 SNR * M^3} \tag{3.3}$$

where σ_f is the standard deviation of the range rate error, f_s is the sampling frequency of the receiver, and M is the number of pulses received.

Some types of radar units are also able to generate measurements of the angle to the target. In modern digital radar systems, this is achieved by steering the beam of the EM wave and mapping the returned power to the scanned space. The lower bound of the theoretical covariance of the angle-of-arrival measurements are given in Equation (3.4) [22]

$$\sigma_\lambda^2 >= \frac{\theta_{3dB}^2}{N * SNR * \alpha^2} \tag{3.4}$$

where σ_λ is the standard deviation of the angular error, θ_{3dB} is the three decibel beamwidth, or the inclusive angle that contains half of the transmitted power, N is the number of pulses in the angular scan, and α is a scale factor related to the beamwidth.

In general, it is acceptable to model the noise on all three radar measurements as zero mean white noise [23]. Therefore, the measurements are given as

$$\tilde{r} = r + \nu_r \tag{3.5}$$

$$\tilde{\dot{r}} = \dot{r} + \nu_{\dot{r}} \tag{3.6}$$

$$\tilde{\lambda} = \lambda + \nu_\lambda \tag{3.7}$$

where r, $\dot{r}$, and λ are the true target range, range rate, and angular values. ν_r, $\nu_{\dot{r}}$, and ν_λ are the additive white noise terms on the corresponding measurements, and˜indicates the measurements of the corresponding values. In this thesis, the measurements are assumed to be relative

to the CG of the vehicles. In reality, the measurements would be made to the body of the vehicle which could cause a varying offset from the CG depending on the size and shape of the vehicle, and on the relative attitude. These errors are relatively small compared to the accuracy of the radar unit, so this assumption is deemed acceptable. It's also possible that the radar unit could be measuring to multiple points on the vehicle. These effects are neglected in this work.

3.2 IMM Algorithm

The IMM algorithm consists of three main operations: state interaction, probability update, and state combination [24]. Within the state interaction operation, the interaction state and covariance for each model is found. This process is shown in Equations (3.8 - 3.9) below

$$\underline{\hat{X}}^{0j} = \sum_{i=1}^{N} \underline{\hat{X}}^{i} \underline{\tilde{\mu}}^{i|j} \tag{3.8}$$

$$\hat{\boldsymbol{P}}^{0j} = \sum_{i=1}^{N} \underline{\tilde{\mu}}^{i|j} [\hat{\boldsymbol{P}}^{i} + (\underline{\hat{X}}^{i} - \underline{\hat{X}}^{0j})(\underline{\hat{X}}^{i} - \underline{\hat{X}}^{0j})^{T}] \tag{3.9}$$

where i and j denote the component model index as they are stepped through, N is the number of component models, $\underline{\hat{X}}^{0j}$ is the interaction state, $\hat{\boldsymbol{P}}^{0j}$ is the interaction covariance, and $\underline{\tilde{\mu}}^{i|j}$ is the conditional model probability for each model. This conditional model probability is based on the previous model probability and the state switching matrix. The expression that defines the conditional model probability is given as

$$\underline{\tilde{\mu}}^{i|j} = \frac{1}{\bar{\psi}^{j}} \boldsymbol{\Pi}_{(i,j)} \underline{\hat{\mu}}^{i} \tag{3.10}$$

where $\boldsymbol{\Pi}$ is the state switching matrix, $\underline{\hat{\mu}}$ is the model probability, and $\bar{\psi}$ is the conditional probability normalization term which is shown in Equation (3.11).

$$\bar{\psi} = \sum_{i=1}^{N} \boldsymbol{\Pi}_{(i,j)} \underline{\hat{\mu}}^{i} \tag{3.11}$$

Selection of the values of the state switching matrix $\boldsymbol{\Pi}$ is important in determining the performance of the IMM algorithm. This matrix represents the probability of the tracked trajectory

switching modes during its motion. The state switching matrix is dimension (NxN), where the diagonals represent the likelihood of the trajectory staying in the corresponding mode, and the off-diagonals represent its likelihood to 'switch' to another mode. For example, an identity state switching matrix would indicate that it is extremely unlikely that the trajectory would switch modes, whereas a state switching matrix where every term is made up of the value 1/N would imply that no matter which mode the trajectory is currently undergoing, it is equally likely to stay in that mode as it is to switch to any of the other modes. Unfortunately, there are no well documented resources which describe the process of tuning this matrix, so the tuning is generally done 'by hand' until the desired performance is met [24]. It is common to have the first model in the system be a default mode in which the trajectory is most likely to be found, and have the following models decrease in likelihood.

During the probability update, the likelihood of each model is computed as shown in Equation (3.12)

$$\underline{\Lambda}^j = \frac{1}{\sqrt{2\pi \boldsymbol{S}^j}} \exp\left[-\frac{1}{2}(\underline{Z}^j)^T(\boldsymbol{S}^j)^{-1}\underline{Z}^j\right] \tag{3.12}$$

where $\underline{\Lambda}$ is the model likelihood, $\underline{Z}$ is the model innovation, and $\boldsymbol{S}$ is the innovation covariance. Equation (3.12) describes the Gaussian distribution from which the model weights are drawn. This equation is very important to the operation of the IMM algorithm. A visual representation of this operation is shown in Figure 3.2. This figure represents the likelihood selection of two models. The first model residual, represented by the black line, has a mean far from zero and a relatively high 1σ value. The second model residual, represented by the red line, has a mean relatively close to zero and a 1σ value much lower than the first model. Upon inspection, it can be seen that Equation (3.12) is in the form of a Gaussian distribution evaluated at $\mu = 0$. This can be seen in Figure 3.2 as the two distributions are sampled at zero. This example makes it clear that the models with residuals closer to zero mean and with lower 1σ values are weighted more heavily than otherwise. This idea is the keystone of the weight selection within the IMM algorithm.

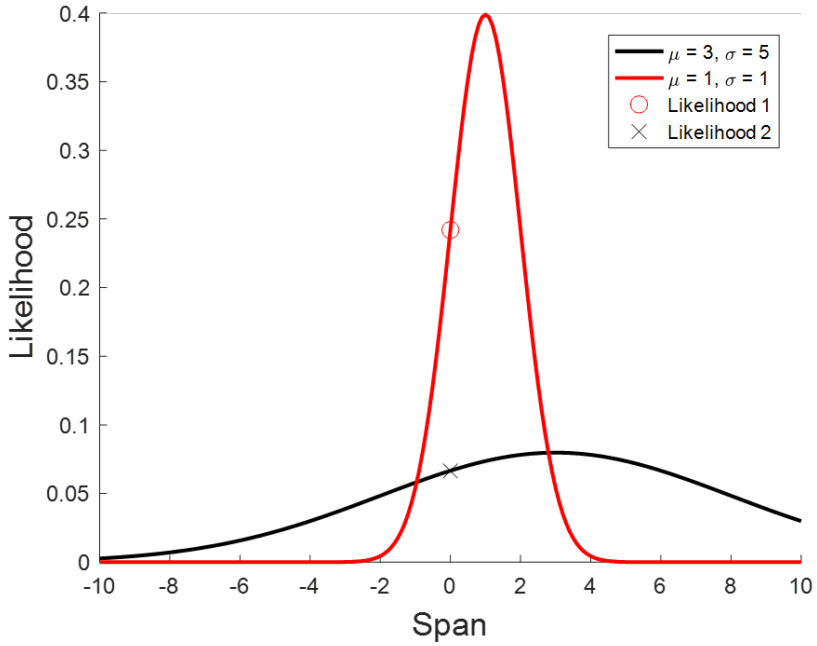

Figure 3.2: IMM Likelihood Selection Example

These likelihoods, or weights, are then normalized with the following expressions

$$\underline{\hat{\mu}}^j = \frac{1}{c}\underline{\Lambda}^j \underline{c}^j \tag{3.13}$$

$$c = \sum_{i=1}^{N} \underline{\Lambda}^i \underline{c}^i \tag{3.14}$$

where c is the normalizing magnitude and $\underline{c}$ is an Nx1 matrix which weights the model probabilities directly. This matrix is generally set to unity, but can be used to fine tune the mixing process after setting the state switching matrix.

Finally, the component model states are mixed together using the newly generated model probabilities. This mixing process is shown in Equation (3.15)

$$\underline{\hat{X}}_{mix} = \sum_{i=1}^{N} \underline{\hat{X}}^i \underline{\hat{\mu}}^i \tag{3.15}$$

where $\underline{X}_{mix}$ is the mixed state. Then, the mixed state covariance is computed as shown in Equation (3.16), completing the IMM operations.

$$\hat{\boldsymbol{P}}_{mix} = \sum_{i=1}^{N} \underline{\hat{\mu}}^{i} [\hat{\boldsymbol{P}}^{i} + (\underline{\hat{X}}^{i} - \underline{\hat{X}}_{mix})(\underline{\hat{X}}^{i} - \underline{\hat{X}}_{mix})^{T}] \tag{3.16}$$

At this point, the component filter states can be set to the mixed state or the interacted states, or they can be left to run independently.

3.3 Illustrative IMM Example Trajectory

In order to more clearly describe the IMM algorithm, a simple two dimensional simulation was generated. The system used in the simulation is a simple ground vehicle model which moves through four different phases of operation. The first phase is the constant acceleration phase, where the vehicle travels in a straight line, but undergoes a constant linear acceleration of 0.8 m/s^2. The second phase of the maneuver is the constant velocity phase, where the vehicle continues to travel in a straight line and maintains a velocity of 20 m/s throughout this phase. In the third phase, the vehicle is given a constant steer angle and maintains constant longitudinal velocity, therefore undergoing a constant turn maneuver of 3 rad/s. The fourth and final phase consists of the vehicle undergoing a sinusoidal maneuver with a frequency of 0.5 Hz. This trajectory can be seen in Figure 3.3. The vehicle starts at the origin, and travels due east during the constant acceleration and constant velocity portions of the trajectory. The vehicle then performs the constant turn rate maneuver in the counter-clockwise direction. Finally, the vehicle stops the constant rate turn and performs the sinusoidal maneuver to the southeast. This trajectory is used in the following sections to demonstrate each component model, and to illustrate the operation of the IMM algorithm.

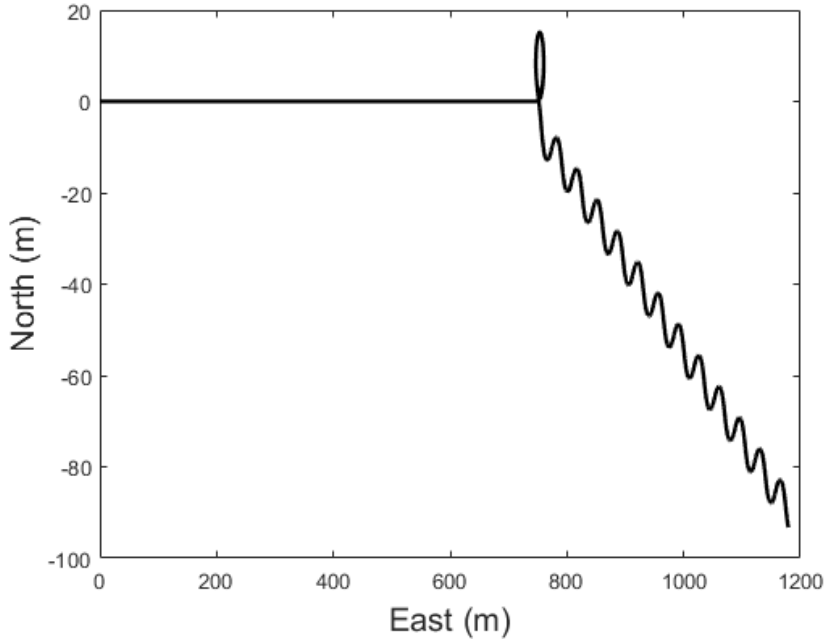

Figure 3.3: IMM Example Trajectory

3.4 IMM Model Development

As mentioned previously, three models were developed for use in this thesis. All three models are constructed of similar dimensions, and all are propagated through the same Kalman filter equations with non-linear measurement models. Each model operates with the state shown in Equation (3.17)

$$\underline{\hat{X}}_{IMM} = \begin{bmatrix} \underline{P} & \underline{V} & \underline{A} \end{bmatrix}^T \tag{3.17}$$

where $\underline{P}$ is the estimate of position, $\underline{V}$ is the estimate of velocity, and $\underline{A}$ is the estimate of acceleration. The state dimensions for each model do not necessarily need to be the same, but integration of models with different states can complicate and add unwanted transients to the IMM response [25]. The IMM state can be expressed in either the radar coordinate frame or the navigation coordinate frame, depending on the implementation. In this thesis, the IMM state is expressed in the radar coordinate frame and rotated into the navigation frame later. The CV, CA, and CT models are described in the following sections.

3.4.1 Constant Velocity Model

The CV model is based on the discrete differentiation of the position. The acceleration is modelled as zero mean with Gaussian white noise. The state transition matrix is shown in Equation (3.18).

$$\boldsymbol{A}_{CV} = \begin{bmatrix} \boldsymbol{I}_3 & \Delta t \boldsymbol{I}_3 & \varnothing_3 \\ \varnothing_3 & \boldsymbol{I}_3 & \varnothing_3 \\ \varnothing_3 & \varnothing_3 & \varnothing_3 \end{bmatrix} \tag{3.18}$$

The process noise covariance of all three models are represented simply as a diagonal matrix with the corresponding terms populated with tunable variance values in order to simplify operation, as shown in Equation (3.19).

$$\boldsymbol{Q}_{IMM} = \begin{bmatrix} q_P \Delta t \boldsymbol{I}_3 & \varnothing_3 & \varnothing_3 \\ \varnothing_3 & q_V \Delta t \boldsymbol{I}_3 & \varnothing_3 \\ \varnothing_3 & \varnothing_3 & q_A \Delta t \boldsymbol{I}_3 \end{bmatrix} \tag{3.19}$$

The variables q_P, q_V, and q_A are the process noise densities on position, velocity, and acceleration respectively. These values can be tuned for each model to achieve the desired steady-state performance of each component filter, but were set to unity in this work. Tracking performance of the CV filter using the 2 degree of freedom example trajectory shown from the previous section is shown in Figure 3.4. It is clear that the CV model tracks relatively well during the CA and especially the CV portions of the trajectory, but is highly degraded during the CT segment. In the final segment of the trajectory, it is clear that the CV model tracks better in the East direction as the vehicle is traveling with essentially constant velocity, but the estimates in the North direction are poor due to the sinusoidal motion.

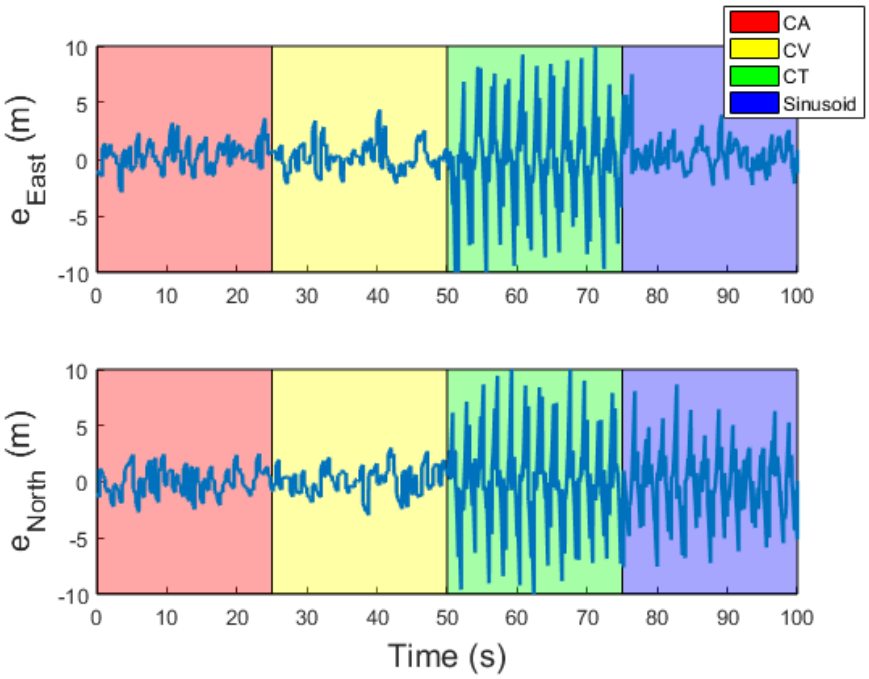

Figure 3.4: Constant Velocity Kinematic Model Performance

3.4.2 Constant Acceleration Model

The CA model is in a similar form to the CV model. The velocity terms from the CV state transition matrix are differentiated once more, resulting in position and velocity estimates which are coupled with the acceleration estimate, which is assumed constant [13]. The state transition matrix of the CA model is given in Equation (3.20).

$$\boldsymbol{A}_{CA} = \begin{bmatrix} \boldsymbol{I}_3 & \Delta t \boldsymbol{I}_3 & 0.5\Delta t^2 \boldsymbol{I}_3 \\ \varnothing_3 & \boldsymbol{I}_3 & \Delta t \boldsymbol{I}_3 \\ \varnothing_3 & \varnothing_3 & \boldsymbol{I}_3 \end{bmatrix} \tag{3.20}$$

The process noise covariance for the CA model is of the same form as that described in Equation (3.19), but with different noise densities in order to achieve desirable steady-state results with the different model. Tracking performance of the CA filter using the 2 degree of freedom example trajectory shown from the previous section can be seen in Figure 3.5. The CA model performs slightly better during the CA portion of the trajectory, but otherwise performs almost exactly the same as the CV model. This is expected, since the addition of the non-zero

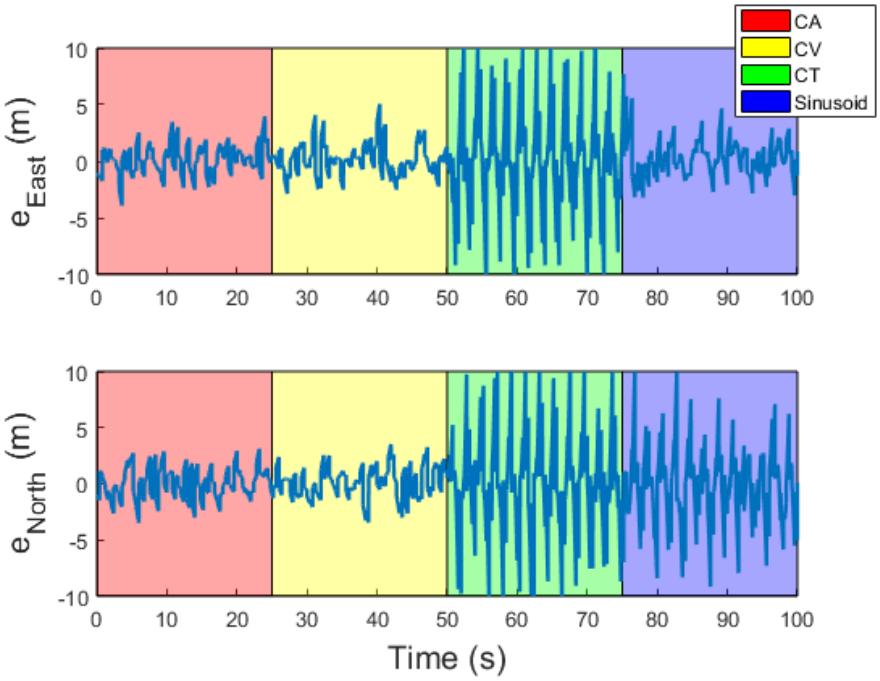

Figure 3.5: Constant Acceleration Kinematic Model Performance

acceleration estimates improves the velocity estimates but has little to no effect on the position estimates.

