GNSS for Vehicle Control

For a list of titles in the *Artech House GNSS Technology and Applications Series*,
please turn to the back of the book.

GNSS for Vehicle Control

David M. Bevly
Stewart Cobb

ARTECH HOUSE
BOSTON | LONDON
artechhouse.com

Library of Congress Cataloging-in-Publication Data
A catalog record for this book is available from the U.S. Library of Congress.

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library.

ISBN 13: 978-1-59693-301-9

Cover design by Vicki Kane

© 2010 ARTECH HOUSE
685 Canton Street
Norwood, MA 02062

10 9 8 7 6 5 4 3 2 1

Contents

Preface

As Global Navigation Satellite Systems (GNSS) such as GPS have grown more pervasive, the use of GNSS to automatically control ground vehicles has drawn increasing interest. From autonomously driven vehicles such as those demonstrated in the DARPA "grand challenges" to automatically steered farm tractors, automated mining equipment, and military unmanned ground vehicles (UGVs), practical and potential applications of GNSS to ground vehicles abound. This text provides an introduction to the concepts necessary to understand and contribute to this field.

It has been said that *navigation* is "knowing where you are," *guidance* is "knowing where you're going," and *control* is "knowing how to get there." For example, suppose you are sitting at home one hot afternoon and you decide to ride your bicycle to the store to get an ice cream cone.

First, consider *navigation*. You know that you're starting at home. Furthermore, you know the names of all the streets all the way to the ice cream store and the names of all the nearby streets as well. At any point during the trip, you can look up at a street sign and know exactly where you are. You will have no trouble with navigation.

Second, consider *guidance*. You know where the store is, and you can think of many routes to get from your house to the store. Some routes would keep you on the paved streets, but it might be faster to take a shortcut across the park. It would be even faster to take a shortcut across the river, but the water is too deep; you can only cross at the bridge. You choose a feasible route in your mind which meets all the constraints you know of. If you have to leave your original route along the way, due to a parade or a street fair, you can choose a new route and still reach the store. Guidance will not be a problem.

Finally, consider *control*. You wheel your bike out the gate, remembering when you learned to ride it. At first, it wasn't easy to keep your balance; you had to learn to steer in the direction you were falling as soon as you started to fall. You also had to learn to steer briefly away from any turn you wanted to make, to change your balance so that you could lean into the turn. But you understand those algorithms now, and you can implement them fast enough to get where you want to go without falling over. Your control of the bicycle is adequate. You set off for the store.

Perhaps this example seems overly complicated. We humans perform most of these tasks unconsciously, most of the time. If we want to program automated systems to perform them, however, we must first understand them in detail. The purpose of this book is to develop an understanding of the navigation and control tasks for the special case of ground vehicles.

The result of a navigation algorithm is the same for every vehicle; your position is your position, whether you're on a bike, a bus, or a boat. However, each vehicle can have a different set of navigation sensors, and its navigation algorithm must be able to use the data from those sensors to compute the best possible position. Chapters 1, 2, and 7 of this book describe navigation sensors applicable to ground vehicles. Chapter 4 describes navigation algorithms that combine the data from these sensors to provide the best available estimate of position and velocity.

Control algorithms are different for every vehicle; buses, boats, and bicycles are steered in very different ways. Chapter 3 describes mathematical models for various categories of vehicles, with a particular emphasis on common four-wheeled vehicles. These models are used in Chapters 5 and 6 to develop vehicle control and estimation algorithms for tasks such as autonomous steering and electronic stability control (ESC).

The problem of guidance is not so easy to generalize. Each type of vehicle has different constraints on the routes it can take. In addition, a given vehicle's mission imposes other goals and constraints not shared even by similar vehicles on different missions. Guidance is therefore largely beyond the scope of this text.

The authors expect this book to be a useful introduction, for a graduate engineer or perhaps an advanced undergraduate, to the problems of navigating and controlling ground vehicles automatically. Because Global Navigation Satellite Systems (GNSS) are now widely available, Chapter 1 discusses concepts and navigation algorithms related to GNSS and to ground-based navigation transmitters known as "pseudolites." Chapter 1 also introduces inertial measurement instruments, which sense acceleration and rotation directly; compasses, which sense heading relative to the Earth's magnetic field; and odometers, which measure distance by counting the rotations of a wheel. Higher-level navigation algorithms, which combine data from these sensors, are presented in Chapter 4.

Chapter 2 describes the use of "machine vision" algorithms to detect a vehicle's position relative to road features such as lane markers, as seen through vehicle-mounted cameras and laser scanners (Lidar). These are especially useful for lateral navigation of a vehicle on a road for Lane Departure Warning (LDW) or lane-keeping assistance. These navigation algorithms combine locally generated measurements from a camera or Lidar with global measurements from GPS and a map database to form a robust measurement of position in the lane.

Chapter 3 introduces models for vehicles, tires, suspensions, and trailers. These models describe the behavior of highway vehicles such as passenger cars and SUVs as well as off-road vehicles such as farm tractors and unmanned ground vehicles (UGVs). This chapter describes the lateral and longitudinal dynamics that arise from these models and compares model data with measured data for a particular vehicle.

Chapter 4 describes methods for creating and updating an estimate of the navigation state using various combinations of the sensor measurements discussed in Chapter 1. These methods are based on the concept of the Kalman filter (a tutorial review of filtering and estimation techniques is provided as the Appendix to this book). Again, results from a simulation model are compared to data measured on a particular vehicle.

Chapter 5 describes methods for estimating parameters specific to four-wheeled vehicles, which are critical for modeling and control of vehicles. These estimated parameters can then be incorporated into a mathematical model of that specific vehicle for online vehicle modeling and vehicle control. Chapter 6 develops and analyzes control algorithms tuned to the models developed in the previous chapters.

Finally, Chapter 7 gives a detailed description of "pseudolites," ground-based transmitters of signals similar to GNSS signals, which are useful for navigating ground vehicles in restricted areas such as open-pit mines.

Acknowledgments

The authors would like to thank their editor, Mark Walsh, for his patience and guidance through the publication process.

David M. Bevly would like to thank the research team of current and former students in the GPS and Vehicle Dynamics Laboratory at Auburn University, specifically those who contributed to the chapters in this book. Additionally, I give thanks to the Lord my God, and His son Jesus, through whom all things are possible (Phil. 4:13).

Stewart Cobb would like to thank his colleagues at Stanford University, IntegriNautics, and Novariant for providing a stimulating, challenging, but always friendly work environment; and his daughter for her tolerance and understanding when Daddy was playing with his computer rather than with her.

1

GNSS and Other Navigation Sensors

Stewart Cobb and Benjamin Clark

Ground vehicles can navigate using signals from external navigation systems such as the Global Positioning System (GPS) (Figure 1.1), or by using signals from internal devices such as a compass, an odometer, a gyroscope, an accelerometer, or a full-blown inertial navigation system (INS). In practice, the most reliable and accurate navigation is obtained by combining data from all available sources, including static databases such as a digitized map. This chapter will discuss many of these sources of navigation data, and methods for combining them.

1.1 Global Navigation Satellite System (GNSS)

There are at least two Global Navigation Satellite Systems (GNSSs) currently in existence, and several more are proposed. These systems are all similar in concept, differing in small details of signal frequencies, signal structure, and orbit design. The concepts presented in this chapter should apply to any GNSS system, regardless of those differences. Where specific details must be cited, the United States' GPS will be used as an example. GPS is the most thoroughly studied GNSS, and the most useful at present.

1.1.1 Description of a Typical GNSS

A GNSS receiver navigates by precisely measuring the range between its antenna and a set of transmitter antennas at precisely known locations, and then performing a triangulation algorithm to determine its position. This is not quite as easy as it sounds, because the transmitters are aboard satellites moving rapidly through space, and the measurements must be made with nanosecond precision.

The space segment of a GNSS consists of a group, or *constellation*, of satellites in orbits that circle the Earth about twice per day. In order to provide adequate signal coverage to the whole earth, a constellation typically consists of 20 to 30 satellites in three to six different orbital planes. Some systems also include satellites in geosynchronous orbits.

The satellites broadcast microwave signals toward the Earth. Each satellite is far enough from Earth that its signal covers most of a hemisphere. Each signal consists of a carrier wave at a frequency near 1.6 GHz, modulated by a stream of digital bits at a rate of about 1 million bits per second (1 Mbps). The digital bits are generated in a way that is actually systematic but which appears random, and are called a *pseudorandom noise* code or PRN code. Each satellite has its own specific PRN code. The PRN code is itself modulated by digital *navigation data* at a slow rate (typically 50 bits per second).

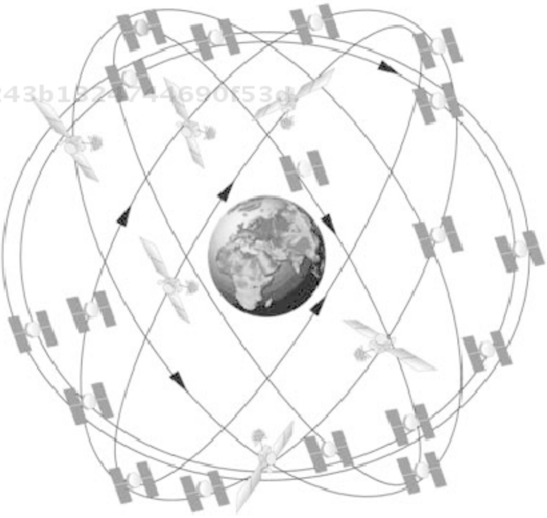


Figure 1.1 GPS satellite constellation, approximately to scale.

The frequency of each satellite's signal and the bit rate of its PRN code are controlled by an extremely precise clock (an *atomic clock*) on board the satellite. The uncompensated drift rate of each satellite's clock is typically a few nanoseconds per day. The satellite signal is designed so that a receiver which "hears" the signal can read the exact time of the satellite's clock at the instant the signal was transmitted, with an error of a few nanoseconds.

Each GNSS has a master control center, which constantly listens to the satellite signals through receivers in several different locations. It uses this information to compute the exact orbits and clock drift corrections for all the satellites, and transmits this information to each satellite in turn. This information is then broadcast by the satellite as part of the navigation data message. Each user receiver can interpret the navigation data to determine the precise time (according to the whole GNSS system, not just a particular satellite clock) that a signal was transmitted from a satellite, and the precise position of the satellite (within a meter or so) when it was transmitted.

At a particular instant (chosen according to its own internal clock), the receiver takes a snapshot of the clock readings of each satellite it can "hear." It then subtracts the reading of its own internal clock from the readings of the satellite clocks. The difference is the time taken by each signal to travel from its satellite to the receiver. Assuming that the signal traveled at the speed of light from the satellite directly to the receiver, the receiver can divide that time by the speed of light to compute the distance, or *pseudorange*, between that satellite and the receiver. (This distance is called "pseudorange" rather than simply "range" because it is in error by the amount that the receiver's own internal clock is in error. The next section shows how this error is corrected.)

The receiver then computes the position of each satellite at the time its signal was transmitted, using information contained in the navigation data message. Finally, knowing the distances (at the sampling instant) between its own position and several known points (the satellite positions), the receiver computes its own position, as shown in the next section.

For the remainder of this text, the GNSS receiver will be treated as a black box that provides pseudorange measurements and satellite positions. The details of the technology by which this is done are fascinating but are outside the scope of this book. The interested reader is referred to [1, 2].

1.1.2 Simple (Pseudorange) GNSS Navigation

Figure 1.2 illustrates the geometry of a pseudorange measurement. The range vector vector $\mathbf{r}_{iu}$ is the difference between the position of the user receiver ($\mathbf{r}_u$)

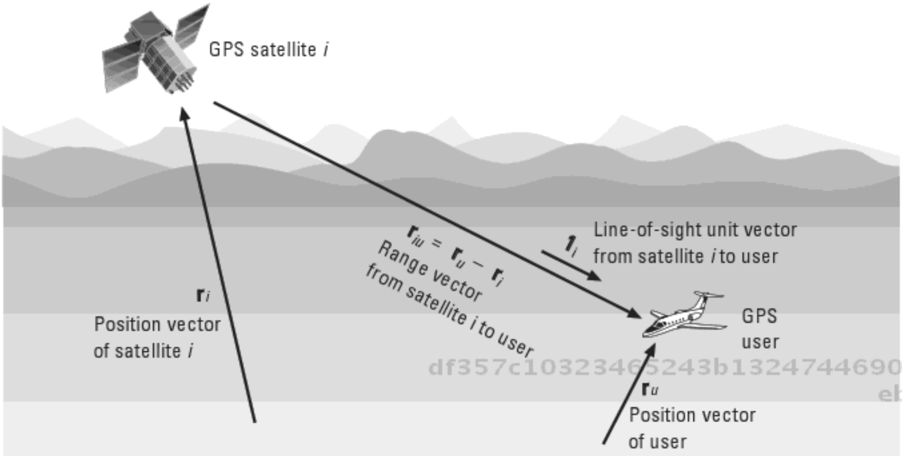


Figure 1.2 Geometry of a pseudorange measurement.

and the position of satellite i ($\mathbf{r}_i$). The pseudorange measurement includes the delay due to the length of this range vector along with various errors. Each pseudorange can be written as

$$\rho_i = \left| \mathbf{r}_u \quad \mathbf{r}_i \right| + c \times b_u + \varepsilon_i \tag{1.1}$$

where

ρ_i is the pseudorange measurement for satellite i;
$\mathbf{r}_u$ is the position of the user receiver;
$\mathbf{r}_i$ is the position of satellite i;
b_u is the receiver's clock bias;
c is the speed of light;
ε_i is the sum of measurement errors associated with satellite i.

To find its position, the receiver applies the principle of triangulation. Three known points (the satellite locations) define a plane, and the ranges to these points uniquely define two possible receiver locations, one above and one below the plane. One of these is generally far out in space and can be eliminated by inspection, leaving the other as the true receiver position.

In practice, the receiver must solve for its internal clock bias as well as for the three dimensions of its position, a total of four unknowns. The receiver needs pseudorange measurements from four different satellites in order to solve for those four unknowns. The solution process typically updates an initial position estimate $\hat{\mathbf{r}}_u$ using new measurement data. If no better initial position estimate is available, one can be obtained by averaging the subsatellite points for all satellites currently being tracked.

Given this initial position estimate and the known satellite positions, the difference ρ_i between the estimated and measured pseudoranges can then be written as

$$\Delta\rho_i = \hat{\rho}_i - \rho_i = \begin{bmatrix} \hat{\mathbf{1}}_i^T & 1 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{r}_u \\ c \cdot \Delta b_u \end{bmatrix} + \Delta\varepsilon_i \tag{1.2}$$

where

$\hat{\mathbf{1}}_i$ is the unit vector from satellite i to the receiver's estimated position;

$\mathbf{r}_u = \hat{\mathbf{r}}_u - \mathbf{r}_u$ is the difference between the receiver's estimated and actual positions;

$\Delta b_u = \hat{b}_u - b_u$ is the difference between the receiver's estimated and actual clock errors.

Four (or more) of these pseudorange equations can be stacked to form the matrix equation

$$\Delta\boldsymbol{\rho} = \mathbf{G}\Delta\mathbf{x} + \Delta\boldsymbol{\varepsilon} \tag{1.3}$$

where

$$\Delta\boldsymbol{\rho} \equiv \begin{bmatrix} \Delta\rho_1 \\ \Delta\rho_2 \\ \vdots \\ \Delta\rho_n \end{bmatrix}, \quad \mathbf{G} \equiv \begin{bmatrix} \hat{\mathbf{1}}_1^T & 1 \\ \hat{\mathbf{1}}_2^T & 1 \\ \vdots & \vdots \\ \hat{\mathbf{1}}_n^T & 1 \end{bmatrix}, \quad \Delta\mathbf{x} \equiv \begin{bmatrix} \Delta\mathbf{r}_u \\ c \cdot \Delta b_u \end{bmatrix}, \quad \text{and} \quad \Delta\boldsymbol{\varepsilon} \equiv \begin{bmatrix} \Delta\varepsilon_1 \\ \Delta\varepsilon_2 \\ \vdots \\ \Delta\varepsilon_n \end{bmatrix}$$

A correction $\hat{\mathbf{r}}_u$ to the initial position estimate $\hat{\mathbf{r}}_u$ can be obtained from this matrix equation by the method of least squares:

$$\Delta\hat{\mathbf{x}} = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\Delta\boldsymbol{\rho} \tag{1.4}$$

The result $\hat{\mathbf{x}}$ contains the position correction $\hat{\mathbf{r}}_u$ as well as a correction b_u to the clock bias estimate. The updated, accurate position $\hat{\mathbf{r}}_u^{new}$ is then computed as

$$\hat{\mathbf{r}}_u^{new} = \hat{\mathbf{r}}_u - \mathrm{D}\hat{\mathbf{r}}_u \tag{1.5}$$

This process is repeated until the correction $\hat{\mathbf{x}}$ is negligibly small.

When more than four pseudoranges are used in the navigation algorithm, a least-squares residual ρ^{new} can be calculated from the final position and the measured pseudoranges. When available, this residual is often used as a measure of the quality or accuracy of the new position estimate, in a process known as receiver autonomous integrity monitoring (RAIM) [2, Chapter 5].

Another quality measure, which is always available, is provided by the position covariance matrix $\mathbf{A}$ $(\mathbf{G}^T\mathbf{G})^{-1}$. The square root of the trace of $\mathbf{A}$ is known as the geometric dilution of precision (GDOP). GDOP describes the accuracy degradation of the position solution due solely to the relative positions of the satellites. Various components of GDOP describe the degradation in particular dimensions: horizontal DOP (HDOP), vertical DOP (VDOP), position DOP (PDOP), and time DOP (TDOP). The GDOP concept assumes that the errors in the individual pseudorange measurements are uncorrelated and have the same statistics. This concept is explained further in [1, Chapter 11].

This abbreviated discussion of the unassisted GNSS pseudorange navigation algorithm is provided as background for this chapter's discussion of enhancements to GNSS pseudorange navigation. For a more detailed explanation of this algorithm, please refer to [1, Chapter 9], which this presentation closely follows.

1.1.3 Differential GNSS Navigation

The GNSS signals measured by the user's receiver contain a number of errors or deviations from the mathematical ideal. The actual position of the GNSS satellite is only known to within a meter or two, and the timing of its clock may be off by a few nanoseconds. The radio signal from the satellite is delayed as it travels through the ionosphere and troposphere. The receiver can be fooled by signal reflections from nearby objects, known as *multipath*. All these errors except multipath are spatially correlated; that is, the sum of these errors will be similar for all receivers within a given area.

1.1.3.1 Local Area Differential GNSS Navigation

The local area differential GPS (LADGPS) technique reduces spatially correlated errors in the GNSS satellite signals to negligible levels. A LADGPS reference receiver, installed at a well-known location, computes an assumed pseudorange for each satellite signal it detects. It then measures the pseudorange for that satellite signal and subtracts the assumed pseudorange, forming a *differential correction*. The LADGPS reference station transmits these corrections as digital data to nearby user receivers.

Each user receiver adds this correction to the pseudorange it measures for the same satellite before performing the navigation algorithm described in the previous section. Errors common to both receivers, such as satellite clock errors, are entirely removed by this procedure. Other errors, such as ionosphere and troposphere delays and satellite ephemeris errors, are removed to the extent that they are spatially correlated. Uncorrelated errors, such as multipath and receiver noise, add directly to the user's navigation error, but a high-quality DGPS reference receiver will minimize them. User receivers are often within line-of-sight to the reference receiver, so LADGPS corrections are often broadcast by the reference receivers on short-range digital radio links.

DGPS concepts are further described in [2, Chapter 1].

1.1.3.2 Network Differential GNSS Navigation

Local area DGPS, as its name implies, requires a reference receiver within the local area of the user receiver. If the area of interest is large, many LADGPS reference receivers are required. Attempts to reduce the number of reference receivers required led to the concept of network differential GNSS navigation (NDGPS).

In network DPGS, a small number of widely spaced reference receivers are connected together with high-speed datalinks. By comparing their simultaneous measurements of a given satellite, they can estimate which portions of their measured signal errors are spatially correlated, and which are not. When a user receiver contacts the network and asks for differential corrections, the network projects the spatially correlated errors to the user's position, adds the uncorrelated errors, and transmits a set of differential correction data specific to that user's location. Because the area covered by the system is large, a user receiver is rarely within line-of-sight to a reference receiver. Network DPGS corrections are often distributed via data networks such as the cellular telephone system. Networks are often operated by state governments in the United States, by small countries elsewhere in the world, and by commercial organizations.

1.1.3.3 Wide Area Differential GNSS Navigation

The network GPS concept can be extended to cover an entire continent. This is known as wide area differential GNSS navigation (WADGPS). With many widely spaced reference receivers, the network can track a GNSS satellite over a large fraction of its orbit. By doing so, the network can compute accurate ephemeris and clock information for each satellite, regardless of the navigation data broadcast by the satellite. By tracking many satellites at once through many widely spaced receivers, the network can accurately compute the delays due to the ionosphere (and in some cases the troposphere) over most of the continent. The network can then transmit a set of processed data that any receiver on the continent can use to compute its own differential corrections. This data can even be distributed over a satellite broadcast channel to all user receivers on the continent. This is known as a space-based augmentation system (SBAS). An international standard for SBAS has been defined, and at least three compatible SBAS systems currently exist.

The United States operates an SBAS called the Wide Area Augmentation System (WAAS), which has been operational for several years. The European Union (EU) operates another SBAS, called European Geostationary Navigation Overlay Service (EGNOS), which is in the process of becoming operational. Japan operates yet another SBAS, the MTSAT Satellite-based Augmentation System (MSAS), which became operational in 2007. All these systems broadcast their corrections through geostationary satellites, using a signal format that is compatible with the GPS navigation signal format. This means that a user receiver needs no additional hardware, only some software, to take advantage of the SBAS corrections. Because the incremental cost of software is very low, and the signals are free to use, most new civilian receivers have SBAS capability built in. Although the accuracy varies with time, most receivers using an SBAS signal can expect to navigate within 2 meters of their actual position, most of the time.

There are also commercial providers of WADGPS services. The commercial systems transmit differential correction data using higher data rates than the government systems use, and claim higher accuracy as a result. These services are available by subscription and can cost tens or hundreds of dollars per month. The best-known of these services are OmniStar, operated by Fugro, and StarFire, operated by John Deere.

1.1.4 Precise (RTK) GNSS Navigation

As it travels through space at the speed of light, each bit of a GNSS satellite's PRN code is about 300 meters long. Each cycle of the carrier frequency is about 19 cm long. These are the features of the GNSS signal that receivers can measure. A good receiver can measure either feature with a precision of a fraction of 1 percent. The precision in range is about 0.5 meter for the PRN code and about 1 mm for the carrier phase. This improvement in measurement precision can permit a corresponding improvement in position accuracy—giving real-time positions with an error of only a few centimeters—once the problem of carrier-phase ambiguity is solved.

The PRN code is designed to be unambiguous; each bit of the code has a distinct signature and cannot be confused with its neighbors. Because of this, a receiver's PRN code measurement gives the pseudorange directly. This is not true for carrier phase measurements. Carrier cycles are not unique; each cycle looks just like every other cycle. The receiver can measure the fractional phase plus an arbitrary number of whole cycles, but cannot directly determine the exact number of whole cycles in the pseudorange. This number, known as the *integer cycle ambiguity*, must be determined by means other than direct measurement. Figure 1.3 illustrates these concepts.

As the fractional carrier phase passes through zero in the positive or negative direction, the receiver can increment or decrement an integer counter as appropriate. The *relative carrier phase* measurement consists of the instantaneous value of the integer counter plus the fractional phase. This measurement is also known as *integrated Doppler* or *carrier beat phase* or *accumulated delta range* (ADR). The integer cycle ambiguity is the difference between this relative carrier phase measurement and the actual pseudorange

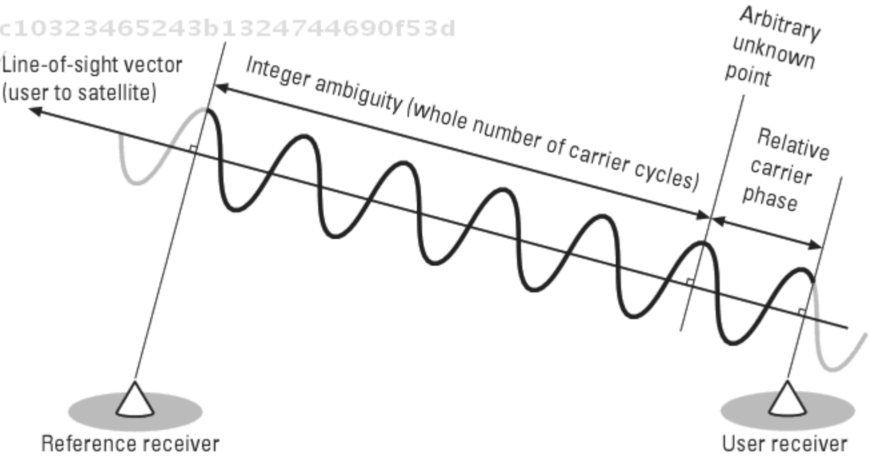


Figure 1.3 Carrier-phase measurements and integer ambiguity.

at any given instant. This integer ambiguity remains a constant for each signal as long as the receiver maintains continuous tracking of that signal.

Although it is theoretically possible to navigate using carrier-phase pseudoranges to the various satellites, carrier-phase navigation in practice is always done differentially. A reference station (or network) computes relative carrier phase measurements for each satellite in view at a fixed location, and transmits those measurements to the user receiver. The user receiver subtracts the reference measurements from its own similar measurements, forming a set of differential carrier-phase pseudorange measurements of the form

$$d\rho_i = (\mathbf{r}_u - \mathbf{r}_d) \times \mathbf{1}_i + N_i + d\varepsilon_i \tag{1.6}$$

where

$d\rho_i$ is the differential carrier phase measurement for satellite i;

$\mathbf{r}_u$ is the position of the user receiver;

$\mathbf{r}_d$ is the position of the reference receiver;

$\mathbf{1}_i$ is the unit vector from the user to satellite i;

N_i is the integer ambiguity associated with satellite i;

$d\varepsilon_i$ is the sum of differential measurement errors associated with satellite i.

The integer ambiguities N_i cannot be measured from the instantaneous GNSS signals but must be determined by other means. Methods for accomplishing this quickly and reliably remain an active area of research, but filtering sets of satellite measurements over time can generally determine the integer ambiguities with high confidence. Once the ambiguities are known, the position solution can be found using the algorithms of Section 1.1.2. This process is frequently called real-time kinematic (RTK) navigation because it was first developed for kinematic surveying applications. When done correctly, it can provide real-time GNSS positions with an accuracy of 1 to 2 cm.

In practice, most LADGPS and network DPGS systems are currently being used as reference stations for RTK navigation. WADGPS systems use reference stations that are too widely spaced to provide accurate carrier-phase reference data.

GPS carrier measurements can also be used to provide accurate three-dimensional velocity measurements. For ground vehicles, the velocity measurements give the direction of travel, also known as course over ground or simply vehicle course. A GPS receiver with multiple antennas can also measure the vehicle attitude (heading, pitch, and roll) in two or three dimensions [3, 4].

A two-antenna GPS receiver provides noisy, but unbiased, measurements of vehicle course ν and heading ψ as follows:

$$\nu^{(GPS)} = \nu + \eta_\nu \tag{1.7}$$

$$\psi^{(GPS)} = \psi + \eta_\psi \tag{1.8}$$

The sampled noise on these measurements for a typical vehicle installation can be modeled as:

$$\sigma_\nu^2 = \left(\frac{0.05 \text{ m/s}}{V}\right)^2 \tag{1.9}$$

$$\sigma_\psi^2 = \left(\frac{0.4\pi \text{ radians}}{180}\right)^2 \tag{1.10}$$

Note that the accuracy of the course estimate will improve with vehicle speed [5].

1.1.5 Current and Future GNSS Constellations

1.1.5.1 GPS

The best-known GNSS constellation is the United States' GPS system. This system became operational some 15 years ago, after almost two decades of testing, and has continued to evolve and improve ever since. It is built and operated by the U.S. Air Force. It was originally designed as a military system, but the main navigation signal was made available to civilians from the beginning. The published policy of the U.S. government is to keep the civil signals available and free to use worldwide. The specified accuracy of the civil navigation signals is 9 meters horizontal (95%) and 15 meters vertical (95%) under certain specified assumptions [6].

At present, the number of operational GPS satellites is generally in the high 20s, giving worldwide navigation availability with generally good PDOP.

The official technical documents describing the GPS system are known as Interface Standards (IS). The primary document is IS-GPS-200 [7].

1.1.5.2 GLONASS

The other existing GNSS constellation at present is the Russian GLONASS system ("GLONASS" is a Russian-language acronym meaning "global navigation satellite system"). It has been in development for almost as long as GPS, but has achieved somewhat less operational success. Many GLONASS satellites have been launched, but they tend to fail more quickly than the GPS satellites. As a result, a full constellation of GLONASS satellites has rarely been seen. An improved satellite design has recently entered service, and many more such satellites are scheduled to be launched soon, so that a full GLONASS constellation may be available in the next year or two. The official technical document describing the GLONASS system is [8].

1.1.5.3 Galileo

The European Union plans to build a GNSS to be known as "Galileo." Both the United States and the EU have worked hard to ensure that Galileo will be compatible with GPS. While Galileo's technical details have been fairly clear for a while, the program was stalled for years by confusion regarding business models and sources of funding. This has now been resolved, and two experimental satellites have been launched. The first validation spacecraft are expected to be launched in late 2010. The official technical document describing the Galileo system is [9].

1.1.5.4 Compass

The People's Republic of China has announced plans to build a GNSS of their own, known as "Beidou" in Mandarin or "Compass" in English. No official technical documents describing the Compass system have been released to date. What little information is available tends to indicate that Compass is similar to, and generally compatible with, GPS and Galileo. At least one experimental Compass satellite has been launched.

1.1.5.5 QZSS

Japan has announced the Quasi-Zenith Satellite System to improve navigation within the Japanese islands. Unlike the other constellations, QZSS consists of three satellites in geosynchronous (but not geostationary) orbit. The satellites are in inclined elliptical orbits, so that they appear to "hang" almost directly over the Japanese islands for more than half of each orbit (Figure 1.4). With three satellites following one another around the same ground

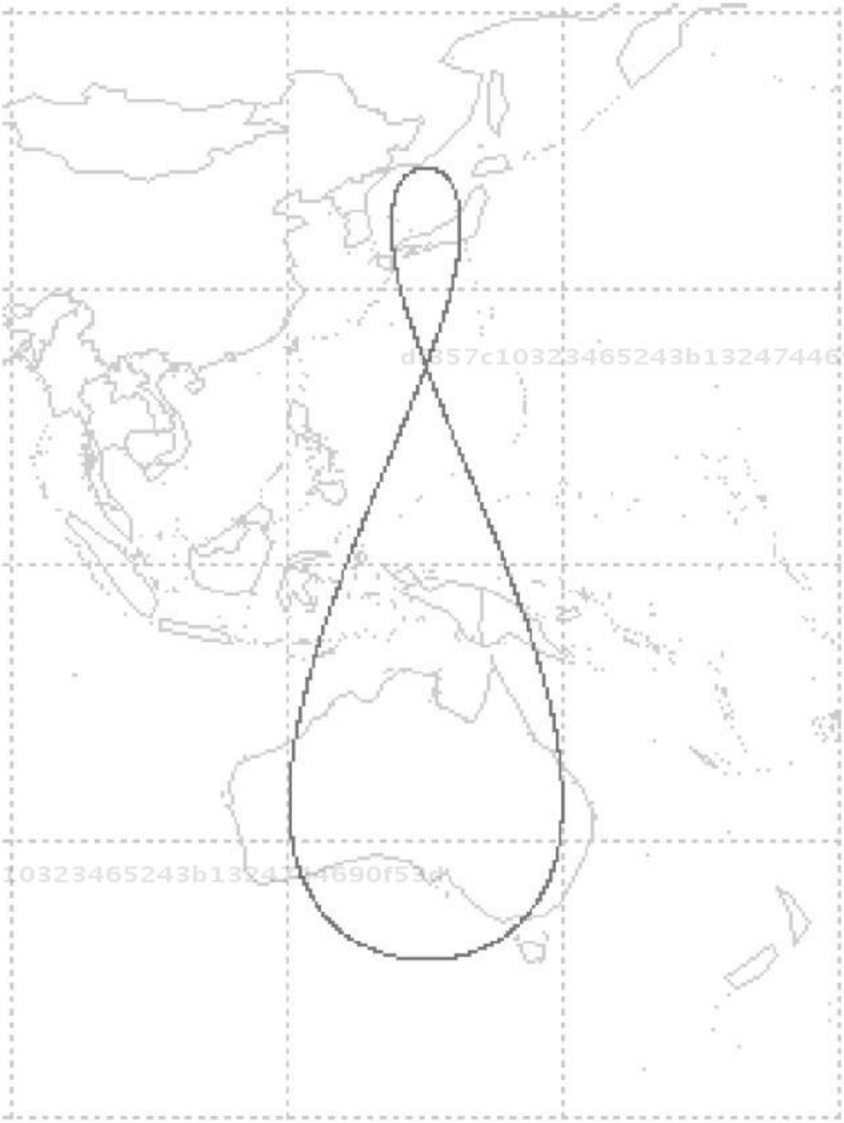


Figure 1.4 Japanese QZSS constellation ground track.

track, at least one will be nearly overhead (“quasi-zenith”) at any given time. The QZSS broadcast signals are intended to be compatible with GPS signals. The official technical document describing the QZSS system is [10]. The first QZSS satellite is expected to be launched in 2010.

1.2 Pseudolites

The previous section illustrated that GNSS receivers generally must measure pseudoranges from at least four different satellites simultaneously in order to navigate effectively. Four equations (the pseudoranges to each satellite) are used to solve for four unknowns (X, Y, Z, and time).

It is possible in some cases to navigate with less than four satellite signals. For example, it may be reasonable in a ground vehicle to assume that altitude (Z) is constant for short periods, so that navigation can proceed temporarily with only three signals. In other systems, individual pseudorange measurements may be incorporated into a model of vehicle dynamics that is also being updated with other data. For some applications, navigation with such a model is "good enough."

Other applications, however, will require at least four pseudoranges at all times. In many locations, foliage or terrain features block the signals from most of the satellite constellation. For example, Figure 7.1 in Chapter 7 shows a diagram of a deep open-pit or open-cast mine, typical of many copper mines worldwide, in which the walls of the pit block many satellite signals. Nevertheless, the mine's profitability depends on accurate positioning of each drill bit and each shovelful of ore. It is for these applications, in these locations, that *pseudolites* are useful.

1.2.1 Pseudolite Basics

The term pseudolite is derived from "pseudo-satellite." Pseudolites are ground-based devices that transmit navigation signals similar to those transmitted by a GNSS satellite. A pseudolite receiver uses these signals to calculate its position, just as conventional GNSS receivers use the GNSS satellite signals.

Some pseudolite systems are designed so that the receiver normally navigates with GNSS signals, and needs only one or two pseudolite signals to supplement the satellite signals when satellite visibility is poor. Other pseudolite systems are designed to operate in a stand-alone manner, entirely separate from any GNSS system. Both types have their uses, as explained in the following section.

1.2.2 Pseudolite/GNSS Navigation

Pseudolites can be designed to supplement conventional GNSS navigation. Such a pseudolite transmits a signal very similar to a GNSS satellite signal, and a user receiver can use the signals from one or more pseudolites—with or without the signals from GNSS satellites—to find its position. The user's navigation algorithm is similar to the satellite-only navigation algorithm described above, but it must be extended to cover certain differences between satellites and pseudolites.

Like the satellites, such a pseudolite contains a very precise clock synchronized to the GNSS system time. Achieving this synchronization can be difficult. The GNSS satellite clocks are monitored and adjusted by the GNSS system control center, which generally has an estimate of the amount by which each satellite clock is in error at any time. A pseudolite can only observe a fraction of the satellites at any time, and unless it is linked the control center, it has no knowledge of the instantaneous error of any satellite. The control center defines the GNSS system time, but the pseudolite must estimate it, so the pseudolite's clock will generally contain more error than the satellites' clocks. The navigation algorithm must weight the pseudolite and satellite pseudorange measurements appropriately to reach an optimum position solution.

A pseudolite transmits its signal, not from orbit, but from a fixed location on the Earth's surface. The navigation algorithm must be made aware of the pseudolite's location, either through the navigation data within the pseudolite's signal, or through a database or a separate transmission.

The satellite-only navigation algorithm implicitly assumes that the satellites are infinitely far away, so that the line-of-sight vectors to the satellites do not change significantly as the algorithm converges on the receiver's position. This is a reasonable assumption, because the distance from a satellite to the user is generally several times the radius of the Earth. This assumption is not valid for pseudolites, which may be quite close to the receiver. The navigation algorithm must recalculate the line-of-sight vectors to the pseudolites during each iteration. This makes the navigation problem nonlinear, which can cause a simplistic algorithm to converge slowly or to fail to converge at all.

All of these problems have been solved in various pseudolite systems. In fact, the GPS concept itself was originally tested using pseudolites placed on desert mesas, before any satellites had been launched.

1.2.3 Differential Pseudolite/GNSS Navigation

Precise clocks are expensive, and synchronizing them to GNSS system time can be difficult. An alternative is to build pseudolites with less expensive, less precise clocks, and to use a separate fixed receiver to provide differential corrections. This is an extension of the "differential GNSS" concept described above. The reference receiver takes simultaneous measurements of the pseudolite signals and the GNSS satellite signals and broadcasts the set of measurements to other "rover" receivers nearby. This set of measurements contains the information needed to determine the pseudolites' clock errors relative to GNSS system time at the instant of measurement. The rover receivers can use this data to determine their own positions more accurately, just as in conventional differential GNSS navigation.

It can be difficult to find a location for the reference receiver which allows it to monitor signals from all the pseudolites and all the satellites that every rover receiver wants to use for navigation. In such cases, two or more reference receivers can be used. If at least one signal (typically from a high-elevation satellite) is measured by all reference receivers, that signal can be used as a common time reference to determine the differences between the internal clocks of all the reference receivers. Once these differences are known, all of their measurements can be projected to a common time and used as if there were only one reference receiver taking all the measurements.

1.2.4 Pseudolite Self-Synchronization

An extension of this concept is to provide the pseudolites themselves with receivers capable of measuring signals from other pseudolites as well as from the GNSS satellites. Each individual pseudolite can then serve as a reference receiver, transmitting its measurements as data on its own pseudolite signal. This simplifies the installation of a pseudolite system, because a separate reference receiver (with its difficult location constraints) is no longer needed.

1.2.5 Stand-Alone Pseudolite Navigation

Taking the concept above to its logical extreme, a pseudolite system can be built that does not depend on GNSS at all. The pseudolites synchronize their clocks to each other rather than to a satellite reference. It is also possible for a fraction of the pseudolites to determine or refine their own positions during this process. The locations of a few pseudolites must be determined by external means, because the synchronization technique can only determine a pseudolite's location relative to other pseudolites in the system, not its absolute location.

1.2.6 Conflicts with GNSS Frequencies

There are few technological restrictions on the signal format or frequency that a pseudolite can transmit, but there are practical and legal restrictions. It might seem attractive to design a pseudolite system to transmit a signal on the same frequency as a GNSS system, with a similar signal format. This would seem to allow a simple GNSS receiver to take advantage of pseudolite signals as well, without requiring new radio hardware or much new software. Unfortunately, the real world is not so simple.

All GNSS satellite signals are weak—typically well below the level of thermal noise—so that the receiver must integrate its input for a long time to detect a usable signal. An incoming signal much stronger than the weak satellite signals can disrupt this integration process, obliterating the weak signals. The strength of a radio signal changes with the inverse square of the distance between the transmitter and the receiver. By design, the GNSS satellites are all at large and relatively constant distances away from a ground-based receiver, so the receiver does not see a satellite's signal strength vary widely while it is in view. Because of this and other clever features of the GNSS satellite signals, a user receiver can measure signals from many satellites at once, all transmitting on the same frequency.

A ground-based pseudolite can be designed and adjusted to present a similar received signal level to a user receiver a fixed distance away. As the user receiver moves, however, the signal level it receives from the pseudolite will change. It is not hard to imagine scenarios in which the distance between a pseudolite and a receiver can change by a factor of 100 or more, leading to a change in signal strength of 40 dB or more. Most nonmilitary GNSS receivers will be jammed into uselessness by an incoming signal 40 dB stronger than the satellite signals. It is also possible for a receiver to move far enough away from a pseudolite that it can no longer measure the pseudolite signal. This is less of a problem in practice, because more pseudolites can usually be added to fill in dead spots in a coverage area.

The variation of pseudolite signal strength with distance to a given receiver is known as the near/far problem in pseudolite research. It has been extensively studied [11], but no general solution exists. The best partial solution is to transmit a strong pseudolite signal in short pulses. The receiver can integrate and measure the weak satellite signals in between the pulses, and the pulses are strong enough (well above the noise level) that the pseudolite signal can be measured directly, without integration. Nevertheless, a pulsed pseudolite can still jam a GNSS receiver which is not designed to cope with pseudolite signals.

Recognizing this problem, most governments have set very tight limits on the amount of signal power that can be deliberately transmitted on the GNSS frequencies. These limits effectively prohibit the use of pseudolites on the GNSS frequencies in any outdoor application. In response, recent pseudolite systems have been designed to transmit their signals in other frequency bands, notably the license-free band near 2.45 GHz.

1.3 Inertial Navigation Systems (INS)

Since the motion of objects is governed by only a few physical rules, having some measure of these quantities allows self-contained navigation to be possible. Inertial sensors provide a link between electronics and the motion of a body they are monitoring by relating motion to signals. One advantage of inertial sensors over ranging systems such as GPS is the independence from external systems. Whereas a GPS signal can be blocked, jammed, or spoofed, inertial measurement units (IMUs) are immune to these external effects. This means that so long as the unit itself is operating correctly, motion information will be available to the user. Also, this information will be accurate to the specifications of the sensor itself. This advantage has been used for decades in various scenarios such as vehicle, aircraft, and missile navigation. The costs of manufacturing inertial sensors has been decreasing rapidly, so that sensors such as those shown in Figure 1.5 are now available at prices appropriate for vehicle navigation and control systems. Inertial sensors generally fall into two categories: accelerometers and gyroscopes.

1.3.1 Linear Inertial Instruments: Accelerometers

Accelerometers are used to relate signal levels to sensed specific forces along a particular sensor axis. These specific forces can be due to field reactions between bodies such as the attraction due to gravity, or inertial forces because of a change in motion. These latter accelerations are the values that are usually desired since keeping track of motion change allows the navigation system to track change in velocity and thus change in position. Therefore, given some initial position/velocity, these navigation states can be tracked by integrating the accelerations sensed by the accelerometer [12].

Most modern accelerometers in use in the vehicle dynamics field are microelectromechanical systems (MEMS). These devices operate on a proof-

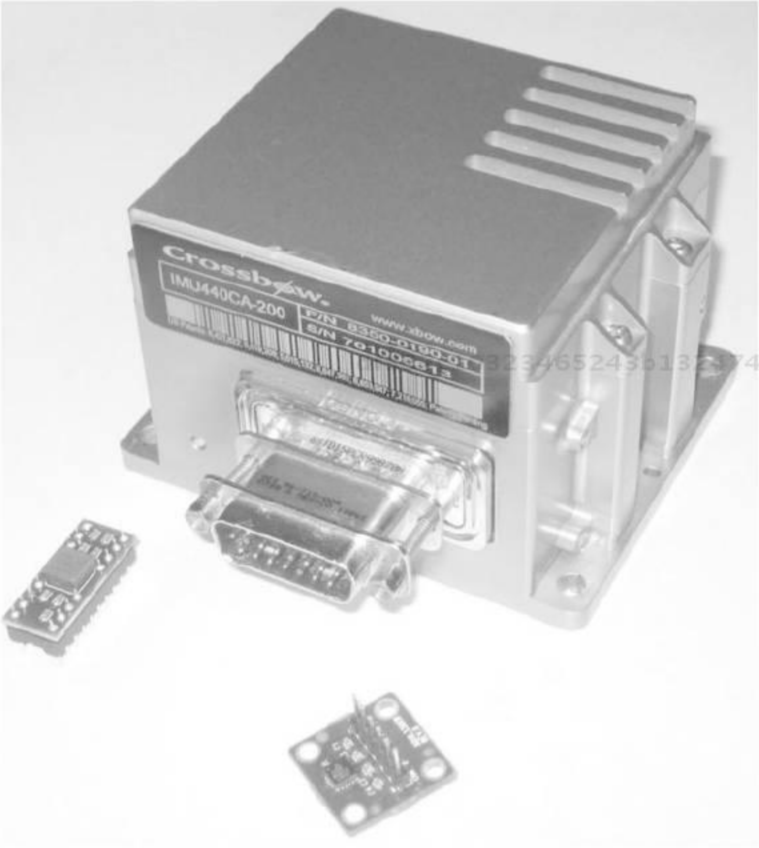


Figure 1.5 Single-axis accelerometer and gyroscope made with MEMS technology (foreground) and a three-axis IMU built with similar sensors (background).

mass concept relating to Newton's second law. A diagram of a simplified model for an accelerometer is shown in Figure 1.6.

This example accelerometer can be modeled as a dynamic system with an equation of motion

$$m\ddot{x} + b\dot{x} + kx = b\dot{u} + ku \tag{1.11}$$

This yields a system transfer function relating relative displacement $d = x \quad u$ to housing motion, u,

$$\frac{D}{U} = \frac{-\,m}{ms^2 + bs + k} \tag{1.12}$$

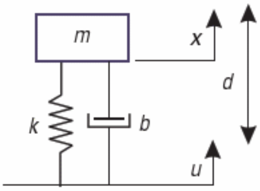


Figure 1.6 Accelerometer proof-mass model along a sensitive axis.

Therefore in steady state, the displacement is a measure of the acceleration of the accelerometer housing. Since this is the desired measurement, steady state conditions are desired, and therefore the accelerometer dynamics are much faster than those expected to be experienced by the housing. In this case, the accelerometer can be considered a scaled measurement of the housing acceleration, which is a function of the specific forces applied to the accelerometer housing.

1.3.2 Angular Inertial Instruments: Gyroscopes

Gyroscopes measure the rotation rate about a sensitive axis. Several types of gyros exist of varying accuracy, cost, and mechanics. Two typical gyroscope types found in vehicle applications include optical and vibratory type gyroscopes. Optical gyros operate by detecting changes in the path traveled by light as shown in Figure 1.7. When the gyroscope rotates along the same direction as the light, the light must travel further along each path segment, resulting in a longer path. The same is true in the opposite direction [13, 14].

Vibrational gyroscopes generate harmonic motion in a structure (typically MEMS) at a known high frequency. Considering the relative acceleration of two points, the relative acceleration can be expressed as

$$a = WWr - 2W\dot{r} + \ddot{r} \tag{1.13}$$

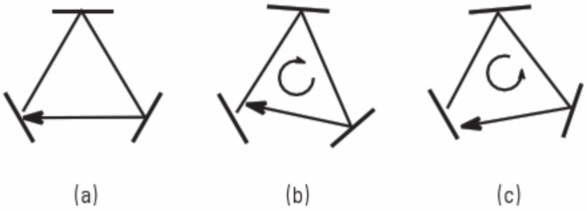


Figure 1.7 Light path for optical gyroscope to detect rotation with (a) no rotation, (b) rotation along the path, and (c) rotation opposite the path.

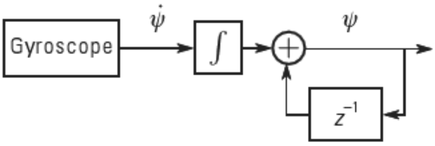


Figure 1.8 Generating attitude from ideal gyroscope measurement.

Since the relative position is induced by a periodic vibration, the r, $\dot{r}$, and $\ddot{r}$ terms are sinusoidal functions with factors of distance, distance times frequency, and distance times frequency squared. At a high vibration rate, the terms without frequency can be neglected and the resulting sensed output is proportional to , the angular rate to be detected [13].

1.3.3 Ideal Inertial Navigation

The use of inertial sensors allows for a navigation system to actively monitor its motion. In the field of vehicle control, however, the motion is not the only effect that needs to be provided to navigation and control subsystems. Often it is desired to know position and attitude from these motion measurements. Since only derivatives of these quantities are provided by inertial sensors, integration must occur to have useful position, velocity, or attitude information. Figure 1.8 shows the steps necessary to generate attitude from a single gyroscope output. Similarly, Figure 1.9 shows the steps to calculate velocity and position from a single accelerometer output.

As an example of the use of inertial sensors in navigation solution calculation, consider a simplified two-dimensional terrain [15]. A vehicle is equipped with longitudinal and lateral accelerometers and a yaw gyroscope positioned in the terrain with respect to an origin point as shown in Figure 1.10.

The position and heading of the vehicle are desired in a north-east coordinate frame, but the measurements are given in a coordinate frame aligned with the vehicle (body frame). In the ideal case, the gyro measurements are equivalent to the heading rate. Therefore, given the initial vehicle heading,

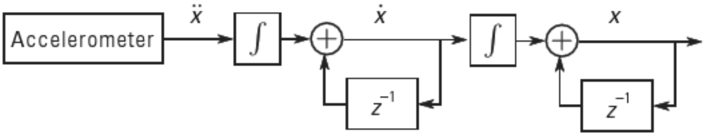


Figure 1.9 Generating velocity and position from ideal accelerometer measurement.

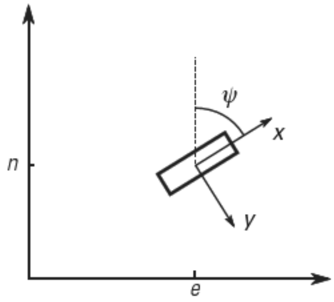


Figure 1.10 Position and heading of a vehicle in a two dimensional terrain.

the heading can be tracked by numerically integrating the yaw gyro output, g_z.

$$\dot{\psi} = g_z \tag{1.14}$$

The accelerometer outputs are not the north and east accelerations, but rather the longitudinal and lateral accelerations. Therefore these measurements must be rotated into the north-east frame by the heading angle.

$$a_n = a_x \cos(\psi) - \ a_y \sin(\psi) \tag{1.15}$$

$$a_e = a_x \sin(\psi) + a_y \cos(\psi) \tag{1.16}$$

These accelerations can then be integrated to get the north and east velocities.

$$\dot{v}_n = a_n \tag{1.17}$$

$$\dot{v}_e = a_e \tag{1.18}$$

These velocities are then integrated to get the north and east positions.

$$\dot{n} = v_n \tag{1.19}$$

$$\dot{e} = v_e \tag{1.20}$$

As an example, given the yaw gyro measurement shown in Figure 1.11, an initial heading of 0 degrees is integrated for the heading shown in Figure 1.12.

Example accelerometer measurements are shown in Figure 1.13, which can be combined with the previously calculated heading to generate the vehicle position plot in Figure 1.14.

1.3.4 Sensing Earth Effects

Since the sensor is measuring the force applied to the proof mass, placing the sensor on a level surface in the Earth's gravitation field will give a reading of approximately 9.81 $^{m}/_{s^2}$ in the up direction. This is because the mass is experiencing a force from the body of the sensor to keep it in place that counteracts the force due to gravity. If the sensitive axis of the accelerometer is in the direction of travel of a vehicle, a positive acceleration will give a measurement in the direction of travel since the body of the sensor is forcing the mass into motion in the direction of the vehicle. This concept is important in understanding that the sensor does not measure "gravity," but can register it due to the reaction forces applied to the proof mass.

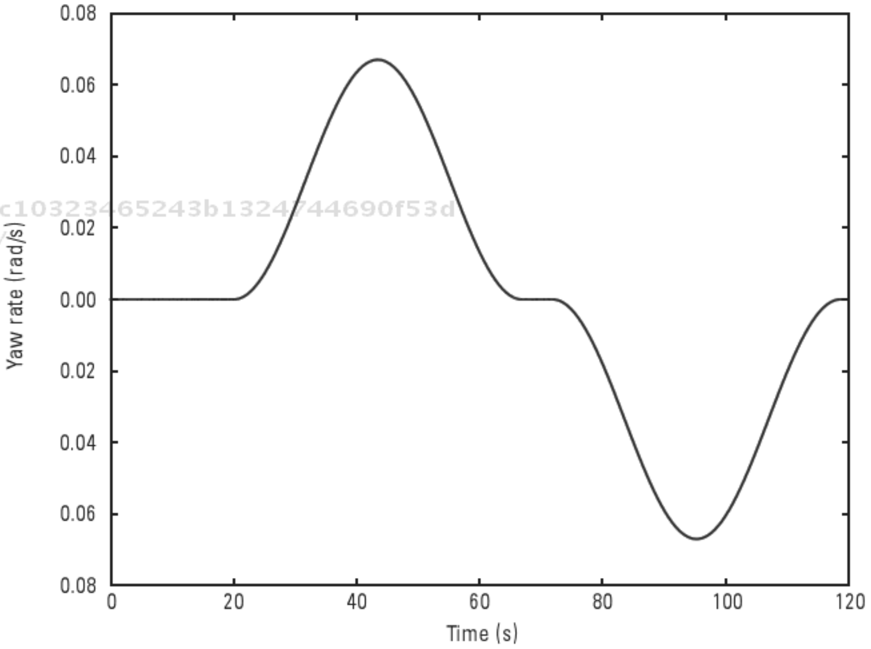


Figure 1.11 Example yaw gyro output for turn maneuver.

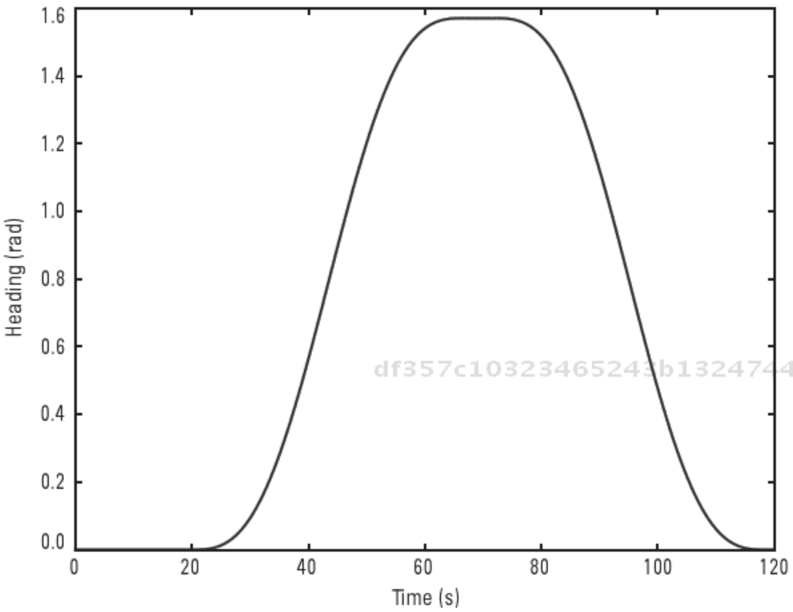


Figure 1.12 Example heading for turn maneuver.

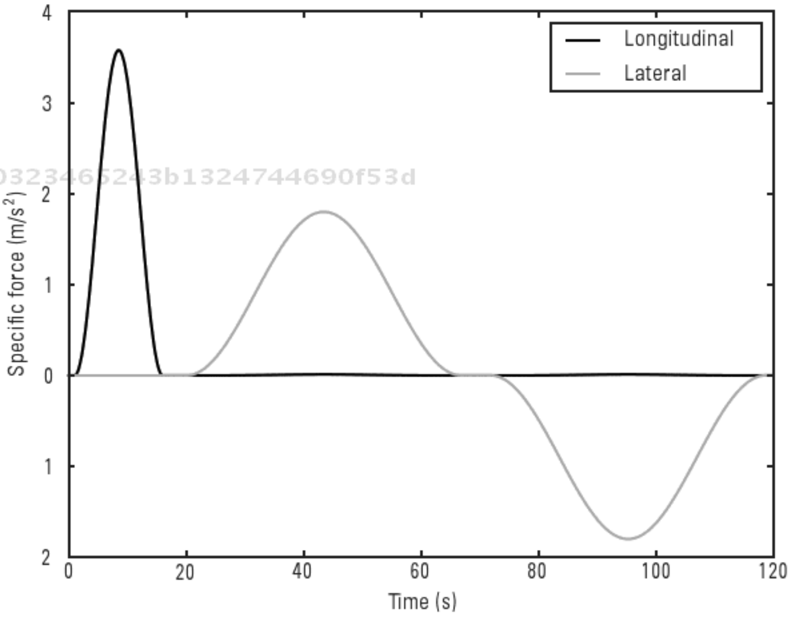


Figure 1.13 Example accelerometer measurements for turn maneuver.

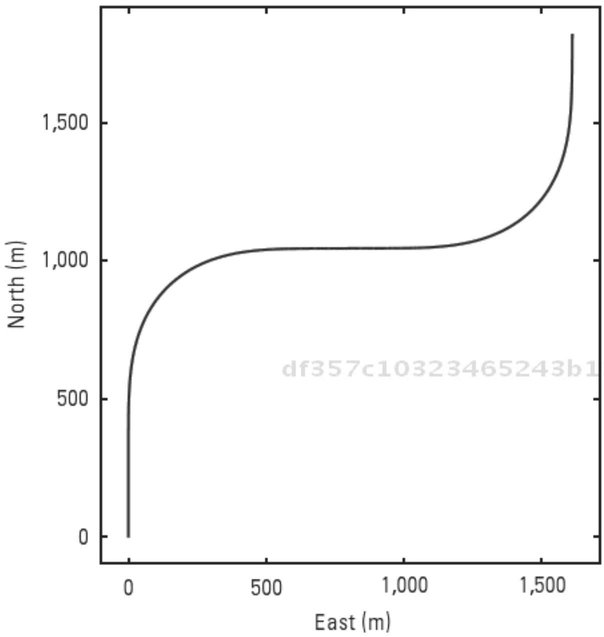


Figure 1.14 Resulting calculated position for turn maneuver.

This effect is also important to account for if any vehicle roll or pitch occurs. Since this happens frequently in vehicle operation, steps should be taken to compensate for this. A case of this error introduction is shown in Chapter 4. Due to unmodeled errors like the gravity direction, either a higher fidelity model is needed and/or inclusion of other sensors in the navigation solution. The latter method is discussed in detail in Chapter 4.

Since the Earth is rotating with respect to some inertial frame, this rotation will be included in the measured rotations of the gyroscope. Unlike the gravity effect on accelerometers, however, the effect is usually considered negligible in MEMS devices since it is indistinguishable from the gyroscope bias and noise. Use of higher accuracy gyroscopes actually allow the measurement of this rotation rate and would therefore require inclusion of this effect in the sensor model.

1.3.5 Inertial Instrument Errors

Most inertial sensors suffer from similar error sources. These error sources may be the result of different causes but can result in the same error trends.

Inertial sensor errors are typically grouped by the systematic effect they have on the resulting measurement. The errors later considered in Chapter 4 are the noise and bias errors. Other errors can be modeled and are briefly described.

1.3.5.1 Noise

The high-frequency variations of the inertial measurements are considered noise. Noise is constantly present in measurements of physical quantities due to many reasons. Unmodeled dynamic effects such as vehicle vibrations and thermal noise due to converting physical phenomenon to electric signals are examples of noise sources. Regardless of the source of the noise, its presence in the measurement means that subsequent measurements of the same motion will not produce a consistent measurement. (One exception to this arises due to quantization error, described in more detail in Section 1.4.1.) The noise on an inertial sensor is typically considered white, zero mean, and Gaussian distributed so that simplified filtering techniques can be used in the processing of these measurements. Therefore a noisy sensor measurement can be modeled as

$$s = x + \nu \tag{1.21}$$

where s is the measurement, x is the quantity measured, and ν is the noise that is modeled as a zero mean Gaussian distributed white noise with variance σ_{ν}^{2}. For an automotive grade accelerometers, this variance can be taken as 1e-2 m/s^2 Hz$^{1/2}$, and for automotive grade gyroscopes, as 2.4e-7 rad/s^2 Hz$^{1/2}$ [16]. These values must be scaled by the sampling frequency since the measurements are used in discrete time. Inclusion of these errors in the example path in Section 1.3.3 gives position errors after 2 minutes shown in Figure 1.15.

1.3.5.2 Bias

Due to inaccurate calibration or slowly drifting error processes, the reported measurement may be biased from the true value. Drifting biases can result from unmodeled effects such as temperature variation. In such cases, the bias can change slowly over time as the temperature of the sensor changes. Bias removal is important in the application of inertial sensors since the reported quantity is integrated to relate to another navigation value. This error propagation is described in greater detail in Section 1.3.6. Often at least part of the bias can be considered to come from the same error mode as the noise. This violates the assumption of zero mean noise. Since the navigation algorithm

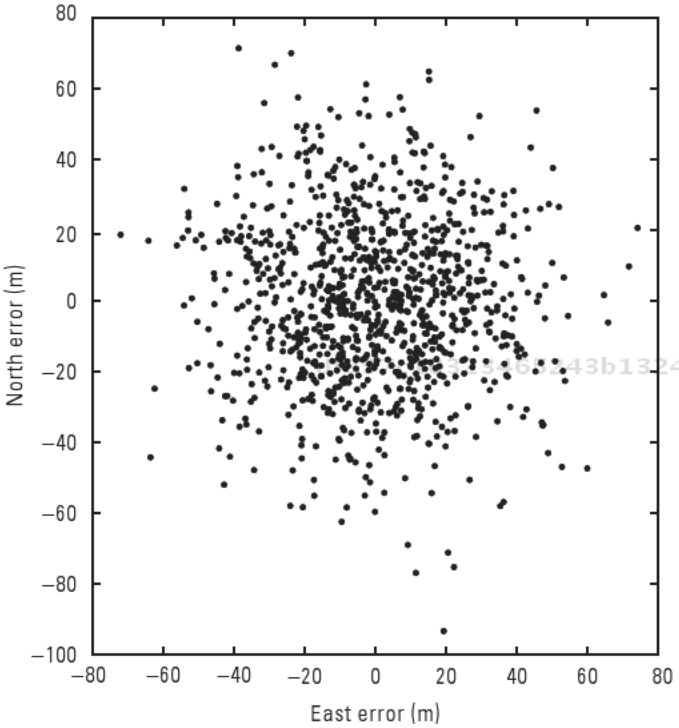


Figure 1.15 Final position error when automotive grade noise is included, sampled at 10 Hz.

requires zero mean noise, the bias is often estimated as part of the navigation state. This process is described in greater detail in Chapter 4. Accurate estimation of the bias allows the navigation system to remove the bias from the measurement and more closely approximate the Gaussian zero mean assumption. Therefore the sensor model can be extended with a bias *b*.

$$s = x + b + v \tag{1.22}$$

Another difficulty in implementing sensors that suffer with a bias is that often the bias does not remain constant over time. The bias stability is a factor that goes into selection of an IMU grade. Navigation grade IMUs have very slow bias drifts while low-cost sensors can have more significant changes in a short time period [17]. In many applications the time period over which the bias is used is short enough that this effect can be neglected. However, if the bias drift is fast enough for the application, a nonconstant model should be used in estimating the bias. In this case, the bias can be modeled as driven by its own noise as

$$\dot{b} = \mu \tag{1.23}$$

where μ is a zero mean Gaussian random variable with variance σ_{μ}^2. Alternatively, the bias can be modeled as a Markov process with a model shown here.

$$\dot{b} = \frac{-b}{\tau} + \upsilon \tag{1.24}$$

Here τ is the sensor time constant and υ is a zero mean Gaussian random variable with variance σ_{υ}^2. For an automotive grade gyroscope, values

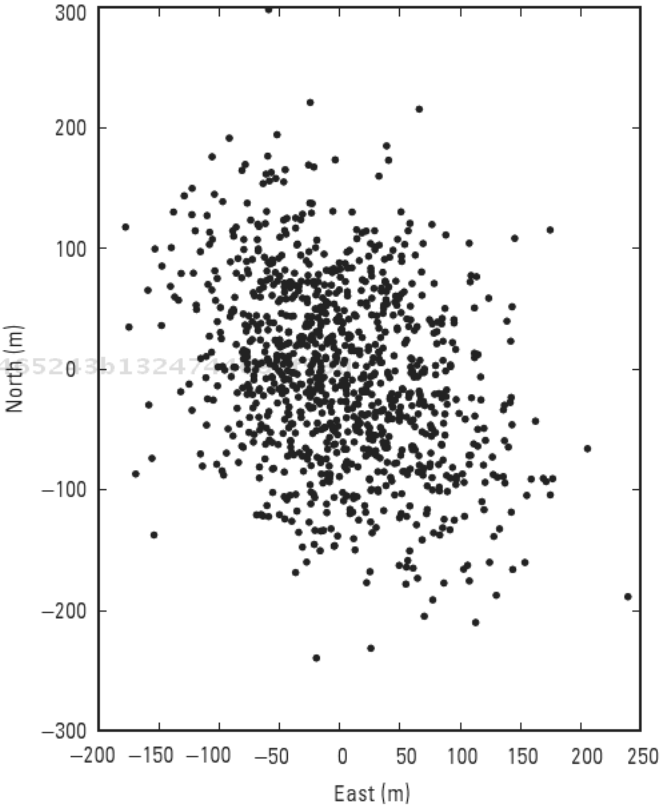


Figure 1.16 Final position error with noise and bias error included in example path.

of 8.7e-4 $^{rad}/_{s}$ and 300 *s* can be used for the noise standard deviation and time constant, respectively. For an automotive grade accelerometer, values of 1.2e-2 $^{m}/_{s^2}$ and 100 *s* can be used [16]. Addition of these errors along with noise gives final position errors for the example path in Section 1.3.3 shown in Figure 1.16. More detail of these processes is given in Chapter 4.

1.3.5.3 Integration

Two problems arise when integrating inertial sensors: the accumulating nature of integration and the unknown initialization. Any error that has previously been introduced by either initialization or integration remains in the quantity being generated by the integration. As there are errors in the integrated quantity, this error is introduced into the propagated quantity. These errors will also be found in all subsequent values of the propagated value. For example, if imperfect velocity information is available to be integrated to find position, this error will cause an apparent drift in the position that will not necessarily be corrected by subsequent velocity errors. For example, given a velocity sensor that has some measurement noise, the integrated position could look like that in Figure 1.17. In this case, there is zero velocity and

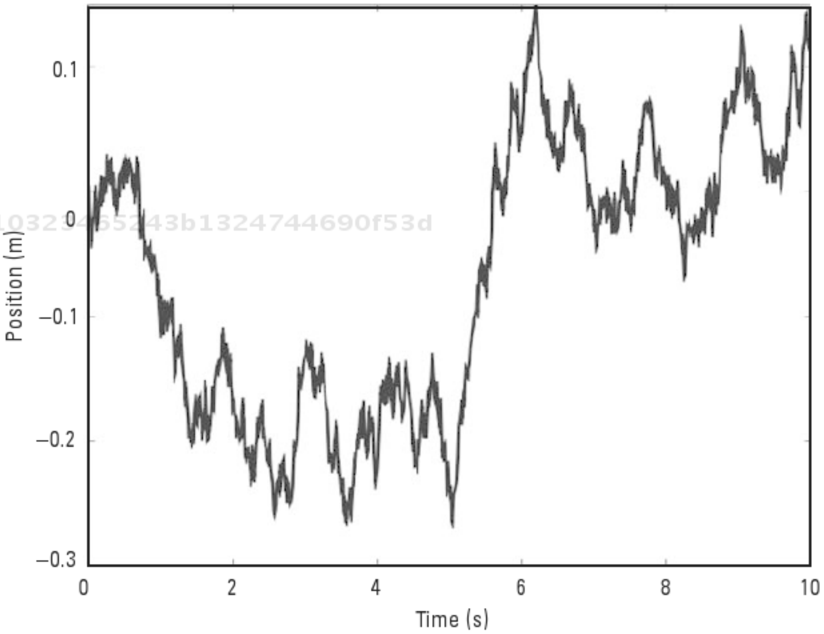


Figure 1.17 Example of accumulated position error due to numerical integration of noisy velocity measurement.

the initial position is known, therefore the resulting reported position is the position error.

Similarly, if the initial value of the propagated quantity is unknown, there is no way to determine it from the integrated quantity. This error will remain in all subsequent values of the propagated quantity.

1.3.5.4 Other Errors

While noise and bias are typically the major error contributors in the use of automotive grade inertial sensors, there are other effects that can be modeled in an inertial sensor's output. Due to calibration error, there is often an unknown (though close to unity) scale factor between the actual quantity being measured and the output given by the sensor. For small variations in sensor output, this effect could appear as a bias. Therefore, large platform dynamics are necessary to effectively observe and estimate the scale factor errors on inertial sensors [18]. Improved estimation performance is possible if the scale factor can be estimated. However, due to the difficulty in dynamically exciting the sensor, the scale factor is sometimes assumed to be unity.

Many IMU packages include sensors with orthogonal sensitive axes allowing for a single unit to measure motion in multiple directions. Errors in the orthogonality of the sensors can be estimated and then included whenever the sensor is used since this alignment will not change over time. Similarly, a mounted sensor may be misaligned from the assumed directions. These errors can be significant depending on the application. Temperature variations can also affect inertial sensor outputs but can be calibrated into the navigation system.

1.3.6 Inertial Error Propagation

Inertial sensors include deterministic and stochastic measurement errors so their unaided use as a navigation system will quickly degrade over time. As described earlier, integration of these errors causes error to grow in the other navigation states. Since the measurement noise has a stochastic part, it is impossible to predict exactly how this error will propagate. However, bounds on the growth of the error can be deterministically found based on sensor class and error type [17]. A bias in the integrated quantity will severely affect the performance of the navigation system. Therefore these biases must be calibrated out (in the case of high fidelity inertial sensors) or estimated out (using alternative sensors).

Even with deterministic errors removed, the propagation due to integration of a noisy measurement yields increasing uncertainty. Figure 1.18

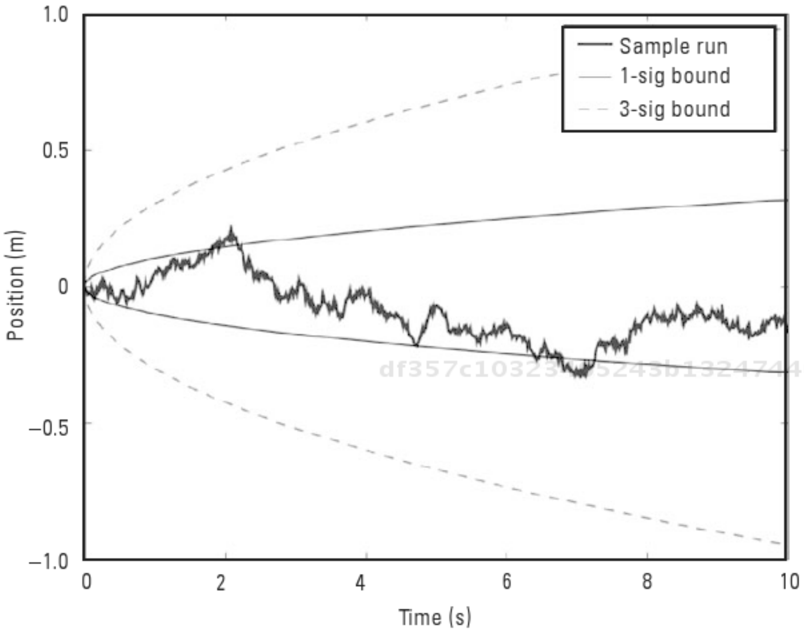


Figure 1.18 Error propagation of single-integrated white noise.

shows one such bound for single-integrated white noise. This type of error is observed when using an integrated accelerometer to calculate velocity or an integrated gyro to calculate heading. The bounds are a function of the variance of the sensor noise (σ_s^2), the sampling period (δt), and the number of integration steps (k) as

$$\sigma_i^2 = \sigma_s^2 \times \delta t \times k \tag{1.25}$$

However, this error growth does not include unremoved bias, scale factor, misalignment, and other errors. Each of these error sources would increase the growth rate of uncertainty.

Specification sheets distributed by inertial sensor manufactures give values that correspond to the accuracy of the sensor by describing how these errors are distributed [17].

1.4 Odometer Technology

Information about speed and heading change can be taken by measuring wheel speeds. Therefore, odometer technology can be used as additional sensors for vehicle applications. This is particularly useful as these sensors are not dependent on surrounding conditions other than the tire-surface contact. There are several types of odometers that may be lumped into active and passive groupings. Both of these types have advantages and disadvantages in their use. Active sensors include Hall-effect transducers that can operate well at low speeds. Passive sensors require sufficient relative motion between the sensor teeth to generate a signal [19]. For other applications, optical encoders may be more appropriate such as in robotic platforms. All of these modules use the fact that given a constant wheel radius, the distance traveled is the product of the radius and the rotation in radians. The wheel speed is also the radius times the wheel rotation rate in radians per second. Using these values in addition to GPS, the variations in GPS can be filtered and changes in the wheel radius can be monitored [19].

There are several specific applications for odometry use in vehicle or tracked vehicle control. Just as inertial sensors were used to dead reckon, wheel sensors can be used to dead reckon a vehicle's position and orientation given an initialization. Particularly when used with GPS, the wheel speeds can be used to update the vehicle velocity and also indicate a static vehicle so that sensor biases can be estimated with static assumptions. The measurements provided by odometers are taken in the axis of the wheel, which may be in a frame that is not coincident with the vehicle frame. Therefore, other effects may be required to relate wheel rotations to motion detected by other sensors.

1.4.1 Quantization

Quantization is a common error in any sensing situation in which continuous values are translated into finite-precision values. Due to the nature of the odometer, there are infinitely many physical angles that result in a single quantized measurement. This effect is most noticeable when the step size in the odometer is larger than the precision available in the discretization. Comparison of quantized output versus true quantity is shown in Figure 1.19.

1.4.2 Wheel Slip

Another error that will affect the accuracy of odometers is wheel slip. Although some wheel slip is expected in most normal operation [20], there are cases of catastrophic slip in which the motion of the wheel in contact with the ground is not acting to move the vehicle. This is the case when sliding,

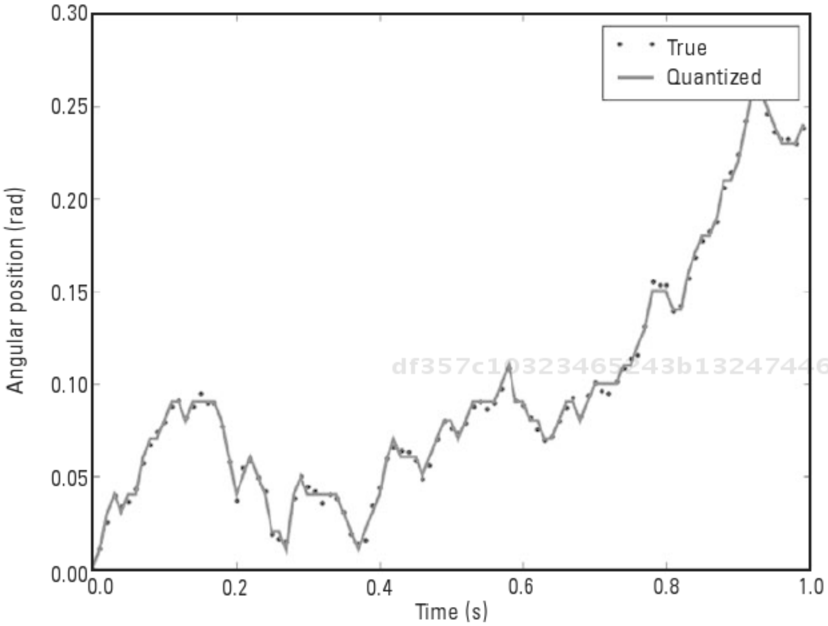


Figure 1.19 Quantized output of a continuous value.

spinout, and so forth occur on surfaces like ice and gravel. This will affect the resulting dead reckoned solution from the wheel sensors since there is more angular change than positional change. Other than these cases, some amount of slip is expected, particularly with drive wheels. Since slip is often quoted as a percentage, the amount of slip acts as a scale factor error on the odometer. This error type often changes quickly and can be a function of wheel speed and drive force. Estimation of this quantity can be accomplished with other sensors such as GPS [21].

1.4.3 Wheel Radius Error

Similar to the wheel slip error, an error in the wheel radius will act as a scale factor on the odometry measurement. Assuming this error is slowly changing, this can be estimated by the navigation filter in ways similar to the biases and drift rates of the inertial sensors. Since rapid changes to wheel radius can occur (as in a flat tire), there can be resulting large changes in the odometry error.

1.5 GNSS/Inertial Integration

Due to the complementary nature of GNSS and inertial sensors, their combination can yield improved navigation results. The result is a navigation system that builds on the strengths of each sensor type and mitigates the errors inherent in them. Typically the combination is in the form of an extended Kalman filter (EKF) although many alternative implementations have been investigated [22]. Many types of integration can be made with several combinations of sensor measurements. An example case (and perhaps the most straightforward) is a loosely coupled EKF. Here the state would include some attitude reference (i.e., heading if dealing with a planar case), velocity states, position states, and augmented states to account for sensor biases, drifts, and so forth. The inertial sensors are used to propagate the navigation portion of these states since their measurements are of the derivatives of these states. Gyroscopes are used to propagate attitude angles as shown in (1.26) and accelerometers are used to propagate velocities in (1.27), each with their respective error terms.

$$g = \dot{\psi} + b_g + \nu_g \tag{1.26}$$

$$a = \dot{v} + b_a + \nu_a \tag{1.27}$$

In these equations, (ψ) represents an attitude angle, (v) represents velocity in the accelerometer axis, (b) represents the sensor bias drift, and (ν) represents the sensor noise. Since the bias terms are included in the states, their rate-of-change models are also included in this propagation. Traditional models include simple constants or a Gauss-Markov processes to model a slowly drifting bias. Modeling the bias as a Gauss-Markov process allows the model to handle the drift of the bias. The bias can be modeled as propagating according to (1.24).

Along with the propagation of the states, the estimate of the state covariance is also propagated in this step using the models of the states (in state transition matrix form). Note that these covariances are simply the estimates of the covariances and not the true. Their proximity to the true covariances is based on several factors including model accuracy and accurate sensor variances.

This state propagation can be performed indefinitely but the resulting covariance (and true covariance) will grow without bound since no correction will be applied to the state. To remedy this, periodic measurement up-

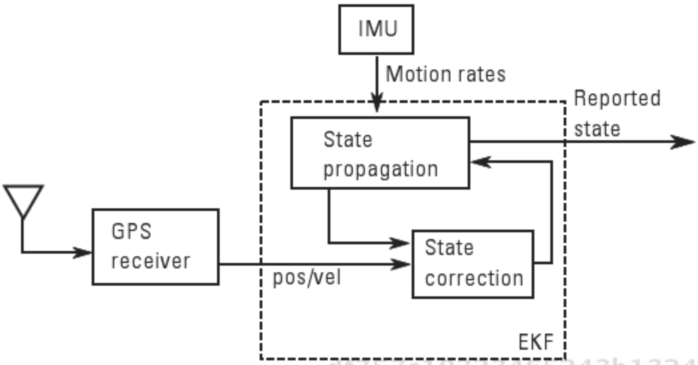


Figure 1.20 Flowchart of a loosely coupled system.

dates are made in which GPS and/or odometry sensors are used to measure state values. These sensors also have error terms inherent in their use that are typically modeled as additive noise. The propagated state is used to generate the expected measurement that is then differenced with the actual measurement taken. This difference is called the innovation, which is effectively the amount of new information made available by the measurements. The propagated state covariance is then used to generate a weighting (called the Kalman gain) to transfer the measurement innovation into state corrections. Once the states have been corrected, the state covariance is then updated to reflect this correction. The process is repeated, giving a rapidly updated navigation solution (due to the state propagation) that is corrected (by the measurement update). The general flow of the filter is shown in Figure 1.20. Subsequent chapters will show performance of this sensor combination. While there are many other GNSS/inertial combination methods, they are beyond the scope of this text and the reader is referred to other works [13, 15].

References

[1] Parkinson, B. W., et al., (eds.), *Global Positioning System: Theory and Applications, Volume I,* Reston, VA: AIAA, 1996.

[2] Parkinson, B. W., et al., (eds.), *Global Positioning System: Theory and Applications, Volume II,* Reston, VA: AIAA, 1996.

[3] Cohen, C. E., et al., "Flight Test of Attitude Determination Using GPS Compared Against an Inertial Navigation Unit," *Navigation: Journal of Institute of Navigation,* Vol. 41, No. 1, Spring 1994.

[4] Kornfeld, R. P., et al., "Single Antenna GPS Based Aircraft Attitude Determination," *Proceedings of the ION National Technical Meeting 1998*, Long Beach, CA, January 1998.

[5] Daily, R., and D. M. Bevly, "The Use of GPS for Vehicle Stability Control Systems," *IEEE Transactions on Industrial Electronics,* Vol. 51, No. 2, April 2004.

[6] *Global Positioning System Standard Positioning Service Performance Standard,* 4th ed., United States Department of Defense, September 2008.

[7] *IS-GPS-200-D: Navstar GPS Space Segment/Navigation User Interfaces*, United States Department of Defense, March 2006.

[8] *GLONASS Interface Control Document, edition 5.1*, Russian Institute of Space Device Engineering. Moscow, 2008.

[9] *GAL OS SIS ICD/D.0: Galileo Open Service Signal in Space Interface Control Document,* European Space Agency/Galileo Joint Undertaking, May 2006.

[10] *IS-QZSS: Quasi-Zenith Satellite System Navigation Service,* Japan Aerospace Exploration Agency, July 2009.

[11] Cobb, H. S., "GPS Pseudolites: Theory, Design, and Applications," Ph.D. Dissertation, Stanford University, 1997.

[12] Chatfield, A. B., *Fundamentals of High Accuracy Inertial Navigation*, Reston, VA: Progress in Astronautics and Aeronautics, Vol. 174, 1997.

[13] Groves, P. D., *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, Norwood, MA: Artech House, 2008.

[14] Ginsberg, J. H., *Advanced Engineering Dynamics,* 2nd ed., New York: Cambridge University Press, 1995.

[15] Farrell, J. A., and M. Barth, *The Global Positioning System & Inertial Navigation,* New York: McGraw-Hill, 1999.

[16] Misra, P., and P. Enge, *Global Positioning System: Signals, Measurements, and Performance,* 2nd ed., Lincoln, MA: Ganga-Jamuna Press, 2006.

[17] Wall, J., and D. M. Bevly, "Characterization of Inertial Sensor Measurements for Navigation Performance Analysis," *Proceedings of the ION GNSS Conference 2006*, Fort Worth, TX, September 26–29, 2006.

[18] Shao, Y., and D. Gebre-Egziabher, "Stochastic and Geometric Observability of Aided Inertial Navigators," *Proceedings of the ION GNSS Conference 2006*, Fort Worth, TX, September 26–29, 2006.

[19] Hay, C., "Turn, Turn, Turn: Wheel-Speed Dead Reckoning for Vehicle Navigation," *GPS World,* Vol. 16, No. 10, October 2005, pp. 37–42.

[20] Gilespie, T., *Fundamentals of Vehicle Dynamics,* Warrendale, PA: Society of Automotive Engineers, 1992.

[21] Bevly, D. M., J. C. Gerdes, and C. Wilson, "The Use of GPS Based Velocity Measurements for Measurement of Sideslip and Wheel Slip," *Vehicle System Dynamics*, Vol. 38, No. 2, August 2002, pp. 127–147.

[22] Wendel, J., et al., "A Performance Comparison of Tightly Coupled GPS/INS Navigation Systems Based on Extended and Sigma Point Kalman Filters," *Proceedings of the ION GNSS Conference 2005*, Long Beach, CA, September 13–16, 2005.

2

Vision Aided Navigation Systems

John Allen

Many vehicle manufacturers have developed lane departure warning (LDW) systems to reduce the number of traffic fatalities that occur due to unintentional lane departures. LDW systems alert the driver when the vehicle has driven outside the lane markings of the current lane of travel. Most of the LDW systems in production now are solely based off camera measurements. An LDW camera uses feature extraction to determine lateral position in the current lane. The feature used for a camera-based LDW system is the painted lane lines.

Lane positioning can also be accomplished using a light detection and ranging (Lidar) scanner. Unlike the camera, a Lidar scanner can provide three-dimensional ranging information; however, Lidar does not provide color information. To overcome this, some Lidar scanners also provide reflectivity data. The reflectivity data can be used to extract lane markings; and the ranging information can be used to provide an estimate of a vehicle's lateral position in the lane.

The objective of this chapter is to provide a method of using a lateral lane position measurement to aid a traditional navigation system [1–5]. Two types of vision integration are discussed in this chapter. The first is integration of vision to aid a position, speed, and heading navigation filter. The second is integration of vision to aid a closely coupled navigation filter. Both of these filter architectures are discussed in detail in Chapter 1.

This chapter will start with a brief overview of how the camera and Lidar are used to measure a vehicles lateral position in a lane. Next, the necessary coordinate frame rotations are discussed. Two types of rotations are covered. The first rotation is a two-dimensional rotation that is necessary for integration with the position, speed, and heading navigation filter. The second rotation is a three-dimensional rotation that is necessary for integration with the closely coupled navigation filter. The next section will discuss vision integration with the position, speed, and heading navigation filter. The final section of the chapter will discuss vision integration with the closely coupled navigation filter.

2.1 Lane Positioning Methods

Two methods of lane positioning are presented in this section. The first is using a Lidar for lane positioning. Currently, Lidar scanners are very expensive. The cost of Lidar scanners has prevented implementation of Lidar-based LDW systems on civilian vehicles; however, the cost of Lidar scanners is expected to decrease as demand for these devices increases. The second method of lane positioning involves using a camera. Ideally, both cameras and Lidar scanners will be used in future LDW systems. Each device has different strengths and weaknesses. Using both devices will improve the robustness of both the LDW system and navigation system aiding.

2.1.1 Lidar-Based Positioning

Lidar measures the range to an object by pulsing a light wave at the object and measuring the time between transmission and reception. The light wave for Lidar applications is a laser. Lidar is very similar to sonar, but instead of using a sound wave, Lidar uses a light wave. Lidar scanners combine the laser with a moving mirror that rotates the laser's beam. This can provide ranging information in multiple directions, both vertically and horizontally. Lidar can provide reflectivity measurements, a measurement known as echo width. Reflectivity data can be used to classify objects that the Lidar encounters.

A Lidar scanner with reflectivity measurements can be used to search for and range off of lane markings. Painted lane markings are more reflective than the asphalt that surrounds the lane marking, therefore the echo width data from a Lidar scanner can be used to find lane markings, and the ranging information from the Lidar can then be used to determine the vehicles lateral

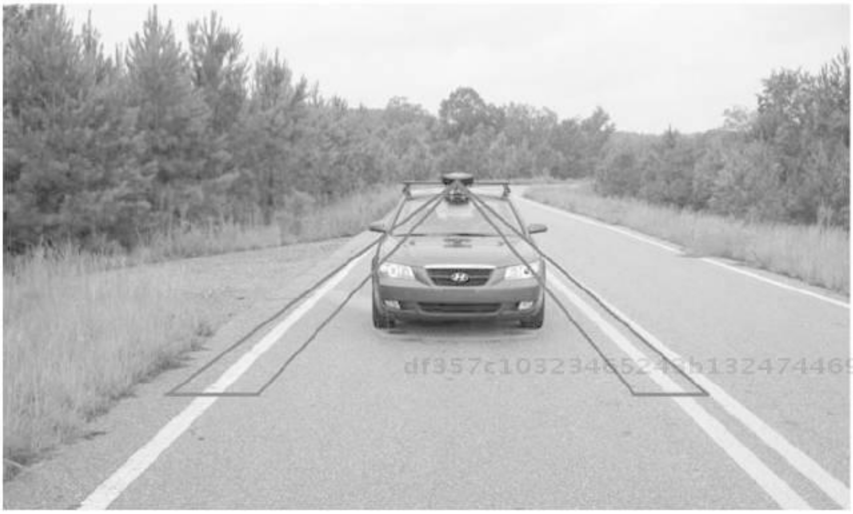


Figure 2.1 Visual representation of a Lidar scanner searching for lane markings.

distance from the lane marking. Figure 2.1 shows a Lidar scanner mounted atop a vehicle and a representation of its scan pattern. Figure 2.2 shows a sample reflectivity data from a Lidar scanner, and Figure 2.3 shows a picture taken from a camera aligned with the Lidar scanner.

The distinct spikes in the echo width are a result of the laser hitting a lane marking. At the start and end of the laser scan, when the laser is not contacting the road, the echo width measurements become noisy. This will result in spikes in the echo width that do not correspond to a lane marking. Therefore, it is necessary to use search bounds on the echo width measurements.

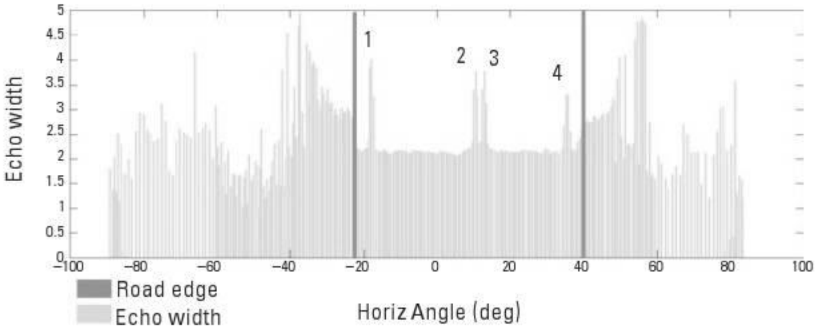


Figure 2.2 Average echo width for 100 scans. (*From:* [1].)

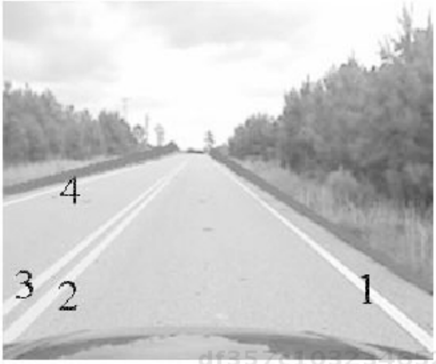


Figure 2.3 Corresponding echo width to lane markings. (*From:* [1].)

Testing has shown that the standard deviation of a Lidar-based lane position measurement under ideal circumstances is around 0.044 meter [1].

One advantage of using a Lidar scanner–based LDW system is the robustness of Lidar scanners to varying lighting and weather conditions. Unlike a camera, Lidar scanners work independent of surrounding lighting conditions. Currently, the largest disadvantage of Lidar-based LDW systems is the cost of the hardware. Lidar is a relatively new technology with a limited number of manufacturers. Also, using Lidar scanners to detect lane markings is a new and undeveloped science.

2.1.2 Camera-Based Positioning

Cameras are the most popular type of hardware used to determine lane position for LDW systems. Camera-based LDW systems are available as an option on some production vehicles. Many methods of camera-based lane detection are present in the literature; the following is one method. The first step of camera-based lane positioning is lane marking extraction. Thresholding is the process of filtering out unwanted features in the image and is used to extract desired areas of the image such as lane markings. Then, edge detection and the Hough transform are used to extract areas of the image that have lines or edges, as in Figure 2.4. Other camera-based lane detection systems employ optical flow, neural networks, and alternatives to the Hough transform such as peak and edge finding. The vehicles position in the lane can be estimated after the image is searched for lane markings. One simple method of estimating lateral lane position is counting the number of pixels

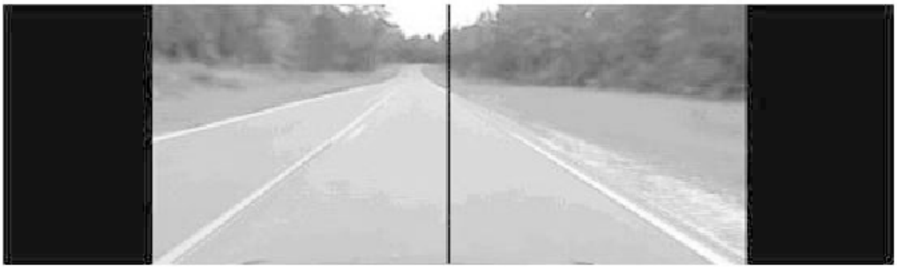


Figure 2.4 Lane lines extracted from the Hough transform. (*From:* [4].)

from the center of the image to the lane marking at the bottom of the image. The number of pixels is multiplied by a scale factor to determine the distance from the center of the vehicle to the lane line. Testing has shown that standard deviation of a camera-based lane position measurement under ideal circumstances is around 0.059 meter [2].