3.4.3 Constant Turn Model

The CT model emulates a trajectory moving along a curve which is flat along the East-North plane with a constant turning rate [26]. This model has been expanded in order to yield the proper sized state, but was initially intended for solely planar operation. The state transition

matrix for the CT model is shown in Equation (3.21).

$$\boldsymbol{A}_{CT} = \begin{bmatrix} 1 & \varnothing & \varnothing & \frac{\sin(\omega\Delta t)}{\omega} & -\frac{1-\cos(\omega\Delta t)}{\omega} & \varnothing & \varnothing & \varnothing & \varnothing \\ \varnothing & 1 & \varnothing & \frac{1-\cos(\omega\Delta t)}{\omega} & \frac{\sin(\omega\Delta t)}{\omega} & \varnothing & \varnothing & \varnothing & \varnothing \\ \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing \\ \varnothing & \varnothing & \varnothing & \cos(\omega\Delta t) & -\sin(\omega\Delta t) & \varnothing & \varnothing & \varnothing & \varnothing \\ \varnothing & \varnothing & \varnothing & \sin(\omega\Delta t) & \cos(\omega\Delta t) & \varnothing & \varnothing & \varnothing & \varnothing \\ \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing \\ \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing \\ \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing \\ \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing \end{bmatrix} \tag{3.21}$$

where ω is the turn-rate of the maneuver.

Three different methods of setting the value of the turn-rate in the CT model were explored. The first, and simplest method, is to set the turn-rate to a constant value which ideally lines up with the expected turn rate in the maneuvers being tracked. This method has some merit, in that on the off-chance the turn-rate happens to line up, the results are quite good, as shown in Figure 3.6.

The second method involves estimating the current turn-rate based the current estimates of velocity and acceleration [27]. The computation of this estimate of turn-rate is expressed as

$$\omega = \frac{\parallel A \parallel}{\parallel V \parallel} = \frac{\sqrt{\ddot{x}^2 + \ddot{y}^2 + \ddot{z}^2}}{\sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}} \tag{3.22}$$

where $\ddot{x}$, $\ddot{y}$, $\ddot{z}$, $\dot{x}$, $\dot{y}$, and $\dot{z}$ are the component accelerations and velocities respectively. This method improves the estimate drastically during the non-turning maneuvers, but provides an estimate during the CT phase of the trajectory that is only slightly better than the CV and CA models, as shown in Figure 3.7. The estimated turn-rate is shown in Figure 3.8. While the estimated turn rate follows the trend of the true turn rate relatively well, there is substantial error which is the apparent cause of this method not performing well.

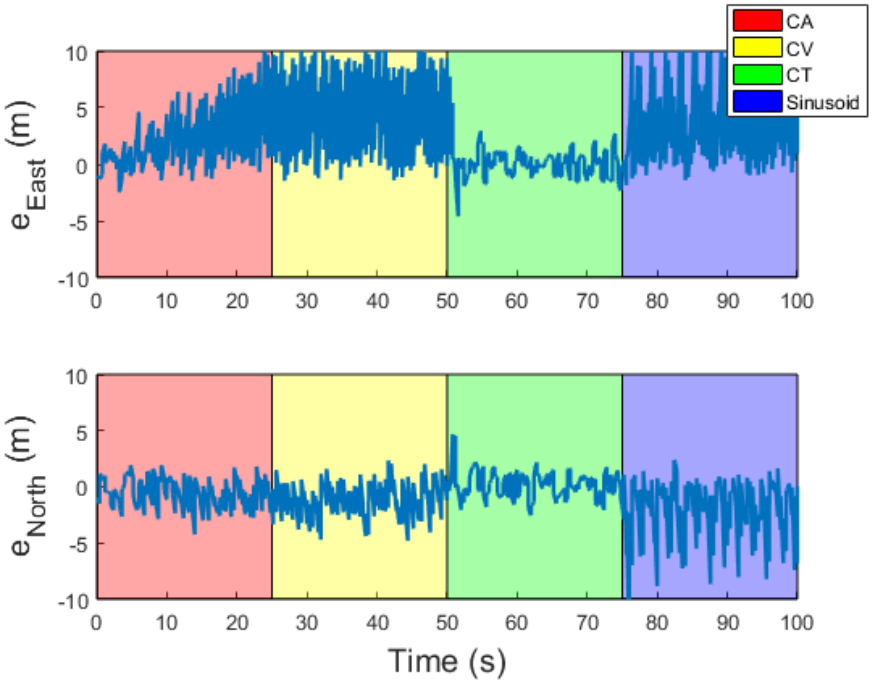

Figure 3.6: Constant Turn Kinematic Model Performance with Single Turn Rate Value

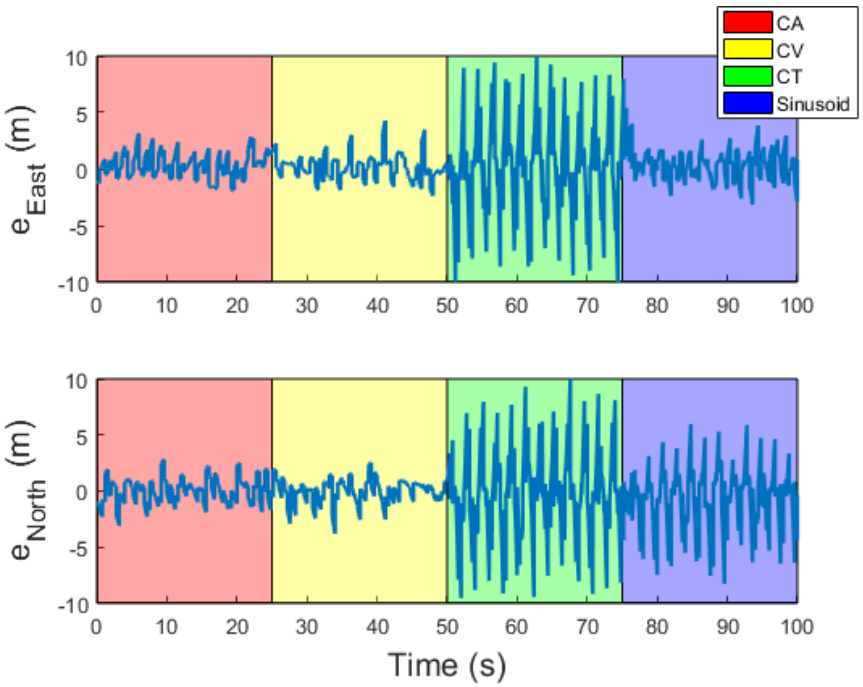

Figure 3.7: Constant Turn Kinematic Model Performance with Varying Turn Rate Value

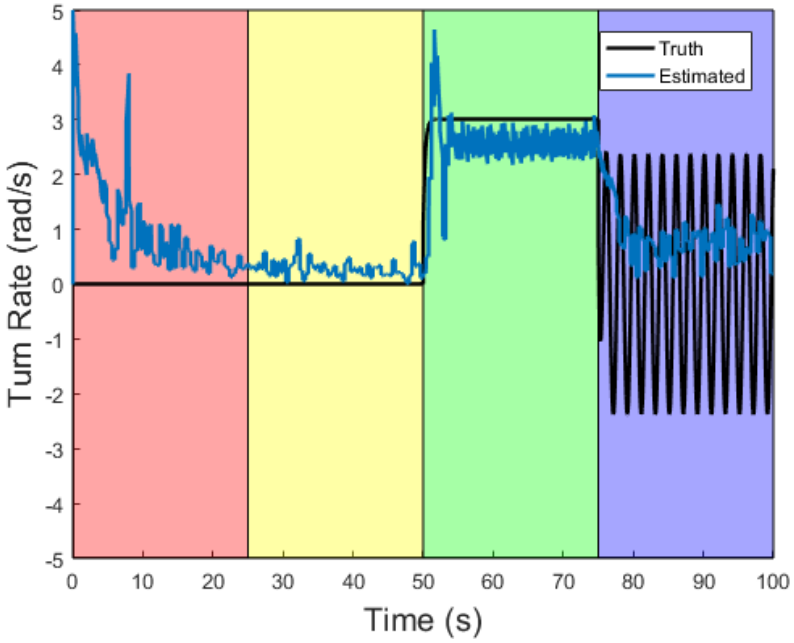

Figure 3.8: Estimated Turn Rate

The third and final method is to use multiple instances of the CT model with a span of selected turn-rates within the IMM framework. This method is by far the most computationally intensive of the three options described, but also produces the best results. The results of the IMM example using four instances of the CT model with turn-rate values of -3 rad/s, -1 rad/s, 1 rad/s, and 3 rad/s are shown in figure 3.9. The four separate CT models are constructed with one of the mentioned turn rates, and are combined with the IMM algorithm. This allows the state estimates from each of the CT models to be summed together with weights related to their component residuals. Therefore, the IMM algorithm essentially chooses the model with the closest turn rate to the true value. This effect can be seen in Figure 3.10. This third method is used in this thesis because the extra computational costs are outweighed by the performance.

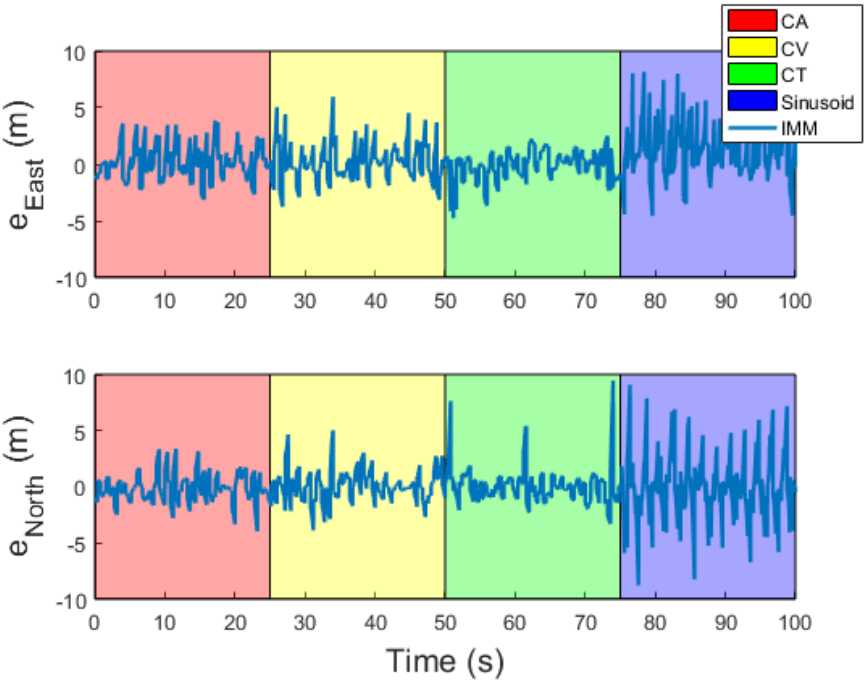

Figure 3.9: Constant Turn Kinematic Model Performance with Multiple Turn Rate Values within IMM

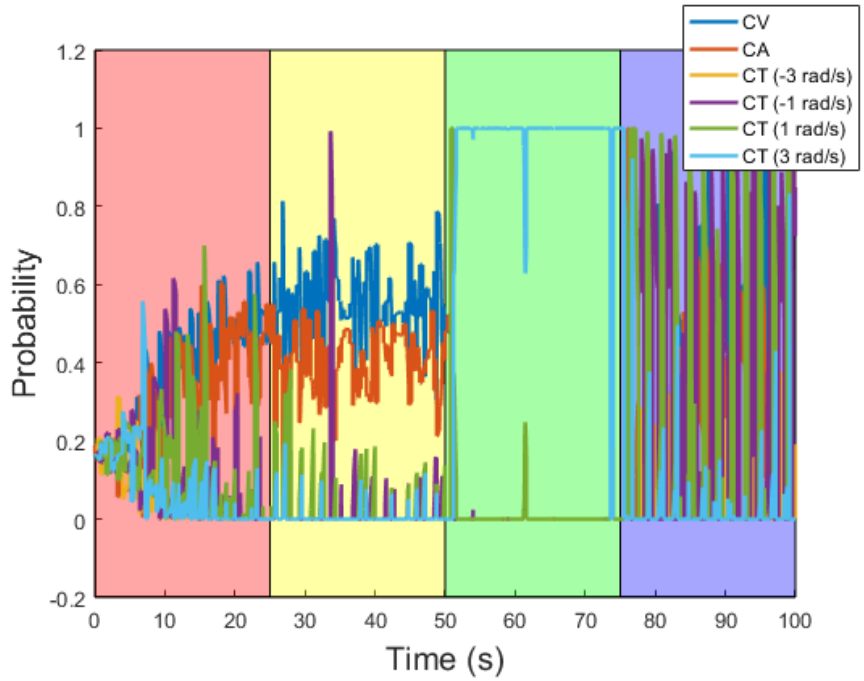

Figure 3.10: IMM Model Probabilities with Multiple Turn Rate Method

3.5 IMM Propagation

In the implementation of the IMM algorithm used in this thesis, the component models are kept independant of each other, and the IMM process described in Section 3.2 is used to mix the states. Each model is propagated in the same process, using a modification of the standard Kalman filter equations in order to include the non-linear radar measurement equations. The process used for all three models is described in the following sections.

3.5.1 IMM Time Update

Similar to the process described in Chapter 2, the standard Kalman Filter equations are used to perform the time update for the IMM algorithm. However, unlike the process described in Chapter 2, the models used in the IMM filter are discrete, so integration is not necessary. The state is propagated with the state transition matrices described in Section 3.4 as shown in Equation (3.23).

$$\underline{X}^{-}_{j,k} = \boldsymbol{A}_j \underline{X}_{j,k-1} \tag{3.23}$$

The subscripts $_k$ and $_{k-1}$ indicate the values at the current and previous discrete time steps. The state covariance is then propagated as shown in Equation (3.24).

$$\boldsymbol{P}^{-}_{j,k} = \boldsymbol{A}_j \boldsymbol{P}_{j,k-1} \boldsymbol{A}_j + \boldsymbol{Q}_j \tag{3.24}$$

If there are measurements available from the radar, then the measurement update is performed at this point. If there are no measurements available, then the propagated states from each model are sent to the IMM algorithm to generate a mixed state estimate using the residuals from the most recently performed measurement update.

3.5.2 IMM Measurement Update

When measurements from the radar unit are available, the measurement update is applied to each model. The first step of this process is to compute the residuals of each state estimate

compared to the measurements. In order to do this, the state must be mapped to the measurement domain. In this thesis, the radar measurements of range, range rate, and azimuth are in a polar coordinate system, and therefore require a non-linear function to map the state. This function is given as

$$\underline{Z}_j = \underline{Y} - h(\underline{X}_{j,k}^-) \tag{3.25}$$

$$\underline{Y} = \begin{bmatrix} \tilde{r} \\ \tilde{\dot{r}} \\ \tilde{\lambda}_Z \end{bmatrix} \tag{3.26}$$

where $\underline{Z}$ is the vector of residuals, $\underline{Y}$ is the vector of radar measurements as defined in Equation (3.26), r, $\dot{r}$, and λ_Z are the range, range rate, and azimuth measurements, and h is the non-linear function of the state estimate which maps the estimate to the measurement domain.

This non-linear mapping function is given as

$$h(\underline{X}) = \begin{bmatrix} R(\underline{X}) \\ \dot{R}(\underline{X}) \\ \Lambda_Z(\underline{X}) \end{bmatrix} \tag{3.27}$$

where $R(\underline{X})$ is the range, $\dot{R}(\underline{X})$ is the range rate, and $\Lambda_Z(\underline{X})$ is the azimuth as functions of the state $\underline{X}$. The functions are defined in Equations (3.28 - 3.30) [28].

$$R(\underline{X}) = \sqrt{\Delta X^2 + \Delta Y^2 + \Delta Z^2} \tag{3.28}$$

$$\dot{R}(\underline{X}) = \frac{\Delta X \Delta \dot{X} + \Delta Y \Delta \dot{Y} + \Delta Z \Delta \dot{Z}}{r} \tag{3.29}$$

$$\Lambda_Z(\underline{X}) = \arctan\left(\frac{\Delta Y}{\Delta X}\right) \tag{3.30}$$

The Δ terms represent the relative values of the components of the state between the radar unit and tracked object.

The Jacobian of the function $h(\underline{X})$ is then found, as it is required to correct the state. This is given in Equation (3.31)

$$\boldsymbol{H}(\underline{X}) = \begin{bmatrix} H_1 & H_2 & H_3 & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing \\ H_4 & H_5 & H_6 & H_1 & H_2 & H_3 & \varnothing & \varnothing & \varnothing \\ H_7 & H_8 & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing & \varnothing \end{bmatrix} \tag{3.31}$$

where

$$H_1 = \frac{\Delta X}{\sqrt{\Delta X^2 + \Delta Y^2 + \Delta Z^2}} \tag{3.32}$$

$$H_2 = \frac{\Delta Y}{\sqrt{\Delta X^2 + \Delta Y^2 + \Delta Z^2}} \tag{3.33}$$

$$H_3 = \frac{\Delta Z}{\sqrt{\Delta X^2 + \Delta Y^2 + \Delta Z^2}} \tag{3.34}$$

$$H_4 = \frac{\Delta \dot{X}}{\sqrt{\Delta X^2 + \Delta Y^2 + \Delta Z^2}} - \frac{\Delta X(\Delta X \Delta \dot{X} + \Delta Y \Delta \dot{Y} + \Delta Z \Delta \dot{Z})}{(\Delta X^2 + \Delta Y^2 + \Delta Z^2)^{3/2}} \tag{3.35}$$

$$H_5 = \frac{\Delta \dot{Y}}{\sqrt{\Delta X^2 + \Delta Y^2 + \Delta Z^2}} - \frac{\Delta Y(\Delta X \Delta \dot{X} + \Delta Y \Delta \dot{Y} + \Delta Z \Delta \dot{Z})}{(\Delta X^2 + \Delta Y^2 + \Delta Z^2)^{3/2}} \tag{3.36}$$

$$H_6 = \frac{\Delta \dot{Z}}{\sqrt{\Delta X^2 + \Delta Y^2 + \Delta Z^2}} - \frac{\Delta Z(\Delta X \Delta \dot{X} + \Delta Y \Delta \dot{Y} + \Delta Z \Delta \dot{Z})}{(\Delta X^2 + \Delta Y^2 + \Delta Z^2)^{3/2}} \tag{3.37}$$

$$H_7 = \frac{-\Delta Y}{\Delta X^2 + \Delta Y^2} \tag{3.38}$$

$$H_8 = \frac{\Delta X}{\Delta X^2 + \Delta Y^2} \tag{3.39}$$

$\boldsymbol{H}(\underline{X})$ is the Jacobian of $h(\underline{X})$, which must be re-evaluated at each measurement update with the most recent state estimate.

The state is then corrected as shown in Equation (3.40).

$$\underline{X}^+_{j,k} = \underline{X}^-_{j,k} + \boldsymbol{L}_j(\underline{Z}_j) \tag{3.40}$$

The Kalman gain $\boldsymbol{L}$ is defined by

$$\boldsymbol{L}_j = \boldsymbol{P}^-_{j,k} \boldsymbol{H}^T_j(\underline{X}^-_{j,k}) \underline{S}^{-1}_{j,k} \tag{3.41}$$

$$\underline{S}_{j,k} = \boldsymbol{H}_j(\underline{X}^-_{j,k})\boldsymbol{P}^-_{j,k}\boldsymbol{H}^T_j(\underline{X}^-_{j,k}) + \boldsymbol{R} \tag{3.42}$$

$$\boldsymbol{R} = \begin{bmatrix} \sigma_r^2\boldsymbol{I}_3 & \varnothing_3 & \varnothing_3 \\ \varnothing_3 & \sigma_{\dot{r}}^2\boldsymbol{I}_3 & \varnothing_3 \\ \varnothing_3 & \varnothing_3 & \sigma_{\lambda_Z}^2\boldsymbol{I}_3 \end{bmatrix} \tag{3.43}$$

where $\underline{S}$ is the covariance of the innovation and σ_r, $\sigma_{\dot{r}}$, and σ_{λ_Z} are the 1σ standard deviation values of the noise on the range, range rate, and azimuth measurements respectively. The state covariance is then propagated with the expression in Equation (3.44).

$$\boldsymbol{P}^+_{j,k} = (\boldsymbol{I}_9 - \boldsymbol{L}_j)\boldsymbol{P}^-_{j,k} \tag{3.44}$$

The updated component model elements are then mixed using the IMM algorithm described in Section 3.2

3.6 Illustrative IMM Example Results

The IMM algorithm with the three models described in Section 3.4 is exercised using the example trajectory introduced in Section 3.3 in order to more clearly demonstrate the operation of the IMM algorithm. The true position values of this trajectory were corrupted with white noise ($\sigma = 1m$) and used as measurements in an IMM system utilizing a CV, CA, and CT motion model. These models were then propagated through the IMM algorithm as described in the previous sections. The results from this simulation can be seen in Figure 3.11. As expected, each model performed best during the phases of the trajectory which involved their component motion types. This is especially notable in the CT results. In the first and second phases, the CT model performs quite poorly when compared to the other models. During the phases involving constant rate turning, the CT model excels. The mean error of the position estimates is 1.98 m, and the 1σ standard deviation of the error is 1.45 m. It can also be seen that the IMM solution does not always track with one model, but instead changes its weighting to use the best performing models at any given time.