One advantage of using a camera-based LDW system is the cost of the hardware involved. Digital cameras have been in production for decades, and the cost of these devices is relatively cheap. Also, methods of lane positioning using a camera were pioneered in the 1990s; therefore, the algorithms used for lane positioning using a camera are well established. Some disadvantages to camera-based LDW systems include vulnerability to lighting and weather conditions. At dawn and dusk, when the sun is low in the sky, a camera may be blinded by the sun. Also, camera-based lane detection can be difficult in urban environments where lane markings are in poor condition or visibility of lane markings are blocked by surrounding traffic.

2.2 Coordinate Frame Rotation and Translation

Lane position measurements are given in the road-based coordinate frame. The road-based coordinate frame can be approximated with a waypoint-based map. A rotation matrix is needed to rotate coordinates in the navigation's coordinate frame to the road coordinate frame because the navigation filter's coordinate frame is not oriented the same way as the road coordinate frame. Since the navigation coordinate frame and the road coordinate frame do not have the same origin, coordinate frame translation must also be covered.

A rotation matrix is a matrix that if multiplied by a vector of values expressed in an initial coordinate frame will result in a vector of values expressed in a new coordinate frame. The new coordinate frame has some different attitude from the initial coordinate frame. The difference in attitude will govern the values in the rotation matrix. A rotation matrix can be constructed using Euler angles. The rotational direction of Euler angles are based off the right-handed coordinate system. In short, rotation about an axis that points towards the observer results in a counterclockwise-positive rotation. Rotation about an axis that points away from the observer results in a clockwise-positive rotation. Rotation matrices are orthogonal; therefore, the inverse of a rotation matrix is equal to its transpose. Equation (2.1) describes this principle.

$$C_{\alpha}^{\beta\ 1} = C_{\alpha}^{\beta^T} = C_{\beta}^{\alpha} \tag{2.1}$$

Rotation matrices are denoted by C_{α}^{β}. This rotation matrix maps coordinates in the α coordinate frame to the β coordinate frame. e denotes the ECEF coordinate frame, n denotes the North East coordinate frame, and r denotes the road coordinate frame.

2.2.1 Two-Dimensional Rotations

The position, speed, and heading navigation filter employs a two-dimensional navigation coordinate frame. Two-dimensional coordinate frame rotation matrices are 2 2 matrices based off one attitude angle. Figure 2.5 shows the principle of a two-dimensional rotation.

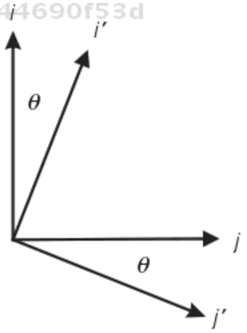


Figure 2.5 A rotation matrix (2.2) can be constructed to map vectors in the initial coordinate frame (*i*,*j*) to a rotated coordinate frame (*i*′,*j*′). Theta is the angle of rotation. This angle is measured from the *i* axis to the *i*′ axis. The right-hand coordinate frame rule governs which direction of rotation is positive. The angle theta in this picture represents a positive rotation.

The (i,j) coordinate frame is the starting coordinate frame, and the (i',j') coordinate frame is the final coordinate frame. For the coordinate frame in Figure 2.5, the right-handed coordinate frame rule defines the axis of rotation as pointing down (or away from the observer); therefore, clockwise rotation results in a positive Euler angle. Equation (2.2) shows the rotation matrix that will map coordinates in the (i,j) coordinate frame to the (i',j') coordinate frame. The rotation matrix used to aid the position, speed, and heading navigation filter will use a rotation matrix of the same format.

$$C_{(i,\,j)}^{(i\ ,\ j\ )} = \begin{matrix} \cos(\theta) & \sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{matrix} \tag{2.2}$$

2.2.2 Three-Dimensional Rotations

Three-dimensional coordinate frame rotation matrices are 3   3 matrices based off three Euler angles. A three-dimensional coordinate frame rotation can be thought of as a series of three two-dimensional rotations. Figure 2.6 shows the rotation series used in this book.

The first rotation is about the z-(k) axis. The second rotation is about the new y-(j') axis. The third rotation is about the new x (i'') axis. Equation (2.3) shows how the three-dimensional rotation matrix is constructed. s_1 is the sine of θ_1 and c_1 is the cosine of θ_1, s_2 is the sine of θ_2, and c_2 is the cosine of θ_2, s_3 is the sine of θ_3, and c_3 is the cosine of θ_3.

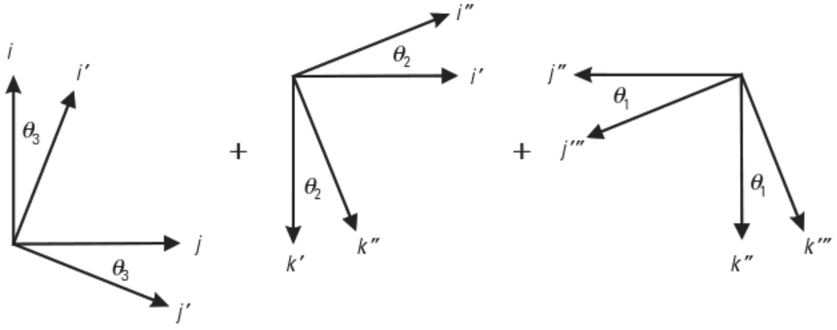


Figure 2.6 The first rotation is about the k axis. This will result in a new coordinate frame (i',j',k'). The second rotation is about the j' axis. This will result in a new coordinate frame (i'',j'',k''). The third rotation is about the i'' axis. This will result in the final rotated coordinate frame.

$$C_{(i,j)}^{(i\ \ ,j\ \ )} = \begin{matrix} 1 & 0 & 0 \\ 0 & c_1 & s_1 \\ 0 & s_1 & c_1 \end{matrix} \quad \begin{matrix} c_2 & 0 & s_2 \\ 0 & 1 & 0 \\ s_2 & 0 & c_2 \end{matrix} \quad \begin{matrix} c_3 & s_3 & 0 \\ s_3 & c_3 & 0 \\ 0 & 0 & 1 \end{matrix}$$

$$= \begin{matrix} c_2c_3 & c_2s_3 & s_2 \\ s_1s_2c_3 \quad c_1s_3 & s_1s_2s_3 \quad c_1c_3 & s_1c_2 \\ c_1s_2c_3 + s_1s_3 & c_1s_2s_3 \quad s_1c_3 & c_1c_2 \end{matrix} \tag{2.3}$$

2.2.3 Coordinate Frame Translation

The origin of the road-based coordinate frame is not located at the same point in space as the origin of the navigation coordinate frame origin. When mapping coordinates in the navigation coordinate frame to the road coordinate frame, the coordinate frame must be moved and rotated. Figure 2.7 shows an example of a coordinate frame translation and rotation.

Equation (2.4) shows how to map coordinates in coordinate frame that has been moved and rotated $\vec{r}_{\alpha p}^{\alpha}$ is the position vector expressed in the initial coordinate frame (α), and $\vec{r}_{\beta p}^{\beta}$ is the position vector expressed in the rotated

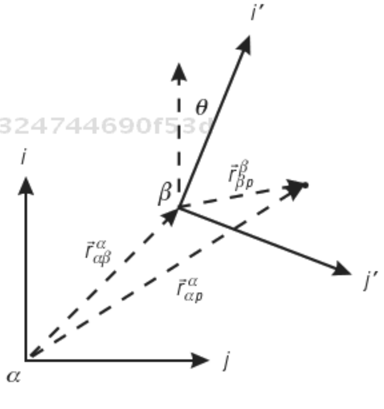

i'
θ
i
$\vec{r}_{\beta p}^{\beta}$
β
$\vec{r}_{\alpha\beta}^{\alpha}$
j'
$\vec{r}_{\alpha p}^{\alpha}$
j
α

Figure 2.7 An example of a two-dimensional coordinate frame translation and rotation. In order to map a position vector in the alpha coordinate frame to the beta coordinate frame, two things must be known. The first is the position vector from the origin of the alpha coordinate frame to the origin of the beta coordinate frame expressed in the alpha coordinate frame. The second is the change in attitude from the alpha coordinate frame to the beta coordinate frame.

and translated coordinate frame (β). $\vec{r}^{a}_{ab}$ is a vector expressed in the initial coordinate that points from the initial coordinate frame to the new coordinate frame, and C^{β}_{α} is the rotation matrix from coordinate frame α to coordinate frame β.

$$\vec{r}^{\beta}_{\beta p} = C^{\beta}_{\alpha} \quad \vec{r}^{\alpha}_{\alpha p} \quad \vec{r}^{\alpha}_{\alpha\beta} \tag{2.4}$$

Coordinates in the final coordinate frame can be mapped back to the initial coordinate frame using (2.5).

$$\vec{r}^{\alpha}_{\alpha p} = C^{\alpha}_{\beta}\vec{r}^{\beta}_{\beta p} + \vec{r}^{\alpha}_{\alpha\beta} \tag{2.5}$$

2.2.4 Global Coordinate Frame Rotations

Sometimes it may be necessary to map coordinates in the Earth Centered Earth Fixed (ECEF) coordinate frame to the North East Down (NED) coordinate frame. The north-east plane of the NED coordinate frame is tangential to a reference ellipsoid. The reference ellipsoid is an ellipsoid that mimics the surface of the Earth. Using the latitude (ϕ) and longitude (λ) of the origin of the NED coordinate frame, a rotation matrix that maps ECEF coordinates to NED coordinates can be constructed. Equation (2.6) shows the rotation matrix that maps coordinates in the ECEF coordinate frame to the NED coordinate frame. The transpose of this matrix will map coordinates from the NED coordinate frame to the ECEF coordinate frame.

$$C^{n}_{e} = \begin{matrix} \sin(\phi)\cos(\lambda) & \sin(\phi)\sin(\lambda) & \cos(\phi) \\ \sin(\lambda) & \cos(\lambda) & 0 \\ \cos(\phi)\cos(\lambda) & \cos(\phi)\sin(\lambda) & \sin(\phi) \end{matrix} \tag{2.6}$$

Equation (2.7) coverts ECEF position coordinates ($\vec{r}^{e}_{ep}$) to NED position coordinates ($\vec{r}^{n}_{np}$). Notice that, along with the latitude and longitude of the origin of the NED coordinate frame, the position of the origin expressed in the ECEF coordinate frame ($\vec{r}^{e}_{en}$) must be known.

$$\vec{r}^{n}_{np} = C^{n}_{e} \quad \vec{r}^{e}_{ep} \quad \vec{r}^{e}_{en} \tag{2.7}$$

Equation (2.8) coverts NED position coordinates to ECEF position coordinates.

$$\vec{r}_{ep}^{\,e} = C_e^{n^T}\,\vec{r}_{np}^{\,n} + \vec{r}_{en}^{\,e} \tag{2.8}$$

Equation (2.9) coverts ECEF velocity coordinates to NED velocity coordinates. Notice that velocity mapping is independent of the position of the NED coordinate frame. The latitude and longitude of the NED coordinate frame is still need to construct the rotation matrix.

$$\vec{v}_n = C_e^n \vec{v}_e \tag{2.9}$$

Equation (2.10) coverts NED velocity coordinates to ECEF velocity coordinates.

$$\vec{v}_e = C_e^{n^T}\,\vec{v}_n \tag{2.10}$$

2.3 Waypoint-Based Maps

The road coordinate frame is the coordinate frame in which the vision measurements are given. The road frame can be approximated using a waypoint map. The waypoints lie in the center of the lane that is being mapped. The distance between waypoints should be defined by the road geometry. Complex road geometry will require waypoints to be close together. For example, in a turn, the waypoints will need to be close together to match the geometry of the road. On a straightaway, the waypoints can be very spread out due to the lack of change in road geometry.

The position states of the navigation filter must be mapped into the road coordinate frame. Equation (2.4) can be used to map position from the navigation coordinate frame to the road coordinate frame. In order to use this equation, the position of the base waypoint in the navigation coordinate frame must be known. Also, the rotation matrix from the navigation coordinate frame to the road coordinate frame must be known. In order to construct this rotation matrix, the attitude of the road coordinate frame with respect to the navigation coordinate frame must be known. These elements can be saved in a map database.

When the navigation filter is initialized, it must check to see where on the map the vehicle is located. GPS can be used to initialize the vehicle's position in the navigation coordinate frame. This position can be used to search the map database.

The distance from the base waypoint to the next waypoint can also be stored in the map database. This value is useful for checking if the vehicle has passed the next waypoint.

2.4 Aiding Position, Speed, and Heading Navigation Filter with Vision Measurements

The position, speed, and heading navigation system is a three-degree-of-freedom navigation filter based in a two-dimensional, rectangular coordinate frame. The navigation coordinate frame for this filter is the north-east coordinate frame. The north-east coordinate frame is denoted by n. The x-axis of the navigation coordinate frame points north and the y-axis of the coordinate frame points east. The road coordinate frame for the position, speed, and heading navigation system is also a two-dimensional coordinate frame. The road coordinate frame is denoted by r. The x-axis of the road coordinate frame points from the last waypoint passed to the next waypoint. The y-axis of the road coordinate frame is perpendicular to the x-axis. If facing with the x-axis, the y-axis points to the left.

Equation (2.11) shows an example map database that can be used to aid the position, speed, and heading navigation system. $N_{r,i}$ and $E_{r,i}$ is the position coordinates of the ith waypoint. $\psi_{r,i}$ is the heading of the ith road coordinate frame. The heading of the road coordinate frame is measured from the north axis with clockwise rotation being positive. $d_{r,i}$ is the distance from waypoint i to waypoint i + 1. This value is used to check if the vehicle has passed the next waypoint. Figure 2.8 depicts a portion of this database graphically.

$$Map\ Database = \begin{matrix} N_{r,1} & E_{r,1} & \psi_{r,1} & d_{r,1} \\ \vdots & \vdots & \vdots & \vdots \\ N_{r,m} & E_{r,m} & \psi_{r,m} & d_{r,m} \end{matrix} \qquad (2.11)$$

The rotation matrix in (2.12) represents the rotation matrix that maps coordinates in the north-east (navigation) coordinate frame to the road coordinate frame. The angle $\psi_{r,i}$ is the heading of the ith road coordinate frame. This angle should not be confused with the heading state of the filter.

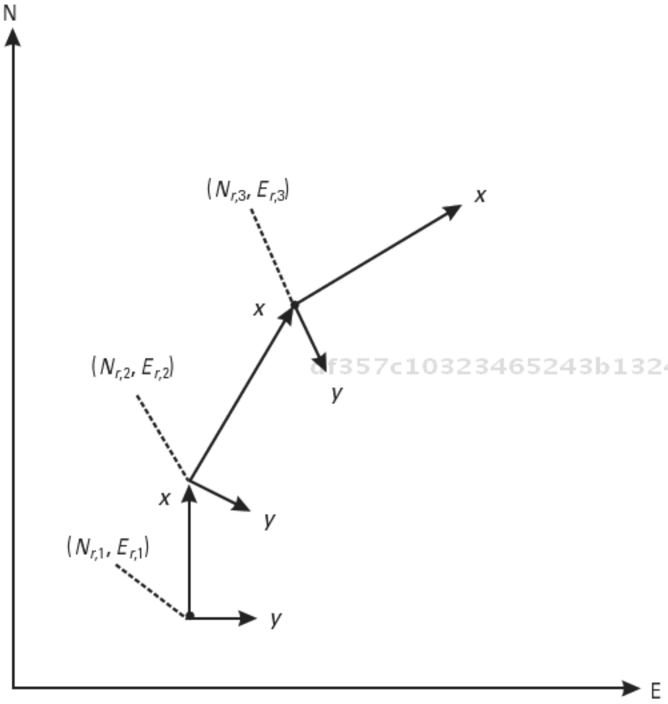


Figure 2.8 Visual example of a two-dimensional waypoint map.

$$C_n^r = \frac{\cos(\psi_r)\sin(\psi_r)}{\sin(\psi_r)\cos(\psi_r)} \tag{2.12}$$

2.4.1 Two-Dimensional Map Construction

In order to aid the position, speed, and heading navigation filter with lane position measurements, the position of the waypoints ($N_{r,i}$, $E_{r,i}$) and the heading ($\psi_{r,i}$) and distance ($d_{r,i}$) between waypoints must be known. The positions of the waypoints ($N_{r,i}$, $E_{r,i}$) are assumed to be given through a survey. The heading between waypoints can be solved with (2.13).

$$\psi_{r,i} = a\tan 2\left(\left[E_{r,i+1} \quad E_{R,i}\right], \left[N_{r,i+1} \quad N_{r,i}\right]\right) \tag{2.13}$$

The distance between the waypoints can be determined using the distance equation.

$$d_{r,i} = \sqrt{(E_{r,i+1} \quad E_{r,i})^2 + (N_{r,i+1} \quad N_{r,i})^2} \tag{2.14}$$

2.4.2 Measurement Structure

In order to use lateral lane position measurements, the lateral lane position with respect to the lane map must be estimated using the current states of the navigation filter. $\hat{N}$ and $\hat{E}$ denote the current position estimate in the north east coordinate frame. N_R and E_R denote the position of the last waypoint passed in the north-east coordinate frame. $\hat{x}$ and $\hat{y}$ denote the estimated position in the road coordinate frame. Equation (2.15) is used to find the position estimates in the road coordinate frame.

$$\begin{matrix}\hat{x} \\ \hat{y}\end{matrix} = C_n^r \begin{matrix}\hat{N} & N_r \\ \hat{E} & E_r\end{matrix} \tag{2.15}$$

Equation (2.16) shows the measurement equation. The measurement equation is a function of the states of the filter and the map parameters. The matrix in (2.17) shows the measurement model, assuming the state matrix takes the form of (4.85) in Chapter 4. The measurement model is created by taking the partial derivative of the measurement equation with respect to each state. $\psi_{r,i}$ is the heading of the current (ith) road coordinate frame, not the heading state of the filter.

$$h(x) = \hat{y} = \quad (\hat{N} \quad N_{r,i})\sin(\psi_{r,i}) + (\hat{E} \quad E_{r,i})\cos(\psi_{r,i}) \tag{2.16}$$

$$H(x) = \begin{bmatrix}0 & 0 & 0 & 0 & & \sin(\psi_{r,i}) & \cos(\psi_{r,i})\end{bmatrix} \tag{2.17}$$

2.4.3 Checking Waypoint Map Position

After every measurement update and time update, the states of the navigation filter will change. Whenever the states of the filter are updated, the distance into the current road frame must be checked to ensure that the vehicle has not passed into the next road coordinate frame. Substituting the position states into (2.18) will result in the current longitudinal distance into the road coordinate frame. If this value is larger than $d_{r,i}$, then the vehicle has passed into the next coordinate frame. The map index i should then be incremented by one.

$$\hat{x} = (N \quad N_{r,i})\cos(\psi_{r,i}) + (E \quad E_{r,i})\sin(\psi_{r,i}) \tag{2.18}$$

2.4.4 Results

Figure 2.9 shows the utility of using vision to aid a GPS-based navigation filter. Figure 2.9 is a plot of the lateral lane position estimates. This data for these results was collected at the NCAT test track, which is a 1.8-mile, two-lane oval track. The track is set up very similar to the interstate system. The thin line represents the GPS measurements. GPS position is a bias measurement. These biases make it difficult to compare a GPS position measurement and a map to get vehicle position within a lane. The thick line is the result of the vision aided navigation filter. Unlike the GPS measurement, the estimated lane position reported by the vision aided navigation filter is not biased.

2.5 Aiding Closely Coupled Navigation Filter with Vision Measurements

The closely coupled navigation filter is a six-degree-of-freedom navigation filter. The filter tracks the global position and velocity of a vehicle. The position and velocity estimates are based in the Earth centered earth fixed (ECEF) coor-

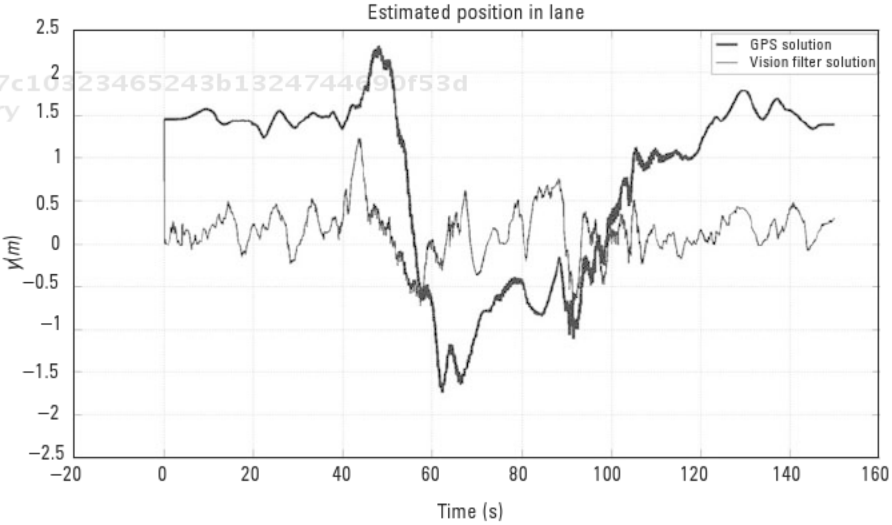


Figure 2.9 Estimated lane position reported by GPS and the vision aided navigation filter.

dinate frame. The ECEF coordinate frame is a three-degree-of-freedom coordinate frame and is denoted by *e*. The closely coupled navigation filter also tracks the attitude of the vehicle. The attitude is represented by the three Euler angles. Along with the pose of the vehicle, the navigation filter tracks accelerometer biases, gyro biases, GPS clock bias, and GPS clock drift. A more in-depth look at the structure of closely coupled navigation filter can be found in Chapter 3.

The goal of this section is to describe how a lateral lane position measurement can be incorporated into the closely coupled navigation filter. Lateral lane measurements from a camera or Lidar can aid the lane navigation solution and provide robust positioning even under limited GPS satellite visibility such as in urban canyons [5]. Similar to the position speed and heading filter, a road map is necessary to include lane position measurements. The road coordinate frame for the closely coupled navigation filter is similar to the road coordinate frame for the position, velocity, heading filter. The road coordinate frame for the closely coupled navigation filter is a three-degree-of-freedom coordinate frame and is denoted by *r*. The *x*-axis of the road coordinate frame points from the last waypoint passed to the next waypoint. The *y*-axis of the road coordinate frame is perpendicular to the *x*-axis. If facing the *x*-axis, the *y*-axis points to the left. The road coordinate frame is assumed to have no bank; therefore, the *y*-axis is always parallel with the plane tangent to Earth's reference ellipsoid. The *z*-axis is perpendicular to the *x*-*y* plane and points down.

Equation (2.19) represents a example map database that can be used. $\vec{r}^{\,e}_{er,i}$ is the position vector of waypoint *i* in the ECEF coordinate frame. $\vec{\varphi}_i$ is the attitude vector of the road coordinate frame. The attitude is represented the same way as the attitude states of the vehicle, with three Euler angles. $d_{r,i}$ is the distance from waypoint *i* to waypoint *i* + 1. This value is used to check if the vehicle has passed the next waypoint. Figure 2.10 depicts this database visually.

$$\textit{Map Database} = \begin{matrix} \vec{r}^{\,e}_{er,1} & \vec{\varphi}_1 & d_{r,1} \\ \vdots & \vdots & \vdots \\ \vec{r}^{\,e}_{er,m} & \vec{\varphi}_m & d_{r,m} \end{matrix} \tag{2.19}$$

The form of the rotation matrix from the ECEF coordinate frame to the road coordinate frame is given in (2.20). The elements from the road coordinate frame attitude ($\vec{\varphi}_i$) are used to construct the rotation matrix. c_1 is the cosine of the first attitude angle in the attitude vector, and s_1 is the sine of the first attitude angel in the attitude vector. Similarly, c_2 and s_2 are the trigonometric functions of the second angle, and c_3 and s_3 are the trigionmec functions of the third angle.

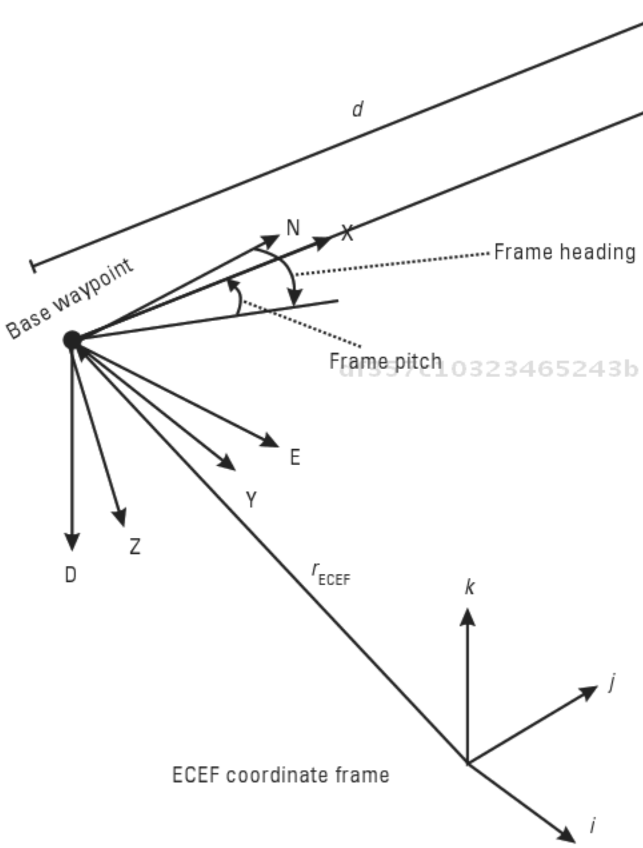


Figure 2.10 Visual example of the three-dimensional road coordinate frame.

$$C_e^r = \begin{matrix} c_2c_3 & c_2s_3 & s_2 \\ s_1s_2c_3 \quad c_1s_3 & s_1s_2s_3 + c_1c_3 & s_1c_2 \\ c_1s_2s_3 + s_1s_3 & c_1s_2s_3 \quad s_1c_3 & c_1c_2 \end{matrix} \tag{2.20}$$

2.5.1 Three-Dimensional Map Construction

The rotation matrix in (2.20) can be thought of as a sequence of two rotations. The first rotation is from the ECEF coordinate frame to the north, east, down (NED) coordinate frame (C_n^r). This rotation matrix is based off two angles. ϕ_i is the latitude of the *i*th coordinate frame, λ_i, and is the latitude of the *i*th coordinate frame. The origin of the NED coordinate frame is at the *i*th waypoint ($\vec{r}^{e}_{er,i}$). The second rotation is from the NED coordinate frame to the road coordinate frame (C^{r}_{n}). This rotation matrix is also based off two angles. ψ_i is the heading of the *i*th coordinate frame. θ_i is the pitch of the *i*th coordinate frame.

$$C^{r}_{e} = C^{r}_{n} C^{n}_{e} \tag{2.21}$$

Equation (2.22) shows the form of the rotation matrix that maps coordinates in the ECEF coordinate frame to the NED coordinate frame. The longitude and latitude of the *i*th road coordinate frame correspond to the global position of the *i*th waypoint. The latitude and longitude can either be surveyed, or solved for using the position of the *i*th waypoint. Transformations between ECEF coordinates and geodetic coordinates can be found in [2].

$$C^{n}_{e} = \begin{matrix} \sin(\phi_i)\cos(\lambda_i) & \sin(\phi_i)\sin(\lambda_i) & \cos(\phi_i) \\ \sin(\lambda_i) & \cos(\lambda_i) & 0 \\ \cos(\phi_i)\cos(\lambda_i) & \cos(\phi_i)\sin(\lambda_i) & \sin(\phi_i) \end{matrix} \tag{2.22}$$

$$C^{r}_{n} = \begin{matrix} \cos(\theta_i)\cos(\psi_i) & \cos(\theta_i)\sin(\psi_i) & \sin(\theta_i) \\ \sin(\psi_i) & \cos(\psi_i) & 0 \\ \sin(\theta_i)\cos(\psi_i) & \sin(\theta_i)\sin(\psi_i) & \cos(\theta_i) \end{matrix} \tag{2.23}$$

Equation (2.23) shows the form of the rotation matrix that maps coordinates in the NED coordinate frame to the road coordinate frame. The pitch and heading angles can be solved by looking at the change in position between waypoint i and waypoint $i + 1$. The first step in solving for these angles is to express the position of the $i + 1$ waypoint in the NED coordinate frame based at the *i*th waypoint. This can be done using (2.24).

$$\begin{matrix} N_{r,i} \\ E_{r,i} \\ D_{r,i} \end{matrix} = C^{n}_{e} \quad \vec{r}^{e}_{er,i+1} \quad \vec{r}^{e}_{er,i} \tag{2.24}$$

Once the position of waypoint $i + 1$ is expressed in the NED coordinate frame based at coordinate frame I, (2.25) and (2.26) can be used to solve for the road coordinate frame heading and pitch.

$$\psi_i = atan2(E_{r,i}, N_{r,i}) \tag{2.25}$$

$$\theta_i = atan \quad \frac{D_{r,i}}{\sqrt{N_{r,i}^2 + E_{r,i}^2}} \begin{matrix}\div\\\div\end{matrix} \tag{2.26}$$

The longitude (λ_i), latitude (ϕ_i), pitch (θ_i), and heading (ψ_i) are all plugged into their corresponding rotation matrices. The result of (2.21) will be a rotation matrix that maps coordinates in the ECEF coordinate frame to the road coordinate frame. The three attitude angles in the map database ($\vec{\varphi}_i$) can be extracted from this rotation matrix. Equations (2.27) through (2.29) show how to solve for the road coordinate frame attitude using the longitude, latitude, pitch, and heading angles. The relationship between attitude angles and rotation matrices is derived in [3].

$$\vec{\varphi}_{i,1} = atan2(\quad \cos(\phi_i)\sin(\psi_i), \cos(\phi_i)\sin(\theta_i)\cos(\psi_i) \quad \sin(\phi_i)\cos(\theta_i)) \tag{2.27}$$

$$\vec{\varphi}_{i,2} = asin(\cos(\phi_i)\cos(\theta_i)\cos(\psi_i) + \sin(\phi_i)\sin(\theta_i)) \tag{2.28}$$

$$\begin{aligned}\vec{\varphi}_{i,3} = atan2(\quad & \sin(\phi_i)\sin(\lambda_i), \cos(\theta_i)\cos(\psi_i) + \cos(\lambda_i)\cos(\theta_i)\sin(\psi_i) \\ & + \cos(\phi_i)\sin(\lambda_i)\sin(\theta_i), \quad \sin(\phi_i)\cos(\lambda_i)\cos(\theta_i)\cos(\psi_i) \\ & \sin(\lambda_i)\cos(\theta_i)\sin(\psi_i) + \cos(\phi_i)\cos(\lambda_i)\sin(\theta_i))\end{aligned} \tag{2.29}$$

The distance between the waypoints can be determined using the distance equation.

$$d_{r,i} = \sqrt{(\vec{r}^{\,e}_{er,i+1} \quad \vec{r}^{\,e}_{er,i})^T (\vec{r}^{\,e}_{er,i+1} \quad \vec{r}^{\,e}_{er,i})} \tag{2.30}$$

2.5.2 Measurement Structure

In order to use lateral lane position measurements, the lateral lane position with respect to the lane map must be estimated using the current states of the navigation filter. $\vec{r}_{eb}^{\,e}$ denotes the current position estimate in the ECEF coordinate frame, $\vec{r}_{er,i}^{\,e}$ denotes the position of the last waypoint passed in the north east coordinate frame, and C_e^r is the rotation matrix given in (2.20). Equation (2.31) is used to find the position estimates in the road coordinate frame. $\hat{y}$ is the lateral lane position estimate, $\hat{x}$ is the distance into the current road coordinate frame, and $\hat{z}$ is the vertical position in the current road coordinate frame. The z-axis of the road coordinate frame points down.

$$\begin{array}{l}\hat{x}\\ \hat{y}\;\; = C_e^r \;\; \vec{r}_{eb}^{\,e} \quad \vec{r}_{er,i}^{\,e}\\ \hat{z}\end{array} \tag{2.31}$$

Equation (2.32) shows the measurement equations. The measurement equations are a function of the states of the filter and the map parameters. The first measurement equation is for the lateral lane position. This measurement is assumed to be provided by a camera or Lidar. The second measurement equation is the height above the road coordinate frame. Since this navigation filter structure is for a ground vehicle, this measurement can be assumed to be a constant. The matrix in (2.35) shows the measurement model, assuming the state matrix takes the form of (4.134) in Chapter 4. The measurement model is created by taking the partial derivative of the measurement equations with respect to each state.

$$h(x) = \begin{array}{l}\hat{y}\\ \hat{h}\end{array}$$

$$= \begin{array}{l} C^r_{e(2,1)}\left(\vec{r}^{\,e}_{eb,1} \quad \vec{r}^{\,e}_{er,i,1}\right) + C^r_{e(2,2)}\left(\vec{r}^{\,e}_{eb,2} \quad \vec{r}^{\,e}_{er,i,2}\right) + C^r_{e(2,3)}\left(\vec{r}^{\,e}_{eb,3} \quad \vec{r}^{\,e}_{er,i,3}\right)\\ C^r_{e(3,1)}\left(\vec{r}^{\,e}_{eb,1} \quad \vec{r}^{\,e}_{er,i,1}\right) \quad C^r_{e(3,2)}\left(\vec{r}^{\,e}_{eb,2} \quad \vec{r}^{\,e}_{er,i,2}\right) \quad C^r_{e(3,3)}\left(\vec{r}^{\,e}_{eb,3} \quad \vec{r}^{\,e}_{er,i,3}\right)\end{array} \tag{2.32}$$

$$e_1 = \;\; C^r_{e(2,1)}, C^r_{e(2,2)}, C^r_{e(2,3)} \tag{2.33}$$

$$e_2 = \quad C^r_{e(3,1)}, \;\; C^r_{e(3,2)}, \;\; C^r_{e(3,3)} \tag{2.34}$$

$$H(x) = \begin{matrix} e_1 & 0 & 0 & 0 & 0 & 0 \\ e_2 & 0 & 0 & 0 & 0 & 0 \end{matrix} \tag{2.35}$$

2.5.3 Checking Waypoint Map Position

After every measurement update and time update, the states of the navigation filter will change. Every time the states of the filter are updated, the distance into the current road frame must be checked to insure that the vehicle has not passed into the next road coordinate frame. Substituting the position states into (2.36) will result in the current longitudinal distance into the road coordinate frame. If this value is larger than $d_{r,i}$, then the vehicle has passed into the next coordinate frame. The map index i should then be incremented by 1.

$$\hat{x} = C^r_{e(1,1)}\left(\vec{r}^{\,e}_{eb,1} \quad \vec{r}^{\,e}_{er,i,1}\right) + C^r_{e(1,2)}\left(\vec{r}^{\,e}_{eb,2} \quad \vec{r}^{\,e}_{er,i,2}\right) + C^r_{e(1,3)}\left(\vec{r}^{\,e}_{eb,3} \quad \vec{r}^{\,e}_{er,i,3}\right) \tag{2.36}$$

2.5.4 Results

Figure 2.11 shows the utility of using vision to aid a GPS-based navigation filter. Figure 2.11 is a plot of the lateral lane position estimates. This data for

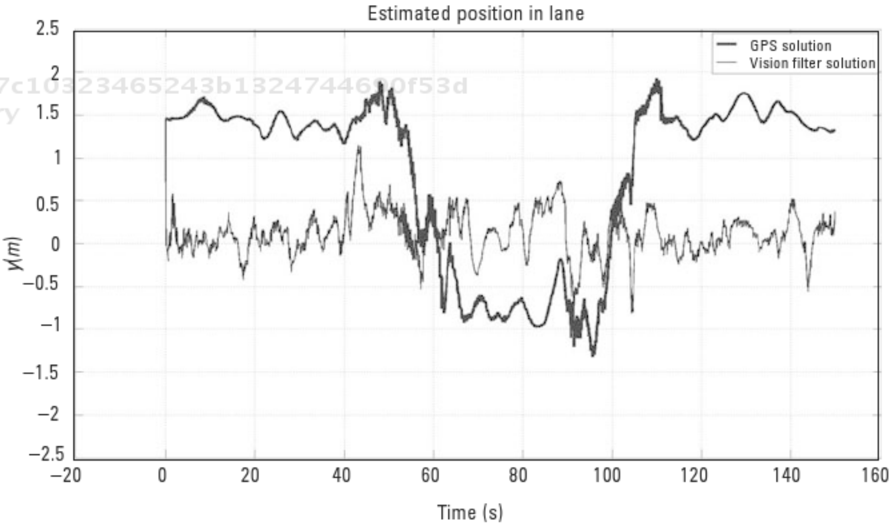


Figure 2.11 Estimated lane position reported by GPS and the vision aided navigation filter.

these results was collected at the NCAT test track, which is a 1.7-mile, two-lane oval. The track is set up very similar to the interstate system. The large line represents the GPS measurements. GPS position is a bias measurement. These biases make it difficult to compare a GPS position measurement and a map to get vehicle position within a lane. The small line is the result of the vision aided navigation filter. Unlike the GPS measurement, the estimated lane position reported by the vision aided navigation filter is not biased [5].

References

[1] Britt, J., and D. Bevly, "Lane Tracking Using Multilayer Laser Scanner to Enhance Vehicle Navigation and Safety Systems," *2009 International Technical Meeting*, Anaheim, CA, 2009.

[2] Farell, J., *Aided Navigation GPS with High Rate Sensors*, New York: McGraw-Hill, 2008.

[3] Groves, P. D., *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, Norwood, MA: Artech House, 2008.

[4] Rose, C., and D. Bevly, "Vehicle Lane Position Estimation with Camera Vision using Bounded Polynomial Interpolated Lines," *2009 International Technical Meeting*, Anaheim, CA, 2009.

[5] Allen, J., and D. M. Bevly, "Use of Vision Sensors and Lane Maps to Aid GPS/INS Under a Limited GPS Satellite Constellation," *Proceedings of the 2009 ION GNSS*, Savannah, GA, September 2009.

3

Vehicle Modeling

Lowell Brown and Dustin Edwards

3.1 Introduction

The goal of any model is to describe physical phenomena. Our interest is vehicular dynamics. There are several models that exist for describing vehicle motion [1–16]. We will explore a few of them. First, though, conventions must be established.

3.2 SAE Vehicle Coordinates

The Society of Automotive Engineers has established the SAE Vehicle Coordinate System. This is the system that will be used for body fixed coordinates. As shown in Figure 3.1 this coordinate system defines the longitudinal axis of the vehicle as x, the lateral axis to be y, and the vertical axis, z, points toward the ground. The coordinate system also defines the direction of roll rate p, pitch rate q, and yaw rate r.

The body fixed SAE coordinate systems is attached to a vehicle that moves about the East, North, Up (ENU) global reference frame, as shown in Figure 3.2.

In Figure 3.2 the horizontal axis is the East direction and vertical axis is the North direction. The positive Up direction comes out pointed at the

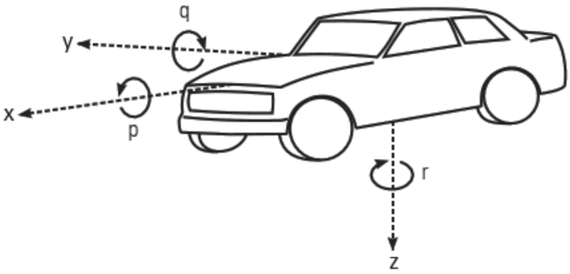


Figure 3.1 SAE Vehicle Coordinate System defined by the SAE [1].

reader and is perpendicular to the E-N plane. The box represents a vehicle with V_x and V_y velocity components. Vehicle heading is ψ, the angle between a line parallel to the North axis and the longitudinal velocity component V_x. Sideslip, β, is defined as the angle between the magnitude of velocity and the longitudinal component of velocity, V_x. Vehicle course, the direction of travel, is defined as ν, where

$$\nu = \psi + \beta \tag{3.1}$$

Note that ψ is equivalent to r, the yaw rate.

Most models in this chapter will be based off the basic "bicycle model." Other models are developed to capture dynamics the bicycle model is unable to describe. Also, simplifications are made to the bicycle model that are valid

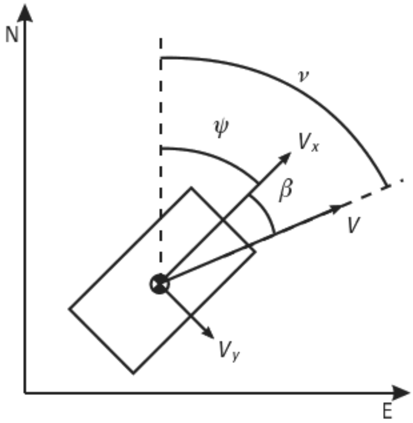


Figure 3.2 ENU global reference plane.

during certain steady-state maneuvers. The models developed in this chapter are important in later sections when developing parameter estimation algorithms and controlling a vehicle autonomously. Therefore, it is important to know the limitations and accuracy of each model.

3.3 Bicycle Model

3.3.1 Basics

3.3.1.1 Assumptions

Now that coordinate conventions have been established, the vehicle models can be explored. The first of these models is the bicycle model. In the bicycle model the inner and outer tires of each axle are represented by a single tire at the center of the vehicle's axle. Thus the front is represented by a single front tire and the rear axle is represented by a single rear tire. So, the assumption is made that the inner and outer slip angles and the inner and outer steer angles are the same. The steering angle of the vehicle is represented by δ. The slip angle of the vehicle is denoted as β and describes the angle between the velocity vector and the longitudinal axis of the vehicle. The bicycle model also neglects weight transfer, the difference in vertical force between the inner and outer tires.

Figure 3.3 represents the bicycle model. The CG is the center of gravity. The body fixed reference frame has an origin at the vehicle's center of gravity. The z-axis is pointing down toward the ground, the x-axis is pointing toward the front of the vehicle, and the y-axis is pointing out to the vehicle's passenger side. In Figure 3.3, δ is the steering angle, and a and b are distances

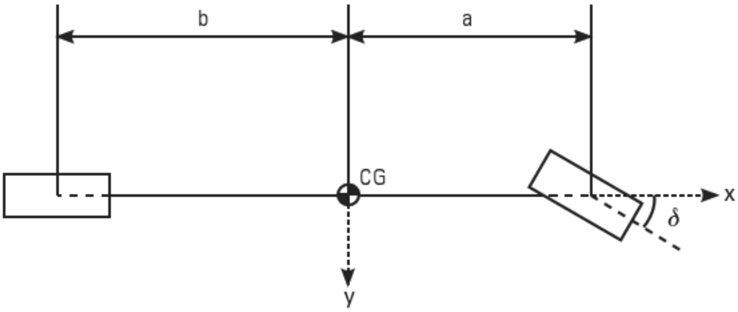


Figure 3.3 Bicycle model.

from the vehicle center of gravity to the front and rear axles, respectively. This model assumes that left and right steer angles are the same for front and rear axles.

3.3.1.2 Kinematic Version

At low speeds with no slip or minimal slip, the slip angles at the tires can be assumed to be zero, which produces the lateral kinematic version of the bicycle model. The perpendicular line from each tire passes through the same point, which is called the center of the turn. Figure 3.4 represents the kinematic bicycle model. Note the perpendicular lines that originate at each axle and pass through the center of the radius, *R*. If the steering angle of the front tire reaches zero, the radius of curvature, *R*, goes to infinity. When the steering angle is not zero the steering angle can be described by (3.2), known as the Ackerman angle.

$$\delta = \tan \ ^{1} \ \frac{L}{R} \div \quad \frac{L}{R} \tag{3.2}$$

By using simple kinematics, the vehicle's velocity, *V*, can be described as the yaw rate, *r*, of the vehicle times the radius of curvature, *R*. Also with no lateral sliding, the lateral acceleration, a_y, of the vehicle is simply the centripetal acceleration developed during the turn.

$$V = Rr \tag{3.3}$$

$$a_y = \dot{V}_y = Rr^2 = \frac{V^2}{R} = Vr \tag{3.4}$$

Substituting (3.3) into (3.2) results in the expected yaw rate of the kinematic model, given a steer input and velocity.

$$r = \ \frac{V}{L} \div \tan(\delta) \qquad \frac{V}{L} \div \delta \tag{3.5}$$

3.3.1.3 Lateral Version

For higher speeds that develop greater slip angles the lateral bicycle model can be used for vehicles that produce minimal roll angles. Figure 3.5 represents the lateral bicycle model where *r* is the yaw rate, $V = [V_x \ \ Y_y]_T$ is the velocity vector acting at the vehicle's center of gravity, ***b*** represents the sideslip angle, ***d*** is the steering angle, and *a* and *b* are distances from the vehicle center of

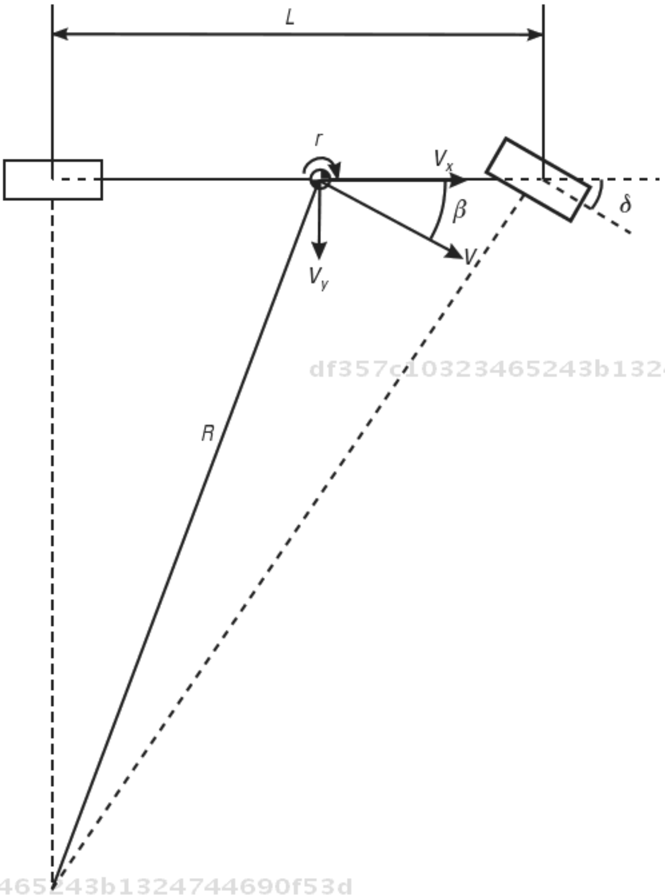


Figure 3.4 Kinematic bicycle model.

gravity to the front and rear axles, respectively. The front and rear tire slip angles are denoted as α_f and α_r, respectively. By summing the forces and moments about the CG on the free body diagram, Figure 3.5, and accounting for bank angle a simple set of dynamic equations can be derived to describe the vehicle's lateral motion. Note that the force due to the bank angle will be the mass times gravity times the sin of the bank angle.

$$F_y = m\ddot{y} = m(V_x\dot{\beta} + V_x r) = F_{yF} + F_{yR} + mgsin(\theta_{bank}) \qquad (3.6)$$

$$M = I_z\ddot{\psi} = I_z\dot{r} = aF_{yF} + bF_{yR} \qquad (3.7)$$

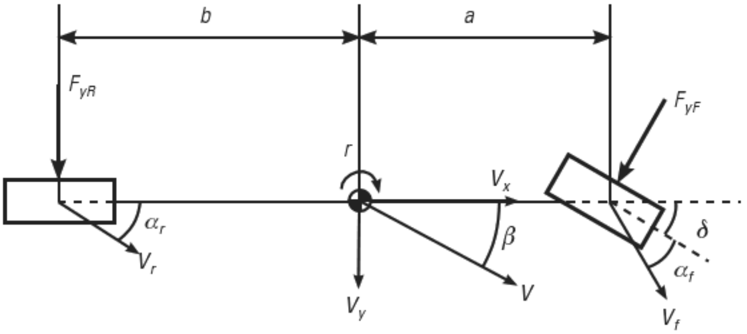


Figure 3.5 Lateral bicycle model.

Solving (3.6) and (3.7) for sideslip rate $\dot{\beta}$ and yaw acceleration $\dot{r}$ yields

$$\dot{\beta} = \frac{F_{yF}\cos(\delta) + F_{yR} + mgsin(\theta_{bank})}{mV_x} \quad r \tag{3.8}$$

$$\dot{r} = \frac{aF_{yF}\cos(\delta) \quad bF_{yR}}{I_z} \tag{3.9}$$

To describe the front and rear tire forces, $F_{yF,R}$, a linear tire model will be used. This model assumes the tire forces remain in the linear region of the tire and are proportional to the tire's respective slip angle times the tire's cornering stiffness, C_α, shown in (3.10) and (3.11). Tire models while be discussed more in depth in Section 3.4.3.

$$F_{yF} = \quad C_{\alpha f}\alpha_f \tag{3.10}$$

$$F_{yR} = \quad C_{\alpha r}\alpha_r \tag{3.11}$$

Substituting (3.10) and (3.11) into (3.6) and (3.7), a state space representation of this model can be developed and is shown by (3.12).

$$\begin{array}{c} a_y \\ \dot{r} \end{array} = \begin{array}{ccc} \frac{C_0}{mV_x} & V_x & \frac{C_1}{mV_x} \\ \frac{C_1}{I_zV_x} & & \frac{C_2}{I_zV_x} \end{array} \begin{array}{c} V_y \\ r \end{array} + \begin{array}{c} \frac{C_{\alpha f}}{m} \\ \frac{C_{\alpha f}a}{I_z} \end{array} \delta \tag{3.12}$$

where

$$\begin{aligned} C_0 &= C_{\alpha f} + C_{\alpha r} \\ C_1 &= aC_{\alpha f} \quad bC_{\alpha r} \\ C_2 &= a^2 C_{\alpha f} + b^2 C_{\alpha r} \end{aligned} \tag{3.13}$$

With a state space representation, this model can be configured for control or estimation purposes. Control and estimation are discussed in proceeding chapters.

Further examination of the lateral bicycle model can reveal the steady-state tire slip. Assuming yaw acceleration is equal to zero and lateral acceleration is equal to (3.4), the centripetal acceleration, then (3.6) and (3.7) are simplified.

$$m\frac{V_x^2}{R} = F_{yF} + F_{yR} \tag{3.14}$$

$$0 = aF_{yF} + bF_{yR} \tag{3.15}$$

Substituting (3.10) and (3.11) into the above equations, the steady-state tire slip can be solved.

$$\alpha_f = \frac{W_f V_x^2}{C_{\alpha f}\, gR} \tag{3.16}$$

$$\alpha_r = \frac{W_r V_x^2}{C_{\alpha r}\, gR} \tag{3.17}$$

For control purposes it is advantageous to look at the relationship between yaw rate, r, and steer angle, δ, or between sideslip, β, and steer angle, δ. Combining (3.10) and (3.11) into (3.6) and (3.7) in an alternate manner and again neglecting the bank angle yields the state space representation:

$$\begin{matrix} \dot{\beta} \\ \dot{r} \end{matrix} = \begin{matrix} \dfrac{C_0}{mV_x} & \dfrac{C_1}{mV_x^2} \quad 1 \\ \dfrac{C_1}{I_z} & \dfrac{C_2}{I_z V_x} \end{matrix} \quad \begin{matrix} \beta \\ r \end{matrix} + \begin{matrix} \dfrac{C_{\alpha f}}{mV_x} \\ \dfrac{C_{\alpha f}\, a}{I_z} \end{matrix} \quad \delta \tag{3.18}$$

The equations for the transfer function from steer angle to yaw rate and from steer angle to sideslip are shown here.

$$\frac{r(s)}{\delta(s)} = \frac{\frac{aC_{\alpha f}}{I_z}s + \frac{(a+b)C_{\alpha f}C_{\alpha r}}{mV_xI_z}}{s^2 + \frac{C_0I_z + mC_2}{mV_xI_z}s + \frac{C_0C_2 \quad C_1mV_x^2 \quad C_1^2}{mV_x^2I_z}} \tag{3.19}$$

$$\frac{\beta(s)}{\delta(s)} = \frac{\frac{C_{\alpha f}}{mV_x}s + \frac{C_{\alpha f}C_2 \quad aC_{\alpha f}C_1 \quad mV_x^2aC_{\alpha f}}{mV_x^2I_z}}{s^2 + \frac{C_0I_z + mC_2}{mV_xI_z}s + \frac{C_0C_2 \quad C_1mV_x^2 \quad C_1^2}{mV_x^2I_z}} \tag{3.20}$$

3.3.1.4 Longitudinal Dynamics

The longitudinal dynamics of a vehicle are influenced by the longitudinal force of the tires. Tires cannot generate force without slipping. This will be discussed in greater detail in Section 3.4.2. In order for a vehicle to move longitudinally slip must occur at the wheels. The percent slip of a wheel is a function of the velocity, V, angular velocity, ω, and effective radius, R_e, of the wheel and is defined as

$$\%\ \text{slip} = \frac{V \quad R_e\omega}{V} \tag{3.21}$$

The effective radius can be defined approximately as a function of the loaded radius, R_L, and the unloaded radius, R_U, of the wheel.

$$R_e \quad R_U \quad \frac{R_U \quad R_L}{3} \tag{3.22}$$

Examining the x-z plane of the SAE coordinate system and using a half car model generates the free body diagram (Figure 3.6). The effect of drag is modeled as ρV^2 acting on the CG where V is equivalent to $\dot{x}$ and the longitudinal velocity ρ is the drag coefficient. Note that wind speed is not taken into account. Also, note that the four wheels are combined into front and rear axles.

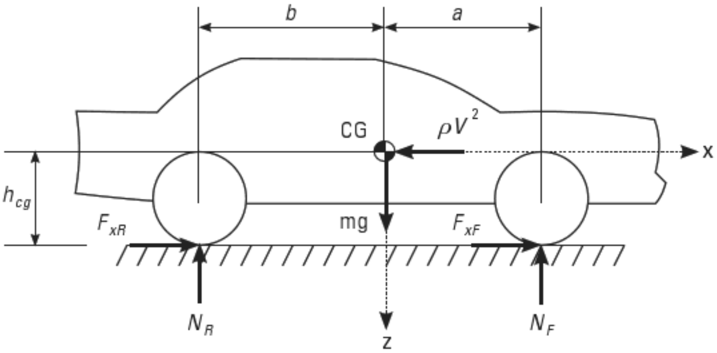


Figure 3.6 Longitudinal free body diagram.

Neglecting air drag and summing the forces and moments yields the longitudinal equations of motion.

$$F_z = m\ddot{z} = \quad N_F \quad N_R = mg \tag{3.23}$$

$$F_z = m\ddot{x} = F_{xF} + F_{xR} \tag{3.24}$$

$$M_{y,cg} = I_y\ddot{\theta} = bN_R \quad h_{cg}(F_{xF} + F_{xR}) \quad aN_F \tag{3.25}$$

Combining the equations of motion produces solutions for the normal forces acting on the front and rear axles.

$$N_F = mg \quad \frac{I_y\ddot{\theta} + h_{cg}m\ddot{x} + amg \quad am\ddot{z}}{a+b} \quad M\ddot{z} \tag{3.26}$$

$$N_R = \frac{I_y\ddot{\theta} + h_{cg}m\ddot{x} + amg \quad am\ddot{z}}{a+b} \tag{3.27}$$

For the case of steady state pitch $\ddot{\theta} = 0$ and $\ddot{z} = g$. The normal forces on the front and rear axles then become

$$N_F = \frac{bmg}{a+b} \quad \frac{m\ddot{x}h_{cg}}{a+b} \tag{3.28}$$

$$N_R = \frac{amg}{a+b} + \frac{m\ddot{x}h_{cg}}{a+b} \tag{3.29}$$

Note that in the steady state pitch case the first term of N_F and N_R is the front and rear weight splits, respectively.

For the case with air drag included, the summation of forces in the x-direction becomes

$$F_x = m\ddot{x} = F_{xF} + F_{xR} \quad \rho\dot{x}^2 \tag{3.30}$$

The effect of air drag is important when designing a velocity controller. The control gains can be linearized about the operating point. This will be discussed in Chapter 6.

3.3.2 Understeer Gradient

In order to develop a better understanding of the turning response of a vehicle, the understeer gradient of the vehicle is defined. Using the steady-state bicycle model, the understeer gradient can be determined from the weight distribution and the cornering stiffness [8]. By including slip angles, a simple kinematic equation between the steer angle and tire slip angles can be developed.

$$\delta \quad \frac{L}{R} + \alpha_f \quad \alpha_r \tag{3.31}$$

Substituting (3.16) and (3.17) into (3.31) gives:

$$\delta \quad \frac{L}{R} + \quad \frac{W_f}{C_{\alpha f}} \quad \frac{W_r}{C_{\alpha r}} \div \frac{V^2}{gR} \quad \frac{L}{R} + K_{us}\alpha_y \tag{3.32}$$

From (3.32), the understeer gradient is labeled as K_{US} and is defined as:

$$K_{US} = \frac{1}{g} \quad \frac{W_f}{C_{\alpha f}} \quad \frac{W_r}{C_{\alpha r}} \div$$

$$K_{US} = \frac{m}{L} \quad \frac{bC_{\alpha r} \quad aC_{\alpha f}}{C_{\alpha r}C_{\alpha f}} \div \tag{3.33}$$

The understeer gradient determines both the magnitude and the direction of the steering inputs required for a given lateral acceleration [16]. The understeer gradient also determines if the vehicle is neutral steer, oversteer, or understeer. Figure 3.7 shows the basic principles with K_{US} being the slope of each line.

3.3.2.1 Neutral Steer

Neutral steer occurs when the understeer gradient is zero, which results in the front and rear steady-state tire slip angles being equivalent. By studying (3.16), during neutral steer the steer angle required to make the turn is approximately the Ackerman angle.

3.3.2.2 Understeer

Understeer occurs when K_{US} is greater than zero, causing larger slip angles to develop in the front tire than the rear. Because there is more slip at the front tire, the steer angle must increase to maintain the radius of the curve. During this condition, the steer angle increases linearly with the speed squared or the lateral acceleration.

3.3.2.3 Oversteer

Oversteer is the opposite of understeer. During oversteer, K_{US} is less than zero causing the rear tire slip angle to be greater than the front. Because the rear is sliding more than the front, less steer angle is required to navigate the turn.

3.3.3 Four-Wheel Bicycle Model

To cope with the effects of roll, roll models are used. This often requires modifying the bicycle model into a four-wheel bicycle model. The four-wheel

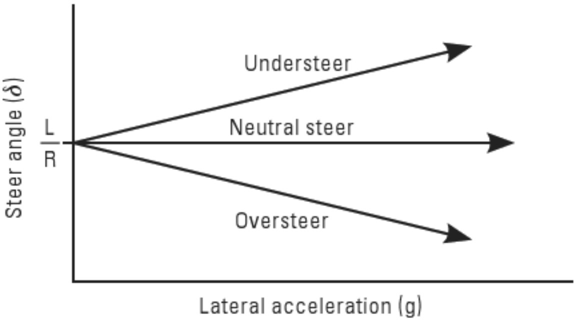


Figure 3.7 Basic understeer gradient plot.

bicycle model accounts for the difference between the inner and outer vertical tire forces. This difference is the weight transfer. Weight transfer is important to understand because the vertical force on tire affects the lateral and longitudinal forces of the tire that dictate the performance of handling dynamics and characteristics. Before exploring roll models, tire models and properties will be discussed.

The dynamic equations of motion for a ground vehicle model are derived for a vehicle that possesses both front and rear steering capabilities as well as individual torque control of each wheel. This general model is easily adapted to simpler vehicles (e.g., vehicles with only front-wheel steering), by the assignment of a constant control input of zero to the appropriate (unavailable) control effectors. The free body diagram for the vehicle under consideration is shown in Figure 3.8. A right-handed coordinate system is used for the derivation of the equations of motion. The body fixed reference frame has an origin at the vehicle's center of gravity. The *z*-axis is pointing down toward the ground, the *x*-axis is pointing toward the front of the vehicle, and the *y*-axis is pointing out to the vehicle's passenger side. In Figure 3.8, *r* is the yaw rate, $V = [\dot{x}\,\dot{y}]^T$ is the velocity vector acting at the vehicle's center of gravity, β represents sideslip angle, δ is the steering angle, *t* is the track width, and *a* and *b* are distances from the vehicle center of gravity to the front and rear axles,

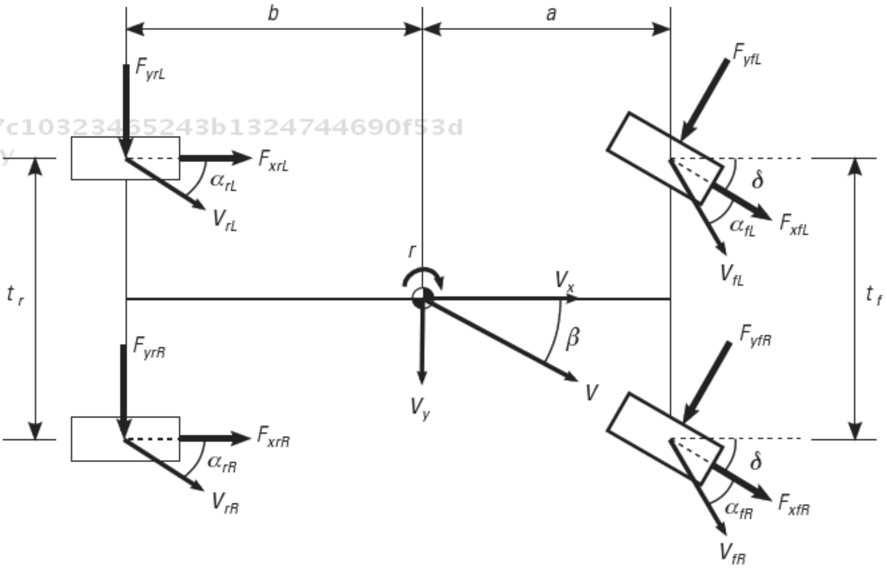


Figure 3.8 Four-wheel bicycle model.

respectively. Subscripts f, r, R, L denote front, rear, right, and left sides of the vehicle, respectively. For clarity, only the left slip angles (α_{fL} and α_{rL}) and the right steer angles (δ_{fR} and δ_{fL}) are shown on Figure 3.8. This model assumes that left and right steer angles are the same for front and rear axles. Note that all forces F and slip angles α are drawn in the positive direction such that the lateral force $F = \quad C_\alpha \alpha$ where C_α represents tire cornering stiffness.

The sum of the moments about the vehicle's center of gravity is written as

$$\begin{aligned} M_{cg} = I_{\mathbf{z}}\ddot{\psi} = a(&F_{yfL}\cos\delta_f + F_{yfR}\cos\delta_f \quad F_{xfL}\sin\delta_f \quad F_{xfR}\sin\delta_f) \\ &\frac{t_f}{2}(F_{ylF}\sin\delta_f \quad F_{yfR}\sin\delta_f \quad F_{xfL}\cos\delta_f + F_{xfR}\cos\delta_f) \\ &b\,(F_{yrL}\cos\delta_r + F_{yrR}\cos\delta_r + F_{xrL}\sin\delta_r + F_{xrR}\sin\delta_r) \\ &\frac{t_r}{2}(F_{yrL}\sin\delta_r \quad F_{yrR}\sin\delta_r \quad F_{xrL}\cos\delta_r + F_{xrR}\cos\delta_r) \end{aligned} \tag{3.34}$$

where I_z is the moment of inertia about the yaw axis. The solution of (3.34) for $\ddot{\psi}$ yields

$$\begin{aligned} \ddot{\psi} = \dot{r} = \quad &a\Big((F_{yfL} + F_{yfR})\cos\delta_f \quad (F_{xfL} + F_{xfR})\sin\delta_f\Big) \\ &\frac{t_f}{2}\Big((F_{ylF} + F_{yfR})\sin\delta_f \quad (F_{xfL} \quad F_{xfR})\cos\delta_f\Big) \\ &b\Big((F_{yrL} + F_{yrR})\cos\delta_r + (F_{xrL} + F_{xrR})\sin\delta_r\Big) \\ &\frac{t_r}{2}\Big((F_{yrL} \quad F_{yrR})\sin\delta_r \quad (F_{xrL} \quad F_{xrR})\cos\delta_r\Big) \;/ I_{\mathbf{z}} \end{aligned} \tag{3.35}$$

which describes the nonlinear yaw dynamics of the vehicle.

The sideslip angle β is the angle between the vehicle's actual velocity vector V and its longitudinal velocity component V_x. Therefore, the vehicle fixed velocities and corresponding accelerations due to sideslip are

$$\begin{aligned} V_x &= V\cos\beta \quad \dot{V}_x = \dot{V}\cos\beta \quad V\dot{\beta}\sin\beta \\ V_y &= V\sin\beta \quad \dot{V}_y = \dot{V}\sin\beta + V\dot{\beta}\cos\beta \end{aligned} \tag{3.36}$$

The effects of yaw rate are then included to give the complete expressions for acceleration in the vehicle fixed reference frame

$$\ddot{x} = \dot{V}\cos\beta \quad V\dot{\beta}\sin\beta \quad rV\sin\beta \tag{3.37}$$

$$\ddot{y} = \dot{V}\sin\beta \quad V\dot{\beta}\cos\beta \quad rV\cos\beta \tag{3.38}$$

The summation of the forces in the y-axis yields

$$\begin{aligned} F_y = m\ddot{y} = (F_{yfL} + F_{yfR})\cos\delta_f + (F_{xfL} + F_{xfR})\sin\delta_f + (F_{yrL} + F_{yrR})\cos\delta_r \\ + (F_{xrL} + F_{xrR})\sin\delta_r \end{aligned} \tag{3.39}$$

Equation (3.38) is substituted into (3.39) and solved for $\dot{\beta}$ to obtain the equation of motion for sideslip

$$\begin{aligned} \dot{\beta} = \Big[(F_{yfL} + F_{yfR})\cos\delta_f + (F_{xfL} + F_{xfR})\sin\delta_f + (F_{yrL} + F_{yrR})\cos\delta_r \\ +(F_{xrL} + F_{xrR})\sin\delta_r\Big] / \big(mV\cos\beta\big) \quad \dot{V}\tan\beta / V \quad r \end{aligned} \tag{3.40}$$

3.4 Tires

3.4.1 Basics

The tire-road relationship is governed by the contact patch. With the exception of aerodynamic forces, all external forces on the vehicle are developed at the tire's contact patch. The tire serves three basic functions:

1. It supports the vertical load, while cushioning against road shocks.
2. It develops longitudinal forces for acceleration and braking.
3. It develops lateral forces for cornering.

3.4.2 Contact Patch and Slip

Due to the elastic properties of the tire the contact patch deforms under loads. This deformation results in the slip angle α. Experimental data shows that the lateral force on a tire displays a nonlinear relationship with the slip angle "alpha." However, there is a linear portion for small angles of α. The slope of this linear portion is called the cornering stiffness C_α. Equations (3.10) and (3.11) can be derived from the linear portion of experimental data similar to the data shown in Figure 3.9.

Figure 3.9 shows typical characteristics of a tire under lateral force generation modeled by the Fiala tire model. More information on this model will be discussed later. As shown in the plot, the lateral tire force remains linear with slip angle, as modeled by (3.10) and (3.11), until the tire becomes saturated. This model relates peak tire force to the tire-road friction, μ, times the normal force, F_z, known as the peak tire force.

An increase in the normal or vertical force of a tire will cause the relationship between the lateral force and the slip angle to change. This increase will yield a higher peak tire force. Vertical forces on the tire are not only important for ride characteristics, but also help to describe the maximum longitudinal and lateral forces developed by the tire. As shown in Figure 3.10, the magnitude of lateral and longitudinal tire force cannot exceed the peak tire force. When the magnitude of tire force reaches this point, sliding occurs. By studying Figure 3.10, it is obvious that the available longitudinal

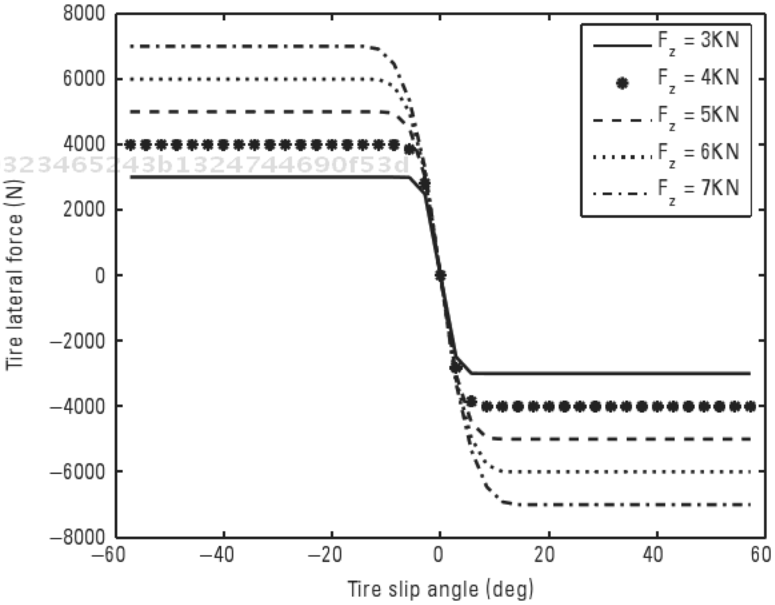


Figure 3.9 Generic tire curve.

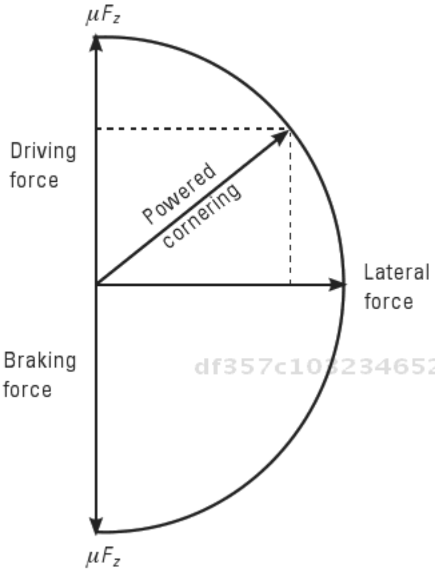


Figure 3.10 Circle of friction for tire forces.

drive and braking force decreases with an increase in lateral force. Because of this effect, both forces must be taken into account during combined lateral and longitudinal tire force generation to develop an accurate vehicle model.

3.4.3 Tire Models

Several researchers have developed models to describe the generated tire forces. One of the most well-known models, called the Magic Formula tire model, was developed by Pacejka [4, 5, 7]. This model is an empirical formula capable of calculating lateral and longitudinal tire forces. Alternatively, the two models used in this book are the Fiala and Dugoff tire model.

3.4.3.1 Fiala Tire Model

The Fiala tire model was originally developed to estimate lateral tire force generation only [2]. The model, however, was transformed to represent both lateral and longitudinal forces [3]. To accomplish this transformation lateral and longitudinal tire stiffness (C_α, C_σ) are assumed to be equal. This assumption though does not always hold true. The total slip σ for this model is simply the magnitude of the lateral and longitudinal slip (σ_y, σ_x), shown by (3.41).

$$\sigma = \sqrt{\sigma_y^2 + \sigma_x^2} \tag{3.41}$$

To calculate the total slip, the individual values for slip must be known. Both the longitudinal and lateral slips are found using (3.42) and (3.43).

$$\begin{aligned} \sigma_x &= \frac{r_{eff}\,\omega_w \quad V_x}{r_{eff}\,\omega_w} \textit{ during acceleration} \\ \sigma_x &= \frac{V_x}{r_{eff}\,\omega_w} \textit{ during braking} \end{aligned} \tag{3.42}$$

$$\sigma_y = \frac{V_x}{r_{eff}\,\omega_w} \tan(\alpha) \tag{3.43}$$

Both slip values can be determined with the use of sensors. Assuming a parabolic pressure distribution on the tire's contact patch, (3.44) is used to describe the magnitude of force on the tire, using the Fiala tire model.

$$\begin{aligned} &\mu F_z \quad 3\theta\sigma \quad \frac{1}{3}(3\theta\sigma)^2 + \frac{1}{27}(3\theta\sigma)^3 \quad \textit{if } \sigma \quad \sigma_m \\ &F = \mu F_z \textit{ if } \sigma \quad \sigma_m \end{aligned} \tag{3.44}$$

The variable, σ_m, is the value of total slip where sliding occurs in the Fiala tire model. As described by the circle of friction, sliding is assumed to begin when the maximum tire force is equal to μF_z.

$$\sigma_m = \frac{1}{\theta} = \frac{3\mu F_z}{C_{\alpha / \sigma}} \tag{3.45}$$

The individual values of lateral and longitudinal tire force (F_y, F_x) can be obtained by breaking up the force magnitude F_t. This is done by multiplying the force magnitude by the ratio of total slip to each forces respective slip, as shown in (3.46) and (3.47).

$$F_x = \frac{\sigma_x}{\sigma} F \tag{3.46}$$

$$F_y = \frac{\sigma_x}{\sigma} F \tag{3.47}$$

In the case of pure lateral slip, set $\sigma_y = \tan(\alpha)$ and $\sigma_x = 0$ in the Fiala tire model. In case of pure longitudinal slip, set $\sigma_y = 0$ [10]. By reducing the combined force generation model to either lateral or longitudinal force generation, simpler calculations can be performed if the amount of noisy measurements is reduced.

3.4.3.2 Dugoff Tire Model

The Dugoff model and Fiala model both allow for tire force estimates during combined lateral and longitudinal tire force generation. However, the Dugoff tire model assumes a uniform vertical pressure distribution on the tire's contact patch [6]. This is a simplification from the Fiala's tire model, but it allows for individual values of lateral and longitudinal tire stiffness. The longitudinal and lateral tire forces are given by (3.48) and (3.49), respectively.

$$F_x = C_\alpha \frac{\sigma_x}{1+\sigma_x} f(\lambda) \tag{3.48}$$

$$F_y = \frac{C_\alpha \tan(\alpha)}{1+\sigma_x} f(\lambda) \tag{3.49}$$

where

$$\lambda = \frac{uF_z(1+\sigma_x)}{2\ \ (C_\sigma\sigma_x)^2 + (C_\alpha \tan(\alpha))^2\ \ ^{\frac{1}{2}}} \tag{3.50}$$

$$f(\lambda) = \begin{array}{l} (2\quad \lambda)\lambda \quad if\, \lambda < 1 \\ \quad\ 1 \quad if\, \lambda \quad 1 \end{array} \tag{3.51}$$

Similar to the Fiala tire model, this model has a transition that occurs when $\lambda = 1$. This transition occurs when the tire leaves the linear region and begins the nonlinear region. If the tire is experiencing lateral slip only, the model may be reduced by setting $\sigma_x = 0$ or for pure longitudinal force generation simply set $\alpha = 0$. This helps to simplify the model during driving conditions where only lateral or longitudinal forces are generated.

3.5 Roll Model

It is very important to understand vehicle roll and rollover. Many researchers have developed models to describe the roll dynamics of vehicles during cornering. Some models are fairly simple while others are very in depth and require more parameters. The less complex roll models do not include the springs and dampers of the suspension and therefore assume the sprung mass is stationary with the axle. Other high-fidelity models take into account forces produced by the springs and dampers.

3.5.1 Free Body Diagram

In order to produce a reliable roll model, a free body diagram (FBD) must be developed. The FBD in Figure 3.11 shows a two-state roll plane model [13, 14]. Three important parameters used in this model include the CG height, h_{cg}, roll stiffness, K_ϕ, and roll damping coefficient, C_ϕ. This model lumps the entire vehicle mass into the sprung mass. This assumption allows a simplified equation for the spring and damper torques, shown in (3.52) and (3.53).

$$T_{spring} = K_\phi \phi \tag{3.52}$$

$$T_{damper} = C_\phi \phi \tag{3.53}$$

Notice that both equations also assume the spring and damper torques are linear with roll, ϕ, and roll rate, $\dot{\phi}$, respectively. Also, note that $\dot{\phi}$ is equivalent to p, the SAE defined roll rate.

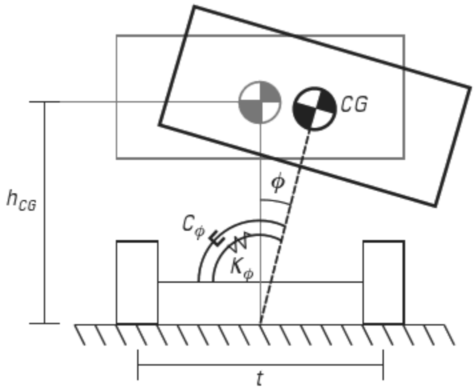


Figure 3.11 Vehicle roll free body diagram.

3.5.2 Equation of Motion

By summing the moments about the roll center on Figure 3.11, a simple equation is derived to describe the roll dynamics of the vehicle. Equation (3.54) assumes the vehicle's sprung mass rotates about a fixed point at the centerline of the lateral axis on the ground.

$$J_{eff}\ddot{\phi} + C_{\phi}\dot{\phi} + K_{\phi}\phi = mh_{cg}(a_y \cos(\phi) + g\sin(\phi)) \tag{3.54}$$

By assuming a steady-state turn and small angles, (3.54) can be simplified to solve for the roll angle with knowledge of the CG height and the spring roll stiffness.

$$\phi = \frac{mh_{cg}a_y}{K_{\phi} \quad mh_{cg}g} = \frac{mh_{cg}V^2}{R(K_{\phi} \quad mh_{cg}g)} \tag{3.55}$$

3.5.3 State Space Representation

Equation (3.55) may be transformed into a state space representation. The state space representation is shown in (3.56).

$$\begin{matrix}\dot{\phi} \\ \ddot{\phi}\end{matrix} = \begin{matrix} 0 & 1 \\ \frac{K_{\phi} \quad mgh_{cg}}{J_{eff}} & \frac{C_{\phi}}{J_{eff}} \end{matrix} \quad \begin{matrix}\phi \\ \dot{\phi}\end{matrix} + \begin{matrix} 0 \\ \frac{mh_{cg}}{J_{eff}} \end{matrix} \; a_y \tag{3.56}$$

Many other models have also been used to analyze roll dynamics. Some models developed do not assume the vehicle's roll center is located at ground height. One model developed in [11, 12] assumes the roll center is not at ground level and that the imaginary roll center derived produces reactionary forces.

3.6 Additional Models Used in this Work

In addition to the bicycle models two other models are used in later chapters to describe two-wheeled vehicles and trailer dynamics.

3.6.1 Two-Wheeled Vehicle

The two-wheeled vehicle motion is modeled as a kinematic relationship to describe the position changes as shown in Figure 3.12.

The variable $\theta = [\theta_L, \theta_R]$ represents angular positions of the robot wheels. The variable d is the distance between the driving wheels, or track width. The vehicle's change in position in the East, North, Up (ENU) Cartesian coordinate frame can be found from the following kinematic equations.

$$\dot{\psi} = r \tag{3.57}$$

$$\dot{e} = V_x \sin\psi \tag{3.58}$$

$$\dot{n} = V_x \cos\psi \tag{3.59}$$

where

$$V_x = \frac{R}{2}(\dot{\theta}_R + \dot{\theta}_L) \tag{3.60}$$

$$r = \frac{R}{d}(\dot{\theta}_L \quad \dot{\theta}_R) \tag{3.61}$$

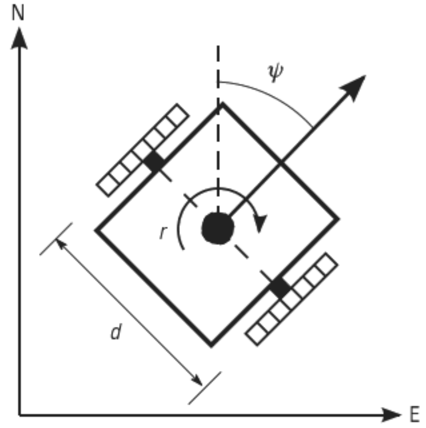


Figure 3.12 Two-wheeled vehicle kinematic diagram.

A state space representation of the transform from body fixed XYZ frame to the global ENU frame is shown in (3.62).

$$\begin{matrix} \dot{e} \\ \dot{n} \\ \dot{\psi} \end{matrix} = \begin{matrix} \sin\psi \\ \cos\psi \\ 0 \end{matrix} V_x + \begin{matrix} 0 \\ 0 \\ 1 \end{matrix} r \qquad (3.62)$$

3.6.2 Trailer Model

A kinematic model is used to describe a trailer or implement towed behind the vehicle. A schematic of a vehicle pulling a trailer is shown in Figure 3.13. The trailer's point of zero lateral velocity, ZLV, as well as its control point,

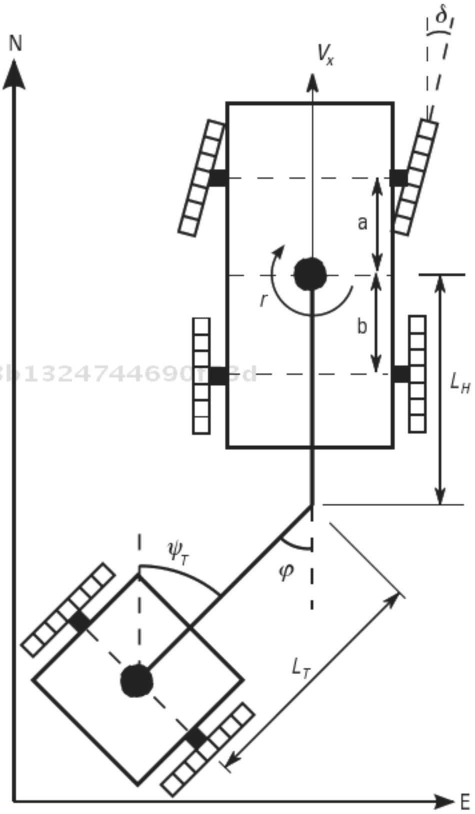


Figure 3.13 Vehicle-trailer free body diagram.

cp, are shown in Figure 3.13. The ZLV is the point on the implement where the lateral velocity is equal to zero. For a two-wheeled trailer, the ZLV point is assumed to be the point at the center of the axle between the two wheels. For farm implements or more complicated trailer designs, more sophisticated system identification techniques may be required to determine the ZLV point. The control point defines the point on the trailer that should follow the specified path. This point can be chosen by the designer and is often chosen to be the location of the GPS antenna on the trailer if one is used to determine the trailer's position.

A kinematic model that relates the vehicle velocity, V_x, and yaw rate, r, to the angle between the vehicle and the trailer is referred to as the hitch angle, φ, and is given by (3.63)

$$\dot{\varphi} = \quad r \;\; 1 + \frac{L_H}{L_T}\cos\varphi \div \quad \frac{V_x}{L_T}\sin\varphi \qquad (3.63)$$

where L_H is the distance from the CG of the vehicle to the hitch point and L_T is the distance from the hitch point to the ZLV point of the trailer. The model is derived by looking at how vehicle velocity and yaw rate translate into velocities at the tow pin. The resulting tow pin velocities can then be used to determine the rotational velocity of the implement assuming zero lateral velocity at the ZLV point.

Table 3.1
Properties of Carsim's G35 Sedan

Wheelbase	L	2.85m
Vehicle Mass	m	940 kg
Dist. from CG to Front Contact Patch	a	1.019m
Dist. from CG to Rear Contact Patch	b	1.831m
Yaw Moment of Inertia	I_z	1,530 kg * m^2
Front Tire Cornering Stiffness	$C_{\alpha f}$	78,311 N/rad
Rear Tire Cornering Stiffness	$C_{\alpha r}$	47,033 N/rad
Injected Noise Value on a_y	$\sigma^2_{\alpha_y}$	(.1 m/s^2)^2
Injected Noise Value on r	σ^2_r	(.02 rad/s)^2
Injected Noise Value on β	σ^2_β	(.02 rad)^2
Injected Noise Value on V	σ^2_v	(.01 m/s)^2

3.7 Vehicle Model Validation

To show the accuracy and limitations of the models, each model is validated with experimental data. The data is from a G35 sedan at the National Center for Asphalt Technology (NCAT) test track. The Infiniti G35 sedan is implemented with a dual antenna GPS unit, wheel speed sensor, a 6 DOF inertial measurement unit (IMU), and an optical encoder for steer angle measurements. Carsim, a high-fidelity vehicle simulation software, is also used for roll model verification.

With the data gathered at the NCAT test track, the kinematic and bicycle model are validated in MATLAB. The parameter values used in the simulations of the G35 sedan are listed in Table 3.1.

By using MATLAB to simulate the dynamic equations presented in this chapter, Figure 3.14 shows that both the kinematic and bicycle model matches the recorded data at 2 m/s, as would be expected. However, for larger slip angles, the assumptions of the kinematic model break down causing the

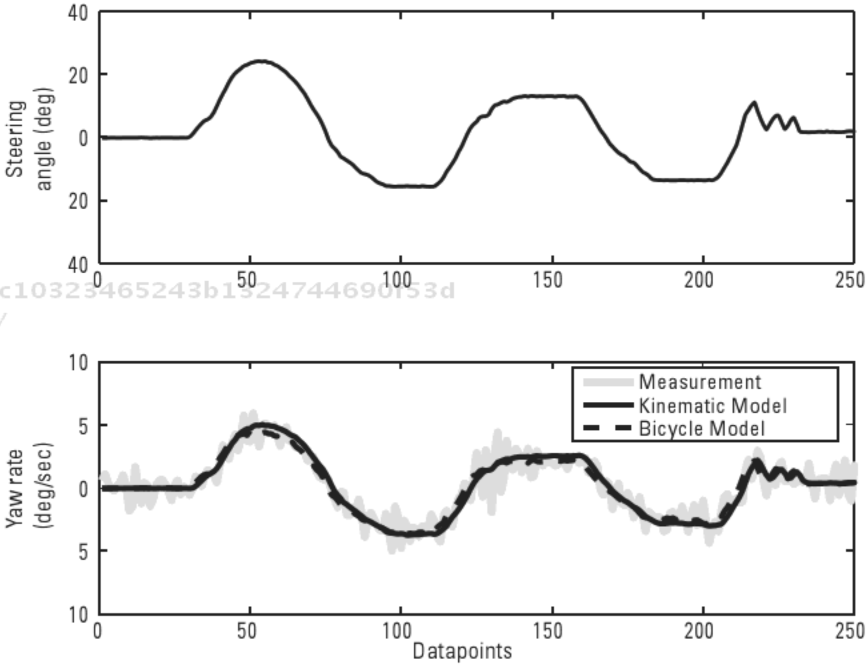


Figure 3.14 Comparison of the kinematic and bicycle model during slow-speed turning in the G35 sedan.

model to perform poorly. In the data logged at NCAT, slip angles remained small enough for both the bicycle and kinematic model to hold true. When vehicles reach higher speeds, as shown in the next experiment, the simplistic kinematic model is not the best choice.

To show the shortcomings of the kinematic model, data was logged in a G35 sedan at higher speeds around the NCAT test track. The inputs were run through both models in MATLAB and the results are shown in Figure 3.15. Notice the difference in the kinematic and bicycle models prediction of yaw rate. While cornering at high speeds with large slip angles, the kinematic model cannot accurately predict the vehicle's dynamics.

To illustrate the shortcomings of the bicycle model with a linear tire model, a maneuver is conducted that saturates the tires enough for the vehicle to begin sliding. This is a very hard maneuver and is conducted to show that

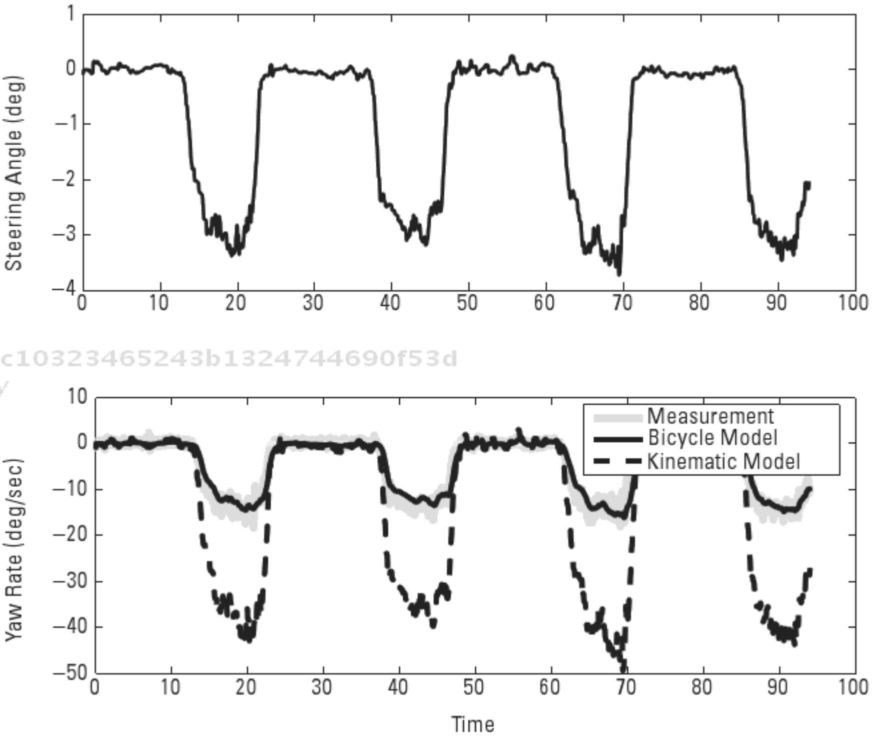


Figure 3.15 Comparison of the kinematic and bicycle model during high-speed cornering in the G35 sedan.

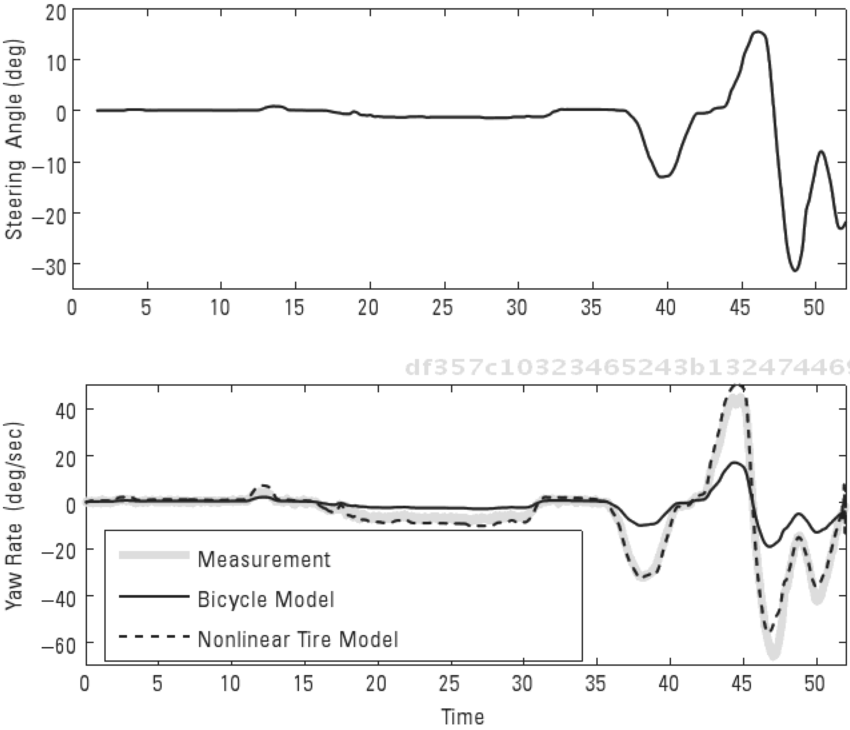


Figure 3.16 Comparison of the bicycle model with linear and nonlinear tire models during high-speed sliding experiments in the G35 sedan.

a linear tire model without saturation cannot describe the vehicle motion at the limits of handling. The more advanced model uses the Dugoff tire model to calculate the lateral tire forces. Figure 3.16 shows the bicycle model with a linear tire failing to match the data when the vehicle loses control. However, the bicycle model with a nonlinear tire model matches the data fairly well, although there is still some mismatch at the highest peak. This is most likely due to the fact that the model does not take into account vehicle roll dynamics.

Finally, the two-state roll plane model is tested. The vehicle used in this simulation is a large SUV from Carsim. Carsim is a high-fidelity vehicle simulation tool that can be used to validate simplified vehicle models. Carsim is chosen in this experiment because a vehicle is needed that produces large roll angles, unlike the G35 sedan used in previous experiments. Table 3.2 provides the parameter values used in the roll plane model during this simulation.