This can be made clear by inspecting the model probabilities over time, which is shown in Figure 3.12. As the vehicle performs the constant acceleration maneuver, the CV and CA

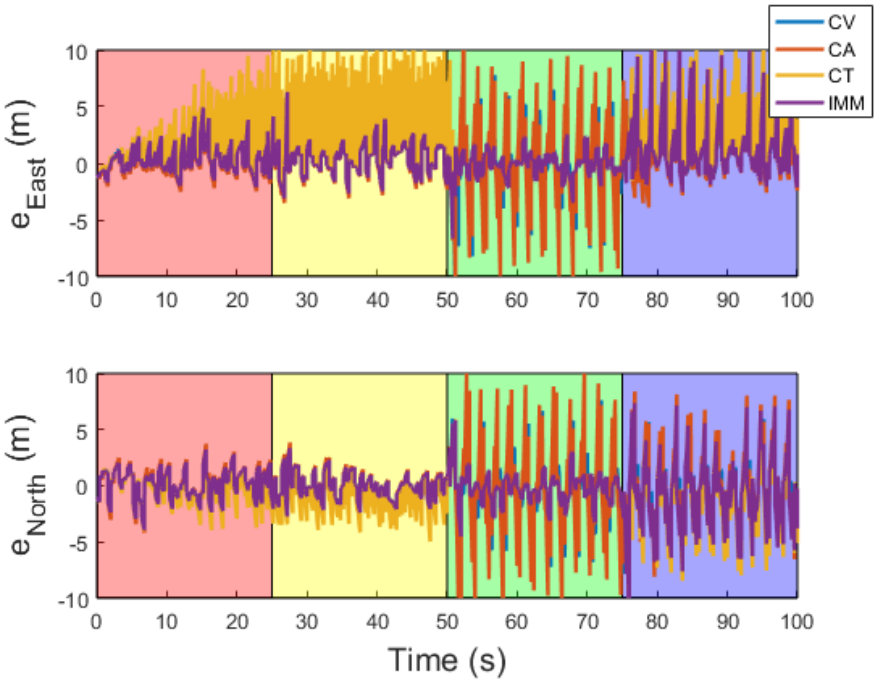

Figure 3.11: Filter Results from IMM Example Trajectory

models track relatively close to each other, but the CT model probability is lowered by the IMM algorithm. During the constant velocity portion of the trajectory, the CV model probability performs slightly better than the CA model, and the CT model probability remains essentially at zero. During the constant turn maneuver, however, the CT model probability increases to nearly one while the other two models are lowered drastically. Then during the sinusoidal portion of the trajectory, all three model probabilities vary. This shows that the IMM algorithm is responding as expected to the changing motion modes the vehicle undergoes in this example trajectory.

In order to further demonstrate the performance of the IMM algorithm, a different configuration of the component filters was exercised with the same example trajectory. In this new configuration, the corresponding model is used by itself during each phase of the trajectory, instead of combining the models within the IMM framework. This means that during the CA phase the CA model was used alone to generate the state estimate, during the CV phase the CV model was used, during the CT phase the CT model was used, and during the sinusoidal phase the IMM algorithm was used combining all three models. This configuration represents the ideal performance of the IMM algorithm, where it weighs the correct model heavily and

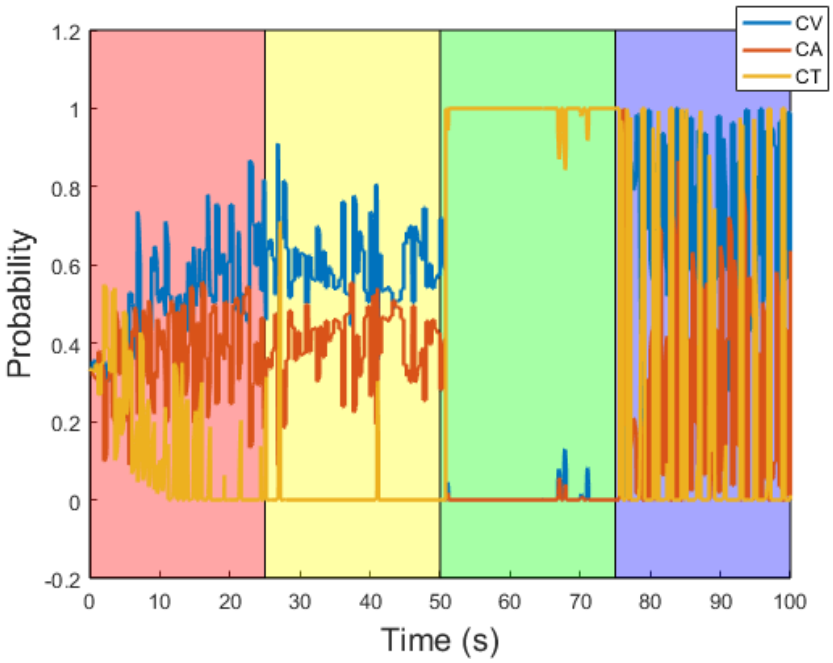

Figure 3.12: Model Probabilities from IMM Example Trajectory

instantaneously as the modes change. The performance of this configuration is shown in Figure 3.13. The mean error of the position estimates for this simulation is 1.93 m, and the 1σ standard deviation of the error is 1.3 m. These results are marginally better than that of the IMM filter, but require perfect knowledge of the motion type at any time throughout the trajectory. This result demonstrates that the IMM algorithm is able to quickly and adequately change its weighting of each model and is suitable for use in this work.

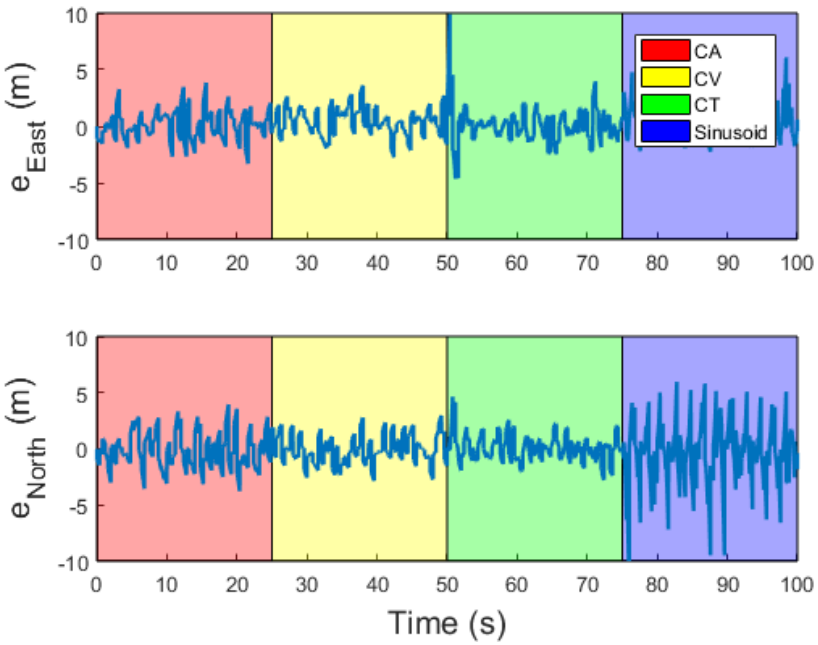

Figure 3.13: Filter Results with Correct Models

Chapter 4

GPS/INS/Radar/IMM Navigation

4.1 GRIMM System Formulation

The proposed GPS/INS/Radar/IMM (GRIMM) system performs relative navigation between two vehicles. The first vehicle is referred to as the base vehicle, and the second vehicle is referred to as the rover vehicle. Both vehicles operate the Loosely Coupled GPS/INS navigation filter described in Chapter 2. The base vehicle is equipped with an automotive grade radar unit which provides the relative measurements to the rover vehicle. Figure 4.1 shows the basic layout of the two vehicles, and the measurements of range, range rate, and azimuth provided by the radar unit on the base vehicle.

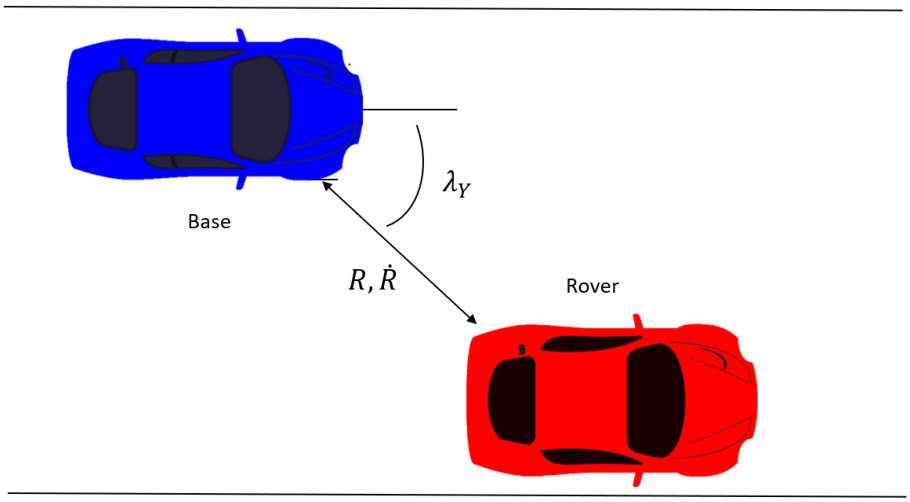

Figure 4.1: Proposed System Layout

The base vehicle takes in measurements from the IMU and GPS sensors and generates an estimate of the state of the base vehicle from the Loosely Coupled GPS/INS filter. Then, the

measurements of relative range, range rate, and azimuth from the base vehicle to the rover are combined with the base state estimate in the IMM to generate relative estimates of the rover with respect to the base vehicle. These relative estimates are then combined to generate the GRIMM estimates of the rover vehicle.

When the GPS measurements for the rover vehicle are acceptable in quality, they are used in conjunction with the IMU measurements in the GPS/INS filter to generate the estimate of the rover state, and the GRIMM measurements are ignored. If the GPS measurements for the rover are unavailable, however, the GRIMM measurements are used in place of the GPS measurements within the GPS/INS filter. In this case, the rover vehicle does not have access to any GPS measurements, and instead uses external measurements generated by the GRIMM algorithm to update its state.

4.2 Base Vehicle

The main operations of the GRIMM algorithm operate with data from the base vehicle. A functional diagram describing these operations is shown in Figure 4.2. The IMU, GPS, and EKF blocks in the top left of the block diagram represent the Loosely Coupled GPS/INS filter described in Chapter 2, except for one difference. The implementation of the GPS/INS filter on the base vehicle only takes in position measurements from the GPS receiver, instead of the position and velocity measurements described previously in order to replicate the sensors used to generate experimental data for this thesis. This means that the measurement update and the generation of the error states must be altered slightly.

Instead of using the measurement vector and mapping matrix described in Equations (2.41 - 2.42), the expressions given in Equations (4.1 - 4.2) must be used.

$$\underline{Y} = \begin{bmatrix} \underline{P}_{GPS} \end{bmatrix} \tag{4.1}$$

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{I}_3 & \varnothing_3 & \varnothing_3 & \varnothing_3 & \varnothing_3 \end{bmatrix} \tag{4.2}$$

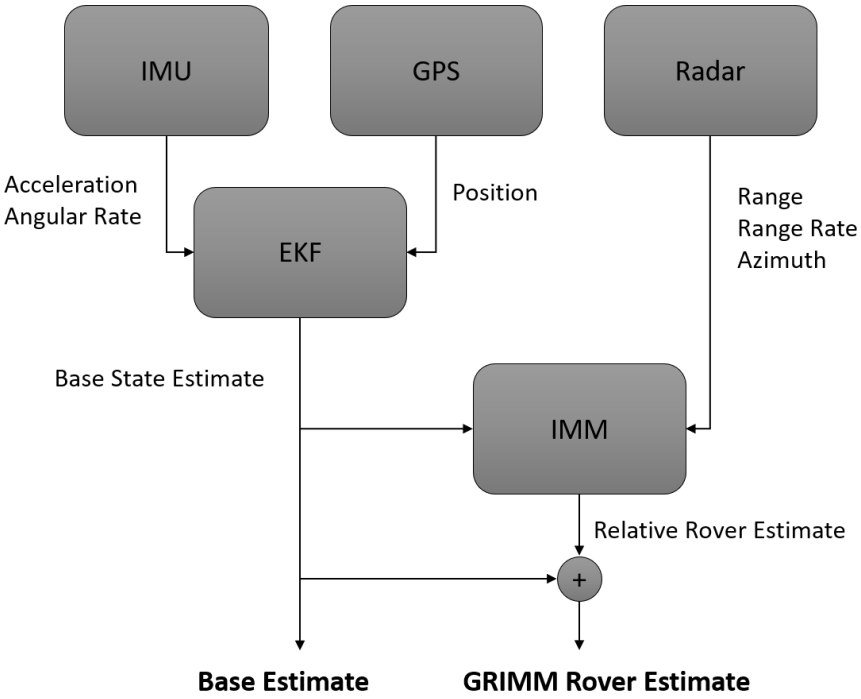

Figure 4.2: Base Vehicle Block Diagram

The dimensions of the measurement covariance matrix must also be changed to accomodate only the GPS position measurements. Equation (2.45) is altered to the expression shown in Equation (4.9).

$$\boldsymbol{R} = \left[\sigma^2_{P_{GPS}} \boldsymbol{I}_3\right] \tag{4.3}$$

Other than those small changes, the GPS/INS filter running on the base vehicle is unaltered from the filter described in Chapter 2.

The Radar and IMM blocks on the right side of Figure 4.2 represent the IMM algorithm described in Chapter 3. The operation of the IMM filter in the GRIMM algorithm is the exact same as described, with the CV, CA, and CT models shown. The IMM algorithm generates relative measurements from the base vehicle to the rover vehicle. Because the radar measurements are in a polar coordinate system aligned with the body frame of the base vehicle, the relative measurements must be rotated into the navigation frame before they can be combined with the base vehicle state estimates. For the position measurements, this is a simple process. The same rotation matrix that is used to propagate the base vehicle GPS/INS filter is used to

rotate the position estimates as given in Equation (4.4)

$$\underline{P}^{rel}_{GRIMM} = \boldsymbol{C}^n_b \underline{P}_{IMM} \tag{4.4}$$

where $\underline{P}^{rel}_{GRIMM}$ is the relative position estimate from the base to the rover vehicle rotated into the navigation frame.

Rotating the velocity estimates into the navigation frame is not as simple, because of the effects of the equivalent lever arm between the base and rover vehicles. In order to counteract this, the relative velocity values caused by this lever arm effect must be removed from the rotated velocities. This process is expressed as

$$\underline{V}^{rel}_{GRIMM} = \boldsymbol{C}^n_b \underline{V}_{IMM} - [\underline{\boldsymbol{\omega}}_{\boldsymbol{INS}} \wedge] \underline{P}^{rel}_{GRIMM} \tag{4.5}$$

where $\underline{V}^{rel}_{GRIMM}$ is the relative velocity estimate from the base to the rover vehicle rotated into the navigation frame and corrected for lever arm errors.

Once the IMM estimates have been rotated into the navigation frame, the base vehicle navigation solution is combined with the rotated IMM estimates in order to generate estimates of the rover in the navigation frame. This combination is shown in Equations (4.6 - 4.7)

$$\underline{P}_{GRIMM} = \underline{P}_{base} + \underline{P}^{rel}_{GRIMM} \tag{4.6}$$

$$\underline{V}_{GRIMM} = \underline{V}_{base} + \underline{V}^{rel}_{GRIMM} \tag{4.7}$$

where $\underline{P}_{base}$ and $\underline{V}_{base}$ are the position and velocity estimates of the base vehicle from the GPS/INS filter. $\underline{P}_{GRIMM}$ and $\underline{V}_{GRIMM}$ are the position and velocity estimates of the rover vehicle in the navigation frame generated by the GRIMM filter.

The GRIMM estimates can be generated at every iteration of the base vehicle GPS/INS filter, or only when radar measurements are available. In this thesis, the GRIMM estimates are generated at every iteration of the base vehicle GPS/INS filter, because the rate of radar measurements is relatively high compared to GPS. If the radar measurements were at a lower update

rate, the GRIMM results may drift heavily between radar measurements as the weighting of the component models is based on the residuals from the previous measurements.

4.3 Rover Vehicle

The rover vehicle portion of the GRIMM filter is less complicated than the base vehicle because it does not operate the GRIMM algorithm on-board. A functional diagram describing the operation of the rover vehicle is shown in Figure 4.3. Like the base vehicle, the IMU measurements are taken in by the EKF block, which represents the operation of the Loosely Coupled GPS/INS filter. The GPS, GRIMM, and IF/ELSE blocks on the right side of the diagram represent the switching between the GPS or GRIMM measurements depending on certain conditions. The logic within the IF/ELSE block essentially states that if the GPS measurements are degraded or completely unavailable, then the measurements from the GRIMM filter will be used in the GPS/INS filter, otherwise the GPS measurements will be used.

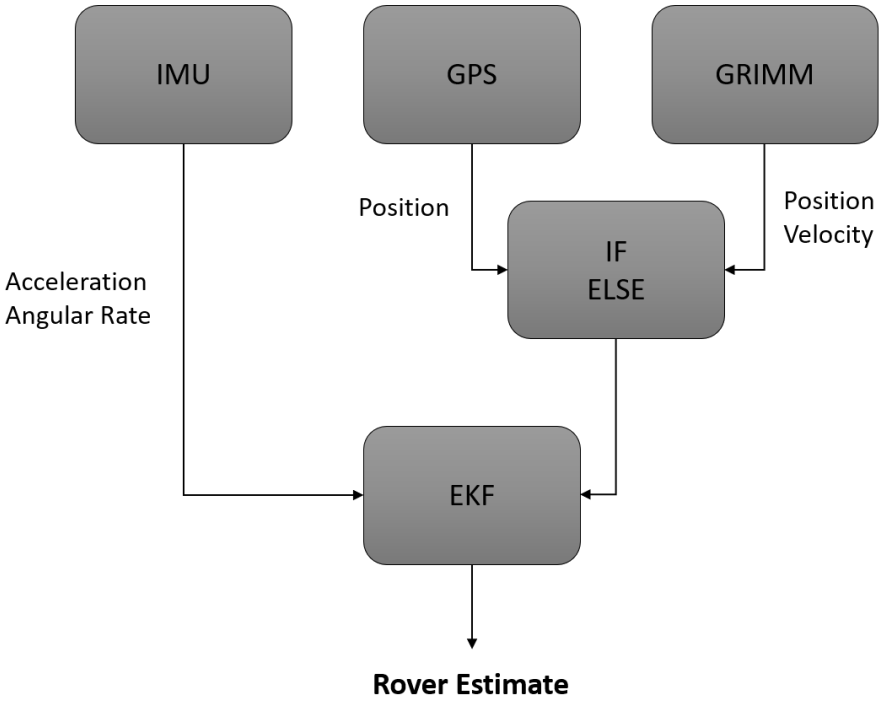

Figure 4.3: Rover Vehicle Block Diagram

In the case that the GPS measurements are acceptable, the measurement vector and mapping matrix described in Equations (4.1 - 4.2) are used. In the case that the GRIMM measurements must be used, however, the mapping matrix described in Equation (2.42) can be used.