Table 3.2
Properties of Carsim's Large SUV

Vehicle Mass	m	2,450 kg
CG Height (Sprung Mass)	h_{CG}	1.1m
Roll Mass Moment of Inertia	J_{eff}	1,243 kg * m^2
Roll Stiffness	K_ϕ	$2{,}527.9\ \frac{\text{N m}}{\text{deg}}$
Roll Damping	C_ϕ	$152.05\ \frac{\text{N m s}}{\text{deg}}$
Track Width	T	1.62m
Injected Noise Value on a_y	$\sigma^2_{a_y}$	$(.1\ \text{m/s}^2)^2$
Injected Noise Value on ϕ	σ^2_{ϕ}	$(.01\ \text{rad})^2$
Injected Noise Value on $\dot{\phi}$	$\sigma^2_{\dot{\phi}}$	$(.005\ \text{rad/s})^2$

The large SUV attempts a double-lane change in order to induce large roll rates and angles. The data from Carsim is used to compare with the simple two-state roll plane model, which is simulated in MATLAB. The data from Carsim matches up well with the simple roll plane model, given

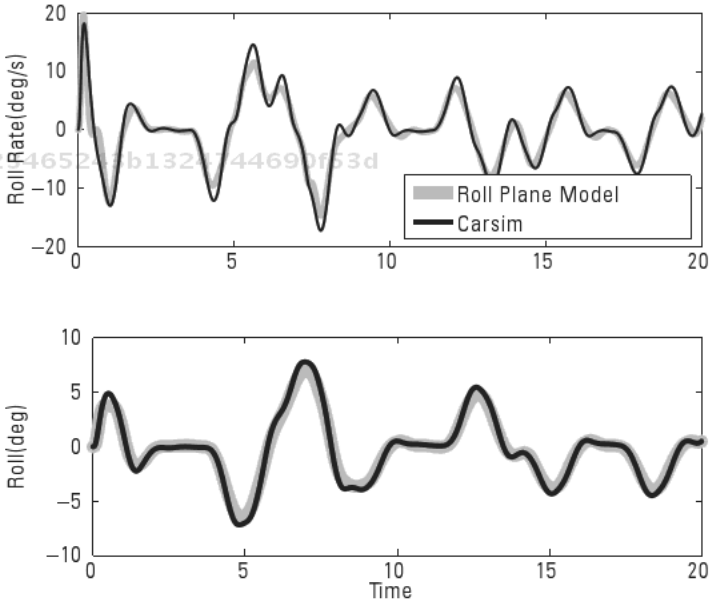


Figure 3.17 Roll plane model validation of a double lane change maneuver in Carsim.

in (3.55), as shown in Figure 3.17. One difference in this model and the Carsim high-fidelity model is the springs used in the Carsim model are highly nonlinear. Shown by (3.54), the simple roll plane model assumes linear springs [9].

You should now have an adequate understanding and introduction to the vehicle models used throughout the proceeding chapters.

References

[1] Milliken, B., and D. Milliken, "Race Car Vehicle Dynamics," *Society of Automotive Engineers*, 1995.

[2] Fiala, E., "Lateral Forces on Rolling Pneumatic Tires," *Zeitschrift,* V. D. I, No. 29:96, 1954.

[3] Pacejka, G. and R. S. Sharp, "Shear Force Development by Pneumatic Tires in Steady State Conditions: A Review of Modelling Aspects," *Vehicle System Dynamics,* Vol. 20, 1991, pp. 121–176.

[4] Pacejka, H. B., E. Bakker, and L. Lidner, "A New Tire Model with an Application in Vehicle Dynamics Studies," SAE Paper, (890087), 1989.

[5] Pacejka, H. B., E. Bakker, and L. Nyborg, "Tyre Modelling for Use in Vehicle Dynamics Studies," SAE Paper, (870421), 1987.

[6] Dugoff, H., P. S. Fancher, and L. Sengal, *Tire Performance Characteristics Affecting Vehicle Response to Steering and Braking Control Inputs*, Technical Report, Office of Vehicle Systems Research, U.S. National Bureau of Standards, 1969.

[7] Pacejka, H. B., and E. Bakker, "The Magic Formula Tyre Model," *Vehicle System Dynamics,* Vol. 21, No. 1, 1992, pp. 1–18.

[8] Ryu, J., "State and Parameter Estimation for Vehicle Dynamics Using GPS," Ph.D. thesis, Stanford University, 2004.

[9] Lambert, K., "A Study of the Properties That Influence Rollover and Their Effect on Electronic Stability Controllers," Master's thesis, Auburn University, 2007.

[10] Rajamani, R., *Vehicle Dynamics and Control,* Springer, 2006.

[11] Whitehead, R. J., "A Study of the Properties That Influence Vehicle Rollover Propensity," Master's thesis, Auburn University, 2005.

[12] Whitehead, R., et al., "A Study of the Effect of Various Vehicle Properties on Rollover Propensity," SAE International, No. 2004-01-2094, Auburn University, 2004.

[13] Solmaz, S., M. Akar, and R. Shorten, "Online Center of Gravity Estimation in Automotive Vehicles using Multiple Models and Switching," *International Conference on Control, Automation, Robotics, and Vision*, 2006.

[14] Solmaz, S., et al., "Realtime Multiple-Model Estimation of Center of Gravity Position in Automotive Vehicles," *Vehicle System Dynamics,* 2007.

[15] Shen, X., J. Dumpert, and S. Farritor, "Design and Control of Robotic Highway Safety Markers," *IEEE/ASME Transactions on Mechatronics,* Vol. 10, No. 5, October 2005.

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

4

Navigation Systems

William Travis

4.1 Introduction

The demand for accurate and reliable navigation for ground vehicles is proliferating as passenger vehicle control systems become more complex and as the market for autonomous capability continues to grow. In addition to vehicle state information, a reliance on a navigation solution will be required as vehicle stability control systems increase their intelligence beyond the current versions of electronic stability control (ESC), vehicle dynamic control (VDC), and so forth. Future intelligent and safety vehicle systems may use this information to provide better lane keeping capability, higher accuracy vehicle tracking, or enhanced driver assistance systems.

Control of unmanned ground vehicles (UGV) requires precise navigation information. The need for reliable navigation information will only increase as passenger vehicles continue to acquire UGV-like capabilities and as UGVs become more common in both military and civilian markets. An increase in the precision of the navigation solution can directly lead to a safer, more capable, and more robust autonomous system.

The goal of this chapter is to provide GPS- and IMU-based navigation algorithms capable of providing the basic information necessary to autonomously control a ground vehicle. Such information can include, and is not limited to, ground speed, direction of travel, position, orientation, and velocity components. The majority of the algorithms can be implemented using the standard NMEA RMC output from any GPS receiver. An RMC message contains the essential navigation data of time, speed over ground, course, latitude, and longitude. A standard IMU provides accelerations along and rotational rates about a three-axis coordinate system, but the algorithms shown will use a reduced IMU.

The following sections discuss several algorithms and their limitations when used on a ground vehicle. The discussion starts with a brief description of a Kalman filter, which is the base algorithm used in navigation state estimation. Two GPS/INS integration architectures are then discussed: loose and close coupling. The differences, advantages, and hindrances are presented. Two simple speed estimation algorithms follow; one incorporates accelerations and GPS speed measurements, and the other adds a wheel speed measurement while accounting for longitudinal wheel slip. A heading estimation algorithm uses a yaw rate gyroscope in conjunction with the RMC course measurement. A full, loosely coupled navigation algorithm is described that produces estimates of vehicle speed, heading, and position using a reduced IMU, an odometer, and data in the RMC message. A brief discussion of modeling deficiencies in the presence of vehicle sideslip urges the system designer to know the limitations of the designed system. An algorithm incorporating a dual antenna GPS system capable of estimating lateral vehicle motion is described. Finally, a closely coupled algorithm is described that incorporates a full IMU and GPS range measurements to produce three dimensional position, velocity, and attitude states.

Readers interested in learning more about navigation systems and their various applications are referred to [1–7]. Specifically, [1–3] focus on multisensor fusion for navigation applications, and [3] provides a detailed overview of sensor operation and error sources. GPS is the main focus in [4–7], where the GPS system, receiver operation, and error sources are described in detail.

4.2 Kalman Filter

The Kalman filter is an unbiased, minimum variance estimator that is the backbone of many navigation systems. The use of this estimator is a proven method for blending measurements to minimize various sensor deficiencies while utilizing the strengths of each sensor by statistically weighting each measurement. For instance, an integrated accelerometer can provide a high-rate velocity measurement, but with poor short-term stability. A GPS-based velocity measurement has a low output rate, but excellent long-term stabil-

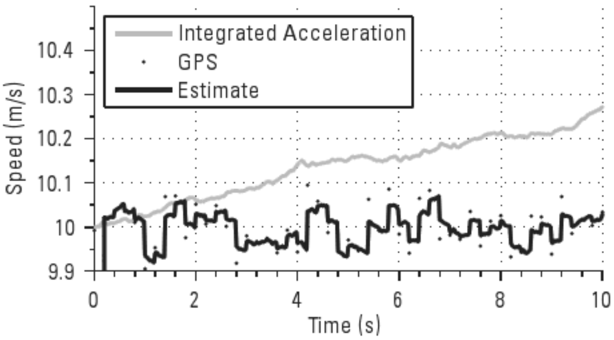


Figure 4.1 An integrated accelerometer (gray) provides a high-rate speed measurement (50 Hz), but quickly drifts off of truth. GPS provides a drift-free speed measurement (dotted line), but at a low rate (5 Hz). A Kalman filter was used to calculate a high-rate, drift-free speed estimate (black).

ity. The two measurements can be combined in a Kalman filter to produce a stable, high rate velocity estimate. This scenario is depicted in Figure 4.1, where the integrated accelerometer quickly drifts from the true value, but the Kalman filter output is a high rate and drift free estimate of speed.

A derivation of least squares methods and the Kalman filter are provided in the Appendix. If more detail is desired, consult the references given in the Appendix.

4.3 GPS/INS Integration Architectures

Two integration architectures are discussed in this chapter. More options exist [4], but for practical purposes and ease of implementation the discussion is limited to loosely coupled and closely coupled GPS/INS integration methods.

4.3.1 Loose Coupling

A loosely coupled GPS/INS integration routine combines the inertial measurements with position and velocity measurements calculated by the GPS receiver. Feedback to the inertial processor calibrates the IMU to remove effects from biases, scale factors, and/or misalignment.

Some benefits to adopting this strategy:

The measurement vector remains a small and constant size because measurements are position and velocity from the GPS receiver. This reduces some complexity in implementation, and the computational burden is low due to the relatively small size of the measurement vector and corresponding measurement covariance matrix.

The routine is easily adaptable to partial IMU integration. A full IMU is not necessary for some applications, so the designer can reduce both financial and computational cost by using the necessary number of accelerometers and/or rate gyroscopes.

Some drawbacks of this strategy are that:

The measurement quality is dependent upon the GPS receiver. The GPS navigation solution can contain large jumps when a satellite comes in or out of view or when multipath is present in the signal. The navigation designer has limited options available to mitigate these effects.

A minimum of four GPS satellites must be in view to solve for the X, Y, Z, and time components of a position solution. A solution might not be provided when fewer than four satellites are visible, and information from the visible satellites is not utilized.

4.3.2 Close Coupling

A closely coupled GPS/INS integration routine combines the inertial measurements with range information to the GPS satellites provided by the GPS receiver. Feedback to the inertial processor is retained, but there is no feedback to the GPS receiver (a tightly coupled architecture uses the navigation solution to aid the GPS receiver's tracking loops, but a specialized receiver is necessary and therefore is not further discussed). Range information includes a user to satellite distance and user to satellite velocity. Range information is given by the pseudorange or carrier phase measurement, and the range rate information is a function of the Doppler frequency or carrier phase measurement.

Benefits of a close coupling architecture are that:

The range measurements are processed individually, so the user has more influence on their impact on the navigation solution. The measurements can be individually weighted, so suspected erroneous measurements can be deweighted or thrown out altogether. This leads to a smoother solution, and when combined with inertial data, a solution that is much less apt to contain large deviations.

The navigation system can utilize range information when three or fewer satellites are visible. A GPS receiver might not provide position and velocity data in such a scenario, but the user position is still partially constrained by the available range information.

Shortcomings of this method:

The measurement vector will fluctuate in size as satellites come in and out of view, so appropriate logical steps need to be installed.

- The measurement vector can be relatively large, which increases the computational requirements. If 12 satellites are visible, then 12 range and 12 range rate measurements will be available; the measurement vector will be 24 × 1 and measurement covariance matrix will be 24 × 24. The user can, however, reduce the number of measurements by using highest quality information that places the best geometrical constraint on the solution.

4.4 Speed Estimation

An accurate measurement of vehicle speed is required to control both the vehicle's speed and position. A good controller requires a quality measurement at a sufficient rate. A designer can have several options when it comes to measuring vehicle speed, but the following discussion will focus on three specific sensors used to obtain a speed measurement: GPS, an accelerometer, and an odometer.

Integration of an onboard accelerometer can provide a high update rate speed measurement, provided an initial speed is well known. Unfortunately, integration of the errors present in the accelerometer output, such as scale factor, bias, and noise, will corrupt the measurement. Often the scale factor can be calibrated, but the accelerometer bias is time varying. The velocity measurement will drift off if the accelerometer is integrated of the over a period of time.

Even the cheapest GPS receivers can provide a quality speed measurement with an error standard deviation of less than 5 cm/s. High-cost receivers can provide slightly improved accuracy and update rates of 10, 20, or even

50 Hz. However, the output rate of a low cost receiver is usually between 1 to 5 Hz, and can be insufficient for control purposes or for other subsystems that might rely on the navigation solution.

A mathematical model of the system must be derived in order to implement a navigation system. Typically, there are two types of models: parameter based and kinematic. A parameter based model is a set of differential equations using physical information about the system. For a parameter based navigation system model of a UGV, the UGV's mass, wheelbase, track width, weight split, and tire cornering stiffness would be incorporated into the navigation system. The shortcoming to this type of model is the lack of certainty and variation in some model parameters. A kinematic model relies solely on kinematic relationships between the sensors and is independent of physical parameters. This improves the robustness of the navigation system, but sacrifices motion constraints of the vehicle.

4.4.1 Accelerometer and GPS

The first estimator example combines a GPS speed measurement with an integrated accelerometer measurement. A simple model of an accelerometer output contains the time rate of change of speed, $\dot{v}$, a moving bias, b, and zero mean, white noise, ω_a ($\omega_a \quad N(0\sigma_a^2)$).

$$a = \dot{v} + b + \omega_a \tag{4.1}$$

The key assumptions of the above model are the vehicle is traveling in a straight line on level ground and the accelerometer is located at the vehicle's center of gravity. Effects due to road grade, vehicle pitch, and centripetal acceleration are neglected. From this equation, a dynamic model for the vehicle's velocity can be constructed.

$$\dot{v} = a_x \quad b \quad \omega_a \tag{4.2}$$

The bias can be modeled as integrated zero mean, white noise (ω_b $N(0,\sigma_b^2)$) to capture its motion.

$$\dot{b} = \omega_b \tag{4.3}$$

A continuous state space representation of the dynamic model is shown here.

$$\dot{x} = \begin{matrix} 0 & 1 \\ 0 & 0 \end{matrix} x + \begin{matrix} 1 \\ 0 \end{matrix} u + \begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix} \omega \tag{4.4}$$

The state vector is defined as $x = [v \;\; b]^T$, the input u is the accelerometer output, and the process noise vector is defined as $\omega = [\omega_a \;\; \omega_b]^T$. The continuous process noise matrix for this system is calculated by evaluating (A.3).

$$Q = \begin{matrix} \sigma_a^2 & 0 \\ 0 & \sigma_b^2 \end{matrix} \tag{4.5}$$

Initial values for σ_a and σ_b are often listed in the sensor specification sheet that is provided by the manufacturer. These values can be altered, or "hand tuned," to adjust estimator performance for custom applications. Often, the accelerometer will record unwanted vibrations in the system, but the value for σ_a can be inflated to account for the additional process noise.

The above system must be discretized for use in a discrete linear Kalman filter. The sample rate, t, is chosen based on the update rate of the input, which is the accelerometer in this instance. This will also be the rate at which estimates of the state vector are available. If no input is used in a Kalman filter, the sample rate can be chosen at the discretion of the system designer. The discrete state matrices and the discrete process noise matrix for this system are as follows:

$$= \begin{matrix} 1 & t \\ 0 & 1 \end{matrix} \tag{4.6}$$

$$= \begin{matrix} \tau \\ 0 \end{matrix} \tag{4.7}$$

$$Q_d = \begin{matrix} t\sigma_a^2 + \frac{1}{3} \;\; t^3\sigma_b^2 & \frac{1}{2} \;\; t^2\sigma_b^2 \\ \frac{1}{2} \;\; t^2\sigma_b^2 & t\sigma_b^2 \end{matrix} \tag{4.8}$$

The GPS speed measurement is modeled at zero mean, white noise (ν_b $N(0,\sigma_v^2)$ added to the true speed.

$$z = v + \nu \tag{4.9}$$

The measurement matrix varies as a GPS measurement is received. When a measurement is available, the update can be calculated with the following matrix:

$$H = [1 \quad 0] \tag{4.10}$$

When a GPS measurement is not available, the measurement matrix is simply a row of zeros. Alternatively, the Kalman gain calculation and measurement update can be skipped.

Next, the measurement covariance matrix must be constructed. The measurement covariance matrix is clear-cut for this example as there is only one measurement available at each epoch. Therefore, it is equal to the variance of the noise on the GPS speed measurement.

$$R = \ \sigma_v^2 \tag{4.11}$$

A GPS speed measurement has a standard deviation of below 5 cm/s. Even the lowest-cost receivers can perform at this level. Some higher-end receivers have quoted performance levels down to 1 cm/s.

Since the measurement in this application is inherently discrete, there is no need for discretization. Equations (4.6) through (4.8), (4.10), and (4.11) can be used in a linear discrete Kalman filter to estimate speed and accelerometer bias given measurements of acceleration and GPS speed.

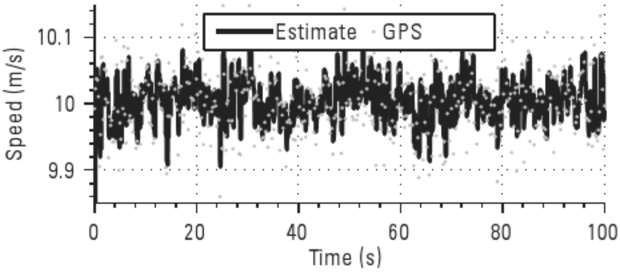


Figure 4.2 A 50-Hz estimated speed is indicated by the line, and a 5-Hz GPS speed measurement is indicated by the dots. The accuracy of the estimated speed is superior to that of the GPS speed.

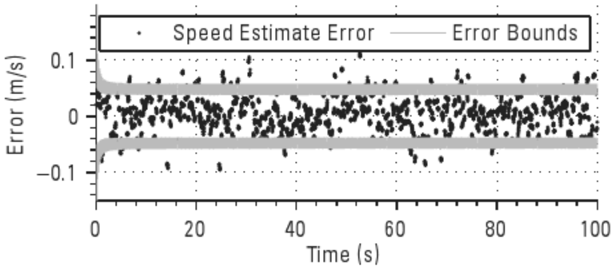


Figure 4.3 The speed measurement residual (dots) appears to be a zero mean Gaussian signal.

The following figures display the estimator output when simulated accelerometer and GPS speed measurements were run through a Kalman filter. Figure 4.2 shows the estimated speed and GPS speed. The true speed during the simulation was a constant 10 m/s. The Kalman filter produced an estimated that accurately tracked the true speed. The estimated speed is output at the same rate as the accelerometer (50 Hz), but without the drift. Also, the variance of the estimated speed is slightly lower than that of the GPS speed measurement.

Figure 4.3 displays the measurement residual for the speed state. The error appears to be zero mean white noise; no noticeable frequency or biases are present. The residual is bounded by the estimated standard deviation of the speed state, where $\sigma_{\hat{x}_1} = \sqrt{P_{11}}$.

Closer examination of the error bounds reveals a saw tooth shape, as depicted in Figure 4.4. This shape is created by the inflation of the estimated

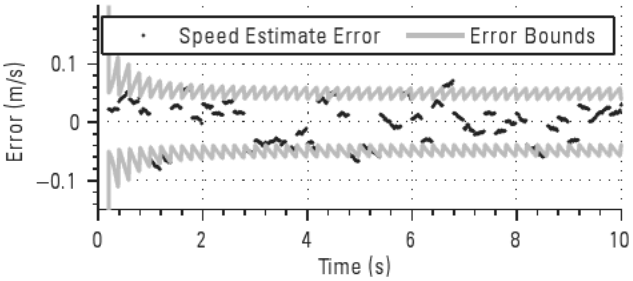


Figure 4.4 The estimated state covariance increases when GPS measurements are not available, but decreases when a new measurement arrives. This creates a saw tooth pattern in the error bounds as the estimator is propagated at a higher rate than which measurements arrive.

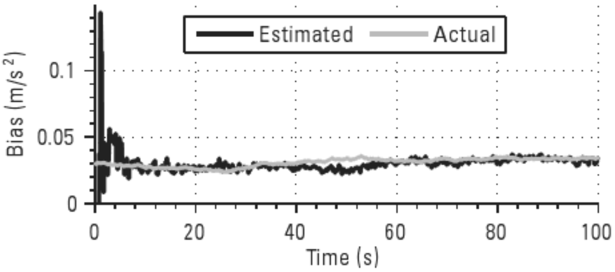


Figure 4.5 The estimated sensor bias (black) is compared to the simulated sensor bias (gray).

state covariance when a GPS measurement is unavailable. The accelerometer is used to propagate the estimate at a 50-Hz rate, but GPS measurements are only available at 5 Hz. Propagation of a state through time perpetuates a loss in confidence of the accuracy of the state. A measurement provides a means of verifying and updating the estimate, and therefore confidence is regained. Therefore in nine out of ten iterations, confidence in the measurement slowly decreases as the accelerometer is integrated. Confidence is regained after the estimate is updated with a new GPS measurement.

The estimated bias is shown in Figure 4.5. Since the data was simulated, truth is known. The bias estimate takes approximately 5 seconds to converge to the true value. The estimated bias accurately tracks the true bias, even as it slowly changes with time.

In practice, a visual confirmation of a correct bias estimate is not available. The designer must resort to other means to verify that the bias estimate is indeed correct. If the bias estimate is incorrect, the error will manifest itself in other states. The measurement residuals should be checked to ensure their

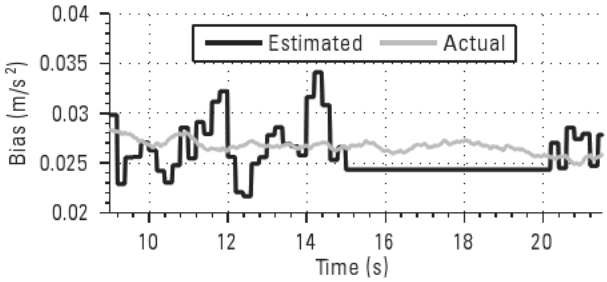


Figure 4.6 The bias estimate is constant during the GPS outage from 15 to 20 seconds. During this time, it is unobservable.

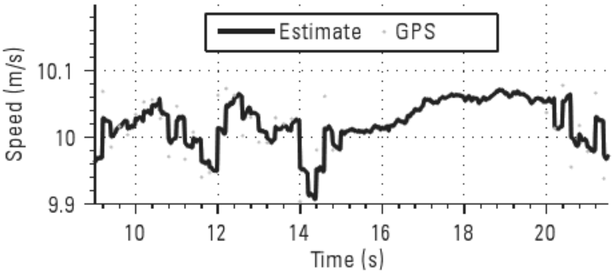


Figure 4.7 The speed estimate exhibits slight drift during the short, 5-second GPS outage.

content is Gaussian. Analysis of the measurement obtained by integrating the accelerometer with instantaneous values of the bias estimate subtracted out also indicates the quality of the bias estimate. Artificial removal of the GPS measurement can also validate the bias estimate. If the bias estimate is correct, the speed estimate will exhibit minimal drift through the outage. An incorrect bias estimate will cause the speed estimate error to grow quickly.

The covariance matrices should be adjusted if poor performance is determined. Values of a sensor's drift rate provided by the manufacturer are a good starting point when tuning the Kalman filter. These values should be altered until the residuals are white and acceptable performance is achieved when GPS is removed for short periods of time.

The GPS speed measurement was artificially removed between 15 and 20 seconds to demonstrate the estimator's dead reckoning performance. The accelerometer bias is unobservable during the GPS outage, and its estimate

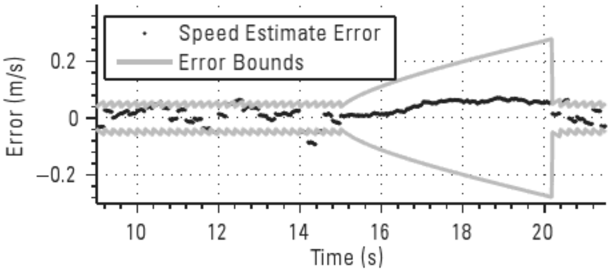


Figure 4.8 The variance of the speed estimate error increases when GPS is unavailable to indicate a loss of certainty in the estimate.

is predicted only with the dynamic model. Figure 4.6 plots the bias estimate. Notice the estimate remains constant during the GPS outage. The Kalman filter is unable to predict any variation in the accelerometer bias. The bias estimate remains reasonably accurate because of the minimal variation over the period of the outage. However, the bias estimate error could get quite large during an extended outage and corrupt other states in the filter.

The accuracy of the speed estimate during the outage is acceptable for many applications. Slight drift is noticeable in Figure 4.7 and is due to the small error in the bias estimate.

As expected, the estimated error variance grows during the GPS outage due to the integration of sensor noise during the time update. The error standard deviation is approximately 5 cm/s at the start of the outage but quickly grows to 25 cm/s in 5 seconds, as seen in Figure 4.8.

A trick the navigation system designer can implement to assist the filter is to input a zero velocity measurement with a low variance when the vehicle is stationary. The zero velocity provides a means to accurately estimate the sensor bias as measurement noise has been eliminated. There must be high confidence that the vehicle is indeed not moving before the speed measurement is set; otherwise small motions cannot be captured by the estimator. The designer should be careful not to divide by velocity as well.

4.4.2 Accelerometer, GPS, and Wheel Speed

The output of an odometer, such as rotary encoders or Hall effect sensors, can be differentiated to provide a speed measurement. The measurement update rate is typically sufficient for navigation, but the output can be noisy due to the differentiation of a quantized signal. Also, changes in wheel radius or longitudinal wheel slip can corrupt the speed measurement. These effects could be neglected (with caution) on a passenger vehicle in a benign highway environment in good weather. However, off-road applications are more conducive to the generation of wheel slip and radius changes due to terrain and typically softer tires.

The following example constructs a navigation system using measurements of acceleration, GPS speed, and wheel speed. The assumptions of the previous model are still in place. The wheel radius will be assumed to be constant, but tire slip will be anticipated. Therefore, the navigation system will be designed to handle intermittent slippage and estimate the amount of slip occurring.

Longitudinal tire slip is expressed as a percentage, and defined as the difference between the wheel velocity and true velocity divided by the true velocity.

$$s = \frac{r\dot{\theta}_w \quad v}{v} \tag{4.12}$$

The true velocity is denoted by v, the wheel radius is r, the rotational wheel speed is $\dot{\theta}_{ws}$, and the percentage of longitudinal slip is s. Simple manipulation of the equation yields and expression for the output of a wheel speed sensor.

$$v_{ws} = r\dot{\theta}_{ws} = (1+s)v \tag{4.13}$$

Knowledge of the slip dynamics must be known if the slip is to be estimated. Approximating the wheel slip as integrated white noise would be a simple starting point. A more complicated model, possibly incorporating applied torque, could be pursued if the simple approximation proves to be relatively successful. This example will focus on the simple case, and model the slip with zero mean white noise ($\omega_s = N(0,\sigma_s^2)$) as follows:

$$\dot{s} = \omega_s \tag{4.14}$$

The previous example can now be incorporated. Assuming a state vector of = $[v \;\; b \;\; s]^T$, an accelerometer for an input, and a process noise vector of ω = $[\omega_a \;\; \omega_b \;\; \omega_s]^T$, a continuous state space representation can be developed for the linear dynamic models.

$$\dot{x} = \begin{matrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix} \; x + \begin{matrix} 1 \\ 0 \\ 0 \end{matrix} \; u + \begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix} \; \omega \tag{4.15}$$

The continuous process covariance matrix contains the independent variances along the diagonal.

$$Q = \begin{matrix} \sigma_a^2 & 0 & 0 \\ 0 & \sigma_b^2 & 0 \\ 0 & 0 & \sigma_s^2 \end{matrix} \tag{4.16}$$

Discretization yields the following:

$$= \begin{matrix} 1 & t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix} \tag{4.17}$$

$$= \begin{matrix} t \\ 0 \\ 0 \end{matrix} \tag{4.18}$$

$$Q_d = \begin{matrix} t\sigma_a^2 + \frac{1}{3} \; t^3\sigma_b^2 & \frac{1}{2} \; t^2\sigma_b^2 & 0 \\ \frac{1}{2} \; t^2\sigma_b^2 & t\sigma_b^2 & 0 \\ 0 & 0 & t\sigma_s^2 \end{matrix} \tag{4.19}$$

The measurement vector is constructed using the same GPS speed model and the wheel speed model given by (4.9).

$$z = \begin{matrix} v \\ (1+s)v \end{matrix} + \begin{matrix} v_{GPS} \\ v_{ws} \end{matrix} \tag{4.20}$$

The measurement model for wheel speed contains two states multiplied together. This is a nonlinear model; therefore the extended Kalman filter must be used. A measurement matrix can be calculated by linearizing the model with respect to the state vector.

$$H = \begin{matrix} 1 & 0 & 0 \\ 1+s & 0 & v \end{matrix} \tag{4.21}$$

The measurement covariance contains the individual measurement variances along the diagonal with zeros in the off diagonal elements.

$$R = \begin{matrix} \sigma_{v_{GPS}}^2 & 0 \\ 0 & \sigma_{v_{ws}}^2 \end{matrix} \tag{4.22}$$

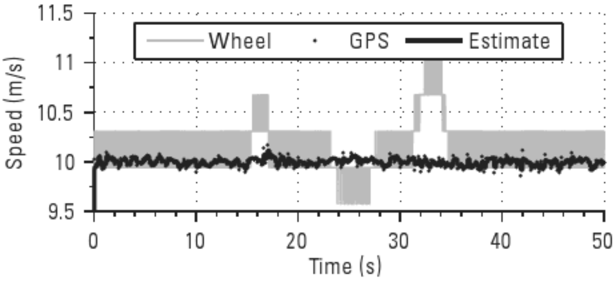


Figure 4.9 The wheel speed (gray), GPS speed (dots), and speed estimate (black) are depicted.

The differentiated encoder noise can be approximated as Gaussian in many cases, especially when the encoder resolution is high.

The simulation of the previous example was regenerated with the augmented matrices and additional wheel speed measurement. Figure 4.9 shows the speed estimate with the two speed measurements. The longitudinal slip causes the wheel speed measurement to deviate from the true speed. However, the error does not propagate through to the speed estimate because the slip is also estimated.

Figure 4.10 is a close-up view of Figure 4.9. The presence of slip is clear, but the magnitude is hard to discern because of the low resolution of the wheel speed measurement.

The slip estimate tracks the true wheel slip, as depicted in Figure 4.11. Shown in the picture is the true wheel slip in gray, the estimated wheel slip in black, and a measurement of wheel slip in black dots. The measurement was constructed using the odometer and GPS, and therefore is only available at

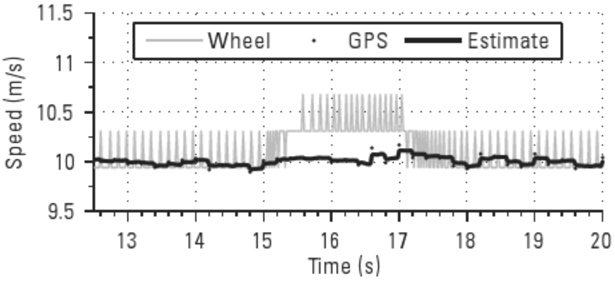


Figure 4.10 A closer look reveals the quantized nature of the wheel speed measurement.

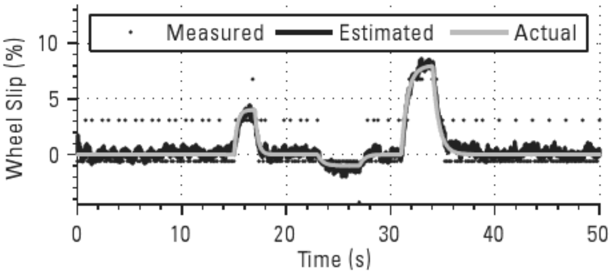


Figure 4.11 The actual slip is shown in gray, the estimated slip is shown in black, and a measured value of slip using the wheel speed and GPS is shown in dots. The estimator tracks the wheel slip adequately for many applications and is superior to the slip measurement. The resolution of the slip measurement is poor and its update rate is slow.

5 Hz. The resolution of the measurement is severely limited by the quantization of the wheel angle measured by the odometer. The slip estimate, on the other hand, has a much higher resolution and is available at 50 Hz. The accuracy of the estimate is suitable for some applications, but systems requiring less noise would have to configure a different model.

The bias estimate, plotted in Figure 4.12 with the actual bias, is slightly improved by the availability of an additional 50-Hz speed measurement. If GPS is lost, the bias will still be observable provided the wheel does not slip. Any slip of the wheel will corrupt the wheel slip estimate and bias estimate because the two signals are indistinguishable without a GPS speed measurement. The error cannot be attributed to wheel slip or a change in bias with any certainty.

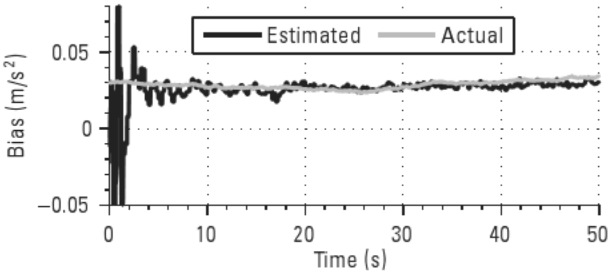


Figure 4.12 The bias estimate is slightly improved due to the additional 50-Hz speed measurement.

4.5 Heading Estimation

The RMC message contains a course or track over ground measurement. Integration of a yaw rate gyro provides a measurement of vehicle heading, if properly initialized. An equivalency between these two measurements exists if and only if ground vehicle exhibits no lateral motion. Any amount of lateral motion will inject a discrepancy between these two measurements. Vehicle heading is the pointing direction of the vehicle, while vehicle course is the direction of travel. The difference between vehicle heading and course is known as the vehicle sideslip.

Under the assumption of zero sideslip, the GPS RMC course measurement can be used to calibrate a yaw rate gyroscope and provide a high rate vehicle heading estimate. The estimation model is similar to the speed estimation model. A yaw rate gyroscope provides a measurement of the true turning rate with an additive moving bias and zero mean, white noise. The true yaw rate and bias can be modeled as follows:

$$\dot{\psi} = r \quad b_g \quad \omega_g \tag{4.23}$$

$$\dot{b}_g = \omega_b \tag{4.24}$$

In the above expressions, $\dot{\psi}$ is the true yaw rate of the vehicle, r is the measured yaw rate, b_g is the yaw rate gyroscope bias, $\omega_g(\omega_g \;\; N(0,\sigma_g^2))$ is the yaw rate gyro noise, and $\omega_b(\omega_b \;\; N(0,\sigma_b^2))$ is the noise driving the bias drift. Assuming a state vector in the form $x = [\psi \;\; b_g]^T$, process noise vector $\omega = [\omega_g \;\; \omega_b]^T$, and input $u = r$, the state space representation can be formulated like the first speed estimation example.

$$\dot{x} = \begin{matrix} 0 & 1 \\ 0 & 0 \end{matrix} \; x + \begin{matrix} 1 \\ 0 \end{matrix} \; u + \begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix} \; \omega \tag{4.25}$$

$$Q = \begin{matrix} \sigma_g^2 & 0 \\ 0 & \sigma_b^2 \end{matrix} \tag{4.26}$$

$$= \begin{matrix} 1 & t \\ 0 & 1 \end{matrix} \tag{4.27}$$

$$= \begin{matrix} t \\ 0 \end{matrix} \tag{4.28}$$

$$Q_d = \begin{matrix} t\sigma_g^2 + \frac{1}{3} \quad t^3\sigma_b^2 & \frac{1}{2} \quad t^2\sigma_b^2 \\ \frac{1}{2} \quad t^2\sigma_b^2 & t\sigma_b^2 \end{matrix} \tag{4.29}$$

The measurement model for GPS course consists of the true heading plus zero mean white noise ($\upsilon = N(0,\sigma_\psi^2)$). This model assumes sideslip is zero.

$$\psi_{GPS} = \psi + \upsilon \tag{4.30}$$

The noise amplitude of a GPS track over ground measurement is speed-dependent due to the nature by which it is generated. The course is based off the direction of travel; therefore, a course does not exist for a stationary object. The course is better defined as the speed of the object increases. The standard deviation of a GPS course measurement is inversely proportional to the speed over ground at the antenna, as seen in Figure 4.13.

The accuracy metric is defined below, where $\sigma_{v_{GPS}}$ is the GPS speed accuracy.

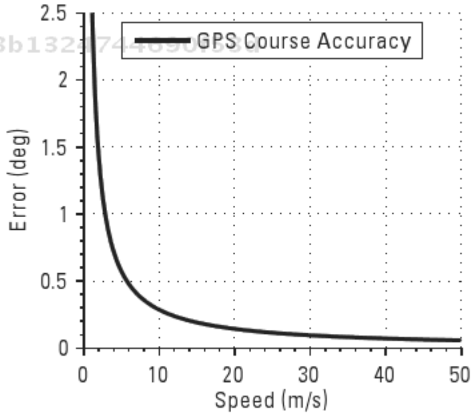


Figure 4.13 GPS course accuracy is inversely proportional to the speed over ground.

$$\sigma_{\psi_{GPS}} = \frac{\sigma_{v_{GPS}}}{v} \tag{4.31}$$

Therefore, the vehicle's speed must be known to compute the GPS course accuracy, and the measurement covariance must be updated each iteration. In practice, a logical step based on a speed threshold can be implemented to prevent a divide-by-zero or unnecessarily large covariance. The measurement covariance consists of the only the GPS course variance (this example assumes that knowledge of speed is readily available, so only the heading and yaw gyro bias are estimated).

$$R = \sigma^2_{\psi_{GPS}} \tag{4.32}$$

Integration of the rate gyroscope yields an unbounded measurement of heading. Heading measurements would increment by 2π after each complete revolution about some fixed point. The GPS course measurement, however, is bounded between 0 and 2π. This discrepancy must be addressed to correctly determine the error between the two measurements. The designer can either bound the integrated rate measurement (and estimated state) or unbound the GPS course measurement, neither of which is overly tedious.

Removing the bound on the GPS course measurement is a single step installed in the filter. An incremental counter tracks the number of complete clockwise or counterclockwise revolutions. The estimate can be bounded after processing for analysis or use in other subsystems.

Three steps must be added in the filter if the GPS course measurement is not unbounded, but the estimate is ready for immediate analysis or further processing. The estimate must be checked and altered after the time update and after the measurement update to keep it within 0 to 2π or $\pm\pi$. The measurement residual must also be placed within the desired boundary.

Real data was processed in a linear discrete Kalman filter to produce a vehicle heading estimate using the algorithm described above. The vehicle started from rest, drove in a counterclockwise manner, and returned to rest after a single revolution. Figure 4.14 shows the heading estimate, the GPS course measurement, and the integrated yaw rate gyro output after initialization.

The GPS course measurement is only noise at the beginning and end of the run because the vehicle is stationary. The heading estimate cannot be

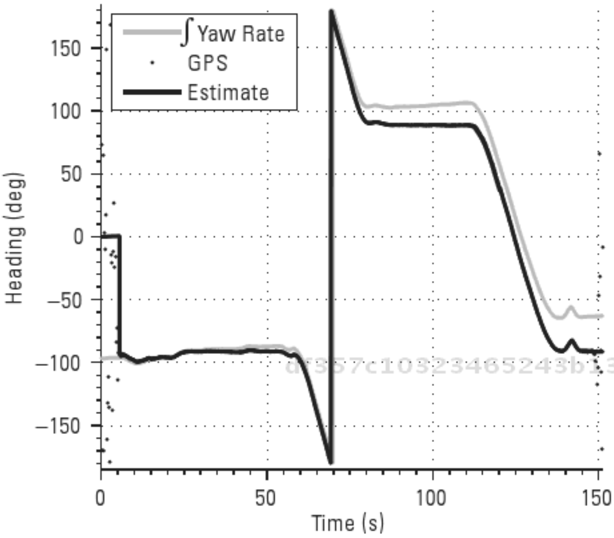


Figure 4.14 The heading estimate (solid line) is compared to the GPS course measurement (dots) and an integrated yaw rate gyroscope (dashed line).

properly initialized until the vehicle moves. The estimate drifts from the user initialized value due to the integration of sensor bias and noise. A similar situation is encountered if the vehicle comes to rest.

Caution must be taken if the estimator is allowed to run during long periods of stationary activity. The estimate error covariance will grow during successive iterations of the time propagation and can reach unreasonably large values. A logical step can be installed to maintain the estimate

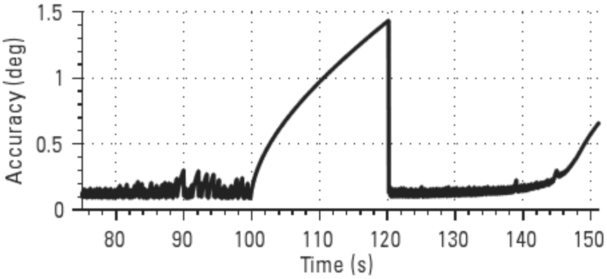


Figure 4.15 The heading accuracy estimate degrades during a GPS outage to account for the integration of noise and a drifting bias.

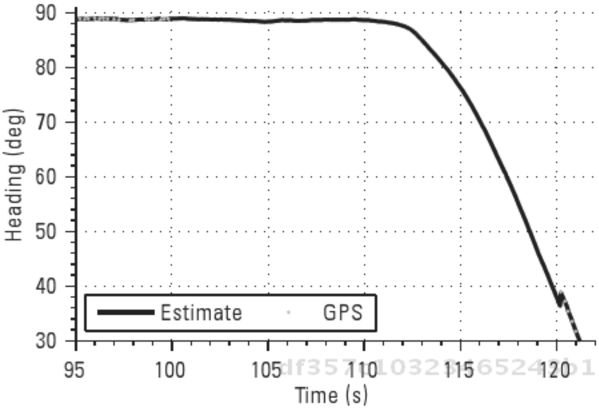


Figure 4.16 The heading estimate slowly drifts from the true value during the 20-second GPS outage.

integrity during motionless periods if the heading estimate has been initialized (i.e., the vehicle has been in motion previously). The last accurate measurement or estimate can be fed into the Kalman filter as a measurement since the GPS course measurement is unreliable, and bias observability will be retained.

The heading estimate is able to maintain some precision if GPS is lost if the bias is correctly estimated. GPS was artificially removed from 100 to 120 seconds. The estimator integrated the yaw rate measurement with the estimated bias removed during this time. The covariance inflates as confidence in the heading estimate is lost, as seen in Figure 4.15. Figure 4.16 shows the heading estimate during the outage.

4.6 Position, Speed, and Heading Estimation

The NMEA RMC message contains all measurements necessary for basic navigation: time, course, speed, and position. Previous sections discussed the integration of a partial IMU with components of the RMC message. This section will develop navigation system equations that combine a partial IMU with GPS course, speed, and position. The position measurement provides additional information about the vehicle's velocity, so the overall navigation system robustness is improved.

Table 4.1
WGS84 Parameters

Equatorial radius	a	6,378,137 meters
Reciprocal flattening	$1/f$	298.257223563
Eccentricity	e	$\sqrt{f(2\quad f)}$
Normal radius	R_N	$R_N = \dfrac{a}{\sqrt{1\quad e^2 \sin^2 \phi}}$

4.6.1 Coordinate Conversion

The units of latitude and longitude, as output from an NMEA RMC message, are in degrees. It is often beneficial to express position in a local coordinate frame to reduce the complexity of the Kalman filter matrices. The following is a brief overview of the equations necessary to convert geodetic latitude, longitude to a local coordinate frame, where position is expressed as a distance to a defined reference point.