The GRIMM measurements are arranged in the measurement vector as

$$\underline{Y} = \begin{bmatrix} \underline{P}_{GRIMM} \\ \underline{V}_{GRIMM} \end{bmatrix} \tag{4.8}$$

and the measurement covariance matrix must be described as

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{P}^{GRIMM}_{(1:3,1:3)} & \varnothing_3 \\ \varnothing_3 & \boldsymbol{P}^{GRIMM}_{(4:6,4:6)} \end{bmatrix} \tag{4.9}$$

where $\boldsymbol{P}^{GRIMM}$ is the covariance of the GRIMM estimates. Because the GRIMM estimates are functions of both the radar IMM filter and the base vehicle GPS/INS filter, their covariances must be functions of those component estimates'. In this thesis, the GRIMM estimate covariance is defined as shown in Equation (4.10)

$$\boldsymbol{P}^{GRIMM} = \boldsymbol{P}^{GPS/INS}_{(1:6,1:6)} + \boldsymbol{P}^{IMM}_{(1:6,1:6)} \tag{4.10}$$

where $\boldsymbol{P}^{GPS/INS}$ and $\boldsymbol{P}^{IMM}$ are the covariances of the GPS/INS and IMM state estimates. Once these changes are made to the rover GPS/INS filter, the operation is exactly the same as that described in Chapter 2.

Chapter 5

GRIMM Algorithm Performance Analysis

5.1 Simulation Description

The GRIMM algorithm was evaluated with three scenarios generated with CarSim, a high-fidelity ground vehicle simulation tool. The first scenario involves the base vehicle driving in a straight line along-side the rover vehicle, which is also driving in a straight line. The layout of the scenario can be seen in Figure 5.1. The second simulated scenario is shown in Figure 5.2. This scenario also has the base vehicle driving in a straight line, but the rover vehicle is now performing a single lane change maneuver, in which it quickly shifts over approximately three meters. The final simulated scenario is shown in Figure 5.3. Similarly to the last two scenarios, the base vehicle is travelling in a straight line. However, the rover performs a double lane change maneuver, where the vehicle shifts over a lane and then back to the original lane quickly.

These scenarios were selected to represent some common situations in everyday driving. Realistic error characteristics for automotive grade IMUs and GPS receivers were used, along with the performance characteristics of the Delphi Electronically Scanning Radar (ESR), which was used in the experimental portion of this work. The values used to simulate these sensors are given in Table 5.1.

Table 5.1: Simulation Sensor Values

σ_{GPS}	1.5	m
σ_a	0.1	m/s^2
σ_g	0.001	rad/s
τ_a	120	s^{-1}
τ_g	60	s^{-1}
σ_{range}	0.5	m
$\sigma_{rangeRate}$	0.12	m/s
$\sigma_{azimuth}$	0.0087	rad

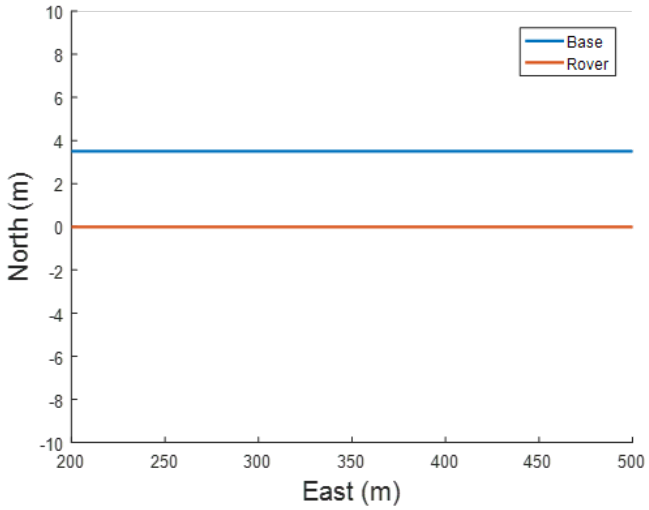

Figure 5.1: Simulation Scenario 1

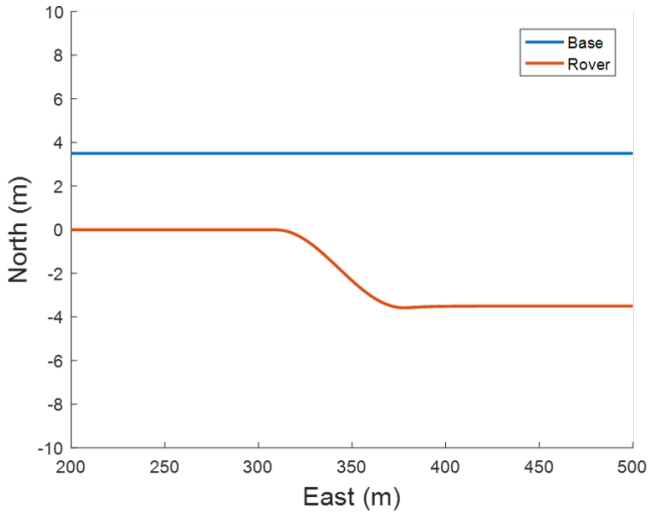

Figure 5.2: Simulation Scenario 2

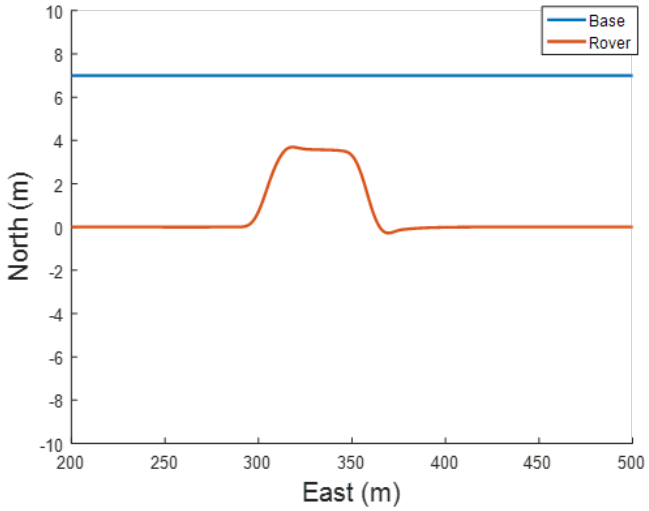

Figure 5.3: Simulation Scenario 3

5.2 Experimental Description

The GRIMM algorithm was further evaluated with the use of experimental data collected using two vehicles operated by the Auburn University GAVLAB: an Infiniti G35 instrumented with real time data collection systems built in Robotic Operating System (ROS), and a Lincoln MKZ instrumented by AutonomouStuff. Both vehicles can be seen in Figure 5.4. All data analysis was performed in post processing. The sensor data local to each vehicle is synchronized with ROS time stamps, and the data between vehicles is synchronized with GPS time.

Figure 5.4: Experimental Ground Vehicle Platforms

The goal of the experimental scenarios was to replicate the corresponding simulated trajectories as closely as possible. The experiments were performed in a parking lot near the

National Center for Asphalt Technology (NCAT) test track. Experimental data sets were generated for two scenarios. The first scenario replicates the first simulated scenario, where both the rover and base drive in a straight line.

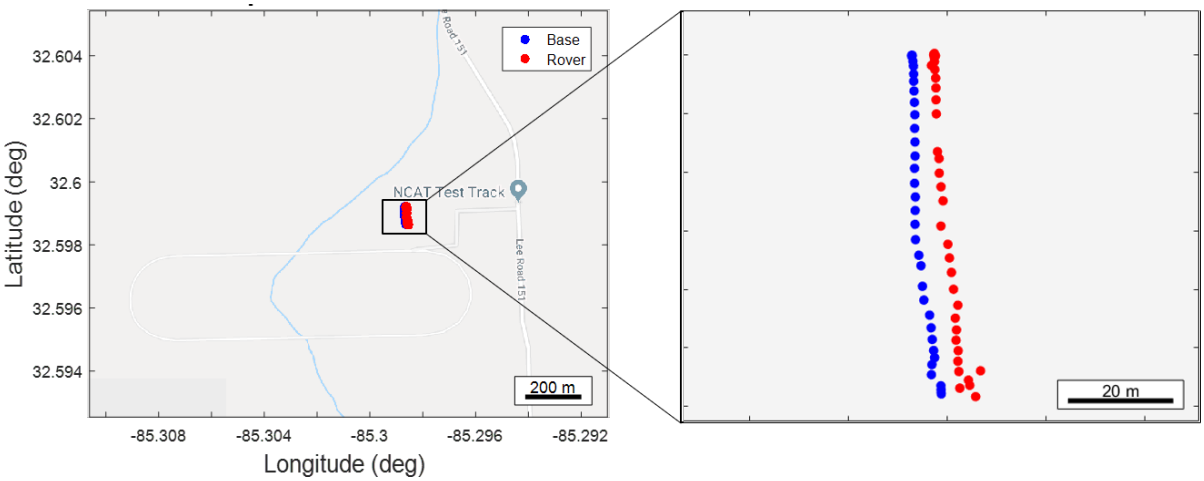

Figure 5.5: Experimental Scenario 1 (Straight Line Driving)

The second scenario is shown in Figure 5.6, where the base vehicle was driven in a straight line, and the rover vehicle was driven in a simple single lane change maneuver. The trajectories do differ slightly to the simulated scenario but are similar enough to allow valid comparisons.

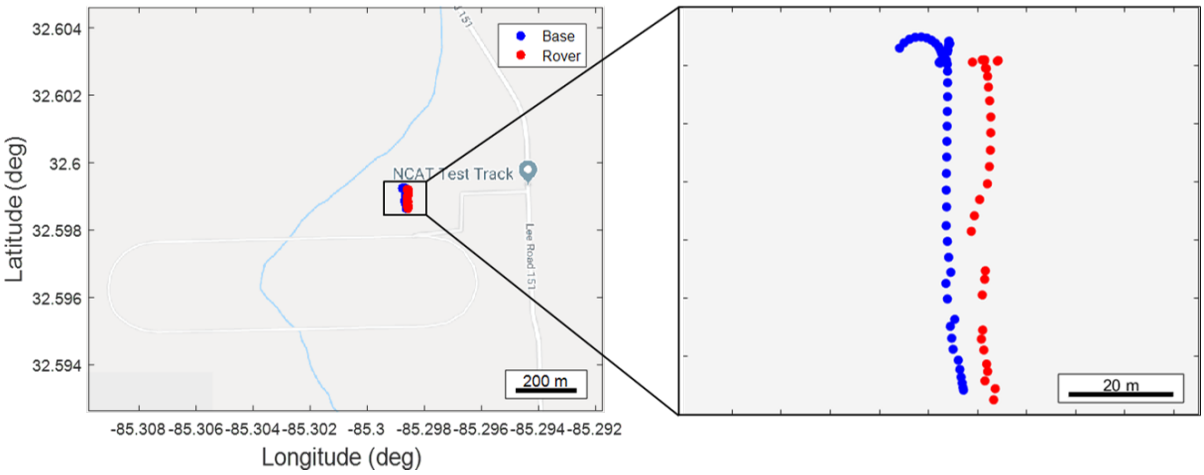

Figure 5.6: Experimental Scenario 2 (Single Lane Change)

5.2.1 Radar Channel Selection

The Delphi ESR used in this thesis has the ability to track up to 64 objects simultaneously. The tracking functionality of the radar unit is processed independently in 64 different channels.

An example of radar data provided by the Delphi ESR is shown in Figure 5.7. The data shown in this figure was generated during experimental Scenario 2. The varying colors of the markers in Figure 5.7 correspond to each channel of the radar data. The difficulty in using this data is that the channels that are tracking the object of interest do not necessarily persist. For example, in Figure 5.7 the trajectory of interest near the bottom of the plot is spanned by various different channels over time.

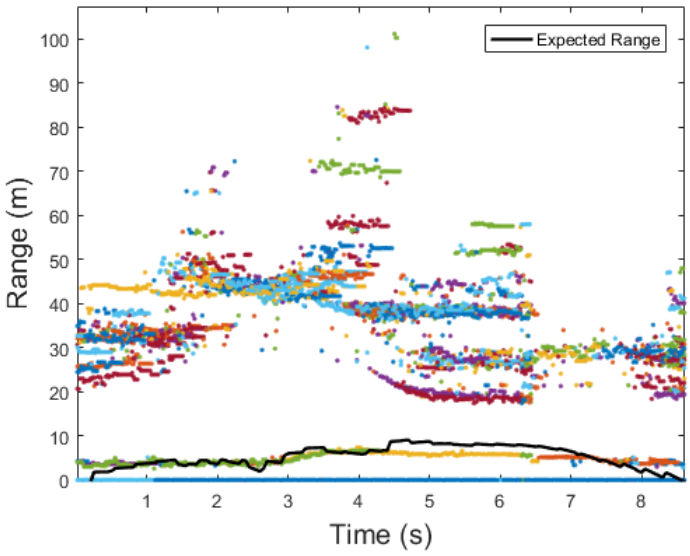

Figure 5.7: Delphi ESR Example Output

Because of the multi-channel nature of the reported radar measurements, some method of selecting the 'correct' channels must be developed. There are multiple methods of selecting channels in the literature. The ideal standalone method is the Probabilistic Data Association Filter (PDAF), which is detailed in [29]. Methods to predict interference from surrounding objects and vehicles are also present in the literature [30]. In this thesis, however, a less complex method of channel selection is employed, because information about both vehicles is known.

In order to select the radar channels, the navigation solutions from both vehicles are combined to generate an expected range measurement. This calculation is given as

$$R_{exp} = \sqrt{(\underline{\hat{X}}_{(1)}^{base} - \underline{\hat{X}}_{(1)}^{rover})^2 + (\underline{\hat{X}}_{(2)}^{base} - \underline{\hat{X}}_{(2)}^{rover})^2 + (\underline{\hat{X}}_{(3)}^{base} - \underline{\hat{X}}_{(3)}^{rover})^2} \tag{5.1}$$

where $\underline{\hat{X}}^{base}$ and $\underline{\hat{X}}^{rover}$ are the current navigation estimates of the base and rover vehicle. The black line in Figure 5.7 shows this expected range measurement and how it would correspond to the channel data.

Once this expected range measurement is generated, the radar channel data is compared to an envelope generated around the expected measurement. This envelope is shown in Figure 5.8. The green field is the envelope that is used to select the radar channels. This envelope can

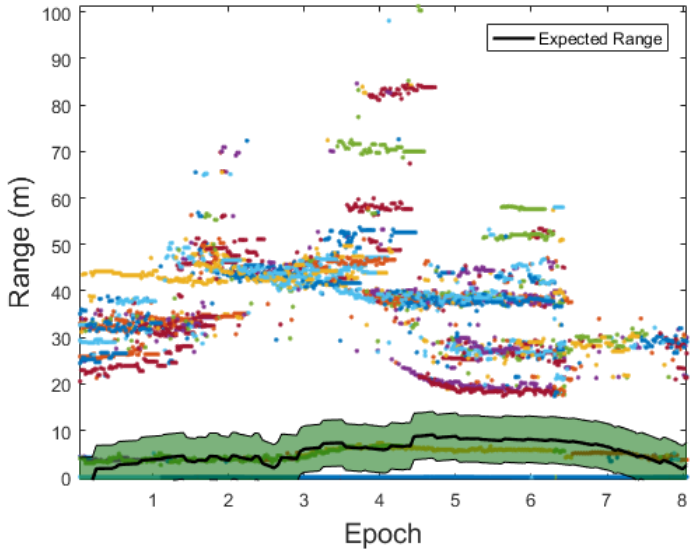

Figure 5.8: Visualization of Measurement Envelope for Radar Channel Selection

be widened or narrowed depending on the confidence in the navigation solutions of the two vehicles.

The channels of the measurements that fall within the set envelope are then stored over time, as shown in Figure 5.9. These channels are used to select the measurements from the range, range rate, and azimuth channels. There are points where multiple channels fall into the envelope. Because of this, measurements from every selected channel at each time step are averaged together. The data from the selected channels, and the effective measurement generated from the averaging of the channels is shown in Figure 5.10.

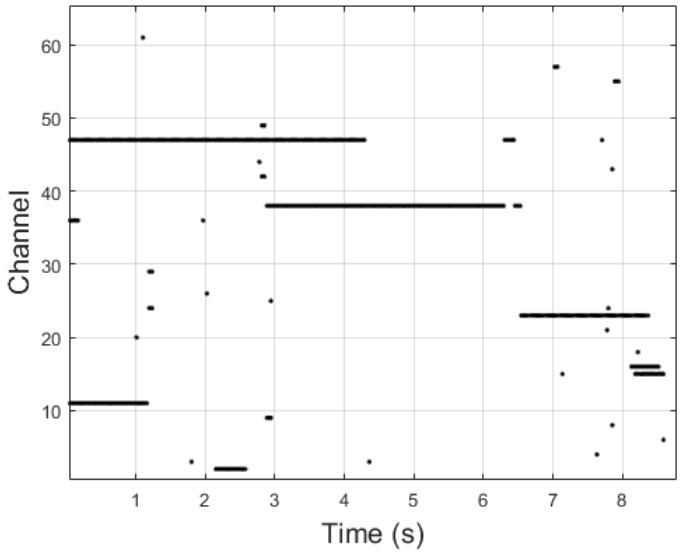

Figure 5.9: Selected Channels Using Expected Measurement Envelope Method

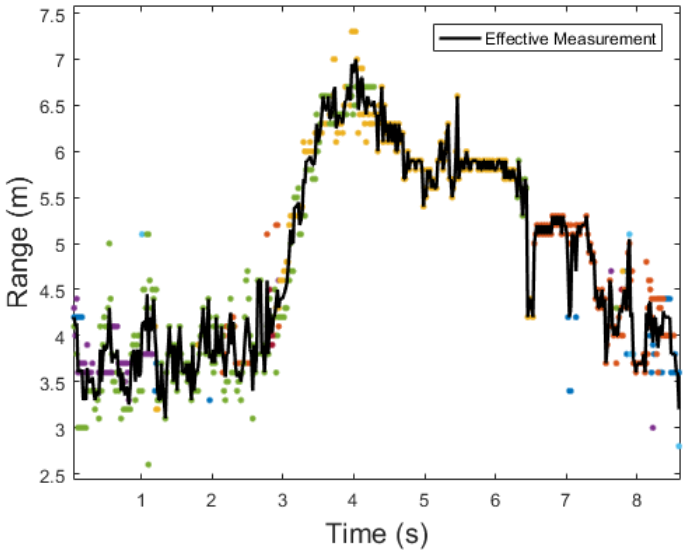

Figure 5.10: Effective Measurement Generated from Selected Channels

5.3 Simulation Results

For each scenario, results from three different navigation filters on the rover are shown. First, the standalone INS solution is shown, where the IMU measurements are propagated forward with the INS algorithm without any external measurements for corrections. The results

from standalone INS represent the worst-case scenario for the rover vehicle without any collaborative aiding or GPS measurements. The second filter shown is the GPS/INS navigation solution, where the rover vehicle has access to GPS measurements and therefore does not require the external measurements from the GRIMM filter. This is the ideal operating condition, and serves as the goal for performance of the GRIMM filter. Finally, the results from the GRIMM filter are shown, where the rover vehicle utilizes the external measurements generated as described in Chapter 4.

The results for simulation Scenario 1 on the rover are shown in Figures (5.11 - 5.13). The standalone INS filter is shown in Figure 5.11. Clearly, the performance of standalone standalone INS is not ideal. The performance is expected though, since standalone INS involves the integration of white noise and moving bias on the IMU measurements. This means that the error of the standalone INS solution is essentially random walk driven by a white noise with the same standard deviation as the noise on the IMU measurements. The rover results using the GPS/INS filter is shown in Figure 5.12. As expected, the GPS/INS filter performs much better than the standalone INS filter. The external measurements from GPS correct the propagated INS solutions in order to bound the growth of the navigation errors to within approximately 2 meters. Finally, the rover using the GRIMM filter is shown in Figure 5.13. The GRIMM filter performs similarly to the GPS/INS filter, and much better than the standalone INS filter. The external measurements from the GRIMM algorithm correct the propagated rover INS solutions in order to bound the growth of the navigation errors. Results for this scenario can be found in Table 5.2.

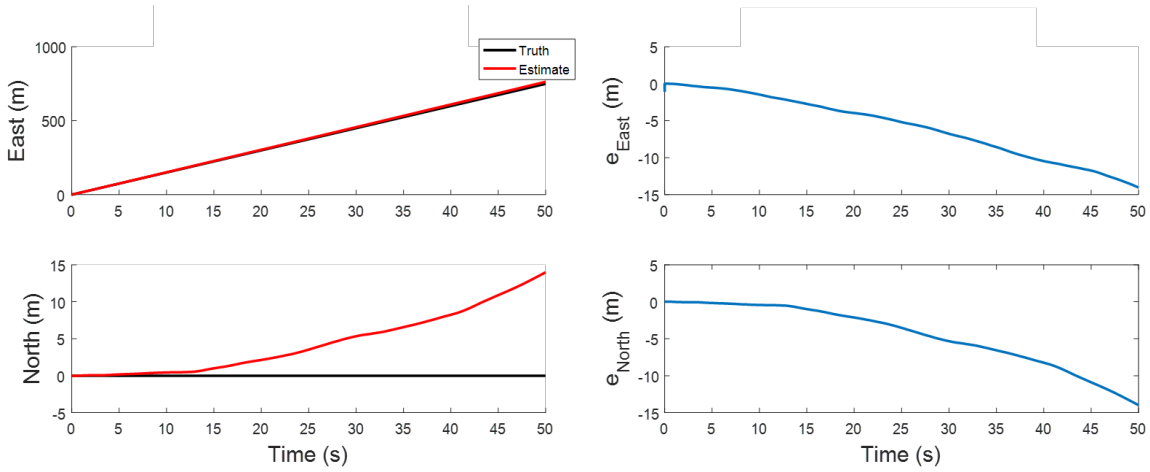

Figure 5.11: Simulation Scenario 1: Standalone INS Rover Navigation Results

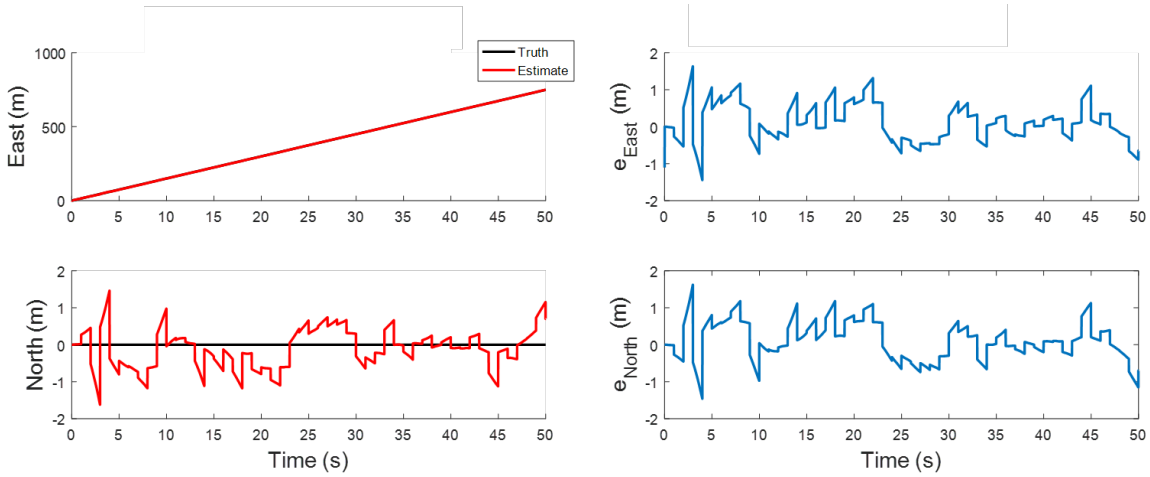

Figure 5.12: Simulation Scenario 1: GPS/INS Rover Navigation Results

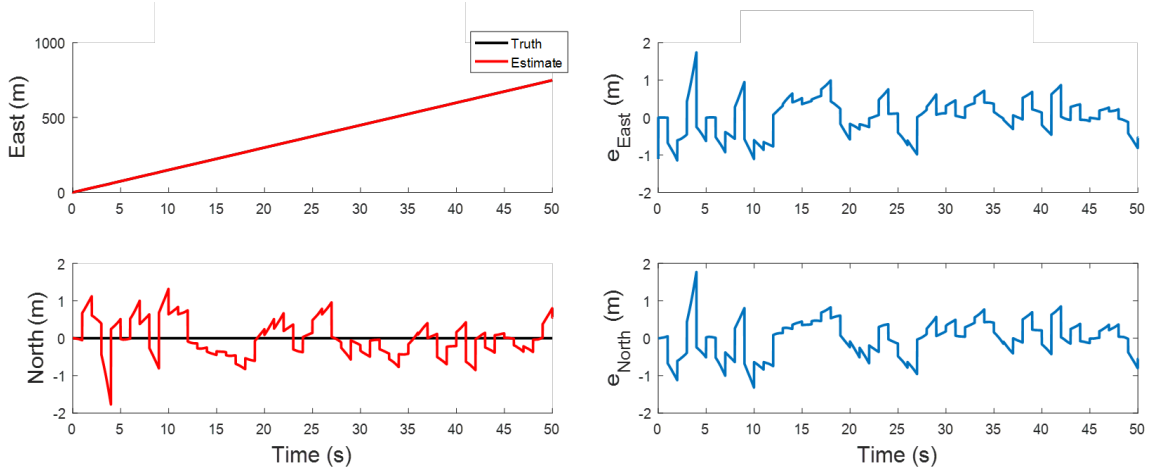

Figure 5.13: Simulation Scenario 1: GRIMM Rover Navigation Results

Table 5.2: Simulation Scenario 1 Results

Filter	μ (m)	**1-σ** (m)
Standalone INS	8.37	6.45
GPS/INS	0.54	0.46
GRIMM	1.08	0.73

The results for simulation Scenario 2 on the rover are shown in Figures (5.14 - 5.16). The standalone INS filter is shown in Figure 5.14. Again, the performance of the standalone INS filter is quite poor, and the navigation solution quickly diverges from truth. The rover results using the GPS/INS filter is shown in Figure 5.15. The GPS/INS filter performs much better than the standalone INS filter again. Finally, the rover results using the GRIMM filter is shown in Figure 5.16. Similar to the first scenario, the GRIMM filter performs well compared to the standalone INS and is comparable to the GPS/INS filter. Results for Scenario 2 are shown in Table 5.3.

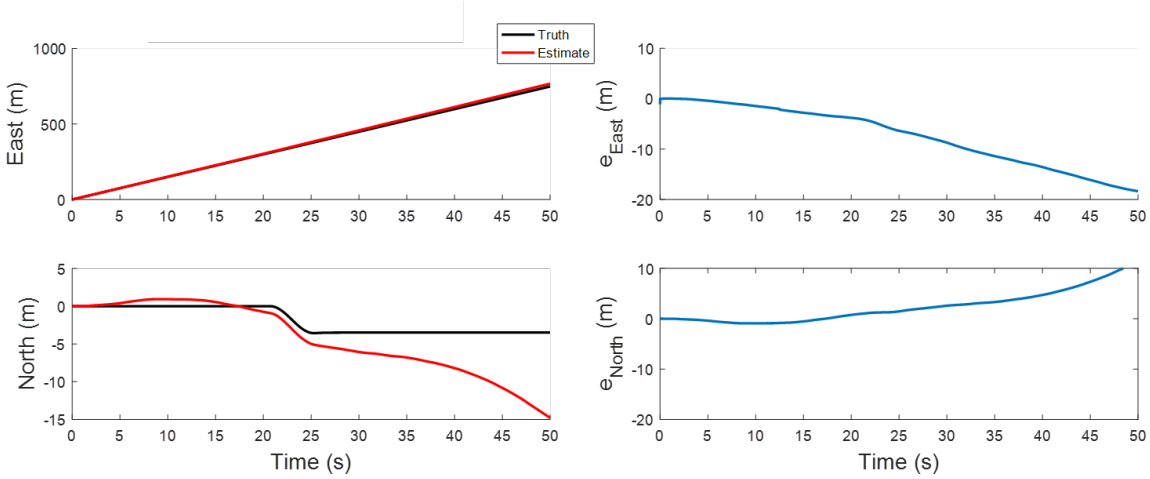

Figure 5.14: Simulation Scenario 2: Standalone INS Rover Navigation Results

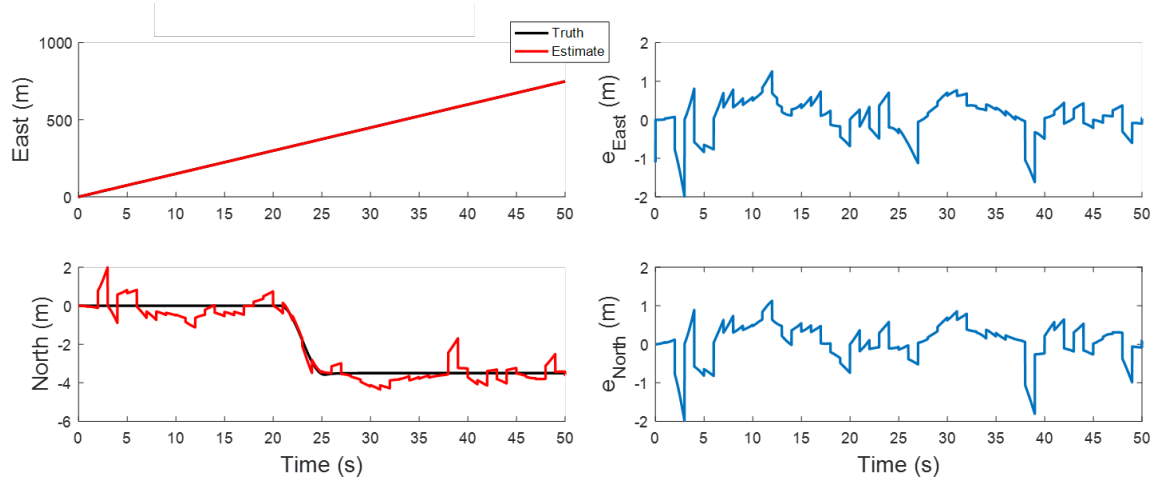

Figure 5.15: Simulation Scenario 2: GPS/INS Rover Navigation Results

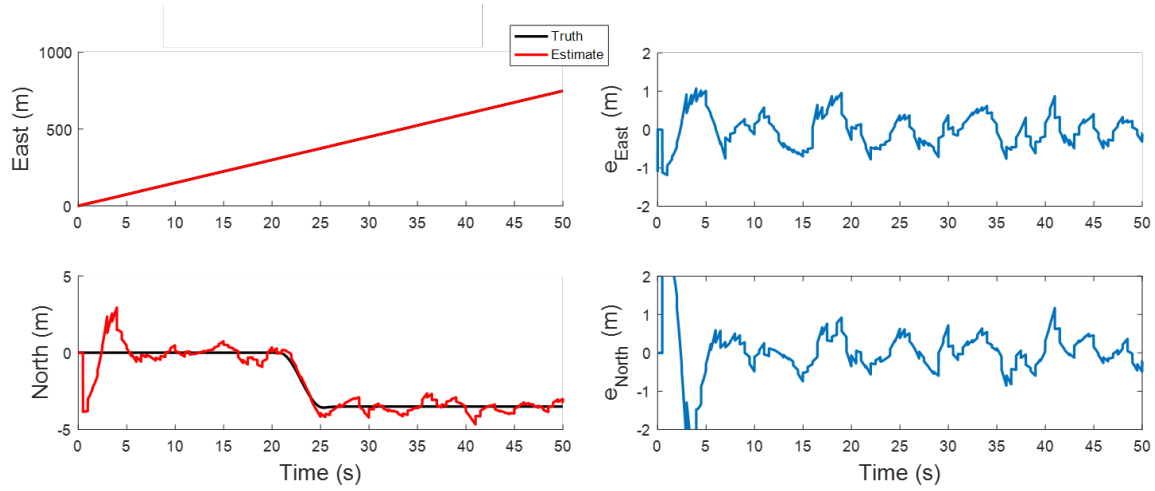

Figure 5.16: Simulation Scenario 2: GRIMM Rover Navigation Results

Table 5.3: Simulation Scenario 2 Results

Filter	μ (m)	**1-σ** (m)
Standalone INS	6.75	4.55
GPS/INS	0.70	0.45
GRIMM	1.06	0.72

The results for simulation Scenario 3 on the rover are shown in Figures (5.17 - 5.19). The standalone INS filter is shown in Figure 5.17. The results of the standalone INS filter are similar to the previous scenarios. The rover results using the GPS/INS filter is shown in Figure 5.18. The results of the GPS/INS filter are much better than that of the standalone INS filter,

just like the previous scenarios. Finally, the rover results using the GRIMM filter is shown in Figure 5.19. The results of the GRIMM filter are similar to the GPS/INS filter and much better than the standalone INS, as has been seen in the previous scenarios. Additionally, a simple tracking filter using only the CA model was exercised with the Scenario 3 data to verify that the IMM algorithm is providing a benefit over simple single-model tracking. The results for this filter are shown in Figure 5.20. This filter performs slightly worse than the GRIMM filter. This shows that although tracking with a single model is a viable option, the IMM algorithm is in fact improving the navigation solution. The results for Scenario 3 are shown in Table 5.4.

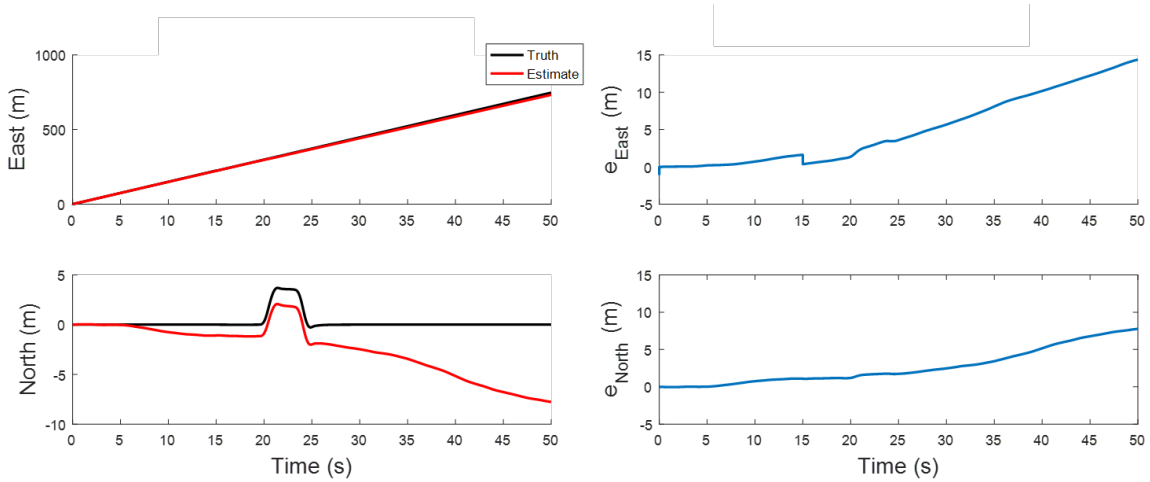

Figure 5.17: Simulation Scenario 3: Standalone INS Rover Navigation Results

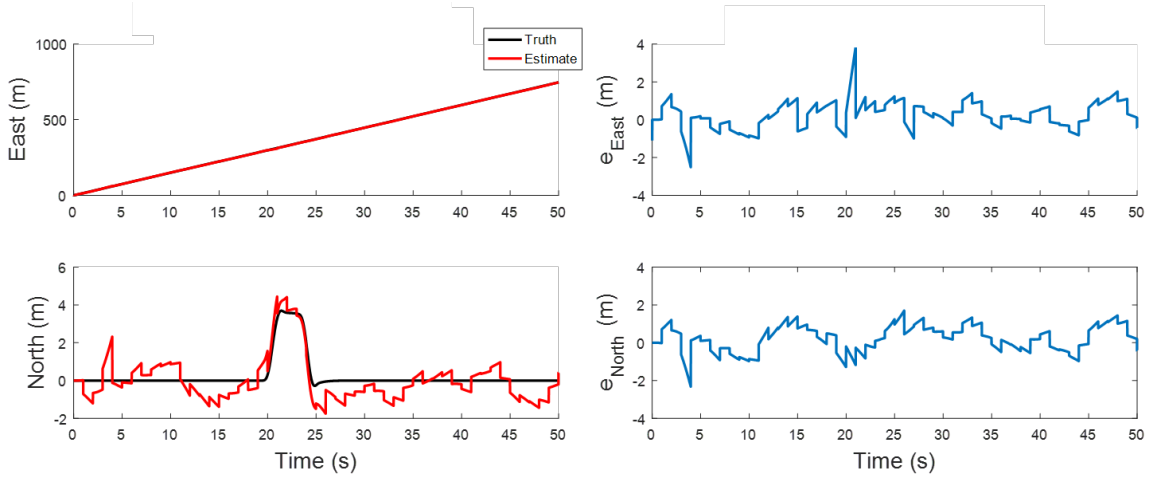

Figure 5.18: Simulation Scenario 2: GPS/INS Rover Navigation Results

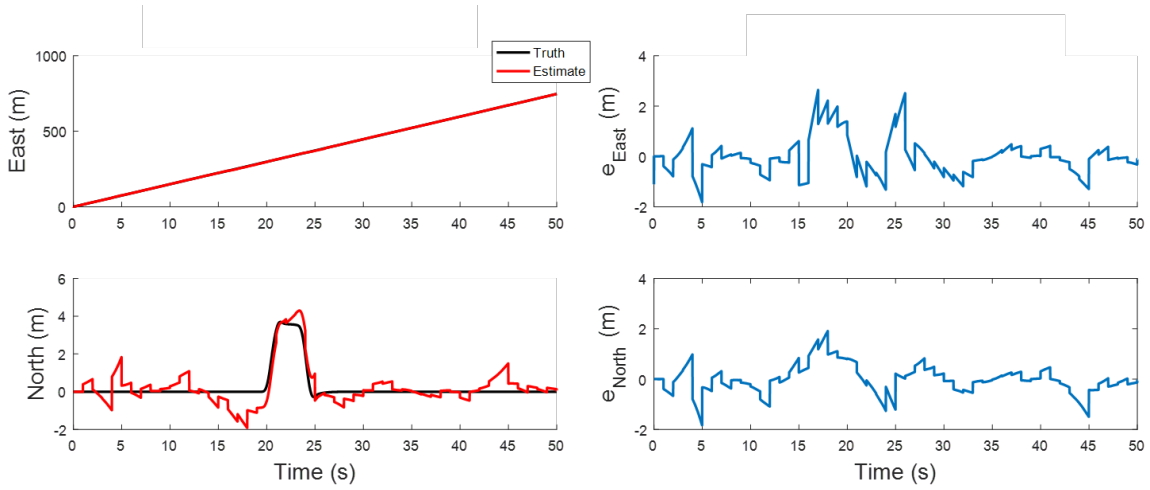

Figure 5.19: Simulation Scenario 3: GRIMM Rover Navigation Results

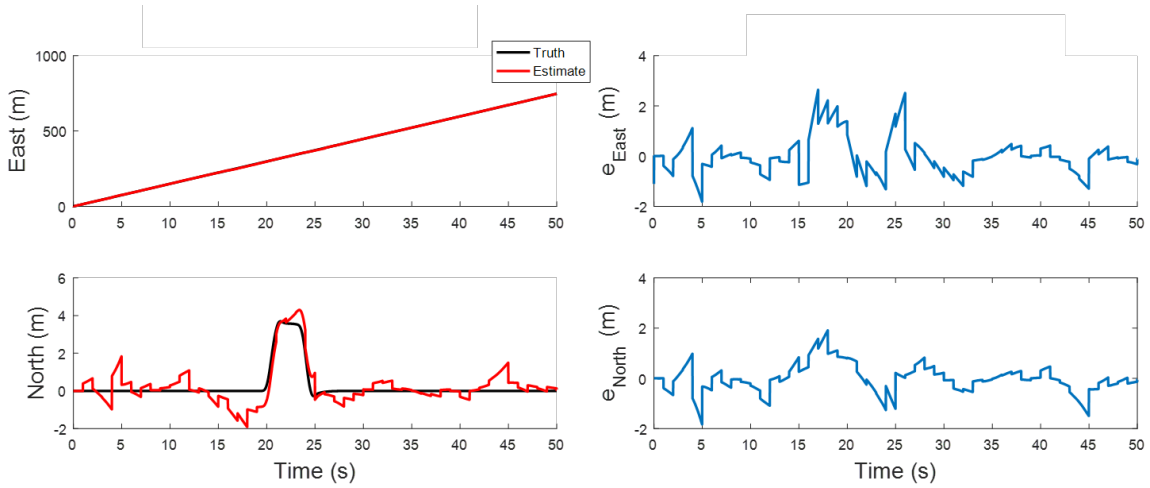

Figure 5.20: Simulation Scenario 3: CA Rover Navigation Results

Table 5.4: Simulation Scenario 3 Results

Filter	μ (m)	**1-σ** (m)
Standalone INS	7.69	5.87
GPS/INS	0.68	0.58
GRIMM	1.24	1.19
CA	1.33	1.40

5.4 Experimental Results

Similar to the simulation data sets, the results from the standalone INS, GPS/INS, and GRIMM filters on the rover are compared for each experimental scenario. The results from the

experimental data are worse than the corresponding simulated scenarios in general. There are a few reasons for this, but the main source of error is likely the inconsistant timing between vehicles and between the sensors on each vehicle locally. Another fact that may effect the deduced reckoning solutions is that the vibrations from the engine running and road noise will essentially increase the noise on the IMU measurements.