The first step requires converting the geodetic coordinates expressed in latitude, ϕ, longitude, λ, and altitude above the WGS84 ellipsoid, h, to the X, Y, and Z coordinates of the Earth Centered Earth Fixed (ECEF) Cartesian frame. The conversion parameters are given in Table 4.1.

$$\begin{aligned} X &= (R_N + h)\cos\phi\cos\lambda \\ Y &= (R_N + h)\cos\phi\sin\lambda \\ Z &= (R_N(1\quad e^2) + h)\sin\phi \end{aligned} \tag{4.33}$$

A reference position is defined (denoted by the subscript o), and the difference between the vehicle position and reference position is computed after their respective conversion to ECEF coordinates.

$$\begin{aligned} X &= X\quad X_o \\ Y &= Y\quad Y_o \\ Z &= Z\quad Z_o \end{aligned} \tag{4.34}$$

A series of rotations based of the reference latitude and longitude produces a position in a local coordinate system meters east, north, and up from the chosen reference point.

$$
\begin{aligned}
\alpha &= 90 \quad \phi_o \\
\beta &= 90 + \lambda_o \\
E &= X \cos\beta \cos\alpha + Y \cos\beta \sin\alpha \quad Z \sin\beta \\
N &= \quad X \sin \quad + Y \cos \\
U &= X \sin\beta \cos\alpha + Y \sin\beta \cos\alpha + Z \cos\beta
\end{aligned}
\tag{4.35}
$$

4.6.2 Accelerometer, Yaw Rate Gyroscope, GPS, and Wheel Speed

This navigation system will incorporate aspects of the previously discussed models to produce a navigation solution. Specifically, longitudinal acceleration and yaw rate from an IMU will be blended with an odometer and GPS measurements contained in the NMEA RMC message. The configuration assumes a vehicle traveling on flat ground, zero slide slip, and zero wheel slip. The previously defined dynamic models will be used and are defined here for completeness.

$$\dot{v} = a_x \quad b \quad \omega_a \tag{4.36}$$

$$\dot{b}_a = \omega_{b_a} \tag{4.37}$$

$$\dot{\psi} = r \quad b_r \quad \omega_r \tag{4.38}$$

$$\dot{b}_r = \omega_{b_r} \tag{4.39}$$

The position is a function of speed and course, and it can be defined with the following nonlinear kinematic relationships under the zero side slip assumption:

$$\dot{N} = v \cos\psi \tag{4.40}$$

$$\dot{E} = v \sin\psi \tag{4.41}$$

A state vector can be constructed containing the necessary states to control a vehicle and IMU calibration states.

$$x = [v \quad b_a \quad \psi \quad b_r \quad N \quad E]^T \tag{4.42}$$

The process noise vector is defined as follows:

$$\omega = [\omega_a \quad \omega_{b_a} \quad \omega_r \quad \omega_{b_r} \quad 0 \quad 0]^T \tag{4.43}$$

Equations (4.36) through (4.41) comprise the dynamic model for this navigation system. Due to the nonlinearities present, an extended Kalman filter must be used. No discretization is required for an extended Kalman filter with a nonlinear dynamic model. The model is propagated via numerical integration. However, the equations must still be linearized about the latest estimate each iteration to produce the required information to propagate the estimate error covariance matrix.

$$F = \begin{matrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \cos\psi & 0 & v\sin\psi & 0 & 0 & 0 \\ \sin\psi & 0 & v\cos\psi & 0 & 0 & 0 \end{matrix} \tag{4.44}$$

$$L = \begin{matrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \tag{4.45}$$

The continuous process noise covariance matrix is defined as follows:

$$Q = \begin{matrix} \sigma_a^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{b_a}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_r^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{b_r}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \tag{4.46}$$

Measurements used will be GPS speed, wheel speed, course, north, and east.

$$z=[v \quad v_{ws} \quad \psi \quad N \quad E]^T \tag{4.47}$$

The GPS measurements will have direct relationships to the estimated states. The wheel speed measurement, obtained by differentiating an odometer with respect to time, will also have a direct relationship to speed under the zero wheel slip assumption. The measurement relationships can be linearly modeled in state space format.

$$z = \begin{matrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{matrix} \; x \tag{4.48}$$

All states are observable when GPS is available. The elements corresponding to the GPS measurements are set to zero when GPS is unavailable. Only the speed and accelerometer bias states remain observable due to the wheel speed measurement during those time periods.

Measurement noise is assumed to be uncorrelated between the measurements. The covariance matrix contains all the individual measurement variances along the diagonal. Note that it must be updated each iteration to recomputed the variance of the GPS course measurement.

$$R = \begin{matrix} \sigma_v^2 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{ws}^2 & 0 & 0 & 0 \\ 0 & 0 & \left(\sigma_v / v\right)^2 & 0 & 0 \\ 0 & 0 & 0 & \sigma_N^2 & 0 \\ 0 & 0 & 0 & 0 & \sigma_E^2 \end{matrix} \tag{4.49}$$

The matrices formulated in this section can be used in the extended Kalman filter. The same logical steps and tricks discussed in previous sections can be implemented in this filter.

Real data was processed in this navigation algorithm. Longitudinal acceleration and yaw rate were output from an IMU at 32 Hz, a NMEA RMC message containing GPS speed, course, and position was logged at 5 Hz, and a wheel speed measurement was available at 32 Hz. The vehicle was driven counterclockwise around a 2.75-km oval test track with 8° banked turns at speeds ranging from 0 to 27 m/s.

The speed and course data for the entire test are shown in Figures 4.17 and 4.18. Magnified plots are shown in Figure 4.19, where the Kalman filter output can be observed accurately tracking the vehicle's motion. The noise on the heading estimate is slightly reduced from the addition of the position measurements to the filter, resulting in a smoother estimate. It can also be seen that the 8° bank minimally affects the heading estimate.

The position estimate sufficiently tracks the vehicle around the test track. Figure 4.20 shows the position estimate around the entire test track, and Figure 4.21 shows the northeast corner of the test track. The top line is the position of the vehicle as it enters the test track, and the bottom line is the position of the vehicle as it exits turn four. It is easily seen that the estimate provides clean information about the vehicle's position between GPS measurements.

Analysis of the position covariance, shown in Figure 4.22, reveals information about the estimate accuracy as the vehicle travels in different directions. The straight sections of the track run east and west, and the east position error is considerably smaller than the north position error while the vehicle is on the straight sections. However, the north position error shrinks and the east position error grows as the vehicle makes a turn from west to

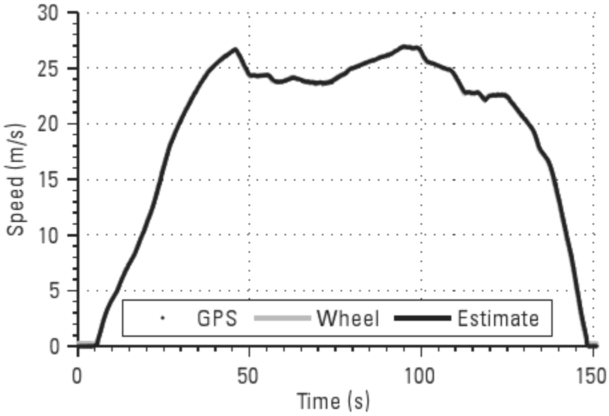


Figure 4.17 The speed profile of the experiment.

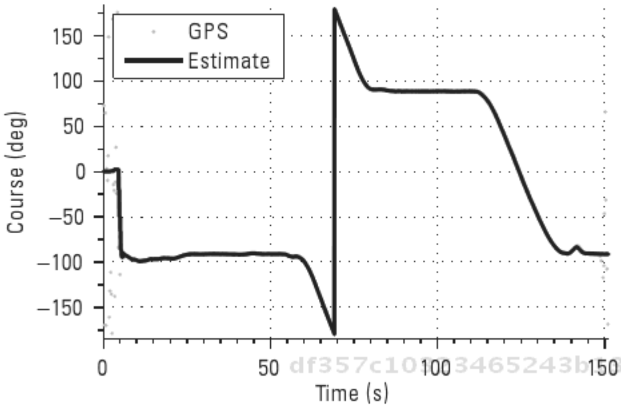


Figure 4.18 The vehicle heading driving counterclockwise around an oval track.

south and from east to north. The accuracy variation is due to the persistence of excitation in the different directions. As the vehicle travels along the straight section, the magnitude of the change in position in the east is much greater than the noise in the east direction. However, the change in the north position is effectively zero, so the noise in the north direction dominates that measurement. The correlation between the north and east position measurements captures the change in direction as well. The

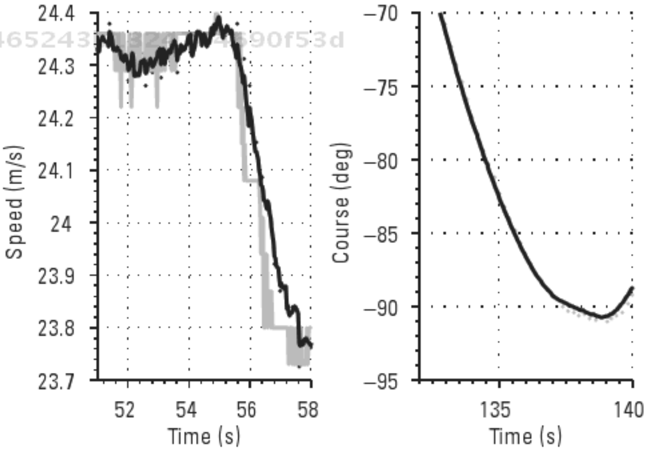


Figure 4.19 Magnification of the speed and heading data shows the estimates accurately tracking the vehicle's velocity.

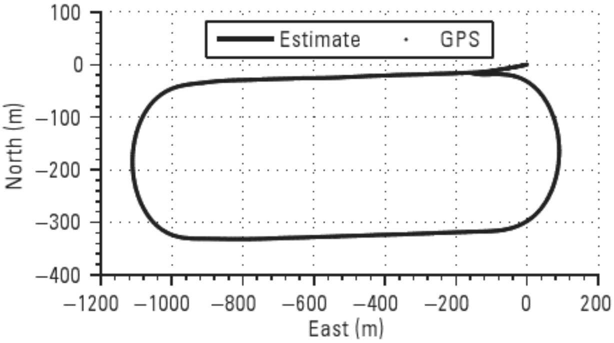


Figure 4.20 The position estimate outlines the 2.75-km test track.

estimates' correlation varies with course; maximum correlation is reached when the course is 45°, 135°, 225°, or 315°, and minimum correlation occurs when the course is 0°, 90°, 180°, or 270°.

GPS was artificially removed from 40 to 100 seconds to simulate a 1-minute outage. Figure 4.23 plots the GPS position estimate during the outage. The estimate is plotted with a thick line, and the GPS position is plotted with a thin line. The GPS measurements used in the Kalman filter are shown by the dots. Initially the estimate tracks the true position well. However, the rate gyroscope bias drifts enough during the outage to generate a significant position error of 60 meters at the end of the outage. The wheel speed measurement improved the performance during the GPS outage as the speed estimate remained observable.

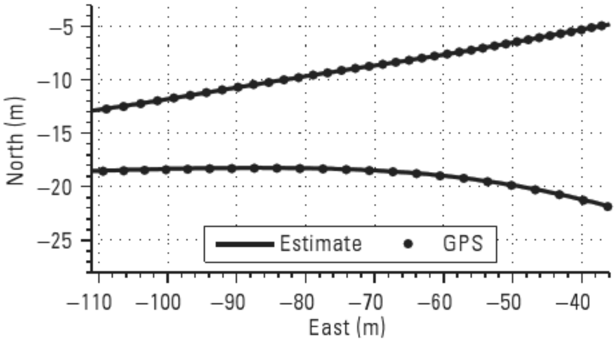


Figure 4.21 Zooming in on the northeast corner of the test track shows the position estimate bridging the gap between the low rate GPS measurements.

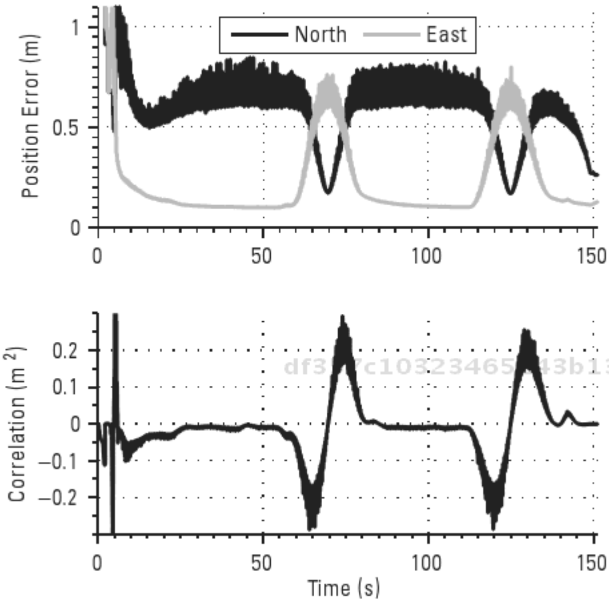


Figure 4.22 Variation in the position accuracies occurs due to the changing signal-to-noise ratio as the vehicle changes direction.

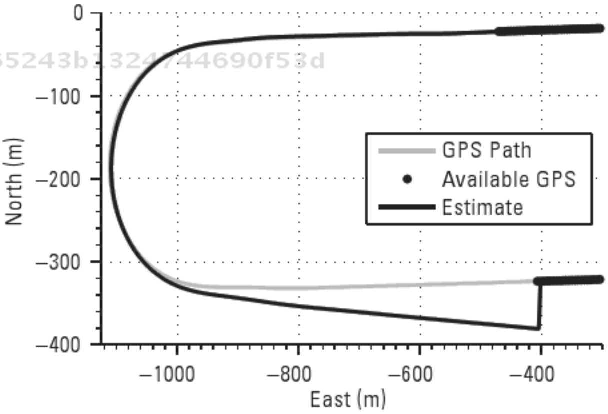


Figure 4.23 The position accuracy degrades with the length of the GPS outage. A 60-meter error exists at the end of the minute outage. The thin line indicates the GPS position, while the dots represent the GPS position used to generate the estimate.

4.7 Navigation in the Presence of Sideslip

Vehicle dynamicists often use models to provide detailed vehicle state information to the driver or a control system. These models are often intricate, requiring parameters that are hard or expensive to measure, and can be excessively complicated for navigational purposes. Navigation specialists sometimes use a simplified vehicle model based on kinematic relationships. However, these models sometimes neglect key states such as lateral velocity present in the actual dynamics of the ground vehicle operating at normal speeds.

It is important for the designer to be familiar with the operating platform's dynamics in its operating environment and to know the limitations of the navigation system implemented on that platform. The following sections discuss the generation of vehicle sideslip and then present a navigation algorithm that utilizes a dual-antenna GPS receiver to incorporate sideslip into the model.

4.7.1 Generation of Sideslip

An explanation of how sideslip occurs is necessary to understand the errors that can potentially arise in a navigation solution. The tire is the vehicle's interface with the road which transfers all drive forces to the ground and generating the required lateral forces to turn the vehicle. Using the explanation presented in [8], as the tire rolls, parts of the tread either adhere to the road or slide. When a force from the vehicle is applied to the tire/road interface, the static coefficient of friction is exceeded in local areas on the tread. Areas where the friction coefficient is surpassed begin to slide, which generates a force to push the vehicle. The generated force is approximated as linear, proportional to slip (α) until the tire saturates. When saturation occurs, the force remains constant as no more lateral or longitudinal force can be applied to direct the vehicle. The linear increase and saturation can be seen in the tire curve shown in Figure 4.24.

The tire cornering stiffness (C_α) is a property that indicates a tire's performance when operating within its linear region (i.e., the tire is not approaching saturation). The cornering stiffness and peak tire force are functions of numerous factors such as terrain, normal force, toe, camber, road conditions, tire materials, and tread pattern, and values for these parameters are rarely known exactly due to the highly complex relationship between these factors. Approximations with look-up tables or empirical models, such as Pacjeka's Magic Model or the Dugoff Tire Model, are often used to sufficiently char-

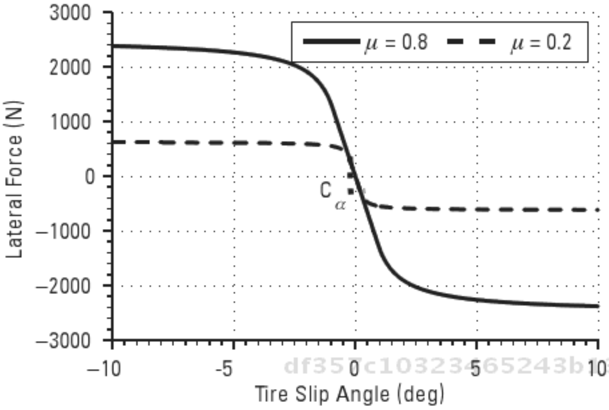


Figure 4.24 Two tire curves with the same cornering stiffness but differing friction coefficients show the available lateral force when a vehicle is on pavement and ice.

acterize the tire [9, 10]. Ultimately, the available vehicle control forces in the linear region are a function of the tire stiffness and lateral slip at the tire.

$$\alpha = \tan^{\ 1}\ \frac{v_{yT}}{v_{xT}} \div \tag{4.50}$$

The lateral slip at each tire can be translated to the vehicle's center of gravity (CG) to obtain overall vehicle slip, more commonly known as vehicle sideslip (β). Sideslip is an angle defined as the difference between the vehicle's course (ν) and its heading (ψ). This difference can be seen in Figure 4.25.

Vehicle heading is the direction the vehicle is pointed, while course is the direction of the vector created by the vehicle's direction of travel. Both of these angles are referenced clockwise from North. In practice, this angle is measured or estimated at the CG, then can be translated to other locations on the vehicle for analysis or to extract further information.

Kinematic relationships between sensors on a vehicle are often used to define state equations in a Kalman filter because they require less model knowledge and are not affected by to changes in the model, such as vehicle mass or tire parameters. The kinematic models presented in the previous sections depict the car as if it were on rails by assuming zero lateral velocity, or by assuming a proportional relationship between steer angle and yaw rate. These assumptions are valid at low speeds for passenger vehicles as the car is generating minute slip angles and the lateral velocity is virtually zero. The

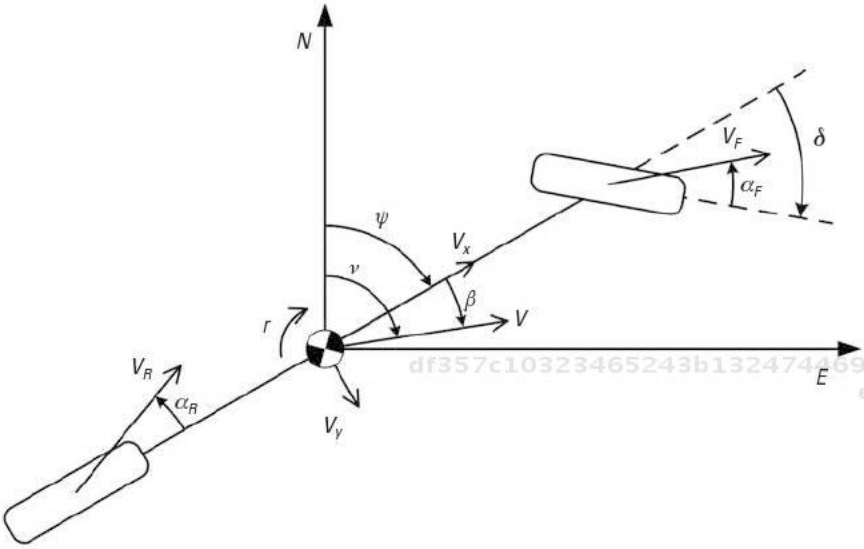


Figure 4.25 The difference between heading and course, or sideslip, can be seen in the bicycle diagram.

assumptions break down at higher speeds as lateral velocities are generated that invalidate the kinematic model, as will be shown in this chapter. When this occurs, a degradation of the performance of navigation or control systems may be observed. For these assumptions, speed also must be considered relative to the vehicle parameters, or the terrain on which the vehicle is operating. A small all-terrain vehicle (ATV) with soft tires maneuvering at 20 m/s on pavement could generate the same slip angles as a sports coupe with performance tires traveling 35 m/s. Similarly, a car turning on dry pavement at 10 m/s would generate very small slip angles, while the same car traveling at the same speed would generate significantly larger slip angles if it were on a low friction surface such as ice.

4.7.2 Sideslip Compensation with a Dual Antenna GPS Receiver

The estimators described in previous sections utilize a single antenna GPS receiver and an IMU to track vehicle position, orientation, and speed. The NMEA RMC message provides a measurement of a vehicle's course, speed, and position, while the IMU measures a vehicle's yaw rate and longitudinal acceleration. The Kalman filter compares the integrated yaw rate gyro with the course measurement, and the integrated accelerometer to the speed measurement, to estimate the vehicle's speed, orientation, longitudinal accelerometer bias, and yaw rate gyroscope bias. The yaw rate gyroscope provides a measurement of the time derivative of heading,

$$\dot{\psi} = r \tag{4.51}$$

and the course measurement is the combination of heading and sideslip.

$$\nu = \psi + \beta \tag{4.52}$$

A heading measurement is not provided in the NMEA RMC message, so the course measurement is substituted for vehicle orientation. Under straight or low-speed driving, the integrated yaw rate, or heading, and course measurements should be similar.

$$\nu = \beta + \quad r \quad dt \tag{4.53}$$

Small sideslip angles fulfill the small angle approximation, so vehicle speed can be approximated with an integrated longitudinal accelerometer. However, vehicle speed gains a lateral component when sideslip is nonzero.

$$\nu_x \qquad a_x dt \tag{4.54}$$

$$\vec{\nu} = \|\vec{\nu}\| \cos(\beta)_{\hat{i}} + \|\vec{\nu}\| \sin(\beta)_{\hat{j}} \tag{4.55}$$

Maneuvers producing sideslip, such as an emergency lane change, introduce a discrepancy in the yaw rate gyro bias estimate because the heading from an integrated yaw rate gyro is compared to a GPS course in the innovation in the extended Kalman filter. The difference between the heading and course is perceived as an input and/or measurement error under the assumptions of the previously given navigation algorithms. The error is propagated through to the heading and yaw gyroscope bias states. Further complications arise because the longitudinal accelerometer in the IMU is measuring a centripetal acceleration neglected in the models, and GPS speed from the NMEA RMC message is measurement of the norm of longitudinal and lateral velocity. These neglected terms can lead to significant state estimate errors if a vehicle is generating even small sideslip angles. The kinematic relationships that account for sideslip should be included in the state equations in the Kalman filter in order to produce a more accurate navigation solution if the appropriate sensors are available.

A dual antenna GPS receiver can provide vehicle attitude information. A heading measurement can be output from the system, and its accuracy is independent of vehicle motion. Another benefit of a dual antenna system is the availability of a roll or pitch measurement. A full attitude system with three (or sometimes four) antennas can provide all three orientation angles: roll, pitch, and yaw.

The previously defined dynamic model in (4.36) through (4.41) is redefined to handle sideslip. The following model assumes the availability of a dual antenna receiver that outputs roll and yaw measurements. The roll rate, p, is added as an input and used to estimate the vehicle roll angle. The velocity estimate is now broken up into longitudinal and lateral components, and centripetal acceleration terms are included.

$$\dot{v}_x = a_x \quad b_{ax} \quad \omega_{ax} + v_y(r \quad b_r) \tag{4.56}$$

$$\dot{b}_{ax} = \omega_{b_{ax}} \tag{4.57}$$

$$\dot{v}_y = a_y \quad b_{ay} \quad \omega_{ay} \quad v_x(r \quad b_r) \quad g\sin(\phi) \tag{4.58}$$

$$\dot{b}_{ay} = \omega_{b_{ay}} \tag{4.59}$$

$$\dot{\psi} = r \quad b_r \quad \omega_r \tag{4.60}$$

$$\dot{b}_r = \omega_{b_r} \tag{4.61}$$

$$\dot{\phi} = p \quad b_p \quad \omega_p \tag{4.62}$$

$$\dot{b}_p = \omega_{bp} \tag{4.63}$$

The position remains a function of speed and course, but course is expressed as the sum of heading and sideslip.

$$\dot{N} = v\cos(v) = \left(v_x^2 + v_y^2\right)^{1/2} \cos \;\; \psi + a\tan \;\; \frac{v_y}{v_x} \div\div \tag{4.64}$$

$$\dot{E} = v\sin v = \left(v_x^2 + v_y^2\right)^{\frac{1}{2}} \sin\quad \psi + a\tan\quad \frac{v_y}{v_x}\div\div \tag{4.65}$$

A state vector can be constructed containing the necessary states to control a vehicle and IMU calibration states.

$$x = \quad v_x \quad v_y \quad \phi \quad \psi \quad b_{ax} \quad b_{ay} \quad b_p \quad b_r \quad N \quad E^{\quad T} \tag{4.66}$$

The process noise vector is defined as follows:

$$\omega = \quad \omega_{ax} \quad \omega_{ay} \quad \omega_p \quad \omega_r \quad \omega_{b_{ax}} \quad \omega_{b_{ay}} \quad \omega_{b_p} \quad \omega_{b_r}^{\quad T} \tag{4.67}$$

The dynamics matrix is generated by calculating the Jacobian of the vector field consisting of (4.56) through (4.65) with respect to the state vector. The result is a 10 10 matrix. The nonzero elements are as follows:

$$F_{2,1} = \quad r + b_r \tag{4.68}$$

$$F_{9,1} = \frac{v_x\cos\quad \psi + a\tan\quad \frac{v_y}{v_x}\div\div + v_y\sin\quad \psi + a\tan\quad \frac{v_y}{v_x}\div\div}{\left(v_x^2 + v_y^2\right)^{\frac{1}{2}}} \tag{4.69}$$

$$F_{10,1} = \frac{v_x\sin\quad \psi + a\tan\quad \frac{v_y}{v_x}\div\div + v_y\cos\quad \psi + a\tan\quad \frac{v_y}{v_x}\div\div}{\left(v_x^2 + v_y^2\right)^{\frac{1}{2}}} \tag{4.70}$$

$$\begin{aligned} F_{1,2} &= F_{2,1} \\ F_{9,2} &= F_{10,1} \\ F_{10,2} &= F_{9,1} \end{aligned} \tag{4.71}$$

$$F_{2,3} = \quad g\cos(\phi) \tag{4.72}$$

$$\begin{aligned} F_{9,4} &= \quad \left(v_x^2 + v_y^2\right)^{1/2} \sin \;\; \psi + a \tan \;\; \frac{v_y}{v_x} \div\div \\ F_{10,4} &= \left(v_x^2 + v_y^2\right)^{1/2} \cos \;\; \psi + a \tan \;\; \frac{v_y}{v_x} \div\div \end{aligned} \tag{4.73}$$

$$\begin{aligned} F_{1,8} &= \quad v_x \\ F_{2,8} &= v_y \end{aligned} \tag{4.74}$$

$$F_{1,5}, F_{2,6}, F_{3,7}, F_{4,8} = \quad 1 \tag{4.75}$$

The process gain matrix is fairly straightforward.

$$L = \begin{matrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \tag{4.76}$$

The process covariance is the variances of the noise sources along the diagonal of an 8 8 matrix.

$$Q = diag(\omega^T \omega) \tag{4.77}$$

Measurements for this system are longitudinal and lateral velocity, heading (not course), roll, and position. Currently, many of these measurements are not available on passenger vehicles but can readily accessible on a UGV platform with the right sensor suite. However, obtaining these measurements on a highway vehicle is not a distant concept as GPS technology continues to become more prevalent in automotive systems. Sideslip has been effectively measured using a yaw rate gyro and GPS, which is currently on many vehicles [11, 12]. As previously stated, this algorithm was derived with a dual antenna GPS receiver in mind, but the source of a heading or sideslip measurement is irrelevant. Once sideslip is known, the GPS velocity measurement can be separated into longitudinal and lateral components using (4.55) and used as separate measurements.

The measurement vector is constructed as follows:

$$z = [v_x \quad v_y \quad \phi \quad \psi \quad N \quad E]^T \tag{4.78}$$

The variances of the measurements are assumed to be uncorrelated, even though that is not the case with both lateral and longitudinal velocity and position. This assumption provides reasonable results.

$$R = diag\left(\sigma_{v_x}^2 \quad \sigma_{v_y}^2 \quad \sigma_\phi^2 \quad \sigma_\psi^2 \quad \sigma_N^2 \quad \sigma_E^2\right) \tag{4.79}$$

The measurement matrix is as follows:

$$H = \begin{matrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{matrix} \tag{4.80}$$

The navigation model using the dual antenna system was compared to the system using a single antenna GPS receiver. An emergency lane change maneuver was simulated; the vehicle swerved suddenly, departing from and returning to the original lane of travel, as if to avoid an object. Figure 4.26 compares the heading estimate. The navigation system with the dual antenna GPS receiver is able to track the vehicle's heading, but the system with the single antenna receiver cannot. The single antenna receiver provides the

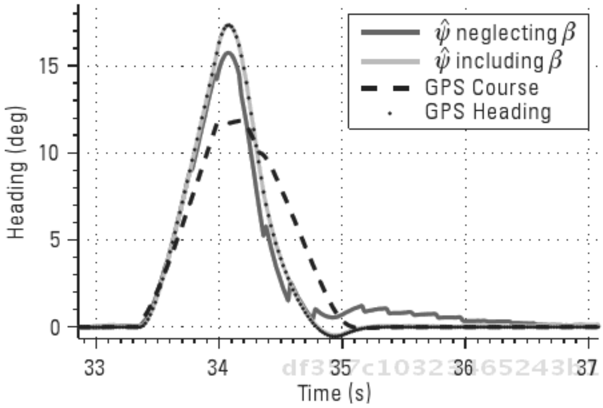


Figure 4.26 The heading estimate of two navigation systems is compared during a maneuver that generates sideslip. The system using the two-antenna system accurately tracks vehicle heading, while the system using a single-antenna system has an error in the heading estimate.

navigation system with a measurement of vehicle course, or the direction of travel, which differs from vehicle heading when the sideslip angle is nonzero.

Error is also present in the yaw gyroscope bias estimate from the algorithm neglecting sideslip effects. However, the bias estimate remains close to the truth when accounting for sideslip, as seen in Figure 4.27.

The error is often negligible in many situations, even during maneuvers that generate sideslip, for two reasons. The discrepancy will be small, on the order of a few degrees, unless complete control of the vehicle is lost. Also, the error will not persist after the slippage as incoming measurements will be used to correctly calibrate the sensor.

The heading and yaw gyroscope bias estimate error can negatively impact the navigation solution in the event of a GPS outage while experiencing sideslip. The yaw gyroscope will be integrated, with the erroneous bias removed, when dead reckoning through an outage. This causes a drastic difference in dead reckoning error, as seen in Figure 4.28.

Another benefit of the algorithm described in this section is the availability of a high rate sideslip estimate. Sideslip is the angle between the longitudinal and lateral velocity components.

$$\hat{\beta} = \tan^{\ 1} \ \frac{\hat{v}_y}{\hat{v}_x} \div \qquad (4.81)$$

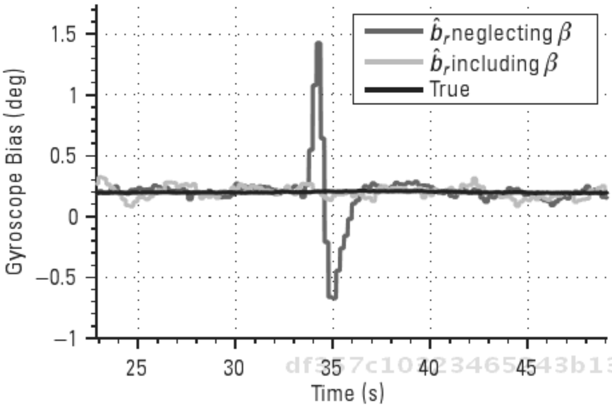


Figure 4.27 Bias estimates from two navigation systems (one neglecting sideslip effects and one incorporating them) are shown during a maneuver that generates a nonzero sideslip angle.

This high rate estimate of sideslip could be used in a stability control system on a UGV. Figure 4.29 displays a 50-Hz sideslip estimate and a 5-Hz sideslip measurement generated using a dual-antenna GPS receiver.

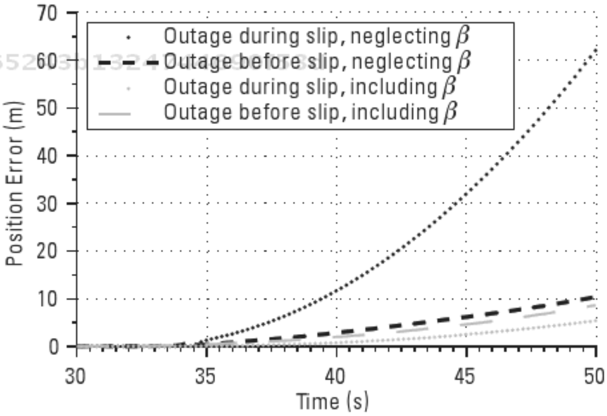


Figure 4.28 Position error during simulated GPS outage is shown. Outages were simulated just before and during a maneuver that generated sideslip. The algorithm accounting for lateral motion exhibited a lower dead reckoned position error than the algorithm that neglected sideslip effects.

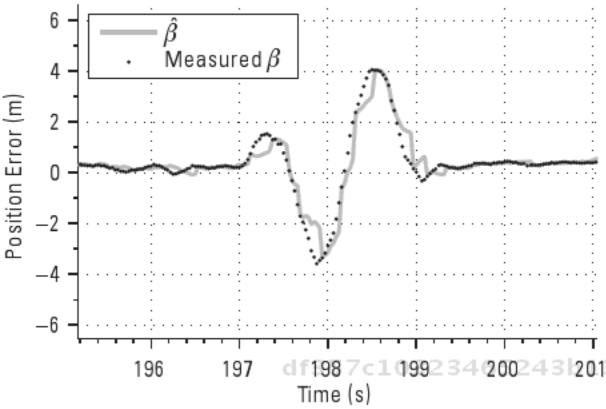


Figure 4.29 A sideslip estimate is compared to the measured value.

4.8 Closely Coupled Integration

A closely coupled integration routine combines the inertial measurements with range measurements from a GPS receiver. The inertial sensors can be calibrated using the measured distances to the GPS satellites, and a navigation solution can be produced. The algorithm is similar to a range processing algorithm used by a GPS receiver to determine user position and velocity. The major change is that receiver dynamics are no longer approximated as a Markov process with a time constant dependent upon the anticipated receiver motion; an IMU provides an actual measurement of the receiver motion.

One advantage to a closely coupled integration is the availability of measurements when a full position solution is not available. Typically, four satellites are needed to solve for three position states and a receiver clock bias state. When fewer than four satellites are visible, these states are not fully observable. A loosely coupled algorithm has no measurement aiding in this scenario, and the dead reckoning error can quickly grow. However, the error of a closely coupled solution drifts at a reduced rate because the navigation processor is still able to incorporate the available range measurements. Note that some receivers operate with a reduced state filter when using fewer than four observations by constraining the height or clock bias states at the cost of an increased error variance. A closely coupled implementation allows the designer to incorporate similar logical “tricks.”

Another advantage to the closely coupled algorithm is realized by the individual weighting and processing of the range measurements combined with the inertial measurements containing information about a receiver's motion. The navigation solution is less prone to large jumps when satellites come in or out of view, or when obstructions shadow or reflect the signal.

The algorithm in this section will utilize all outputs from an IMU (three accelerations and three rotational rates) and combine them with GPS pseudorange and Doppler frequency measurements. The pseudorange is a receiver to satellite range measurement, and the Doppler frequency can be used to form a pseudorange rate, or a receiver to satellite range rate measurement. "Pseudo" is used to indicate the measurement is not a true range or range rate due to the presence of errors. This algorithm is also derived assuming a discrete time system. This gives the reader a look at an alternative method to develop navigation system models. It will utilize an extended Kalman filter with a nonlinear time propagation and a nonlinear measurement update.

A strap-down IMU must be mechanized before integration with the GPS range measurements. Its acceleration and turning rate measurements will be about the IMU frame. The navigation frame and IMU frame are usually noncoincident, and therefore the inertial measurements must be transformed from the IMU frame to the navigation frame. An Euler angle-based approach is presented, although other methods are commonly used. The Euler angle approach is fairly intuitive, but a singularity exists when the pitch angle is 90°. If this presents a problem to the designer, an alternative method, such as a quaternion-based approach, should be sought.

The algorithm presented in this section uses the ECEF frame as the navigation frame. Therefore, the Euler angles (ϕ, θ, ψ) correspond to the three rotations necessary to rotate inertial measurements from the IMU frame to the ECEF frame. The inertial frame is assumed to be coincident with the body frame, so the body to ECEF frame angles can be expressed as ϕ_b^e, θ_b^e, and ψ_b^e. Note that the Greek letter psi is not to be confused with vehicle heading; the Euler angle and vehicle heading do not have to have a direct relationship.

A coordinate transformation matrix is constructed in order to express the IMU accelerometer measurements in the ECEF frame. It consists of operations on the sine and cosine of the angles. The notations s(·) and c(·) denote the sine and cosine of an angle, respectively.

$$C_b^e = \begin{matrix} c\theta\, c\psi & c\phi\, s\psi + s\phi\, s\theta\, c\psi & s\phi\, s\psi + c\phi\, s\theta\, c\psi \\ c\theta\, s\psi & c\phi\, c\psi + s\phi\, s\theta\, s\psi & s\phi\, s\psi + c\phi\, s\theta\, s\psi \\ s\theta & s\phi\, c\theta & c\phi\, c\theta \end{matrix} \qquad (4.82)$$

The expression of the angular rates of the body frame in the ECEF frame requires the time derivative of the rotation matrix.

$$M_b^e = \begin{matrix} 1 & \dfrac{s\phi s\theta}{c\theta} & \dfrac{c\phi s\theta}{c\theta} \\ 0 & c\phi & s\phi \\ 0 & \dfrac{s\phi}{c\theta} & \dfrac{c\phi}{c\theta} \end{matrix} \tag{4.83}$$

Notice the division of terms by the cosine of the pitch angle. Together, these matrices are used to express the IMU output in the ECEF frame. Knowledge of the vehicle's original position and orientation is necessary to initialize these matrices.

A notational difference exists between this section and previous sections. This difference is due to the expression of the same measurement in multiple coordinate systems. Using the notational style in [2], a generic variable can be expressed as

$$\vec{x}_{\alpha\beta}^{\gamma} \tag{4.84}$$

where the vector x is the kinematic relationship from the α-frame to the β-frame expressed in the γ-frame.

The accelerometer output is denoted by the vector f, and it contains the three measurements about the local X, Y, and Z axes. This variable is chosen to reflect the fact that the accelerometer is technically outputting a measurement of specific force. Forces that move the accelerometer bend or move a proof mass, and the output is a ratio of the measured force to the proof mass. A superscript denotes the frame in which the measurement is expressed. For instance, f^e denotes the specific force measurement expressed in the ECEF frame, and f^b denotes the specific force measurement in the body frame.

Rotational rate vectors are expressed using the vector ω. The IMU measures both the body and Earth rotation, but these two can be separated. The body rotation vector about the ECEF frame expressed in the body frame is defined as follows:

$$\vec{\omega}_{eb}^{b} = [p \quad q \quad r]^T \tag{4.85}$$

where p, q, and r are the body roll, pitch, and yaw rates. The Earth rotation vector about the inertial frame expressed in the ECEF frame is defined as follows:

$$\vec{\omega}^e_{ie} = [0 \quad 0 \quad \omega_e]^T \tag{4.86}$$

where ω_e is 0.00007292115 rad/s. A skew symmetric form of the Earth rotation can be also be written.

$$ {}^e_{ie} = \omega_e \begin{matrix} 0 & \omega_e & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix} \tag{4.87}$$

The Euler angles are contained in a vector denoted by a, which is known as the attitude vector.

$$\vec{a} = \phi^e_b \quad \theta^e_b \quad \psi^e_b \quad {}^T \tag{4.88}$$

The specific force expressed in the ECEF frame is integrated to produce the ECEF velocity and ECEF position. The measurement is corrected for gravitational, centripetal, and Coriolis effects so only the specific force corresponding to the translational motion of the receiver is integrated. The accelerometer bias is removed from the specific force in the body frame, and the resulting quantity is rotated to the ECEF frame. The variable w is used to indicate the presence of white noise. The gravity, centripetal, and Coriolis terms are then removed.

$$\vec{f}^e_k = C^e_b\left(\vec{f}^b_k \quad \vec{b}_{f_k} \quad \vec{w}_{f_k}\right) + G\left(\vec{r}^e_{eb_k\ 1}\right) \quad 2 \quad {}^e_{ie}\nu^e_{eb_k\ 1} \tag{4.89}$$

The middle term represents the gravity model. This accounts for the gravitational and centripetal effects. Various gravity models based on the user's location and altitude can be implemented. A simple model using the standard Earth gravitational parameter, GM, where GM = 398,600,441,800,000 m^3/s^2, is shown here.

$$G\left(\vec{r}^e_{eb}\right) = \quad \frac{GM}{\left\|\vec{r}^e_{eb}\right\|^3}\vec{r}^e_{eb} \quad {}^e_{ie} \quad {}^e_{ie}\vec{r}^e_{eb} \tag{4.90}$$

Once the specific force at time k is determined, the position, velocity, and attitude states can be discretely propagated using the sample rate, $\quad t$.

$$\vec{r}^{e}_{eb_k} = \vec{r}^{e}_{eb_{k\ \ 1}} + \vec{v}^{e}_{eb_{k\ \ 1}} \quad t + \vec{f}^{e}_{k} \frac{\quad t^2}{2} \tag{4.91}$$

$$\vec{v}^{e}_{eb_k} = \vec{v}^{e}_{eb_{k\ \ 1}} + \vec{f}^{e}_{k} \quad t \tag{4.92}$$

$$\vec{a}_k = \vec{a}_{k\ \ 1} + M^e_b \left(\vec{\omega}^b_{ib} \quad \vec{b}_{g_k} \quad \vec{w}_{g_k} \right) \quad t \tag{4.93}$$

As with the accelerometer measurement, the rate gyroscope bias is removed before transforming the measurements into the navigation frame. The six bias states, three accelerometer biases and three rate gyroscope biases, are modeled as white noise.

$$\vec{b}_{f_k} = \vec{b}_{f_{k\ \ 1}} + \vec{w}_{bf_k} \tag{4.94}$$

$$\vec{b}_{g_k} = \vec{b}_{g_{k\ \ 1}} + \vec{w}_{bg_k} \tag{4.95}$$

Two additional states are necessary when processing pseudorange and pseudorange rate measurements. The oscillator on the receiver driving its clock will have a phase and frequency error, which produces a bias and drift in the clock output. A common two-state model derived in [6] is used to capture the clock errors. It assumes both the bias and drift exhibit a random walk over time, and that the drift drives the bias state. The speed of light constant (c = 299,792,458 m/s) is multiplied through to convert the time to units of distance.

$$c\delta t_{u_k} = c\delta t_{u_{k\ \ 1}} + c\delta \dot{t}_{uk} \quad t + w_{\delta t_k} \tag{4.96}$$

$$c\delta \dot{t}_{u_k} = c\delta \dot{t}_{u_{k\ \ 1}} + w_{\delta \dot{t}_k} \tag{4.97}$$

There are 17 total states representing a three-dimensional position, velocity, and orientation of the receiver, plus the six IMU calibration states and two receiver clock states.

$$x = \quad \vec{r}^{e}_{eb} \quad \vec{v}^{e}_{eb} \quad \vec{a} \quad \vec{b}_f \quad \vec{b}_g \quad c\delta t \quad c\delta \dot{t} \quad ^T \tag{4.98}$$

Although the nonlinear system model is propagated with (4.91) through (4.93), the state transition matrix is still required to propagate the error covariance. The state transition matrix is determined by calculating the Jacobian of the vector field of nonlinear functions with respect to the state vector. The noise terms are ignored as they are nondeterministic; they will be accounted for in the process noise gain and covariance matrix. The nonzero blocks of the state transition matrix are shown here.

$$ {}_{1:3,1:3} = I + \frac{t^2}{2} \quad 3\frac{GM}{\left\|\vec{r}^{\,e}_{eb}\right\|^5}\vec{r}^{\,e}_{eb}\vec{r}^{\,eT}_{eb} \quad \frac{GM}{\left\|\vec{r}^{\,e}_{eb}\right\|^3} I \quad {}^{e}_{ie} \quad {}^{e}_{ie} \tag{4.99} $$

$$ {}_{4:6,1:3} = \quad t \quad 3\frac{GM}{\left\|\vec{r}^{\,e}_{eb}\right\|^5}\vec{r}^{\,e}_{eb}\vec{r}^{\,eT}_{eb} \quad \frac{GM}{\left\|\vec{r}^{\,e}_{eb}\right\|^3} I \quad {}^{e}_{ie} \quad {}^{e}_{ie} \tag{4.100} $$

$$ {}_{1:3,4:6} = \quad t\left(I \quad {}^{e}_{ie} \quad t\right) \tag{4.101} $$

$$ {}_{4:6,4:6} = I \quad 2 \quad {}^{e}_{ie} \quad t \tag{4.102} $$

$$ {}_{1:3,7} = \frac{t^2}{2} \begin{matrix} (s\phi s\psi + c\phi s\theta c\psi)\overline{f}_y + (c\phi s\psi \quad s\phi s\theta c\psi)\overline{f}_z \\ (\quad s\phi c\psi + c\phi s\theta s\psi)\overline{f}_y + (\quad c\phi c\psi \quad s\phi s\theta s\psi)\overline{f}_z \\ (c\phi c\theta)\overline{f}_y + (\quad s\phi c\theta)\overline{f}_z \end{matrix} \tag{4.103} $$

$$ {}_{1:3,8} = \frac{t^2}{2} \begin{matrix} (\quad s\phi c\psi)\overline{f}_x + (s\phi c\theta c\psi)\overline{f}_y + (c\phi c\theta c\psi)\overline{f}_z \\ (\quad s\phi s\psi)\overline{f}_x + (s\phi c\theta s\psi)\overline{f}_y + (c\phi c\theta s\psi)\overline{f}_z \\ c\theta\,\overline{f}_x + (\quad s\phi s\theta)\overline{f}_y + (\quad c\phi s\theta)\overline{f}_z \end{matrix} \tag{4.104} $$

$$ {}_{1:3,9} = \frac{t^2}{2} \begin{matrix} (\quad c\theta s\psi)\overline{f}_x + (\quad c\phi c\psi \quad s\phi s\theta s\psi)\overline{f}_y + (s\phi c\psi \quad c\phi s\theta s\psi)\overline{f}_z \\ (c\theta c\psi)\overline{f}_x + (\quad c\phi s\psi + s\phi s\theta c\psi)\overline{f}_y + (s\phi s\psi + c\phi s\theta c\psi)\overline{f}_z \\ 0 \end{matrix} \tag{4.105} $$

$$_{4:6,7} = \quad t \quad \begin{array}{c} (s\phi s\psi + c\phi s\theta c\psi)\overline{f}_y + (c\phi s\psi \quad s\phi s\theta c\psi)\overline{f}_z \\ (\quad s\phi s\psi + c\phi s\theta s\psi)\overline{f}_y + (\quad c\phi c\psi \quad s\phi s\theta s\psi)\overline{f}_z \\ (c\phi c\theta)\overline{f}_y + (\quad s\phi c\theta)\overline{f}_z \end{array} \tag{4.106}$$

$$_{4:6,8} = \quad t \quad \begin{array}{c} (\quad s\phi c\psi)\overline{f}_x + (s\phi c\theta c\psi)\overline{f}_y + (c\phi c\theta c\psi)\overline{f}_z \\ (\quad s\phi s\psi)\overline{f}_x + (s\phi c\theta s\psi)\overline{f}_y + (c\phi c\theta s\psi)\overline{f}_z \\ c\theta\,\overline{f}_x + (\quad s\phi s\theta)\overline{f}_y + (\quad c\phi s\theta)\overline{f}_z \end{array} \tag{4.107}$$

$$_{4:6,9} = \quad t \quad \begin{array}{c} (\quad c\theta s\psi)\overline{f}_x + (\quad c\phi c\,\psi \quad s\phi s\theta s\psi)\overline{f}_y + (s\phi c\,\psi \quad c\phi s\theta s\psi)\overline{f}_z \\ (c\theta c\,\psi)\overline{f}_x + (\quad c\phi s\psi + s\phi s\theta c\,\psi)\overline{f}_y + (s\phi s\psi + c\phi s\theta c\,\psi)\overline{f}_z \\ 0 \end{array} \tag{4.108}$$

$$_{7:9,7} = \quad \begin{array}{c} 1 + \quad t \quad \dfrac{c\phi s\theta}{c\theta}(q \quad b_{gq}) \quad \dfrac{s\phi s\theta}{c\ \theta}(r \quad b_{gr}) \div \\ t\Big(\quad s\phi(q \quad b_{gq}) \quad c\phi(r \quad b_{gr})\Big) \\ t \quad \dfrac{c\phi}{c\theta}(q \quad b_{gq}) \quad \dfrac{s\phi}{c\theta}(r \quad b_{gr}) \div \end{array} \tag{4.109}$$

$$_{7:9,8} = \quad \begin{array}{c} t \quad s\phi + \dfrac{s\phi s\theta^2}{c\theta^2} \div (q \quad b_{gq}) + \quad c\phi + \dfrac{c\phi s\theta^2}{c\theta^2} \div (r \quad b_{gr}) \div \\ 1 \\ t \quad \dfrac{s\phi s\theta}{c\theta^2}(q \quad b_{gq}) + \dfrac{c\phi s\theta}{c\theta^2}(r \quad b_{gr}) \div \end{array} \tag{4.110}$$

$$_{7:9,9} = \quad \begin{array}{c} 0 \\ 0 \\ 1 \end{array} \tag{4.111}$$

$$_{1:3,10:12} = \frac{t^2}{2} C_b^e \tag{4.112}$$

$$_{10:12,10:12} = I \tag{4.113}$$

$$_{7:9,13:15} = tM_b^e \tag{4.114}$$

$$_{13:15,13:15} = I \tag{4.115}$$

$$_{16:17,16:17} = \begin{matrix} 1 & t \\ 0 & 1 \end{matrix} \tag{4.116}$$

The process noise on the states comes from noise in the IMU and the receiver clock. The IMU noise is modeled as independent and uncorrelated white noise. The clock noise model was derived in [6, 13] and requires knowledge of the clock quality to determine the spectral variance of the bias and drift.

$$s_b = 2\pi^2 h_{\ 2} c^2 \tag{4.117}$$

$$s_d = \frac{h_0}{2} c^2 \tag{4.118}$$

Values for h_0 and $h_{\ 2}$ can be obtained from an Allan variance plot of the clock output. Values listed in [13] for several clocks of varying quality are listed in Table 4.2. The process noise vector can be defined with statistics of the IMU noise and bias drift and receiver clock quality.