The results for experimental Scenario 1 on the rover are shown in Figures (5.21 - 5.23). The standalone INS filter is shown in Figure 5.21. As mentioned previously, the standalone INS solution is notably worse in with experimental data than it is with simulated data. The rover results using the GPS/INS filter is shown in Figure 5.22. The GPS/INS results are drastically better than the standalone INS filter results, as is expected based on the simulation results. Finally, the rover results using the GRIMM filter is shown in Figure 5.23. As in the results from the simulated data sets, the GRIMM filter performs almost as well as the GPS/INS filter, and much better than the standalone INS filter. Results from the first experimental scenario are given in Table 5.5

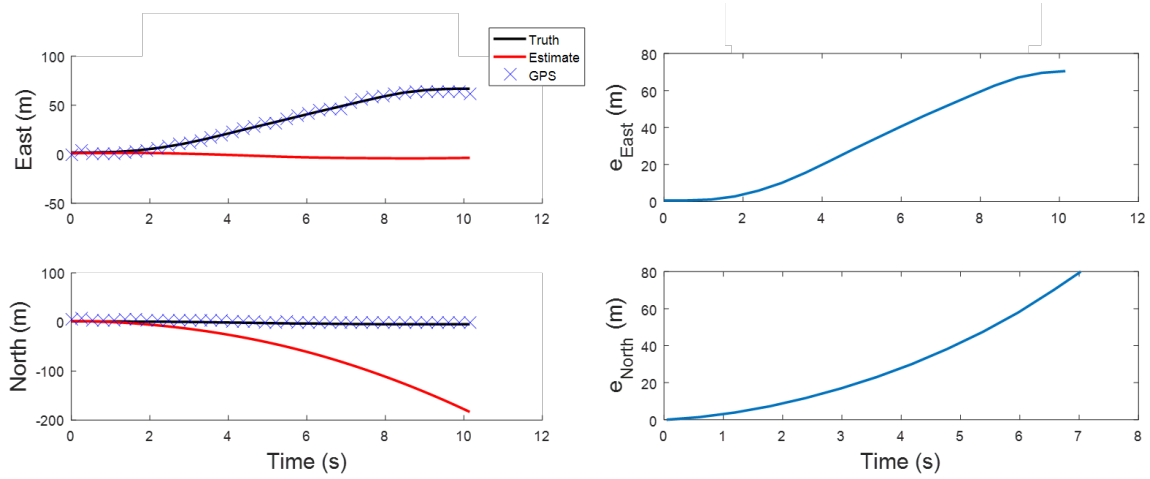

Figure 5.21: Experimental Scenario 1: Standalone INS Rover Navigation Results

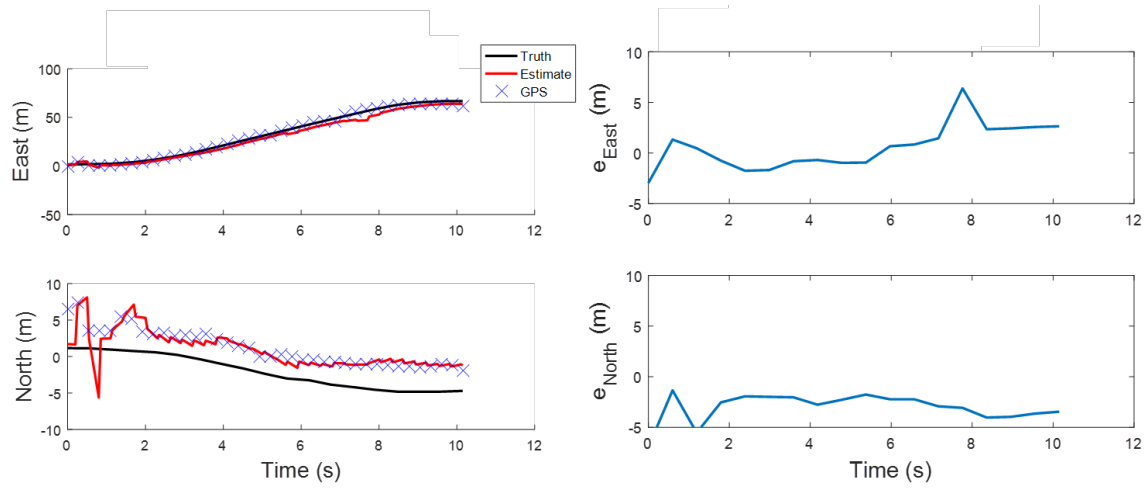

Figure 5.22: Experimental Scenario 1: GPS/INS Rover Navigation Results

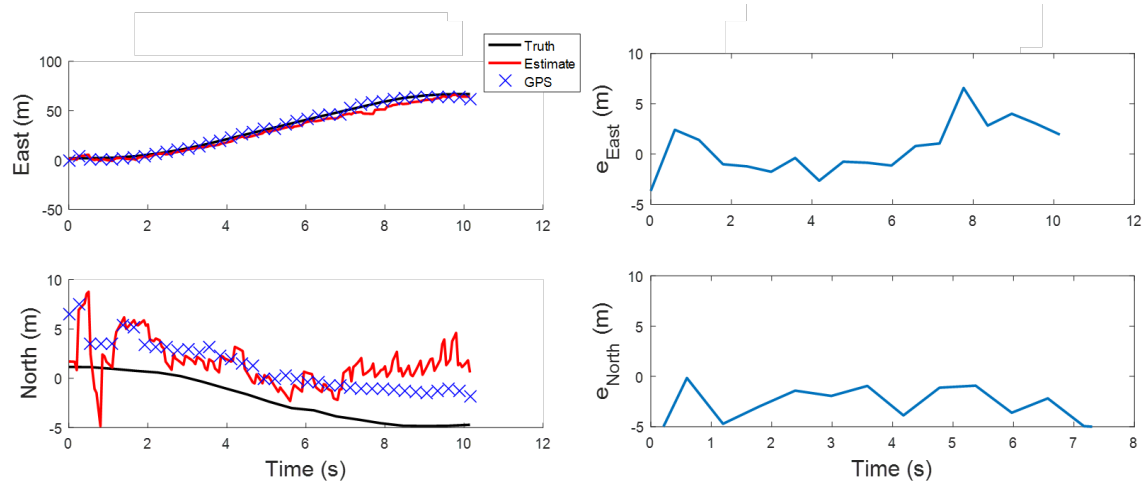

Figure 5.23: Experimental Scenario 1: GRIMM Rover Navigation Results

Table 5.5: Experimental Scenario 1 Results

Filter	μ (m)	**1-σ** (m)
Standalone INS	66.85	59.50
GPS/INS	3.65	1.72
GRIMM	4.12	1.53

The results for experimental Scenario 2 on the rover are shown in Figures (5.24 - 5.26). The standalone INS filter is shown in Figure 5.24. Again, the results from the standalone INS filter are quite bad, and would obviously be unacceptable in an autonomous navigation application. The rover results using the GPS/INS filter is shown in Figure 5.25. The results

with the GPS/INS filter are much better than the standalone INS solution as expected. Finally, the rover results using the GRIMM filter is shown in Figure 5.26. Just as expected, the GRIMM filter performs similarly to the GPS/INS filter and much better to the standalone INS filter in this scenario as well. The results from experimental Scenario 2 can be found in Table 5.6.

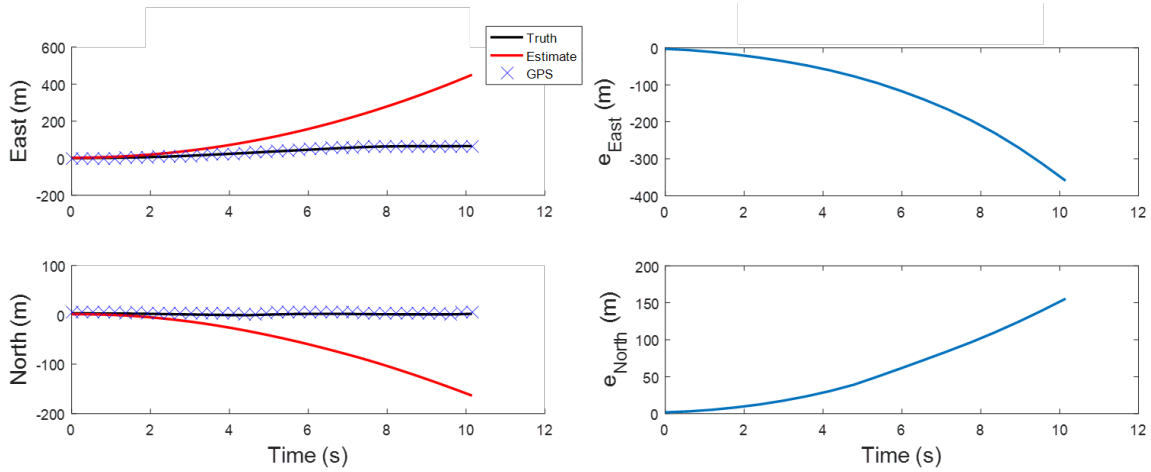

Figure 5.24: Experimental Scenario 2: Standalone INS Rover Navigation Results

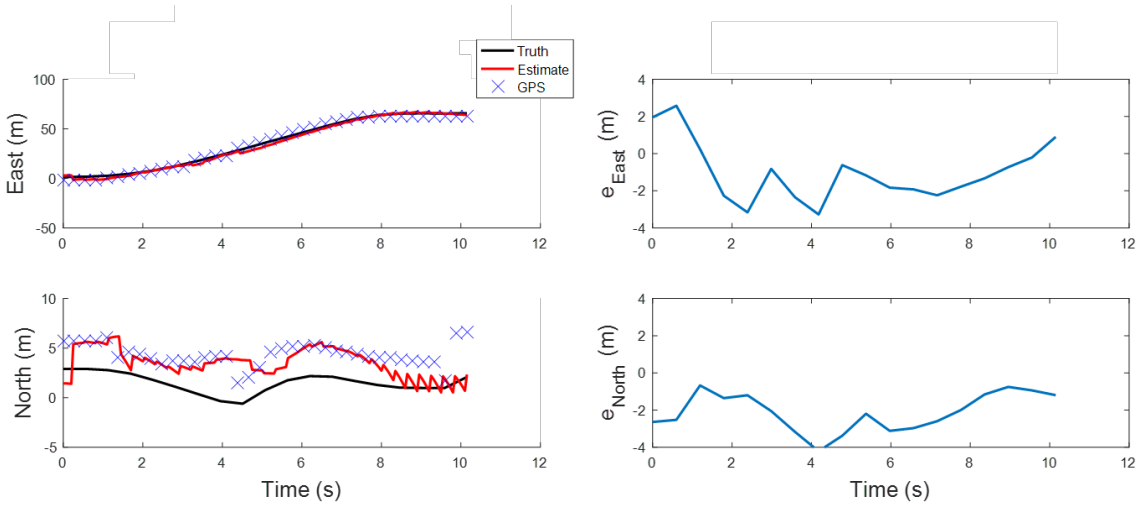

Figure 5.25: Experimental Scenario 2: GPS/INS Rover Navigation Results

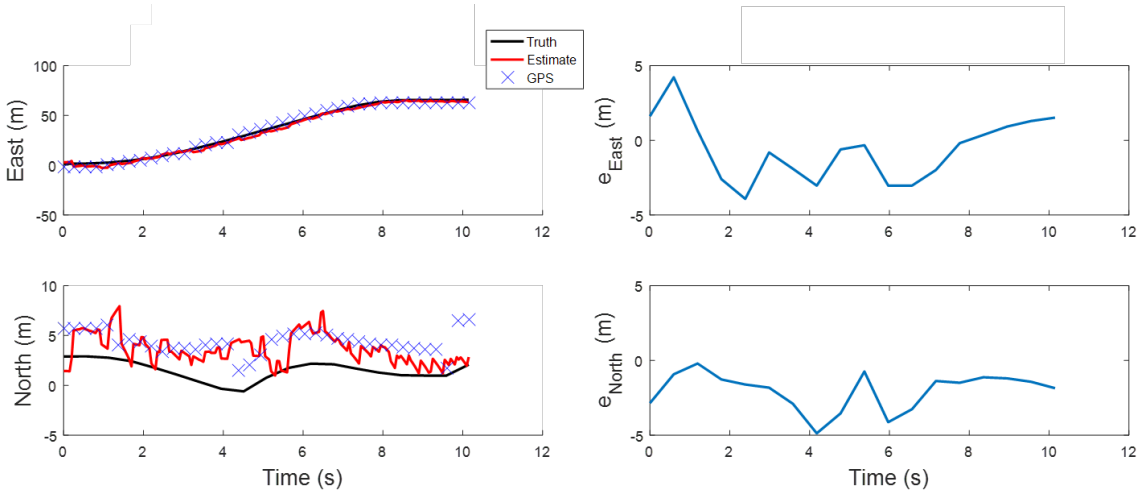

Figure 5.26: Experimental Scenario 2: GRIMM Rover Navigation Results

Table 5.6: Experimental Scenario 2 Results

Scenario	μ (m)	**1-σ** (m)
Standalone INS	133.76	122.88
GPS/INS	2.75	1.21
GRIMM	3.60	1.40

The GRIMM filter was exercised with multiple simulated and experimental data sets. The performance of the GRIMM filter in each of these data sets was compared to the performance of the standalone INS filter and the GPS/INS filter. It has been shown that the GRIMM filter performed drastically better than the standalone INS filter, and more similarly to the GPS/INS filter, in every scenario. A summary of the results described in this chapter is given in Table 5.7. It can be observed in the table that the GRIMM filter consistently performed much better than the standalone INS filter, and only slightly worse than the GPS/INS filter. This trend is consistent between both simulated and experimental data. This shows that the GRIMM filter is a viable option for maintaining acceptable navigational performance even when GPS data is unavailable.

Table 5.7: Summary of GRIMM Algorithm Performance

Scenario	**INS**		**GPS/INS**		**GRIMM**		**% INS**		**% GPS/INS**	
	μ (m)	**1-σ** (m)	μ (m)	**1-σ** (m)	μ (m)	**1-σ** (m)	μ (%)	**1-σ** (%)	μ (%)	**1-σ** (%)
Sim 1	8.37	6.45	0.54	0.46	1.08	0.73	12.9	11.3	199.6	158.0
Sim 2	6.75	4.55	0.70	0.45	1.06	0.72	15.7	15.8	151.4	160.0
Sim 3	7.69	5.87	0.68	0.58	1.24	1.19	16.1	20.3	182.4	205.2
Exp 1	66.85	59.50	3.65	1.72	4.12	1.53	6.2	2.6	112.9	88.9
Exp 2	133.76	122.88	2.75	1.21	3.60	1.40	2.7	1.1	130.9	115.7

Chapter 6

Using GRIMM to Detect Vehicle Cut-In for Autonomous Platooning Applications

Autonomous platooning is a method of coherently transporting multiple ground vehicles in a way that performs better in some aspect compared to normal driving. For industry applications, the best example of this is truck platooning for commercial and military applications. In the long-haul industry, one of the main operational expense is fuel [31]. Because of this, trucking companies have a vested interest in any methods of improving fuel economy they can find. Autonomous platooning has cemented its place in this application, as specific longitudinal truck spacings at highway speeds have been shown to reduce aerodynamic drag on both the lead and following vehicles [32]. Similarly to the civilian trucking market, the military uses a large amount of fuel moving resources and personnel at home and abroad, so they have an interest in improving fuel economy. The military also has unique interests in minimizing used footprint on roads, and minimizing the amount of personnel needed to operate a large convoy. Both of these interests are motivated by the operation in hostile environments where roads can be mined or convoys ambushed, and both are looking to solutions which include autonomous platooning.

No matter the application, effective autonomous platooning requires robust navigation. For the lead vehicle, the navigation can be based on GNSS or optical/local methods. However, the following vehicles must have some knowledge of the lead vehicle's state. There are two common approaches to this necessity; vehicle-to-vehicle communication of waypoint data, or line-of-sight measurements to the lead vehicle. Both cases have a common issue to deal with, which is the unpredictability of surrounding vehicles. In the first case, where waypoint data is shared via some inter-vehicle communication network, surrounding vehicles intruding into

the platoon's lane or driving erratically in general provide obvious safety concerns that need to be addressed. These concerns are the same for the line-of-sight based following systems, with the additional factor that non-platoon vehicles may interfere with the ability to adequately track and follow the lead vehicle. Because of these issues, the ability to predict the behavior of surrounding vehicles is very important, especially predicting a potential cut-in which could cause a safety risk. The following section describes an application of the tracking portion of the GRIMM filter in a lane intrusion (i.e. vehicle cut-in) prediction/detection system.

6.1 Lane Intrusion Prediction and Detection

The proposed lane intrusion prediction/detection system is composed of three ground vehicles. The first vehicle is refered to as the lead vehicle, the second vehicle is refered to as the follow vehicle, and the third vehicle is referred to as the threat vehicle. The lead and follow vehicles make up the platoon and are assumed to follow the same path with some longitudinal spacing between them, and the threat vehicle is tracked in order to predict/detect its danger to the platoon. Two zones surrounding the platoon path are defined for use in determining the danger of the platoon. These zones are referred to as the Threat Zone and the Danger Zone. Both zones bulge directly around the follow vehicle in order to provide an earlier detection when collisions are more likely. The configuration of the proposed system is shown in Figure 6.1.