Table 4.2
Power Spectral Density Coefficients for Various Clocks

Clock Type	h_0	$h_{\ 2}$
Compensated crystal	$2 \quad 10^{19}$	$2 \quad 10^{20}$
Ovenized crystal	$8 \quad 10^{20}$	$4 \quad 10^{23}$
Rubidium	$2 \quad 10^{20}$	$4 \quad 10^{29}$

$$Q_{1:12,1:12} = diag\left(\sigma^2_{w_f} \quad \sigma^2_{w_g} \quad \sigma^2_{w_{bf}} \quad \sigma^2_{w_{bg}}\right) \tag{4.119}$$

$$Q_{13:14,13:14} = \begin{matrix} s_b \quad t + s_d \dfrac{t^3}{3} & s_d \dfrac{t^2}{2} \\ s_d \dfrac{t^2}{2} & s_d \quad t \end{matrix} \tag{4.120}$$

The process gain matrix is determined by calculating the Jacobian of the vector field generated by (4.82) through (4.90) with respect to the process noise vector. The elements of the process gain matrix corresponding to the clock states are set to identity.

$$_{1:3,1:3} = \frac{t^2}{2} C_b^e \tag{4.121}$$

$$_{4:6,1:3} = tC_b^e \tag{4.122}$$

$$_{7:9,4:6} = tM_b^e \tag{4.123}$$

$$_{7:9,7:9} = t^2 M_b^e \tag{4.124}$$

$$_{13:15,7:9} = tI \tag{4.125}$$

$$_{1:3,10:12} = \frac{t^3}{2} C_b^e \tag{4.126}$$

$$_{4:6,10:12} = t^2 C_b^e \tag{4.127}$$

$$_{10:12,10:12} = tI \tag{4.128}$$

$$_{16:14,13:14} = I \tag{4.129}$$

Measurements in the closely coupled system consist of the range information computed in the GPS receiver. Specifically, the pseudorange and pseudorange rate are used. The pseudorange measurements are usually output directly by the receiver. However, the pseudorange rate is typically determined by manipulating the Doppler frequency measurement.

$$\dot{\rho} = \quad \lambda f_D \tag{4.130}$$

In (4.136), the pseudorange rate is a function of the carrier wavelength, λ, and the Doppler frequency, f_D. The phase information can be used in lieu of the pseudorange rate, but using both does not add information to the system because the phase measurement is the integral of the Doppler shift.

The measurement vector is denoted by z, and it contains all of the pseudorange and pseudorange rates to be processed. The system designer must account for the change in size of this vector as the number of observations fluctuates.

$$z = [\rho \qquad \dot{\rho}]^T \tag{4.131}$$

The nonlinear relationship between the states and measurements, or measurement model, is expressed as follows:

$$h(x) = \begin{array}{l} \sqrt{\vec{r}^{\,e}_{eb} \quad \vec{r}^{\,e}_{es} \;{}^{T} \quad \vec{r}^{\,e}_{eb} \quad \vec{r}^{\,e}_{es}} + c\delta t \\ \dfrac{\vec{r}^{\,e}_{eb} \quad \vec{r}^{\,e}_{es} \;{}^{T} \quad \vec{v}^{\,e}_{eb} \quad \vec{v}^{\,e}_{es}}{\sqrt{\vec{r}^{\,e}_{eb} \quad \vec{r}^{\,e}_{es} \;{}^{T} \quad \vec{r}^{\,e}_{eb} \quad \vec{r}^{\,e}_{es}}} + c\delta t \end{array} \tag{4.132}$$

The subscript b gives reference to the body or receiver position and velocity, and the subscript s is used to denote the GPS satellite position and velocity. Computing the Jacobian of $h(x)$ with respect to the state vector yields a linearized measurement model that utilizes the user to satellite unit vectors. The elements relating the velocity measurements to the position states are small enough that they can be assumed to be zero.

$$H = \begin{matrix} e^s_b & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & e^s_b & 0 & 0 & 0 & 0 & 1 \end{matrix} \tag{4.133}$$

$$e_b^s = \frac{\vec{r}_{eb}^{\,e} \quad \vec{r}_{es}^{\,e}}{\sqrt{\quad \vec{r}_{eb}^{\,e} \quad \vec{r}_{es}^{\,e} \quad {}^{T} \quad \vec{r}_{eb}^{\,e} \quad \vec{r}_{es}^{\,e}}} \tag{4.134}$$

Measurement accuracy is determined by the bandwidth of receiver's tracking loops, signal quality, and anticipated multipath effects. Expressions of the accuracy of the delay and frequency lock loops (DLL and FLL, respectively) as a function of the carrier-to-noise ratio, C/N_0, expressed in hertz, are given in [4, 5].

$$\sigma_{tDLL} = \lambda_c \sqrt{\frac{4d^2 B_n}{C/N_0} \quad 2(1 \quad d) + \frac{4d}{T_s C/N_0} \div} \tag{4.135}$$

$$\sigma_{tFLL} = \frac{\lambda}{2\pi T_s} \sqrt{\frac{4B_n}{C/N_0} \quad 1 + \frac{1}{T_s C/N_0} \div} \tag{4.136}$$

The measurement errors consist of the tracking loop errors plus additional terms. The term κ is used to inflate the error bounds to account for unmodeled terms, such as multipath and any residual atmospheric effects. The term f_e is the dynamic stress error and is a function of tracking loop bandwidth and loop order. The accuracy of the pseudorange and pseudorange rate measurements can be modeled with the following expressions.

Table 4.3

Tracking Loop Parameters

Parameter	Value	Unit
B_n, code loop noise bandwidth	2	Hz
d, correlator spacing	0.5	Chips
f_e, dynamic stress error	3	m/s
κ_ρ, unmodeled range error	5	m
λ, carrier wavelength	0.1902	m
λ_c, code chip width	293.05	m
T_s, predetection integration time	0.005	s

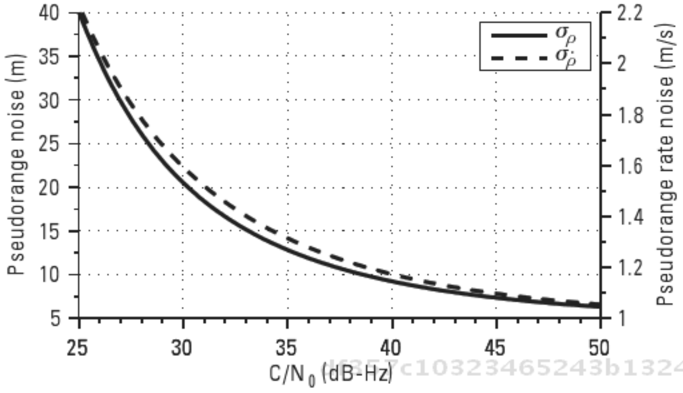


Figure 4.30 Pseudorange and pseudorange rate measurement noise increases as the C/N_0 decreases.

$$\sigma_\rho = \kappa_\rho + \sigma_{tDLL} \tag{4.137}$$

$$\sigma_\rho = \sigma_{tFLL} + \frac{f_e}{3} \tag{4.138}$$

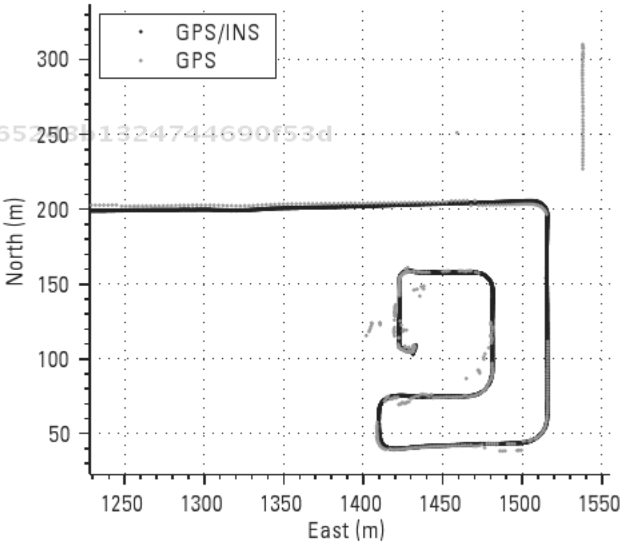


Figure 4.31 GPS and GPS/INS position plotted in the NED frame shows the improvements of adding inertial data into the filter; jumps in the position are filtered out.

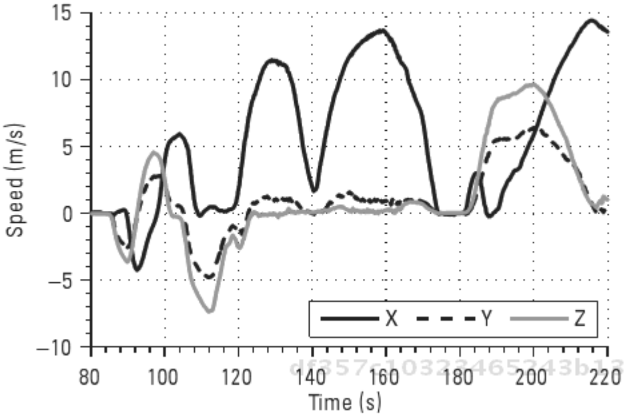


Figure 4.32 Vehicle velocity is provided by the closely coupled algorithm in the ECEF frame at a high rate.

Reasonable values and definitions of the parameters in (4.122) through (4.125) obtained from [5] are given in Table 4.3, and Figure 4.30 displays expected accuracies as a function of C/N_0 in dB-Hz using those parameters.

Data was collected on a vehicle traveling through a typical suburban environment using an automotive grade IMU and a GPS receiver. GPS data was logged at 5 Hz, and IMU data was logged at 50 Hz. Figure 4.31 shows the vehicle position plotted in an NED frame of reference. The 50-Hz position estimate from the closely coupled system is superior to the 5-Hz GPS only

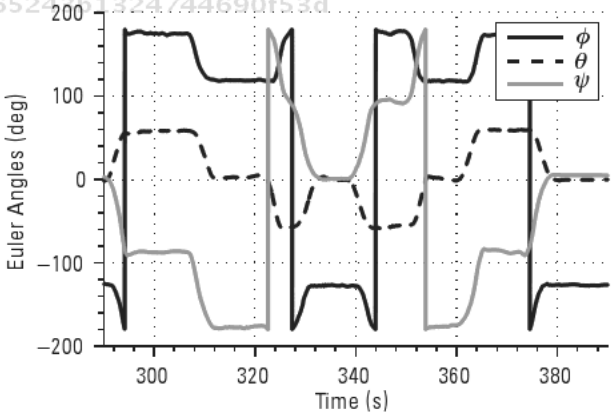


Figure 4.33 Estimates of the Euler angles from the body frame to ECEF frame are provided at a high rate by the closely coupled GPS/INS navigation system.

solution as jumps in the position are eliminated as the vehicle travels between three-story buildings. A high rate, three-dimensional velocity estimate is also provided by the closely coupled system. Figure 4.32 depicts the vehicle velocity in the ECEF frame. Figure 4.33 shows the Euler angles from the body frame to the ECEF frame as estimated by the closely coupled algorithm.

References

[1] Farrell, J. A., *Aided Navigation*, New York: McGraw-Hill, 2008.

[2] Groves, P. D., *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, Norwood, MA: Artech House, 2008.

[3] Grewal, M. S., L. R. Weill, and A. P. Andrews, *Global Positioning Systems, Inertial Navigation, and Integration*, New York: John Wiley and Sons, 2001.

[4] Misra, P., and P. Enge, *Global Positioning System: Signals, Measurements, and Performance*, Lincoln, MA: Ganga-Jamuna Press, 2006.

[5] Kaplan, E. D., and C. H. Hegarty, (eds.), *Understanding GPS: Principles and Applications*, 2nd ed., Norwood, MA: Artech House, 2006.

[6] Parkinson, B. W., and J. J. Spilker, Jr., *Global Positioning System: Theory and Applications Volume I*, Washington, D.C.: American Institute of Astronautics and Aeronautics, 1996.

[7] Parkinson, B. W., and J. J. Spilker, Jr., *Global Positioning System: Theory and Applications Volume II*, Washington, D.C.: American Institute of Astronautics and Aeronautics, 1996.

[8] Milliken, D., and W. Milliken, *Race Car Vehicle Dynamics*, Warrendale, PA: Society of Automotive Engineers, 1995.

[9] Dugoff, H., P. S. Fancher, and L. Segel, "An Analysis of Tire Traction Properties and Their Influence on Vehicle Dynamic Performance," SAE Paper 700377, 1970, pp. 1219–1243.

[10] Pacejka, H., E. Bakker, and L. Nyborg, "Tyre Modelling for Use in Vehicle Dynamics Studies," SAE Paper 870421, 1987.

[11] Bevly, D. M., R. Sheridan, and J. C. Gerdes, "Integrating INS Sensors with GPS Velocity Measurements for Continuous Estimation of Vehicle Sideslip and Tire Cornering Stiffness," *Proceedings of the American Control Conference 2001*, Arlington, VA, June 25–27, 2001, pp. 25–30.

[12] Bevly, D. M., J. C. Gerdes, and C. Wilson, "The Use of GPS Based Velocity Measurements for Measurement of Sideslip and Wheel Slip," *Vehicle System Dynamics*, Vol. 38, No. 2, 2002, pp. 127–147.

[13] Brown, R. G., and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, New York: John Wiley and Sons, 1997.

5

Vehicle Dynamic Estimation Using GPS*

David M. Bevly, Robert Daily, and William Travis

5.1 Introduction

The state feedback that will be described in Chapter 6 requires knowledge of all the states in the vehicle model. Some of these states (specifically lateral velocity or sideslip) are often not measured and therefore must be estimated. Additionally, the full state feedback assumes that vehicle parameters such as tire cornering stiffness are known. This chapter demonstrates various methods for estimating vehicle sideslip (for use in the full state feedback control algorithm presented in Chapter 6) as well as estimating critical model parameters. In order to perform vehicle control, both navigation states (such as vehicle position and orientation) and feedback control states (such as lateral velocity, yaw rate, and steer angles) must be obtained, either from direct measurements or observers. Because observers for estimating missing states in full state feedback control algorithms are well known, this chapter focuses primarily on the estimation of the lateral velocity and vehicle parameters that effect the lateral velocity. Note that the state estimation algorithms presented

*Reproduced with the permission of Inderscience Enterprises Limited, who retain the copyright, from the following source: Daily, R., W. Travis, and D. M. Bevly, "Cascaded Estimators to Improve Lateral Vehicle State and Tire Parameter Estimates," *International Journal of Vehicle Autonomous Systems*, Volume 5, Number 3–4, 2007, pp. 230–255.

in this chapter can be done independently or in conjunction with the navigation estimators presented in Chapter 4. However, it has been shown that it is often best to separate the estimation of the navigation states and feedback control states [1].

5.2 Sideslip Calculation

The body sideslip angle (β in Figure 5.1) is the angle between the longitudinal and lateral velocity components at the center of gravity (CG).

$$\beta = \tan^{\;1} \frac{V_y}{V_x} \div \tag{5.1}$$

A typical method to measure sideslip on test vehicles is using an optical sensor to measure the lateral and longitudinal velocity. However, these sensors are expensive and performance can be surface dependent. For these reasons, on production vehicles, this state is not measured. A newer method to measure sideslip is using GPS. The sideslip angle can also be defined as the difference between the vehicle heading and direction of travel (course) measured at the CG.

$$\beta = \nu \quad \psi \tag{5.2}$$

GPS provides a course measurement. However, the vehicle heading must still be determined. Heading can be estimated using a yaw gyro. The gyro is

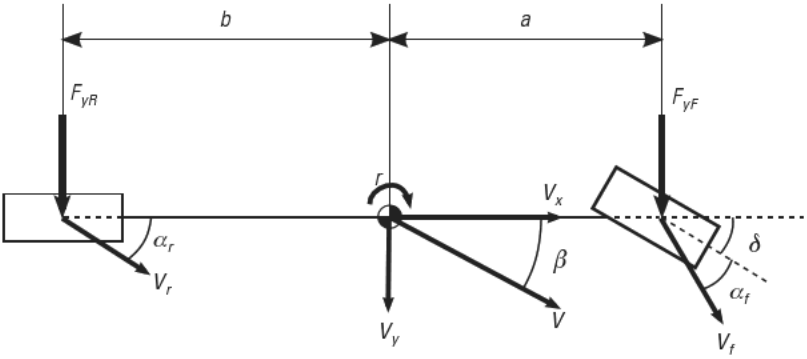


Figure 5.1 Lateral bicycle model showing vehicle sideslip angle.

initialized and its bias estimated during periods of straight driving and then integrated during turning maneuvers to provide the vehicle heading [2].

5.3 Vehicle Estimation

The GPS velocity based course measurements can be combined with the yaw rate gyroscope measurement through two types of estimators, a kinematic estimator or a model-based estimator. The purpose of combining the GPS measurements with the gyroscope measurements is to provide high-update rate, unbiased, accurate estimates of the lateral vehicle dynamics. This chapter utilizes a Kalman filter to integrate the GPS and IMU measurements to provide the state estimates for the vehicle. Note that one could utilize any type of estimator such as a traditional linear estimator with desired closed-loop dynamics designed based on the characteristics of the system dynamics. The Kalman filter does have the distinct advantage of providing the accuracy of the state estimation (assuming all the parameters in the estimator are correct) in the state estimation covariance matrix (P).

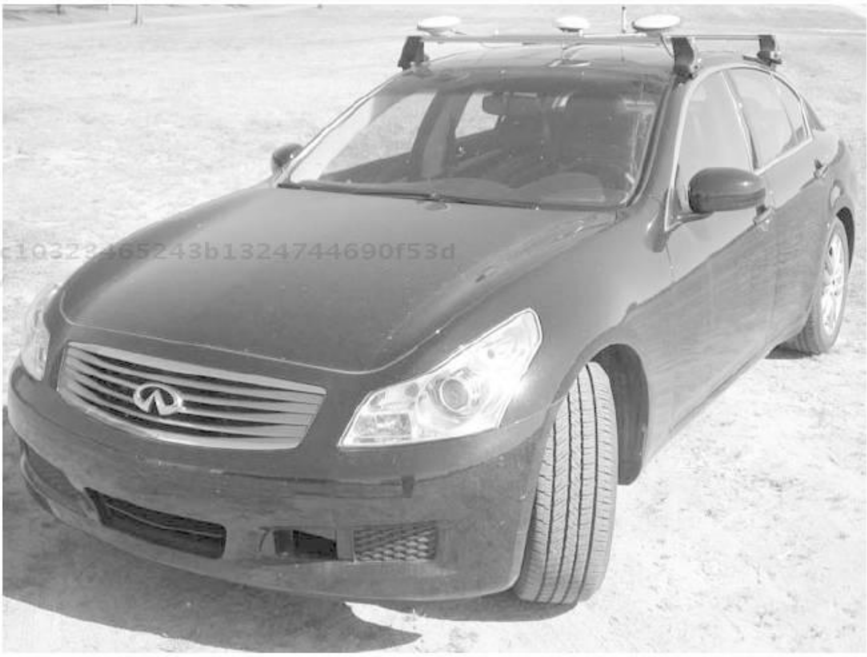


Figure 5.2 Instrumented Infiniti G35 sedan.

5.4 Experimental Setup

The methods described in this chapter are experimentally tested on the fully instrumented G35 sedan shown in Figure 5.2. The G35 is equipped with a dual antenna Novatel Beeline receiver that provides 5-Hz updates of vehicle heading, roll, speed, and course. It is also instrumented with a lateral accelerometer and yaw gyroscope recorded at 30 Hz. Additionally, wheel speed and steer angle are recorded from the onboard sensors through the CAN bus at 30 Hz. With the exception of the dual antenna GPS system, each of these sensors is available on any vehicle with electronic stability control (ESC). The methods presented in this chapter show the benefit of adding dual antenna GPS to ESC. The vehicle's mass (m) is 1,528 kg; the yaw moment of inertia (I_z) is approximated as 2,400 kg m^2 [3].

5.4.1 Test Scenarios

Two experimental test scenarios are used to validate the methods presented in this chapter: hard cornering on a low-μ gravel surface and limit driving

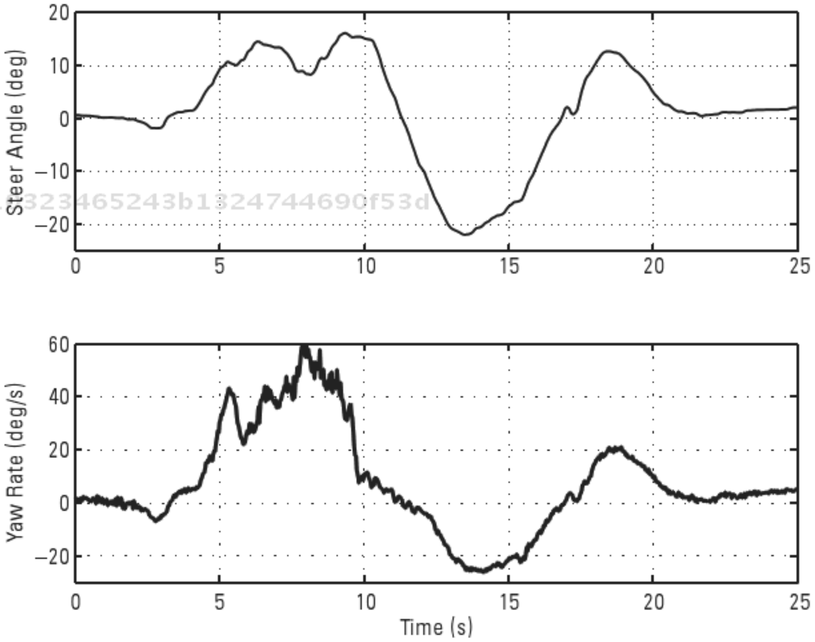


Figure 5.3 Low-μ experiment steer angle and yaw rate.

on asphalt. The low-μ surface experiment steer angle and resulting yaw rate are shown in Figure 5.3. This experiment is run at an average speed of 6 m/s.

The asphalt scenario consists of three segments. The first segment is a high-speed, steady state cornering experiment performed at 37 m/s around a 150-m radius turn banked at 8°. As in (3.18), the bank angle is typically neglected in vehicle models. The second segment is a lane change performed at 20 m/s. The final segment is a slalom at 25 m/s. The steer angle and yaw rate for this experiment are shown in Figure 5.4.

5.5 Kinematic Estimator (Single GPS Antenna)

One of the current ways to combine the GPS heading measurements with the information provided from the yaw gyroscope is through a kinematic or indirect Kalman filter [4–6]. The term *kinematic* comes from the fact that the model used by the estimator is based solely on the kinematic relationship

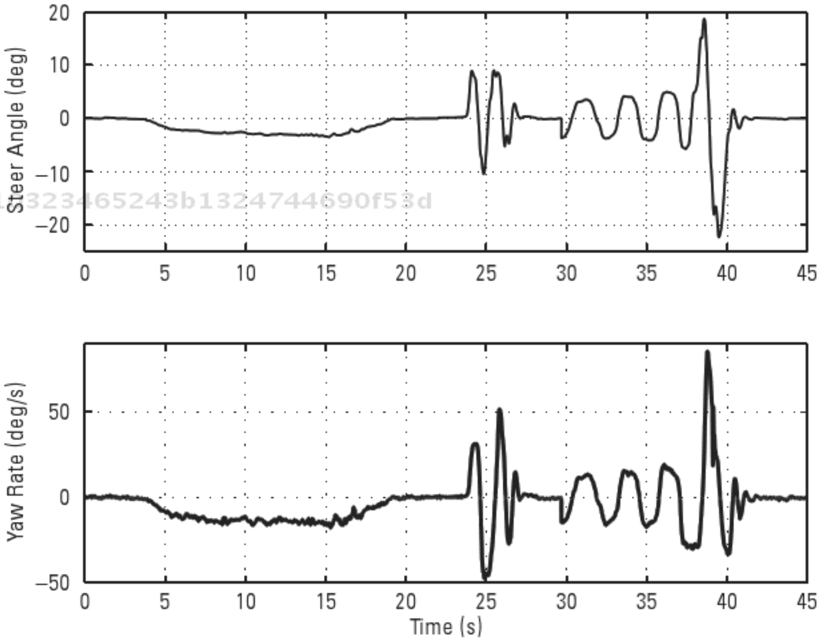


Figure 5.4 Asphalt experiment steer angle and yaw rate.

between the sensors, and not a dynamic model of the vehicle. The state space model for the kinematic Kalman filter is given in (5.3):

$$\begin{matrix} \dot{\psi} \\ \dot{b}_{gyro} \end{matrix} = \begin{matrix} 0 & 1 \\ 0 & 0 \end{matrix} \begin{matrix} \psi \\ b_{gyro} \end{matrix} + \begin{matrix} 1 \\ 0 \end{matrix} r + w$$

$$y = [1 \quad 0] \begin{matrix} \psi \\ b_{gyro} \end{matrix} + \varepsilon \quad \text{or} \quad y = [0 \quad 0] \begin{matrix} \psi \\ b_{gyro} \end{matrix} + \varepsilon \tag{5.3}$$

In the kinematic Kalman filter, the two states being estimated are the heading of the vehicle and the bias on the yaw rate gyroscope which is modeled as a random walk. The yaw rate acquired from the gyroscope is used as an input into the system. During straight driving, the observation matrix (**C**) is set to [1 0] in order to estimate the bias on the gyroscope by using the GPS course measurement. GPS course is the heading of the vehicle plus sideslip which is assumed to be zero during straight driving. The observation matrix is then set to [0 0] during turning or during periods when GPS is lost. Setting the observation matrix to zero during turning effectively turns off the gyro bias estimation and integrates the yaw gyroscope (with the bias removed) in order to provide an estimate of the true vehicle heading. It is important to note that this method can lead to estimation errors in the presence of gyroscope scale factor errors or during periods of long integration. Although the kinematic Kalman filter does not provide a direct estimate of vehicle sideslip, an estimate of sideslip can be obtained by comparing the difference between the GPS course angle and the estimated heading of the vehicle from integrating the bias-free gyro. The equation for this sideslip estimate can be seen in (5.4).

$$\hat{\beta} = \nu_{GPS} \quad \hat{\psi}_{Kf} \tag{5.4}$$

The measurement noise and discrete process noise covariance matrix for the kinematic Kalman filter are given in (5.5):

$$R_d = \sigma^2{}_{gps}$$

$$Q_d = T_s I \begin{matrix} T_s\sigma_r^2 & 0 \\ 0 & \sigma_{b_{gyro}}^2 \end{matrix} I \tag{5.5}$$

Note that the estimator dynamics change with these parameters [6].

Note that although the kinematic Kalman filter is able to account for the bias present on the gyroscope, the method is unable to account for the presence of a scale factor error on the gyroscope. Although methods to estimate the gyroscope's scale factor error exist [7], any error in the gyro scale factor error will lead to a growth in the slip angle estimate error during turning. Also, with a single antenna receiver, the method cannot account for roll or road grade errors, which can lead to a false estimation of sideslip.

5.6 Kinematic Kalman Filter (Dual Antenna)

The single antenna estimation technique presented in the previous section has several drawbacks. Estimating an integrated signal is an unobservable process; mathematically the estimate error cannot be bound. Practically, this means errors such as incorrect biases or scale factors can cause large errors in the sideslip estimate. Another problem is aligning the time that each measurement was taken. Inherent delay occurs between the time a GPS measurement is taken and the time the measurement is received by a computer. Depending on how the gyroscope measurement is taken, it may have a delay as well. Because vehicle sideslip is generally small (on the order of 1–2°), the two signals being subtracted in (5.2) must be aligned at precisely the same time to produce an accurate measurement [5]. Without an external signal (such as the GPS pulse per second), it is not easy to programmatically line up the two measurements.

Therefore, an improved method of estimating the vehicle sideslip angle can be performed using vehicle heading measured using two GPS antennas [8, 9]. By utilizing two antennas, heading estimate errors due to both gyro integration (arising from bias estimate errors, scale factor, integration routine, and integration of a noisy signal) and synchronization of the GPS/INS measurements are eliminated. The GPS velocity measurements (including course) have a delay that is not present in the position measurements (including heading). The GPS velocity vector is calculated by comparing consecutive phase measurements of the carrier wave. When consecutive phase measurements are compared, the measurement has a half sample delay. This delay is similar to that introduced by a backward difference approximation. Unlike the GPS/IMU synchronization, this delay is easy to account for programmatically.

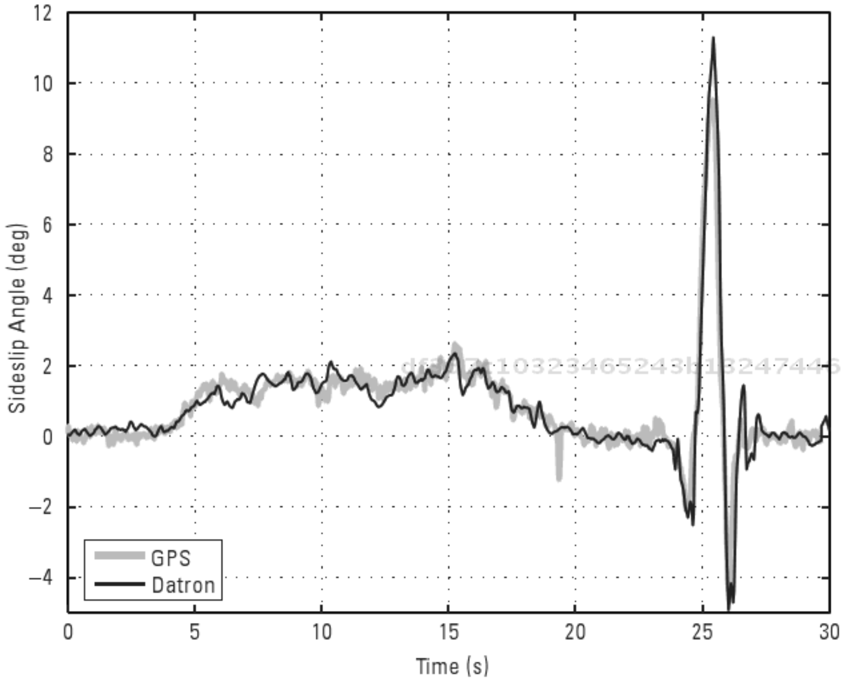


Figure 5.5 Asphalt experiment GPS slip angle validation with optical sensor.

The two methods (GPS and optical) are compared for the asphalt experiment in Figure 5.5 and match quite well; for the sake of scale, the slalom is not included. This validates the GPS method to compute sideslip presented above. Further validation using the optical sensor can be found in [10, 11]. For the remainder of the chapter, the GPS approach will be used for the sideslip measurement.

The first step of the cascaded estimation approach utilizes a kinematic Kalman filter to combine the GPS and inertial measurements to provide high-update rate, unbiased, accurate estimates of the lateral vehicle dynamics [5]. For this filter, the linearized form of the Kalman filter described in the Appendix is used to estimate the states. The models for the gyroscope and lateral accelerometer described in Chapter 1 can be combined with the measurements from a two-antenna GPS system to form the kinematic state space relationships given next.

$$\boldsymbol{f} = \begin{matrix} \dot{\psi} \\ \dot{\beta} \\ \dot{b}_{gyro} \\ \dot{b}_{accel} \end{matrix} = \begin{matrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & \frac{1}{V_x} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{matrix} \begin{matrix} \psi \\ \beta \\ b_{gyro} \\ b_{accel} \end{matrix} + \begin{matrix} 1 & 0 \\ 1 & \frac{1}{V_x} \\ 0 & 0 \\ 0 & 0 \end{matrix} \quad r^{\langle \text{gyro} \rangle} \quad a_y^{\langle \text{corrected} \rangle} \quad + \boldsymbol{w}$$

$$\boldsymbol{y} = \begin{matrix} \psi^{\langle \text{GPS} \rangle} \\ \nu^{\langle \text{GPS} \rangle} \end{matrix} = \begin{matrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{matrix} \begin{matrix} \psi \\ \beta \\ b_{gyro} \\ b_{accel} \end{matrix} + \eta$$

The estimator adjusts the accelerometer bias estimate to compensate for any roll measured by the lateral accelerometer. However, the two-antenna GPS system provides a roll measurement that can be used to correct the lateral acceleration.

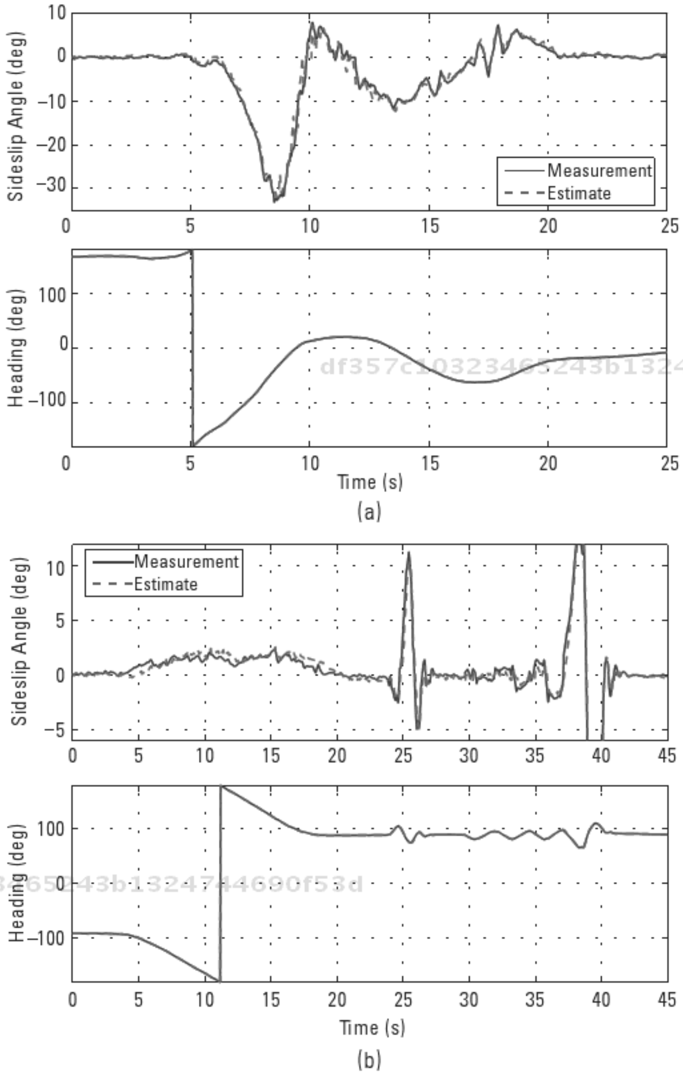
$$a_y^{\langle \text{corrected} \rangle} = a_y^{\langle \text{accel} \rangle} \quad g\sin(\theta) \tag{5.7}$$

The sensor noise and process noise covariance matrices for this system are given here.

$$\boldsymbol{R_d} = \begin{matrix} \sigma_\psi^2 & 0 \\ 0 & \sigma_\nu^2 \end{matrix}$$

$$\boldsymbol{Q_d} = \begin{matrix} T_s^2\sigma_r^2 & T_s^2\sigma_r^2 & 0 & 0 \\ T_s^2\sigma_r^2 & T_s^2 \quad \sigma_r^2 + \frac{\sigma_{a_y}^2}{V_x^2} \div & 0 & 0 \\ 0 & 0 & \sigma_{b_{gyro}}^2 & 0 \\ 0 & 0 & 0 & \sigma_{b_{accel}}^2 \end{matrix} \tag{5.8}$$

T_s is the sample rate of the Kalman filter inputs. Figure 5.6 shows the estimates of sideslip and heading, for the low-μ and asphalt tests described previously.



(a)

(b)

Figure 5.6 (a, b) Estimates using the kinematic Kalman filter.

5.7 Tire Parameter Identification

Using the state estimates from either kinematic Kalman filter, the tire curve for each axle can be generated and tire parameters estimated. Because no small angle approximations are used in the kinematic Kalman filter plant model, the estimates are valid beyond the linear region of the tire. Several researchers have produced experimental tire curves using various methods (such as steering wheel torque) to estimate the lateral force at the tire [12–15]. Other researchers have developed methods to estimate peak coefficient of friction [16, 17], including methods that utilize GPS [18]. In this chapter, simple Newtonian dynamics are used to accomplish the force estimation. The lateral acceleration and yaw rate estimates can be used to determine the front and rear axle lateral forces.

$$\begin{aligned} F_y &= ma_y = F_{yr} + F_{yf}\cos(\delta) \\ M_z &= I_z\ddot{\psi} = \ \ bF_{yr} + aF_{yf}\cos(\delta) \end{aligned} \tag{5.9}$$

Several assumptions are made in these equations. First, longitudinal forces are ignored. The only longitudinal force that might appear is from the front tire due to the steer angle. Unless the vehicle is braking, this force will be negligible for rear wheel drive vehicles (such as the G35). It is possible to add a longitudinal acceleration/force equation and make an assumption about the brake bias to include braking and driving forces. If the vehicle is undergoing significant acceleration or braking, the additional forces should be included so the lateral estimation does not become corrupted by the neglected longitudinal terms. Including longitudinal forces also allows the longitudinal tire curve to be predicted; however, this is beyond the scope of this chapter.

The lateral forces in the equations are per axle, not per tire; the lateral force on an axle is simply the sum of the lateral forces from each of the tires on that axle. The equations include the second derivative of heading. To determine this measurement, the yaw rate gyroscope must be numerically differentiated; the differentiation may add lag and noise that can corrupt the estimation. Using the tire sideslip measurements presented above and these lateral forces, it is possible to predict the lateral tire curves for the vehicle. The tire curves can then be used to estimate parameters for the various tire models available.

Although many researchers have estimated tire cornering stiffness online using linear tire models, to predict the tire behavior outside the linear region, a more complicated model such as the Dugoff model, discussed in Chapter 3, must be used. The Dugoff model has two parameters to identify for each tire: cornering stiffness and peak lateral force (peak coefficient of friction multiplied by vertical load). As with the linear model, it is safe to assume the inner and outer cornering stiffnesses are equal; the inner and outer coefficients of friction are also equal. However, this model includes (and is not linear in) vertical force; assuming no weight transfer is not always practical. To allow the model to capture a wider range of vehicle behavior, a more complicated tire model may be used, such as Pacjeka or Fiala models.

Because the Dugoff model is not linear in vertical force, it is not immediately apparent mathematically that it is possible to identify a "lumped" peak force for the axle. This problem can be visualized in the transition region of the model. With this model it is possible to have the outer tire in the linear region ($\lambda > 1$) and the inner tire beginning to saturate ($\lambda < 1$) due to more vertical load on the outer tire. However, it is normally safe to assume that the concept of the Dugoff model works for an axle as well as for a tire; the axle will have a cornering stiffness and a peak lateral force. The Dugoff model of the axle captures these two characteristics. This assumption may break down in vehicles with large roll angles, however, particularly if the roll dynamics are not significantly faster than the yaw dynamics (i.e., the weight transfer for a given slip angle cannot be considered constant). In this case, the peak lateral force for the axle would not be constant.

For vehicles with large amounts of roll, this presents opportunities as well as difficulties. If cornering stiffness and coefficient of friction are assumed equal for the inner and outer tires and the total vertical load on all the tires is constant (vehicle weight), it should be possible to identify the vertical load on each tire as well as the tire parameters by summing the lateral forces for each axle. However, the vehicle used in this chapter does not exhibit significant weight transfer; thus, there is not enough excitation to identify the tire loads separately.

Because the Dugoff model is nonlinear, simple techniques such as recursive least squares (RLS) do not work to identify the parameters. However, researchers have used a modified Dugoff model to incorporate RLS [19]. A more complicated nonlinear technique, such as an extended Kalman filter, must be used for the nonlinear Dugoff model, where the states (i.e., parameter estimates) vary with time. Both the Kalman filter and extended Kalman filter are presented in the Appendix.

$$\begin{aligned} \boldsymbol{f} &= \begin{matrix} \dot{C}_\alpha \\ (\dot{\mu} F_z) \end{matrix} = \boldsymbol{w} \\ \boldsymbol{y} &= [F_y] = \boldsymbol{C} \begin{matrix} C_\alpha \\ (\mu F_z) \end{matrix} + \eta \end{aligned} \tag{5.10}$$

The measurement is lateral force, so the output matrix ($\boldsymbol{C}$) is the gradient of the Dugoff model with respect to the parameters.

$$\lambda(t_k) = \frac{(\mu F_z)_{k\ \ 1}}{2C_{\alpha k\ \ 1}\left|\tan(\alpha_k)\right|}$$

$$\boldsymbol{C}(t_k) = \begin{matrix} \dfrac{(\mu F_z)^2_{k\ \ 1}}{4C^2_{\alpha k\ \ 1}\tan(\alpha_k)} & \dfrac{(\mu F_z)_{k\ \ 1}}{2C_{\alpha k\ \ 1}\tan(\alpha_k)} & \text{sign}(\tan(\alpha_k)) & \text{if } \lambda(t_k)\ \ 1 \\ & [\ \ \tan(\alpha_k) \quad 0] & & \text{if } \lambda(t_k) > 1 \end{matrix} \tag{5.11}$$

The process noise covariance ($\boldsymbol{Q}_d$) acts like a forgetting factor, allowing the estimated parameters to change over time. For this work, the process and measurement noise covariances are chosen to be

$$\boldsymbol{R}_d = (1{,}000\ \text{N})^2$$

$$\boldsymbol{Q}_d = 10^{\ \ 8} \begin{matrix} (80{,}000\ \ \text{N/rad})^2 & 0 \\ 0 & (15{,}000\ \text{N})^2 \end{matrix} \tag{5.12}$$

Another way to think of the recursion algorithm is as recursive least squares where the regression matrix is defined like the output matrix in the Kalman filter.

The parameter identification algorithm is validated on the two experimental runs described above. Figure 5.7 shows the asphalt experiment. There

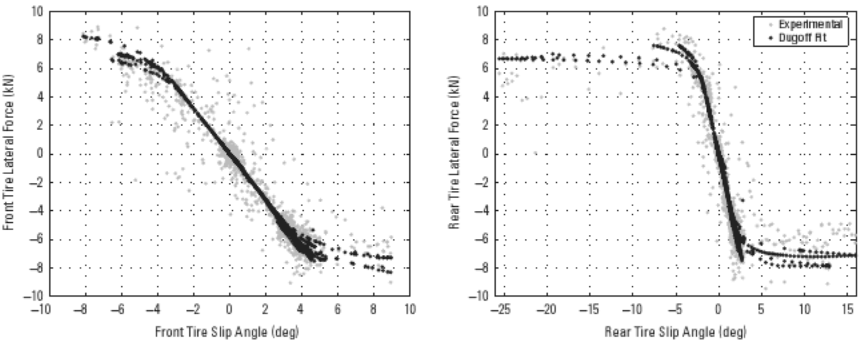


Figure 5.7 Estimated tire curves from the asphalt experiment.

appears to be multiple fits due to the estimation algorithm adapting over time. The raw data also seem to have this behavior implying the parameters are varying with time. In particular, the rear tire saturates at approximately 7 kN but distinctly goes above this lateral force at 5° of slip. This discrepancy could be due to weight transfer. Slip angles beyond 7° are only seen near the end of the slalom portion. This maneuver is violent enough that weight transfer was significant. The other two portions of the experiment (steady state cornering and lane changes) are not that violent and thus do not have the weight transfer. The estimation algorithm is adaptive enough to capture the different behaviors at different times during the experiment.

Figure 5.8 shows the parameter estimation throughout the asphalt experiment. The Kalman filter (recursive) is compared to a nonlinear batch least squares performed over the entire experiment. For the most part, the recursion converges near the batch estimation. The notable exception to this is the rear cornering stiffness; the recursion remains lower than the batch estimation. Although this difference looks significant as plotted, the difference in slope represented on the tire curve is not very noticeable. Either could be judged as a valid fit.