The follow vehicle utilizes the GRIMM filter described previously in order to track the threat vehicle. The follow vehicle has access to GPS measurements and uses an onboard radar unit to take measurements of the threat vehicle. The GRIMM filter is used to generate an estimate of the threat vehicles state in the navigation frame. This is then compared to the platoon path to determine the alert level. A three tier alert level is defined in order to quantify the level of danger that the threat vehicle poses to the platoon. The three levels are safe, caution, and danger which correspond to the colors green, yellow, and red. If the threat vehicle's state meets certain requirements, then the alert level is changed. These requirements are based on two factors. The first factor is the distance of the threat vehicle perpendicular to the platoon path. The second factor is the velocity of the threat perpendicular to the platoon path. These

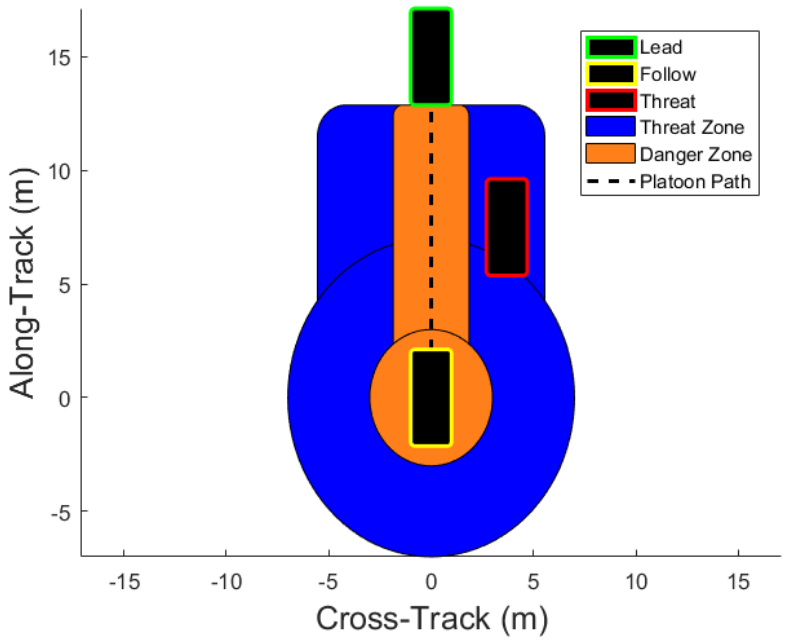

Figure 6.1: Configuration of Lane Intrusion Prediction/Detection Zones

two factors are combined in a logic structure in order to determine the proper alert level. The logic used to determine the alert level is shown in Algorithm 1.

Algorithm 1 GRIMM Lane Intrusion Prediction/Detection Logic

```
1:  procedure AlertLevel(P_⊥, V_⊥)
2:      if P_⊥ < δ_danger then                    ▷ Threat in Danger Zone
3:          alertLevel ← RED
4:      else if P_⊥ < δ_threat then               ▷ Threat in Threat Zone
5:          if V_⊥ > V_threshold then             ▷ Normal Velocity over Threshold
6:              alertLevel ← RED
7:          else
8:              alertLevel ← YELLOW
9:          end if
10:     else
11:         alertLevel ← GREEN
12:     end if
13: end procedure
```

The variables $P_\perp$ and $V_\perp$ are the position and velocity of the threat vehicle normal to the platoon path, δ_{threat} and δ_{danger} are the distances from the platoon path to the edges of the threat and danger zones respectively, and $V_{threshold}$ is the velocity threshold which is compared against the normal velocity of the threat vehicle.

Two scenarios were simulated in order to test the lane intrusion system. In the first scenario, the platoon is travelling in a straight line with a fifteen meter spacing between the lead and follow vehicle. The threat travels beside the convoy and performs an unstable oscillation in order to simulate an instability due to a tire blowout or some other mechanical failure. The layout of the first scenario can be seen in Figure 6.2.

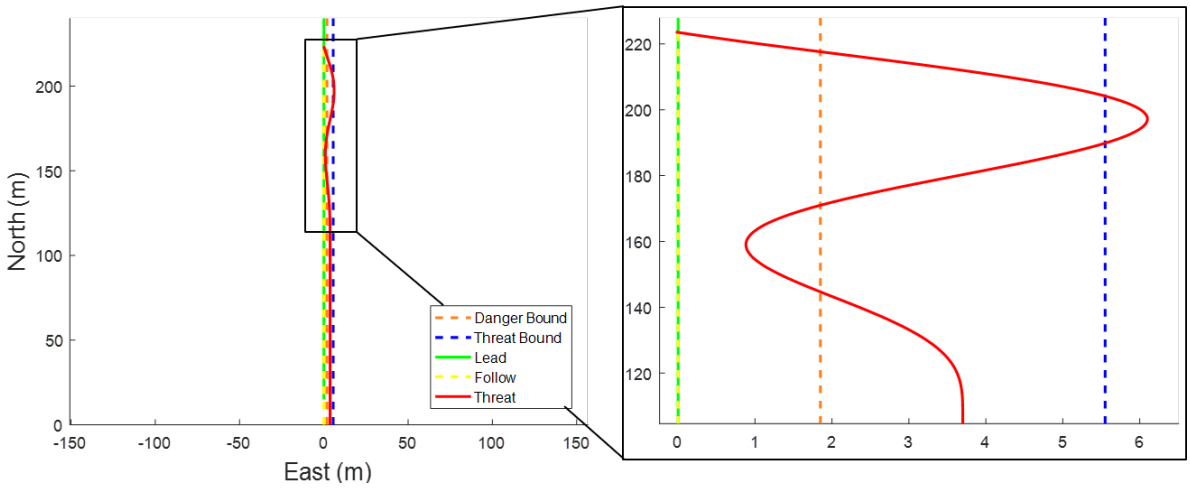

Figure 6.2: Lane Intrusion Scenario 1

In the second scenario, the platoon is travelling around a bend in the road, again with a fifteen meter longitudinal spacing between the lead and follow vehicles. In this case, the threat vehicle is traveling alongside the platoon on the inside of the turn and drifts accross the path of the platoon during the turn, ending up on the opposite side of the platoon after the maneuver. The layout of scenario two is shown in Figure 6.3.

The results from the first lane intrusion scenario can be seen in Figure 6.4. Upon inspection, the logic described in Algorithm 1 is evident in the results. During the first half of the scenario, when the threat vehicle is inside the threat zone but travelling parallel with the platoon, the system is at alert level Caution. When the vehicle begins to lose control, but before it enters the danger zone, the system has already elevated to alert level Danger. This demonstrates the importance of both a position and velocity threshold with which to judge the threat imposed by surrounding vehicles. Then the vehicle moves into the danger zone, where the alert level remains at Danger. Once the vehicle starts swerving back out of the danger zone, the alert is lowered to Caution. Then, the alert level briefly moves to Safe as the threat is outside of the

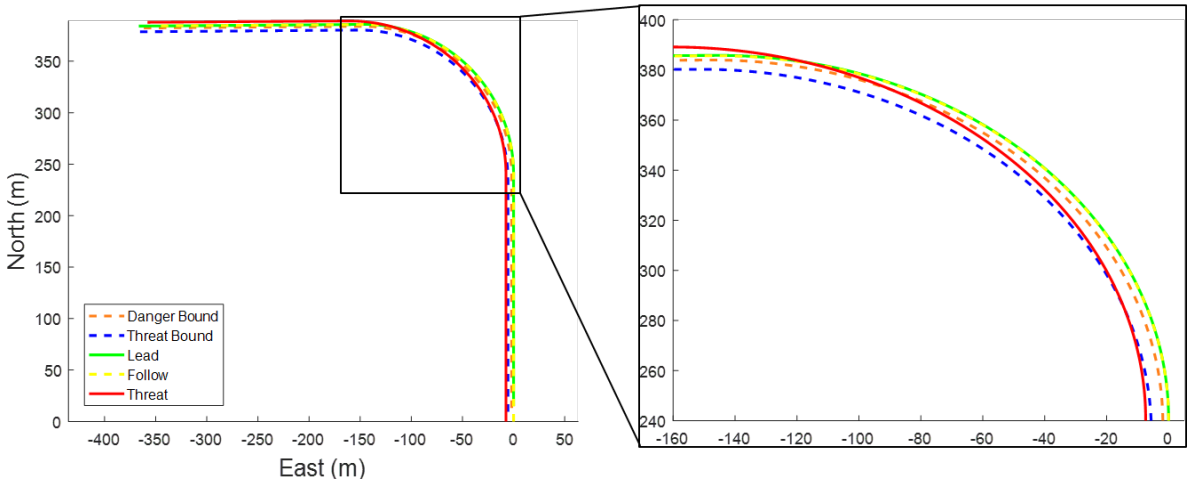

Figure 6.3: Lane Intrusion Scenario 2

threat zone, until it moves back into the threat zone with enough velocity normal to the path to elevate the alert level back to Danger.

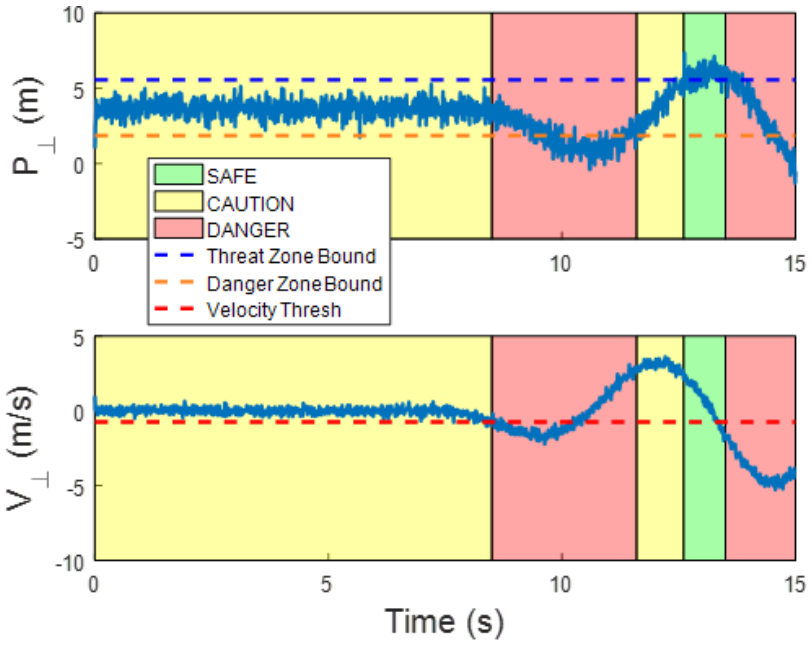

Figure 6.4: Lane Intrusion Scenario 1 Results

The results from the second lane intrusion scenario can be seen in Figure 6.5. During the first third of the scenario, the threat vehicle is outside of the threat zone, and so the system is at alert level Safe. Soon after the start of the turn, the threat vehic le begins moving into the platoon's path. Because the velocity threshold is relatively low, the system is already triggered to move up to alert level Danger as soon as the threat crosses the threat bound. As the threat

moves across the platoon path, the follower continues to track it and maintains the Danger level. As the platoon's path straightens out after the turn heading West, the threat vehicle falls in beside the platoon within the threat zone. Because the threat is travelling parallel with the platoon, however, the alert level gets lowered to Caution.

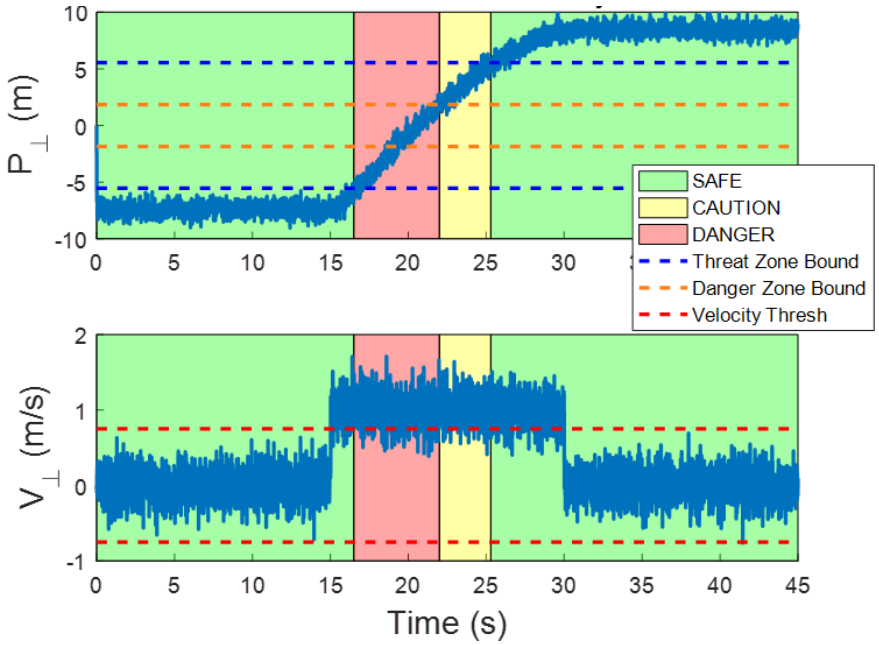

Figure 6.5: Lane Intrusion Scenario 2 Results

The two scenarios shown obviously do not encompass all of the situations in which cut-ins or other poor driving can endanger a platoon, but they do provide a first look at how a possible prediction/detection system may perform. While the alert levels defined here may seem arbitrary, when coupled with other systems within the autonomous platooning realm, they could become very useful. The different levels could provide input for the onboard control systems in order to begin planning crash avoidance maneuvers before it may be too late. These alert levels could also be different for each pair of vehicles if the platoon contained more than two vehicles. This could allow the platoon to maintain different spacings vehicle to vehicle based on the alert levels in that area, leading to safer platoon formations optimizing both safety and fuel efficiency.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

This thesis has discussed the development of vehicle to vehicle radar measurements for use in aiding navigation solutions when GPS data was unavailable. The goal of this thesis was to determine the feasibility of integrating radar measurements with GPS/INS navigation filters to provide external measurements to surrounding vehicles. The work in this thesis was performed assuming low-cost, automotive grade sensors which are already common in modern vehicles.

Collaborative navigation methods and radar tracking applications are common in the literature. However, there has been limited work in fusing GPS, IMU, and radar data in a collaborative navigation application. This thesis focused specifically on the development of a collaborative navigation scheme combining GPS/INS navigation data with radar data to aid vehicles without access to quality GPS measurements.

This thesis contributed the devlopment of a ground vehicle based Interacting Multiple Model (IMM) tracking algorithm, and the development of a collaborative framework in which the relative measurements generated from the tracking algorithm can be used to improve navigation solutions of other vehicles. Analysis of the proposed collaborative system compared to other common navigation methods is also shown in different simulated and experimental scenarios to verify the robustness of the proposed method. Applications in autonomous platooning were also discussed, and a method of lane intrusion prediction and detection was developed.

At the beginning of this thesis, the errors inherent in GPS and INS measurements were discussed in detail, and the models of these errors used throughout the work were introduced.

GPS/INS architectures were then discussed, and the rationale for using the Loosely Coupled GPS/INS filter in this work was given. The operation of the INS propagation and the time and measurement updates of the Loosely Coupled GPS/INS filter were described in detail.

The IMM algorithm was then described, along with the errors inherent in radar measurements. The expressions used to model the radar measurements were given as well. The details of the IMM algorithm were described, and an example application of the IMM filter was shown in order to more clearly illustrate its operation. The Constant Velocity (CV), Constant Acceleration (CA), and three versions of the Constant Turn (CT) models used in this work were given, and simulation results for each model were shown. The propagation of the IMM filter including the component models was then described in detail, including the usage of a non-linear set of measurement equations.

Following the description of the GPS/INS and IMM systems, the combination of the two filters into the GPS/INS/Radar/IMM (GRIMM) algorithm was described. A two vehicle system was used to demonstrate the capabilities of the filter. The operation of the base vehicle navigation and tracking filters, and their combination into external measurements for the rover vehicle, was defined. The rover vehicle navigation system was discussed, and the representation of the GRIMM measurement covariance was given. The simulated and experimental scenarios used to validate the GRIMM performance capabilities were then introduced. A computationally simple radar channel selection method was described, as the radar unit used for the experimental scenarios provided measurements in multiple channels.

Results from three different simulation scenarios and two different experimental data sets were shown. The data was fed into the standalone INS, GPS/INS, and GRIMM filters in order to judge the GRIMM filter performance compared to the alternative approaches. Using both simulation and experimental data, it was shown that the GRIMM filter performed much better than the standalone INS filter, and had performance similar to that of the GPS/INS filter. These results demonstrated that the proposed GRIMM filter is a method of collaborative navigation which is feasible for use in mitigating navigational errors caused by local GPS outages or potentially GPS degradation.

Finally, applications in the autonomous platooning realm were discussed, and a method of utilizing the GRIMM tracking algorithm for lane intrusion prediction and detection was presented. This method was validated with multiple simulated scenarios in order to show the GRIMM filter's value in such an application.

7.2 Future Work

There are many different avenues that can be taken to further this work. An assumption that was central to this thesis was that the communication between vehicles was perfect. As discussed at the beginning of this work, there are many factors that complicate the issue of vehicle-to-vehicle communication. The development of communication channels between vehicles that is reliable and fault tolerant would be very important to the real-time implementation of this work. The development of the navigation filters for real-time use in ROS or some other publisher-subscriber type operating system is another segment of the implementation of this work that was not explored in this thesis. The timing of the sensors local to each vehicle and between vehicles was challenging in the post-processing of the experimental data in this work, but would be more challenging and must be addressed for real-time implementation.

The Loosely Coupled GPS/INS filter was used in this thesis because the common types of GPS receivers used in ground vehicles would not allow access to internal measurements that the higher levels of coupling require. However, the structure of the GRIMM algorithm does allow for different levels of coupling. Therefore, further work could be done incorporating those different navigation filters into the GRIMM architecture. The more closely integrated GPS/INS filters generally provide more accurate results than the Loosely Coupled filter, so their integration into the GRIMM algorithm would likely produce favorable results. Methods such as the Closely Coupled GPS/INS filter generate estimates based on pseudoranges to satellites instead of the GPS solution itself. This would allow for portions of GPS measurements to be used in degraded environments where a full GPS solution is not available.

Another area that could be furthered is the inclusion of different models into the IMM filter. The CV, CA, and CT models used in this work are good base models for the filter, but more specialized models could be added to describe certain specific motions that vehicles commonly

undergo. Variations of the existing models could also be added with different tunings in order to cover more possible dynamics. Different types of Gaussian sum filters could also be explored. As mentioned in Chapter 3, the tuning of the state switching matrix in the IMM is important to the performance of the filter, but is commonly done by hand. Further investigation of the effects of various tunings of this matrix on the overall dynamics of the mixed solution could be a good avenue to further improve this work. The alternative channel selection methods as described in Chapter 5 could also be explored and applied to this work, namely the PDAF. This would be a robust method of channel selection in more complex scenarios with more clutter nearby that the radar may erroneously track.

Finally, applying the GRIMM framework to more multi-vehicle scenarios could also be an area of further research. Single-base multi-rover, multi-base single-rover, and multi-base multi-rover systems all pose unique challenges to the integration of navigation data. The ability channel selection task becomes more challenging in these scenarios, and could provide interesting contributions to the field. The GRIMM algorithm could also be integrated with leader-follower, obstacle avoidance, and other autonomous ground vehicle applications in order to provide better tracking of targets than standalone measurements could provide. Further along those lines, measurements from lidar, Ultra-Wide Band (UWB) radio, or optical sensors could be integrated into the GRIMM system to further improve the estimates of relative states to surrounding objects.

References

[1] M. Puyol, M. Frassl, and P. Robertson, "Collaborative mapping for pedestrian navigation in security applications," 2012.

[2] A. e. A. Kealy, "Collaborative navigation as a solution for pnt application sin gnss challenged environments: Report on field trials of a joint fig/iag working group," *Journal of Applied Geodesy*, 2015.

[3] A. e. A. Kealy, "Collaborative navigation with ground vehicles and personal navigators," *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–8, 2012.

[4] K. Jo, K. Chu, and M. Sunwoo, "Interacting multiple model filter-based sensor fusion of gps with in-vehicle sensors for real-time vehicle positioning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 1, pp. 329–343, 2012.

[5] D.-J. Jwo, C.-W. Hu, and C.-H. Tseng, "Nonlinear filtering with imm algorithm for ultratight gps/ins integration," *International Journal of Advanced Robotic Systems*, vol. 10, no. 5, p. 222, 2013.

[6] L. Perera, P. Oliveira, and C. Soares, "Maritime traffic monitoring based on vessel detection, tracking, state estimation, and trajectory prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1188–1200, 2012.

[7] S.-S. Jan and Y.-C. Kao, "Radar tracking with an interacting multiple model and probabilistic data association filter for civil aviation applications," *Sensors*, vol. 13, no. 5, pp. 6636–6650, 2013.

[8] B. Cook, T. Arnett, O. Macmann, and M. Kumar, "Real-time radar-based tracking and state estimation of multiple non-conformant aircraft," p. 1133, 2017.

[9] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, "Interacting multiple model methods in target tracking: a survey," *IEEE Transactions on aerospace and electronic systems*, vol. 34, no. 1, pp. 103–123, 1998.

[10] Z. Cao, Y. Mao, C. Deng, Q. Liu, and J. Chen, "Research of maneuvering target prediction and tracking technology based on imm algorithm," vol. 9684, p. 968430, 2016.

[11] T. Yuan, K. Krishnan, and Q. e. A. Chen, "Object matching for inter-vehicle communication systems: An imm-based track association approach with sequential multiple hypothesis test," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 12, pp. 3501–3512, 2017.

[12] R. Huang, H. Liang, P. Zhao, B. Yu, and X. Geng, "Intent-estimation-and motion-model-based collision avoidance method for autonomous vehicles in urban environments," *Applied Sciences*, vol. 7, no. 5, p. 457, 2017.

[13] X. Li and V. Jilkov, "A survey of maneuvering target tracking: Dynamic models," vol. 6, no. 4048, pp. 212–235, 2000.

[14] Y.-S. Kim and K.-S. Hong, "An imm algorithm for tracking maneuvering vehicles in an adaptive cruise control environment," *International Journal of Control, Automation, and Systems*, vol. 2, no. 3, pp. 310–318, 2004.

[15] P. Misra and P. Enge, *Global positioning system: signals, measurements, and performance*. Ganga-Jamuna Press, 2011.

[16] P. D. Groves, *Principles of GNSS, intertial, and multisensor integrated navigation systems*. Artech House, 2013.

[17] R. Brown, *Introduction to Random Signals and Applied Kalman Filtering*. Wiley and Sons, 2012.

[18] D. Simon, *Optimal State Estimation*. Wiley and Sons, 2006.

[19] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman filter: particle filters for tracking applications*. Artech House, 2004.

[20] X.-R. Li and Y. Bar-Shalom, "Design of an interacting multiple model algorithm for air traffic control tracking," *IEEE Transactions on Control Systems Technology*, vol. 1, no. 3, pp. 186–194, 1993.

[21] M. Skolnik, *Introduction to Radar Systems*. McGraw Hill, 2001.

[22] G. Galati and F. Struder, "Angular accuracy of the binary moving window radar detector," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-18, no. 4, pp. 413–422, 1982.

[23] M. Richards, J. Scheer, and W. Holm, *Principles of Modern Radar: Basic Principles*. Scitech Publishing, 2010.

[24] A. Genovese, "The interacting multiple model algorithm for accurate state estimation of maneuvering targets," *Johns Hopkins APL technical digest*, vol. 22, no. 4, pp. 614–623, 2001.

[25] K. e. A. Granstrom, "Systematic approach to imm mixing for unequal dimension states," *IEEE Transactions on Aerospace and Electronic Systems*, 2015.

[26] G. Zhai, H. Meng, and X. Wang, "A constant speed changing rate and constant turn rate model for maneuvering target tracking," vol. 14, pp. 5239–5253, March 2014.

[27] W. D. B. Gregory A. Watson, "Imm algorithm for tracking targets that maneuver through coordinated turns," 1992.

[28] J. Kim, "Comparison between three spiraling ballistic missile state estimators," *American Institute of Aeronautics and Astronautics*, 2008.

[29] T. Sherer, "Radar probabilistic data association filter with gps aiding for target selection and relative position determination," 2017.

[30] A. Al-Hourani, R. Evans, S. Kandeepan, B. Moran, and H. Eltom, "Stochastic geometry methods for modeling automotive radar interference," *IEEE Transactions on Intelligent Transportation Systems*, 2017.

[31] A. Hooper and D. Murray, *An Analysis of the Operational Costs of Trucking: 2017 Update*. American Transportation Research Institute, 2017.

[32] H. L. Humphreys, J. Batterson, D. Bevly, and R. Schubert, "An evaluation of the fuel economy benefits of a driver assistive truck platooning prototype using simulation," *SAE International Technical Papers*, 2016.

[33] E. D. Kaplan, *Understanding GPS/GNSS: Principles and Applications*. Artech House, 3 ed., 2017.

Appendices

Appendix A

Description of Coordinate Frames and Transformations

Transformations between various coordinate frames were used in the work described in this thesis. These are especially useful for moving GPS data into useful frames for navigation. A description of the most common coordinate frames and transformation techniques are shown in this appendix.

A.1 Coordinate Frames

In general, positions within coordinate frames are defined by three values, as necessitated by our three dimensional physical world. Most coordinate systems are Cartesian, with a triad of axes marking the origin of the system and the orientation of said triad defining the system. There are some examples of coordinate systems that utilize polar or spherical coordinate systems, in which a combination of one or two angles from some defined triad and a range from the origin are used to define the position of an object within that space. The five most commonly used coordinate systems are described in the following sections.

A.1.1 Earth Centered Inertial Frame

The Earth Centered Intertial (ECI) coordinate frame is a Cartesian frame with the origin at the center of mass of the Earth [33]. The X-Y plane of the ECI system is defined as coplanar with the equatorial plane of the Earth, with the X axis pointing in a direction that is fixed relative to the celestial sphere. The Z axis is taken to be normal to the X-Y plane in the direction of the North pole, and the Y axis is defined in such a way to complete the triad in a right-handed form. Because the orientation of the equatorial plane is not constant, it is common to define

the ECI coordinate frame based on a previous orientation of the Earth. The J2000 system is the most common ECI standard. The ECI coordinate system is useful for defining the motion of satellites and other space vehicles, as it does not rotate with the Earth.

A.1.2 Geodetic and Geocentric LLA Frame

The Latitude-Longitude-Attitude (LLA) frame is a spherical-type coordinate system fixed to the Earth. The user's position is defined by an angle from the Equator (latitude), an angle from the Prime Meridian (longitude), and the height above the reference sea-level (altitude). The latitude is generally defined as positive North of the Equator, and the longitude is generally defined as positive East of the Prime Meridian. These coordinates can be described relative to the center of the Earth (Geocentric), but this method assumes that the Earth is spherical. The most common form of the LLA frame is in a Geodetic configuration.

The Geodetic LLA system assumes the Earth is a shape called an oblate spheroid. This represents the Earth as a surface generated by rotating an ellipse alligned with the plane of the Prime Meridian and rotated about the Earth's axis of rotation. This generates a solid which is bulged slightly at the equator and flattened at the poles. The longitude values are unaffected by this change, but the latitude values must be computed differently since the normal of the Earth's surface does not align with the center of the Earth, as shown in Figure A.1. The variables shown in Figure A.1 are defined in Table A.1.

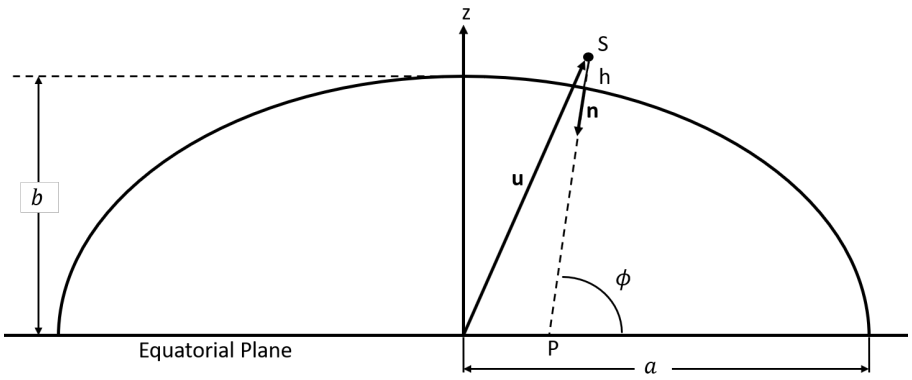

Figure A.1: Ellipsoidal Representation of Latitude and Altitude Coordinates

Because the LLA coordinate frame is 'attached' to the Earth, it is relatively easy to find the Cartesian coordinates of the user in terms of the LLA values. This calculation is shown in

Table A.1: Definitions of Variables within Ellipsoidal LLA Representation

Variable	Definition	Units
S	User Location	N/A
$\boldsymbol{u}$	Vector in ECEF from Origin to User	m
a	Semimajor Axis Length	m
b	Semiminor Axis Length	m
$\boldsymbol{n}$	Unit Vector of Ellipsoid Normal	N/A
P	Intersection of n and Equatorial Plane	N/A
h	Height of S above Ellipsoid Normal	m
ϕ	Geodetic Latitude	deg

Equation (A.1).

$$\boldsymbol{u} = \begin{bmatrix} \frac{a\cos\lambda}{\sqrt{1+(1-e^2)\tan^2\phi}} + h\cos\lambda\cos\phi \\ \frac{a\sin\lambda}{\sqrt{1+(1-e^2)\tan^2\phi}} + h\sin\lambda\cos\phi \\ \frac{a(1-e^2\sin\phi)}{\sqrt{1-e^2\sin^2\phi}} + h\sin\phi \end{bmatrix} \tag{A.1}$$

where λ is the longitude measurement and e is the eccentricity of the reference ellipse. The three components generated in Equation (A.1) are the user coordinates in the Earth Centered Earth Fixed frame, which is described in the next section.

A.1.3 Earth Centered Earth Fixed Frame

The Earth-Centered Earth-Fixed (ECEF) coordinate frame is a Cartesian frame with the origin at the center of mass of the Earth, X axis through the Prime Meridian, Z axis through the reference North pole, and the Y axis perpendicular to the XZ plane in right-hand convention. It is convenient to compute positions of users in the ECEF system as it rotates with the Earth. Satellite position measurements generated by tracking stations are generally found in ECEF, and the transformation from ECEF to the ECI frame is kept track of to a high degree of accuracy. Over small time scales, the transformation from ECI to ECEF is relatively constant, but on longer time scales this transformation becomes complicated due to the uneven motion of the Earth such as rotation and precession. It is common to use the ECEF system as an in-between step when moving to either LLA or local navigation frames such as East North Up or North East Down. These local navigation frames are described in the next section.

A.1.4 Local Navigation Frames

The two common local navigation frames are East North Up (ENU) and North East Down (NED). Both frames are based on the local tangent of the reference ellipsoid. The origin of the local navigation frame is defined by a reference set of LLA coordinates. The X-Y plane of the local navigation frame is then set tangent to the ellipsoid at the reference LLA coordinates. In the case of the ENU coordinate frame, the Z axis set to the normal of the tangent plane pointing away from the Earth, the Y axis is aligned with the local meridian pointing North, and the X axis is perpendicular to the Y and Z axes pointing in the East direction. The NED frame is similar, but with the X and Y axes swapped, and with the Z axis pointing into the Earth instead of away. An Illustration of the ENU frame, which is used in this thesis, is shown in Figure A.2.

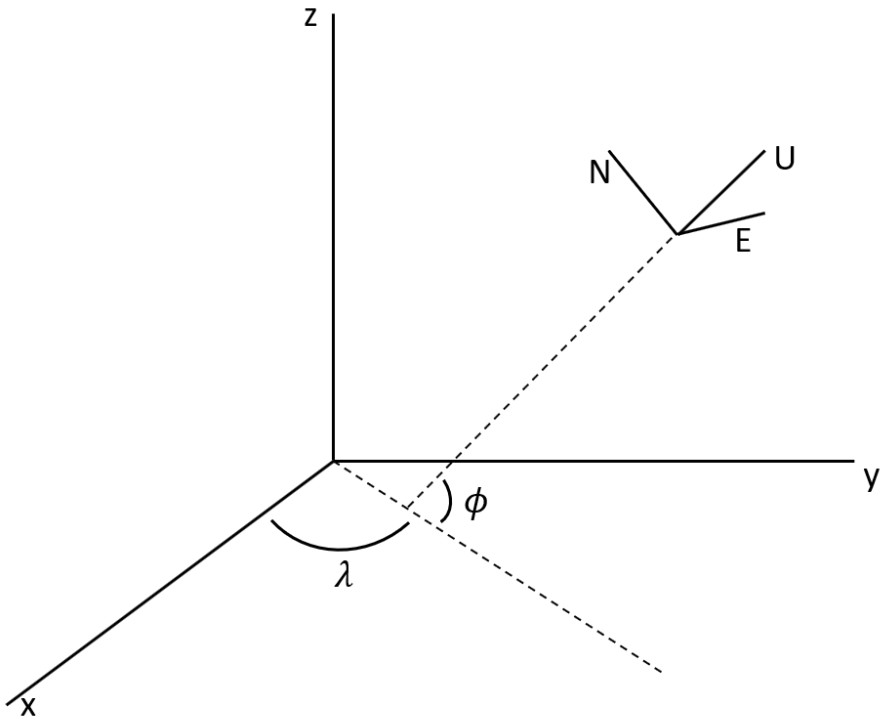

Figure A.2: Relationship of ECEF and Local Navigation Frame Systems

The ENU coordinates can be expressed as functions of the ECEF coordinates and reference latitude and longitude using the expression given in Equation (A.2).

$$\begin{bmatrix} e \\ n \\ u \end{bmatrix} = \begin{bmatrix} -\sin\lambda & \cos\lambda & 0 \\ -\sin\phi\cos\lambda & -\sin\phi\sin\lambda & \cos\phi \\ \cos\phi\cos\lambda & \cos\phi\sin\lambda & \sin\phi \end{bmatrix} \begin{bmatrix} x - x_o \\ y - y_o \\ z - z_o \end{bmatrix} \tag{A.2}$$

The variables x, y, and z are the ECEF coordinates of the user, and x_o, y_o, and z_o are the ECEF coordinates of the origin of the ENU frame.

A.1.5 Body Frame

As the name suggests, body fixed coordinate frames rotate with the dynamic body of interest. Body frames can be defined many ways depending on the system and the common usage in the related field. For ground vehicles, however, there are two common ways to define the body frame. In both, the origin is at the CG of the vehicle and the X axis is aligned with the longitudinal axis of the vehicle pointing forwards. One way to define the Y and Z axes is to have the Y axis pointing out of the right hand side of the vehicle, with the Z axis pointing vertically down. The other way, and the definition used in this thesis, is to have the Y axis pointed out of the left hand side of the vehicle, with the Z axis pointing vertically up. This definition is used in this thesis as it simplifies the rotations into the ENU navigation frame. The orientation of the body frame is shown in Figure A.3.

A.2 Coordinate Transformations

Coordinate transformations are used to transform coordinates from on frame to another. There are various ways to represent this transformation. The most commonly used representation of these transformations is Euler Angles, which are described in the following section.

A.2.1 Euler Angles

Euler angles are three angles that describe the relationship between two coordinate systems. This relationship is described by three component rotations defined by the three Euler

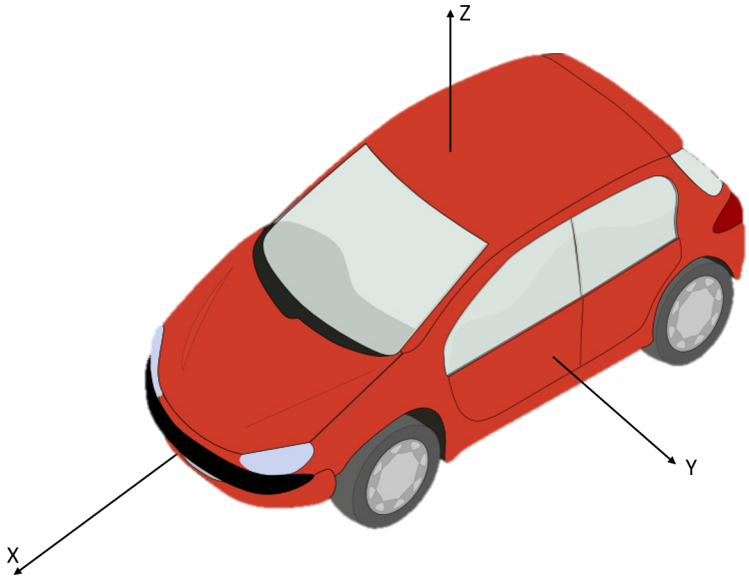

Figure A.3: Definition of the Body Fixed Coordinate Frame

angles. These three angles are defined as roll, pitch, and yaw, and are also referred to as the numbers one through three. Independent rotations can occur about each axis as defined by Equations (A.3), (A.4), and (A.5).

$$\boldsymbol{R}_1(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \tag{A.3}$$

$$\boldsymbol{R}_2(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \tag{A.4}$$

$$\boldsymbol{R}_3(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.5}$$

The variables $R1$, $R2$, and $R3$ represent the rotations about the three different axes. Three rotations are required to represent the difference between two coordinate frames with Euler angles. The most common type of rotation is the 3-2-1 rotation, in which the matrices described

in Equations (A.3)-(A.5) are multiplied together in that order. This multiplation is given in Equation (A.6).

$$\boldsymbol{R}_{321} = \boldsymbol{R}_3(\psi)\boldsymbol{R}_2(\theta)\boldsymbol{R}_1(\phi) \tag{A.6}$$

The variables ψ, θ, and ϕ are the yaw, pitch, and roll respectively. If this multiplication is broken apart, it can be seen that there are two intermediate coordinate frames which are moved between, as shown in Equations (A.7)-(A.9).

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \boldsymbol{R}_3(\psi) \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{A.7}$$

$$\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = \boldsymbol{R}_2(\theta) \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \tag{A.8}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \boldsymbol{R}_1(\phi) \begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} \tag{A.9}$$

The coordinate sets $[x, y, z]$ and $[X, Y, Z]$ are the initial and final coordinate frames, and $[x', y', z']$ and $[x'', y'', z'']$ are the intermediate frames. The spatial representation of this 3-2-1 rotation is given in Figure A.4.

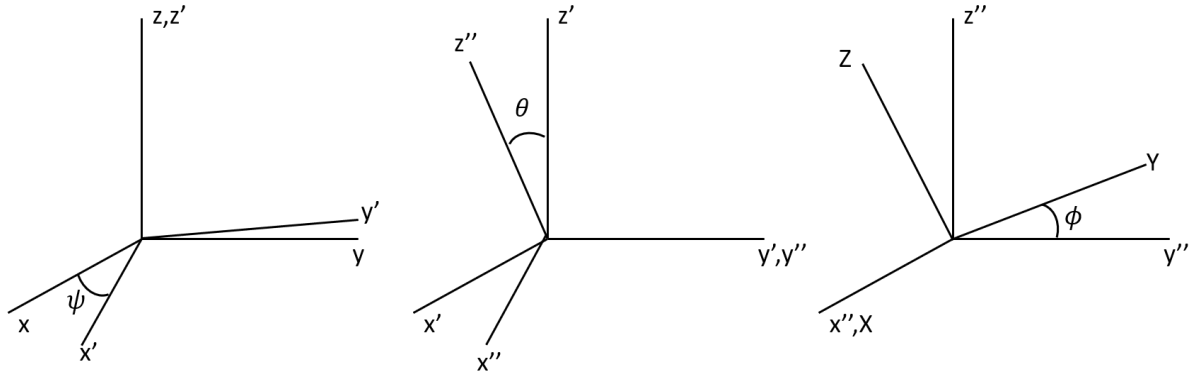

Figure A.4: Spatial Representation of Euler Angle Rotations