The peak lateral force estimate portrays an aspect of the Kalman filter that is quite useful in this application. Most parameter identification schemes require a switch to determine when to estimate solely the cornering stiffness versus estimating both parameters (i.e., when the tire is beginning to saturate). When the tire is in the linear region ($\lambda > 1$), the Dugoff model (and thus the Kalman filter output matrix) does not contain the peak force; this parameter is unobservable. When the Kalman filter encounters an unobservable state, it sets the gain on that state to zero in this case, meaning hold the parameter constant. Therefore, when the tire is operating in the linear region,

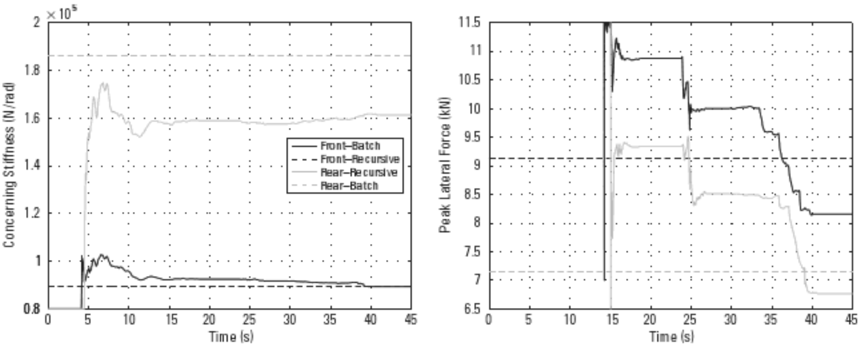


Figure 5.8 Estimated tire parameters from the asphalt experiment.

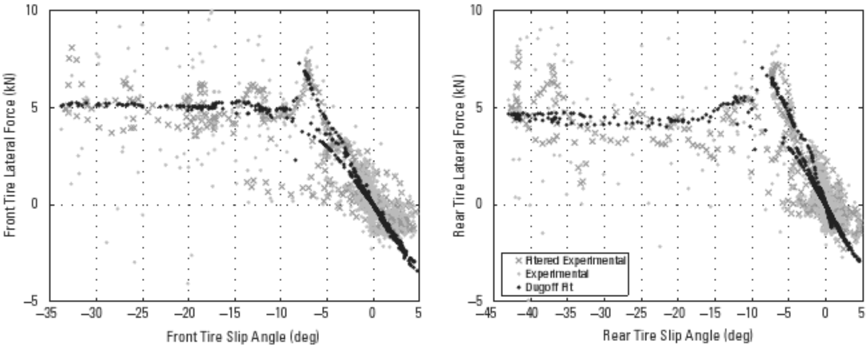


Figure 5.9 Estimated tire curves from the low-μ test.

the Kalman filter automatically holds the peak force at the last estimated value. This is the reason for the many plateaus in the peak force estimate. The tire only enters the nonlinear region for short periods at a time; the first of these is exiting the corner. During these short intervals, the peak force estimate is updated; otherwise, it is simply held constant.

Figure 5.9 shows the measured tire curve along with the estimated Dugoff fit for the low-μ experiment and Figure 5.10 shows the resulting parameter estimates (note the difference in both the cornering stiffness and peak force from the asphalt experiments). Filtered measurements are shown along with the unfiltered measurements and curve fit; the curve fit utilizes the unfiltered measurements. The reason for the filtered measurements is to show graphically that the curve fit appears to identify the most reasonable parameters for the tire. As with the asphalt experiment, the parameters appear to vary with time (there appear to be multiple curve fits); this is due in part to

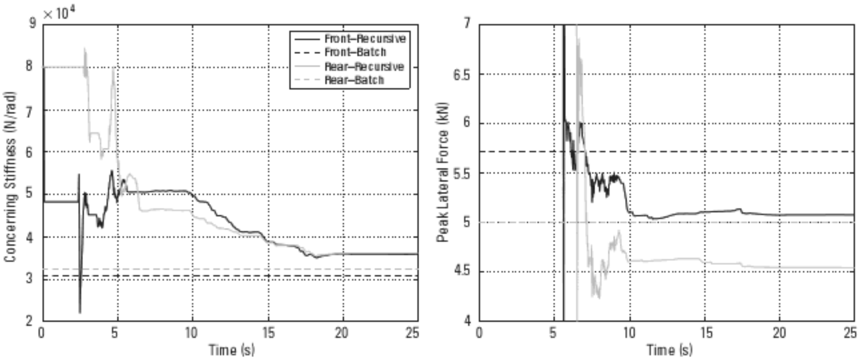


Figure 5.10 Estimated tire parameters from the low-μ test.

weight transfer as it was in the asphalt experiment. The changing parameters are also due in part to the irregularity in the gravel surface itself. In addition, the measured force (even after filtering) is very noisy due to the rough gravel surface, but the estimation scheme effectively averages the noise out. The parameters identified in this experiment are successfully used in the following model-based filter, showing that even in the presence of a large signal-to-noise ratio the identification and estimation schemes perform well together.

Another important consideration when estimating the tire parameters is the initial guess of peak force. To obtain an accurate estimate, the initial peak force guess must be higher than the actual value. In an extreme sense, if the peak force estimate is too low, the estimator is constantly receiving measurements of lateral force above the peak and so continually moves the estimate of peak force up to the measured value. However, the Kalman filter also believes the estimate cannot change quickly (based on $\boldsymbol{Q}_d$), and so it becomes "harder" to move the estimate. Ultimately, the estimate of peak force will always be lower than the truth and even the highest measured forces.

This problem is easy to circumvent by initially guessing the peak force unreasonably high. This does not cause any problems in the estimator because while the tire is in the linear region, the peak force does not come into play. As soon as the tire begins to saturate for the first time (t = 15s in Figure 5.8), the peak force estimate drops immediately to a reasonable value.

5.8 Model-Based Kalman Filter

As discussed in Chapter 3, the understeer gradient (K_{US}) defines whether the vehicle is understeer ($K_{US} > 0$), oversteer ($K_{US} < 0$), or neutral steer ($K_{US} = 0$). The vehicle's understeer characteristics determine the change in steer angle required to hold a steady state turn as velocity increases [20]. In a neutral steer vehicle the understeer gradient is zero; the state transition matrix from (3.18) simplifies to

$$\boldsymbol{A} = \begin{matrix} \dfrac{C_{\alpha f} \quad C_{\alpha r}}{mV} & 1 \\ 0 & \dfrac{a^2 C_{\alpha f} \quad b^2 C_{\alpha r}}{I_z V} \end{matrix} \tag{5.13}$$

Note this allows yaw rate to be independent of sideslip, but sideslip is still dependent upon yaw rate. This results in the sideslip being unobservable with only a yaw rate measurement.

5.8.1 Linear Tire Model

Once the tire parameters have been identified, the GPS measurements can be combined with the inertial sensors using a model based estimator. The estimator's state space model is derived from the linearized bicycle model given in (3.18). This Kalman filter form is called a model-based estimator because the vehicle model is used in the state matrices for the estimator. The state space equations for the model-based estimator, including the heading and gyro bias, are shown in (5.14).

$$
\begin{array}{cccccccccccc}
 & \dot{\beta} & & \frac{C_{\alpha f}\quad C_{\alpha r}}{mV} & \frac{aC_{\alpha f}+bC_{ar}}{mV^2} & 1 & 0 & 0 & 0 & \beta & & \frac{C_{af}}{mV} \\
 & \dot{r} & & & & & & & & r & & \\
\boldsymbol{f}= & \dot{\psi} & = & \frac{aC_{\alpha f}+bC_{\alpha r}}{I_z} & \frac{a^2C_{\alpha f}\quad b^2C_{\alpha r}}{I_zV} & & 0 & 0 & 0 & \psi & + & \frac{aC_{\alpha f}}{I_z} \quad \delta+\boldsymbol{w} \\
 & \dot{b}_{gyro} & & 0 & 1 & & 0 & 0 & 0 & b_{gyro} & & 0 \\
 & \dot{b}_{accel} & & 0 & 0 & & 0 & 0 & 0 & b_{accel} & & 0 \\
 & & & 0 & 0 & & 0 & 0 & 0 & & & 0
\end{array}
$$

$$
\begin{array}{ccc}
 & \beta & \\
 & r & \\
\boldsymbol{y}=\boldsymbol{C} & \psi & +\boldsymbol{D}\delta+ \\
 & b_{gyro} & \\
 & b_{accel} &
\end{array}
\qquad (5.14)
$$

The output equations for this model (as well as which states are observable) depend on the sensors used. Table 5.1 provides the various states (and the associated *C* and *D* matrices) that are observable given the measurements available to the model based estimator. Farrelly has shown [21] how to incorporate a lateral accelerometer measurement into the model based estimator that results in the following state space description of the system. If used, the lateral accelerometer measurement must be compen-

Table 5.1
Kalman Filter Sensor Options

Case	Sensors	*C* Matrix	*D* Matrix	Observable
1	1 ant GPS	$\begin{bmatrix} 1 & 0 & 1 & 0 & 0 \end{bmatrix}$	$[0]$	β, r, ψ
2	1 ant GPS, yaw gyro	$\begin{matrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{matrix}$	$\begin{matrix} 0 \\ 0 \end{matrix}$	β, r, ψ, b_{gyro}
3	2 ant GPS	$\begin{matrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{matrix}$	$\begin{matrix} 0 \\ 0 \end{matrix}$	β, r, ψ
4	2 ant GPS, yaw gyro	$\begin{matrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{matrix}$	$\begin{matrix} 0 \\ 0 \\ 0 \end{matrix}$	β, r, ψ, b_{gyro}
5	yaw gyro	$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 \end{bmatrix}$	$[0]$	β, r, b_{gyro}*
6	yaw gyro, lat accel	$\begin{matrix} 0 & 1 & 0 & 1 & 0 \\ \frac{C_{\alpha f} \quad C_{\alpha r}}{m} & \frac{aC_{\alpha f} + bC_{ar}}{mV} & 0 & 0 & 1 \end{matrix}$	$\begin{matrix} 0 \\ \frac{C_{af}}{m} \end{matrix}$	$\beta, r, b_{gyro},$ b_{accel}
7	1 ant GPS, lat accel	$\begin{matrix} 1 & 0 & 1 & 0 & 0 \\ \frac{C_{\alpha f} \quad C_{\alpha r}}{m} & \frac{aC_{\alpha f} + bC_{ar}}{mV} & 0 & 0 & 1 \end{matrix}$	$\begin{matrix} 0 \\ \frac{C_{af}}{m} \end{matrix}$	$\beta, r, \psi, b_{accel}$
8	1 ant GPS, yaw gyro, lat accel	$\begin{matrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ \frac{C_{\alpha f} \quad C_{ar}}{m} & \frac{aC_{\alpha f} + bC_{ar}}{mV} & 0 & 0 & 1 \end{matrix}$	$\begin{matrix} 0 \\ 0 \\ \frac{C_{af}}{m} \end{matrix}$	$\beta, r, \psi, b_{gyro},$ b_{accel}
9	2 ant GPS, lat accel	$\begin{matrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \frac{C_{\alpha f} \quad C_{ar}}{m} & \frac{aC_{\alpha f} + bC_{ar}}{mV} & 0 & 0 & 1 \end{matrix}$	$\begin{matrix} 0 \\ 0 \\ \frac{C_{af}}{m} \end{matrix}$	$\beta, r, \psi, b_{accel}$
10	2 ant GPS, yaw gyro, lat accel	$\begin{matrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \frac{C_{\alpha f} \quad C_{\alpha r}}{m} & \frac{aC_{\alpha f} + bC_{ar}}{mV} & 0 & 0 & 1 \end{matrix}$	$\begin{matrix} 0 \\ 0 \\ 0 \\ \frac{C_{af}}{m} \end{matrix}$	$\beta, r, \psi, b_{gyro},$ b_{accel}

*If the vehicle is not neutral steer.

sated for roll as shown in (5.7). Additionally, if performing this state estimator in conjunction with the navigation estimator in Chapter 4, the bias compensated gyro can be used for the yaw rate measurement. This eliminates the need to reestimate the gyro bias and also allows for the estimation of a steer angle bias.

These estimators provide a direct estimate of the sideslip (β), yaw rate (r), and vehicle heading (ψ). Some configurations estimate the gyroscope and/or accelerometer biases (b_{gyro} and b_{accel}). The input into the estimators is the steer angle at the wheel (δ). Measurements include yaw rate plus bias (from a gyroscope), the course angle (from GPS), vehicle heading (from two antenna GPS), and lateral acceleration plus bias (from a lateral accelerometer). Note, as shown in (5.13), vehicle sideslip is not observable using only a yaw rate measurement if the vehicle model is neutral steer. However, sideslip remains observable with GPS measurements for a neutral steer vehicle.

The discrete processes noise covariance is shown in (5.15).

$$
\boldsymbol{Q}_d = \begin{array}{ccccc}
\left(\frac{0.1\pi\ \text{rad}}{180}\right)^2 & 0 & 0 & 0 & 0 \\
0 & \frac{0.1\pi\ \text{rad/s}}{180}\div^{2} & 0 & 0 & 0 \\
0 & 0 & \left(\frac{0.1\pi\ \text{rad}}{180}\right)^2 & 0 & 0 \\
0 & 0 & 0 & \left(10^{\ 5}\ \text{rad/s}\right)^2 & 0 \\
0 & 0 & 0 & 0 & \left(10^{\ 5}\ \text{m/s}^2\right)^2
\end{array} \tag{5.15}
$$

Note that the actual transformation from process noise on the steer angle to state process noise must propagate through the input matrix. However, it may be better to "hand-tune" the process noise covariance matrix to account for model parameter errors in the estimator model, resulting in the covariance above. Also, note that the estimator dynamics will change with the magnitude of the noise on the bias states, but that change is not a focus of this chapter. The measurement noise covariance depends on the sensors used; like the other measurement noise covariance shown in the chapter, it is diagonal and made up of the individual sensor noise variances.

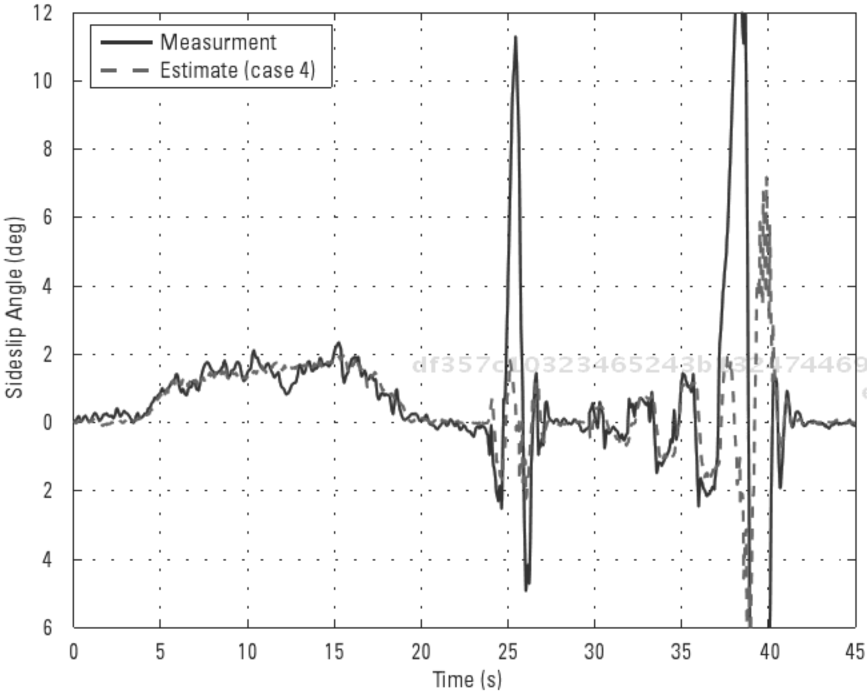


Figure 5.11 Sideslip estimate during the asphalt experiment using the linear model-based estimator (case 4).

Figure 5.11 shows the results of the sideslip estimation using the linear model-based estimator with the two antenna GPS measurements and a yaw rate gyro (Case 4). As seen in the figure, the linear estimator does a good job of estimating the sideslip around the turn of the track, even in the presence of the 8° road bank, and during the first part of the lane change and slalom maneuvers. However, as would be expected with a linear estimator, it does not perform adequately during the more extreme maneuvers when the vehicle begins to operate in the nonlinear region of the tire.

5.8.2 Nonlinear Tire Model

The model-based estimator can be improved by using the Dugoff (or other nonlinear) tire model [22–24] to more accurately capture the nonlinear effects of the tire. The full nonlinear dynamic model given in (3.8) and (3.9), in conjunction with the Dugoff tire model given in (3.45) through (3.48), are used in the state estimation continuous propagation step of the estimator, (A.92) (i.e., to calculate f). For the remainder of the estimator steps, instead of computing the Jacobian of the full nonlinear model, (A.8), the state transition matrix is approximated. For each tire, the slope of the Dugoff model is computed at the current slip angle (α_o). These slopes become the cornering stiffnesses in the state transition matrix; see (3.18). The slope of the Dugoff tire model is

$$C_\alpha(\alpha) = \left.\frac{dF_y}{d\alpha}\right|_{\alpha_o} = \begin{array}{ll} \dfrac{\mu^2 F_z^2}{4C_\alpha}\csc^2(\alpha) & \text{if } \lambda \quad 1 \\ & \\ C_\alpha \sec^2(\alpha) & \text{if } \lambda > 1 \end{array} \tag{5.16}$$

Figure 5.12 shows the slalom portion of the asphalt run compared to the simulated vehicle model using the Dugoff tire model. The peak force

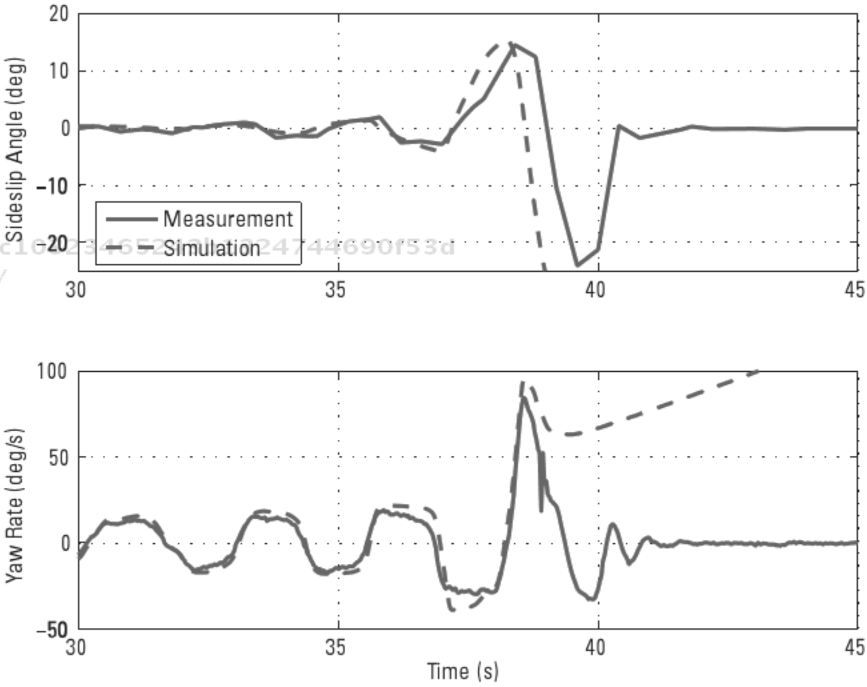


Figure 5.12 Sideslip and yaw rate during slalom section of the asphalt experiment.

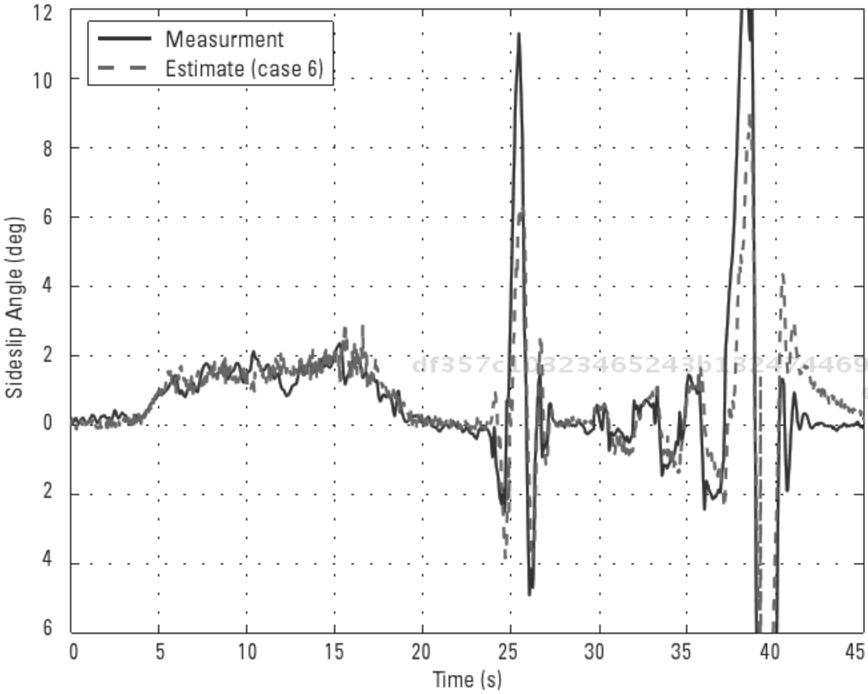


Figure 5.13 Sideslip estimate (case 6) during asphalt experiment using model estimation with only yaw rate and lateral acceleration measurements.

and tire cornering stiffness are taken from the nonlinear tire curve estimation shown in Figure 5.8. In simulation, the vehicle model with the Dugoff tire parameters is able to capture the initial slide of the vehicle (as seen in the sideslip and yaw rate plots), but becomes unstable at the end of the run. However, as seen in Figure 5.13 and Figure 5.14, the nonlinear model-based estimator is still able to accurately estimate the states through the entire maneuver. Figure 5.13 shows the estimation results using only the yaw rate and lateral acceleration measurements (case 6), while Figure 5.14 shows the estimation results with the addition of the two antenna GPS measurements (case 4). Note the improvement of the sideslip estimates with the addition of the GPS measurements.

To test the robustness of the model-based estimator, it is run with incorrect vehicle parameters. The vehicle parameters are chosen such that the incorrect vehicle model has the same understeer gradient, (3.33), as the G35 [since vehicle understeer is fairly easy to estimate online; as seen in Figure

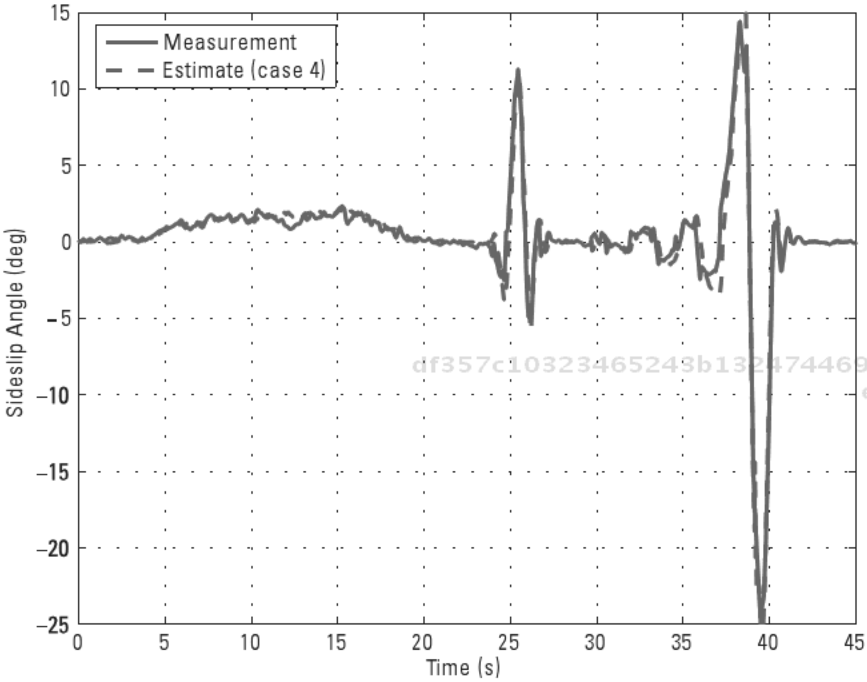


Figure 5.14 Sideslip estimate (case 4) during asphalt experiment using the two-antenna GPS system.

5.15, the model estimator (case 4) with the incorrect model is able to accurately estimate the yaw rate through the turn, lane change] and part of the slalom. However, as seen in the figure, the sideslip estimation error is quite significant. Figure 5.16 shows the measurement residuals (ε_x) from the Kalman filter estimator as described in the Appendix, (A.89). Note that the yaw rate measurement residual appears to be fairly random and centered about zero. However, inspection of the GPS measurement residuals provides a clear indication that there are errors in the Kalman filter model. This shows the benefit of the GPS measurements (from one- or two-antenna systems) to provide a level of confidence in the estimated states and the model associated with the estimator. Additionally, these figures demonstrate the importance of accurate knowledge of the front and rear tire parameters and not simply the understeer gradient of the vehicle.

Finally, the nonlinear model-based estimator is tested on the low-μ gravel surface described previously. Figure 5.17 shows the results of the

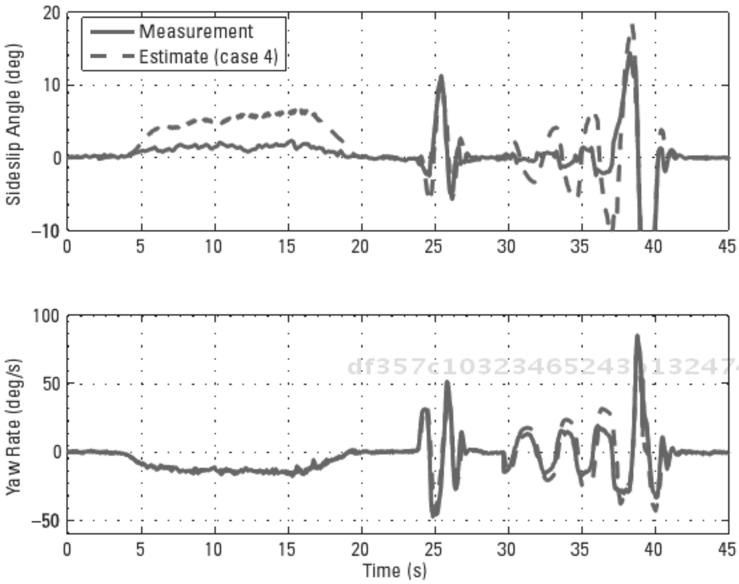


Figure 5.15 Sideslip estimate during a series of maneuvers using an incorrect model (case 4).

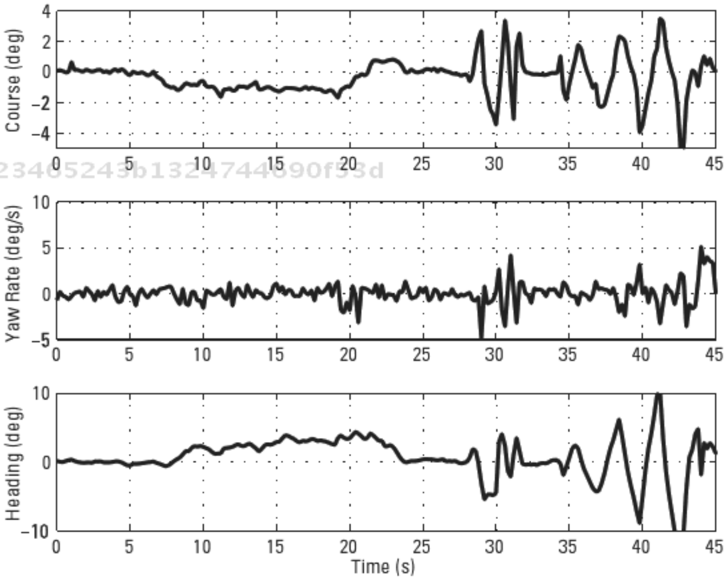


Figure 5.16 Measurement residuals during a series of maneuvers using an incorrect model (case 4).

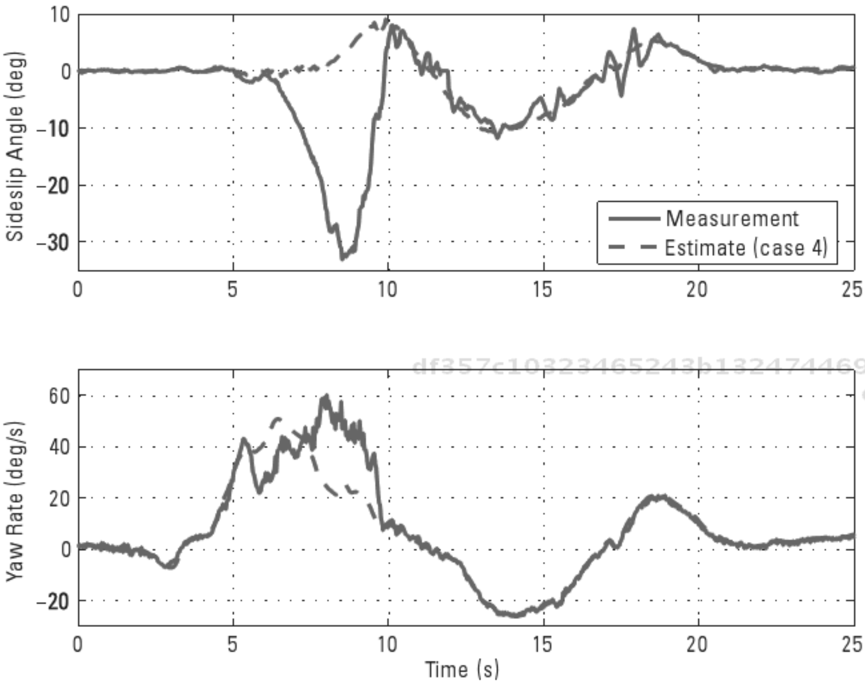


Figure 5.17 Sideslip estimate (case 4) during a hard right turn on a low-μ surface using incorrect Dugoff tire parameters.

nonlinear model-based estimator (case 4) using the Dugoff tire parameters from the asphalt experiment. Notice that although the estimator provides a fairly good estimate of yaw rate during this experiment, it does not capture the large sideslip from the vehicle slide on the low-μ surface. This demonstrates the importance of accurate peak tire force in the model based estimators. Figure 5.18 shows the sideslip estimation (cases 4 and 6) results using the correct Dugoff low-μ tire parameters, estimated in Figure 5.10. The estimation results with only inertial sensor measurements (yaw rate and lateral accelerometer) compared to the results with the two antenna GPS measurements are shown. Note that even with the correct model parameters, the yaw rate gyroscope and lateral accelerometer are unable to estimate the large slip angle produced in this maneuver. However, the addition of the GPS measurements allows the model based estimator to accurately capture the sideslip angle.

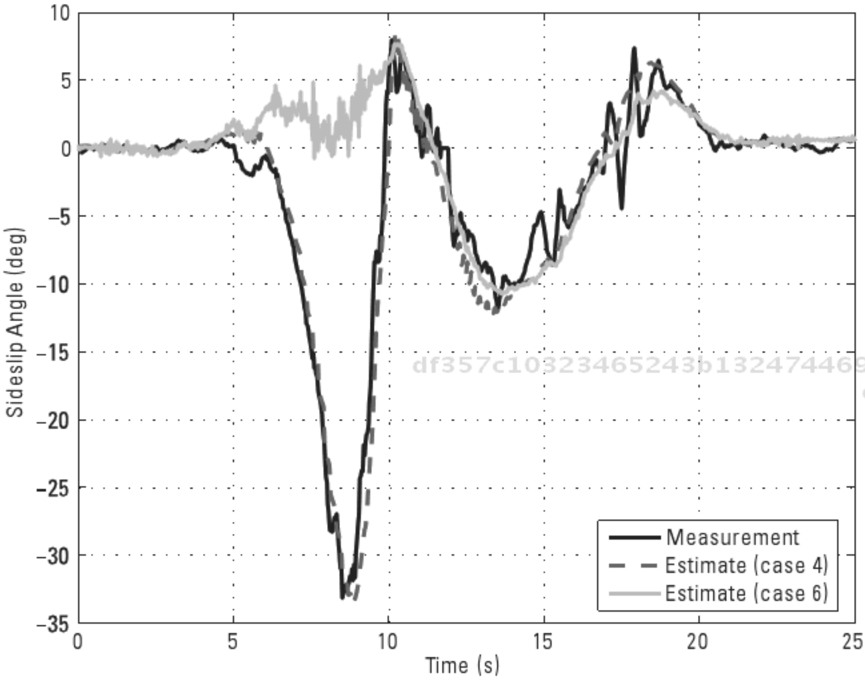


Figure 5.18 Sideslip estimate (cases 4 and 6) during a hard right turn on a low-μ surface using the correct Dugoff tire parameters.

5.8.3 Estimator Accuracies

To show the benefit of identifying the tire parameters and reestimating the vehicle states with a model-based estimator, a simple covariance analysis can be performed to determine the accuracy of the various estimation algorithms [11]. A summary of the covariance analysis (given in terms of the 1-σ values) is shown in Table 5.2. It is important to note that the covariance values for the kinematic Kalman filter assume there are no unmodeled sensor dynamics in the system (since the kinematic Kalman filter only depends on integration of sensor measurements). Also, note that the accuracy of these estimates will improve with speed as seen in (1.9). To improve the overall accuracy of the estimates, more comprehensive kinematic relationships could be added to reduce errors induced by roll and pitch.

Table 5.2
Covariance Analysis Summary

Estimated State (Sensors Used)	Two-Antenna (KKF)	Model-Based Kalman Filter
Vehicle Heading (GPS only)	0.4°	0.03°
Vehicle Heading (GPS and yaw gyro)	0.1°	0.01°
Vehicle Course (GPS only)	0.36°	0.36°
Vehicle Slip Angle (GPS only)	0.54°	0.25°
Vehicle Slip Angle (GPS and yaw gyro)	0.37°	0.2°
Vehicle Slip Angle (GPS, yaw gyro, and lat accel)	0.28°	0.05°

(Max 1-σ Values) at 8 m/s.

The covariance values given for the model-based Kalman filter assume there are no unmodeled vehicle or sensor dynamics in the system (since the model-based Kalman filter depends on the integration of sensor measurements with the vehicle model). The analysis given in Table 5.2 is dependent primarily on the INS sensors, such as sensor noise and bias stability, and not the vehicle model. Also note that while it is not discussed here, the model based estimator provides better estimates in the presence of gyroscopic scale factor errors (if the correct vehicle parameters are used) or in the presence of banked roads [25, 26].

5.9 Conclusions

This chapter has shown the ability of a variety of estimators (utilizing GPS and inertial measurements) to provide accurate estimates of vehicle states for use in vehicle control systems. A method for using these measurements to obtain Dugoff tire parameters was provided. The method was tested using experimental data on high and low friction surfaces to verify the algorithms were robust to parameters changing with ground conditions. A covariance analysis provided the theoretically achievable accuracy, and results showed an improvement when using the estimates from the model based estimator over a kinematic Kalman filter. These improved state estimates can be used to develop a more reliable and robust ESC system. Although the model based estimator can provide vehicle information during a GPS outage, the kinematic Kalman filter can only provide reliable sideslip information for short periods without GPS measurements. Future work will investigate how this impacts the model-based Kalman filter as well as other methods to improve sideslip estimation.

Acknowledgments

The authors would like to thank Nissan Technical Center, North America, for the loan of the Infiniti G35 sedan for the vehicle tests. Additionally, the authors would like to thank Delphi for the loan of the Datron velocity sensor used for validating some of the GPS/INS sideslip measurements. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of Nissan or Delphi.

References

[1] Bevly, D. M., and B. W. Parkinson, "Cascaded Kalman Filters for Accurate Estimation of Multiple Biases, Dead-Reckoning Navigation, and Full State Feedback Control of Ground Vehicles," *IEEE Journal of Control Systems Technology*, Vol. 15, No. 2, 2007.

[2] Bevly, D. M., J. C. Gerdes, and C. Wilson, "The Use of GPS Based Velocity Measurements for Measurement of Sideslip and Wheel Slip," *Vehicle System Dynamics*, Vol. 38, No. 2, 2002, pp. 127–147.

[3] Heydinger, G. J., et al., "Measured Vehicle Inertial Parameters—NHTSA's Data Through November 1998," SAE Paper 1999-01-1336, 1999.

[4] Bevly, D. M., et al., "The Use of GPS Based Velocity Measurements for Improved Vehicle State Estimation," *Proceedings of the American Control Conference 2000*, Chicago, IL, June 2000.

[5] Bevly D. M., R. Sheridan, and J. C. Gerdes, "Integrating INS Sensors with GPS with GPS Velocity Measurements for Continuous Estimation of Vehicle Sideslip and Tire Cornering Stiffness," *Proceedings of the American Control Conference*, 2001.

[6] Bevly, D. M., "GPS: A Low Cost Velocity Sensor for Correcting Inertial Sensor Errors on Ground Vehicles," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 126, No. 2, June 2004.

[7] Ryu, J., and J. C. Gerdes, "Integrating Inertial Sensors with Global Position System for Vehicle Dynamics Control," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 126, No. 2, June 2004.

[8] Hong, S., J. U. Choi, and J. L. Speyer, "Estimation of Errors in INS with Multiple GPS Antennas," *Proceedings of the 2001 Annual Conference of the IEEE Industrial Electronics Society*, 2001.

[9] Ryu, J., E. J. Rossetter, and J. C. Gerdes, "Vehicle Sideslip and Roll Parameter Estimation Using GPS," *Proceedings of the International Symposium on Advanced Vehicle Control (AVEC),* Hiroshima, Japan, 2002, pp. 373–380.

[10] Rock, K. L., et al., "Validating GPS Based Measurements for Vehicle Control," *Proceedings of IMECE 2005*, Orlando, FL, November 2005.

[11] Bevly, D. M., R. Daily, and W. Travis, "Estimation of Critical Tire Parameters Using GPS Based Sideslip Measurements," SAE Paper 2006-01-1965, 2006.

[12] Koo, S., H. Tan, and M. Tomizuka, "Nonlinear Tire Lateral Force Versus Slip Angle Curve Identification," *Proceedings of the American Control Conference*, 2004.

[13] Switkes, J. P., and J. C. Gerdes, "Guaranteed Lanekeeping Performance with Tire Saturation Using Computed Polynomial Lyapunov Functions," *Proceedings of IMECE 2005*, Orlando, FL, November 2005.

[14] Hsu, Y. J., and J. C. Gerdes, "Stabilization of a Steer-by-Wire Vehicle at the Limits of Handling Using Feedback Linearization," *Proceedings of IMECE 2005*, Orlando, FL, November 2005.

[15] M'Sirdi, N. K., et al., "Vehicle-Road Interaction Modeling for Estimation of Contact Forces," *Vehicle System Dynamics*, Vol. 43, 2005, pp. 403–411.

[16] Ray, L., "Nonlinear State and Tire Force Estimation for Advanced Vehicle Control," *IEEE Transactions on Control System Technology,* Vol. 3, No. 1, 1995, pp. 117–124.

[17] Müller, S., M. Uchanski, and K. Hedrick, "Estimation of the Maximum Tire-Road Friction Coefficient," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 125, 2003, pp. 607–617.

[18] Hahn, J., and R. Rajamani, "GPS-Based Real-Time Identification of Tire-Road Friction Coefficient," *IEEE Transactions on Control Systems Technology,* Vol. 10, No. 3, 2002, pp. 331–343.

[19] Kwak, B., Y. Park, and D. Kim, "Design of Observer for Vehicle Stability Control System," *FISITA World Automotive Congress*, 2000.

[20] Gillespie, T. D., *Fundamentals of Vehicle Dynamics*, Warrendale, PA: Society of Automotive Engineers, 1992.

[21] Farrelly, J., and P. Wellstead, "Estimation of Vehicle Lateral Velocity," *Proceedings of the 1996 IEEE Conference on Control Applications*, 1996.

[22] Dugoff, H., P. S. Fancher, and L. Segel, "An Analysis of Tire Traction Properties and Their Influence on Vehicle Dynamic Performance," SAE Paper 700377, pp. 1219–1243.

[23] Fiala, E., "Lateral Forces on Rolling Pneumatic Tires," *Zeitschrift V. D. I.,* No. 29:96, 1954.

[24] Pacejka, H. B., and E. Bakker, “The Magic Formula Tyre Model,” *Vehicle System Dynamics,* Vol. 21, No. 1, 1992, pp. 1–18.

[25] Anderson, R., and D. M. Bevly, “Estimation of Slip Angles Using a Model Based Estimator and GPS,” *Proceedings of the American Control Conference,* 2004, pp. 2122–2127.

[26] Anderson, R., “Using GPS For Model Based Estimation of Critical Vehicle States and Parameters,” M.S. Thesis, Department of Mechanical Engineering, Auburn University, December 2004.

6

GNSS Control of Ground Vehicles

David W. Hodo

6.1 Introduction

Several different methods of vehicle control are commonly used for autonomous vehicles. In this chapter, vehicle models presented in Chapter 3 are used to develop path-following controllers using both classical and modern (state-space) methods. Control methods are presented for two different types of paths. The first path type is composed of discrete locations or waypoints for the vehicle to drive to. The second uses a continuous function to define the path. A controller is developed to follow continuous paths consisting of lines and circular arcs, but the method is easily extensible to any path defined by a closed-form, continuous function. Finally a model for a vehicle towing a trailer or implement (e.g., farm tractor, 18-wheeler, and so forth) is given and a controller is developed for path following of the trailer.

6.2 Vehicle Model

An accurate vehicle model is required before a vehicle can be controlled. The majority of autonomous vehicles fall into two categories: Ackermann steered and differential drive. An Ackermann-steered vehicle is one with steerable front wheels such as a standard passenger car or truck. A differential drive vehicle is one with nonsteerable wheels or tracks, such as a tank, bulldozer, or many small robotics platforms. The inputs for an Ackermann-steered vehicle are typically acceleration and a steering wheel input. The inputs for a differential drive vehicle are typically wheel velocity commands or wheel motor voltages.

Models can be derived for both vehicle types between the vehicle's inputs and the vehicle's forward velocity and yaw rate. Once a model between the inputs, velocity, and yaw rate is found, the differences in the vehicle types can be ignored. Several vehicle models for both categories are developed in Chapter 3. The methods discussed in this chapter can be used with any of the vehicle models previously presented. For demonstration purposes, a four-wheeled, Ackermann-steered vehicle is assumed.

A basic vehicle diagram for an Ackermann steered vehicle is shown in Figure 6.1. The vehicle's center of gravity (CG) is denoted by the black circle in the center of the vehicle. The distances between the front tire's contact patch and the CG and the distance from the CG to the rear tire's contact patch are given by the distances *a* and *b*, respectively. The angle between the direction the vehicle is pointing and the front wheels are pointing (steering angle) is denoted by the angle δ. The vehicle's heading (the direction the vehicle is pointed) is denoted by the angle ψ and is measured clockwise from due north. The vehicle's yaw rate is given by the angular velocity *r*. The longitudinal velocity is given by V_x and the lateral velocity by V_y.

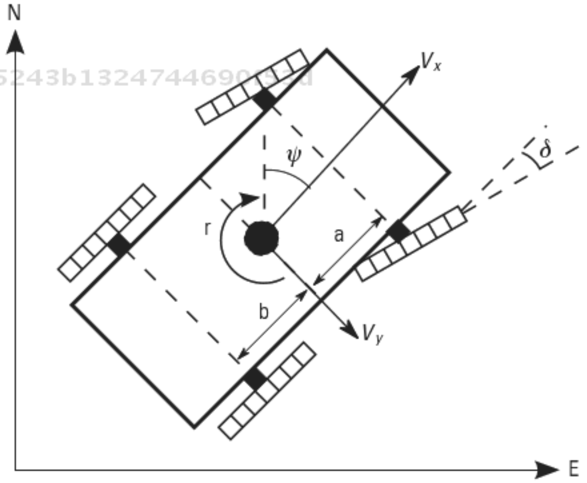


Figure 6.1 Vehicle diagram.

The vehicle's throttle, brake, and steering wheel are assumed to be actuated using DC motors with the input being the voltage applied to the motor (although the model used is suitable for many other actuation methods). Assuming the inductance of the motor's windings is sufficiently small, the model of the motor between input voltage and rotational speed can be approximated by a first-order system with a time constant τ determined by the inertia of the motor's load and a DC gain K, which is assumed to be 1 in the following examples [1].

The steering system is therefore modeled using a first-order lag and an integrator between the steering input u and the steer angle δ. A block diagram of this system including both the steering system and vehicle model is shown in Figure 6.2. The first block in the diagram from u to $\dot{\delta}$ represents the dynamics of the steering actuator. The integrator defines the relationship between steering wheel angular speed and steer angle. The final block defines the vehicle yaw dynamics.

A second-order bicycle model (as presented in Chapter 3) is assumed for the model between steer angle δ and yaw rate r. The bicycle model is given in state space form in (3.12) and in transfer function form in (3.19). The following substitutions are made for brevity and will be used in the remainder of the chapter:

$$\begin{aligned} n_1 &= \frac{aC_{\alpha f}}{I_z} \\ n_0 &= \frac{(a+b)C_{\alpha f}C_{\alpha r}}{mV_x I_z} \\ d_1 &= \frac{C_0 I_z + mC_2}{mV_x I_z} \\ d_0 &= \frac{C_0 C_2 \quad C_1 m V_x^2 \quad C_1^2}{mV_x^2 I_z} \end{aligned} \tag{6.1}$$

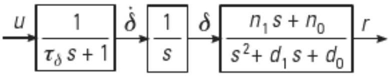

u
$\frac{1}{\tau_\delta s + 1}$
$\dot{\delta}$
$\frac{1}{s}$
δ
$\frac{n_1 s + n_0}{s^2 + d_1 s + d_0}$
r

Figure 6.2 Vehicle model block diagram.

where

$$\begin{aligned} C_0 &= C_{\alpha f} + C_{\alpha r} \\ C_1 &= aC_{\alpha f} \quad bC_{\alpha r} \\ C_2 &= a^2 C_{\alpha f} + b^2 C_{\alpha r} \end{aligned} \tag{6.2}$$

Two control design methods are presented: one based on cascading classical controllers and one using modern, state-space techniques. The classical method relies on models given as transfer functions, such as the model between input and yaw rate given in (3.19). The modern control method relies on models given in state space form [2]. The model between input and yaw rate shown in Figure 6.2 can be converted into state-space form resulting in:

$$\dot{x} = \begin{matrix} \dot{r} \\ \dot{V}_y \\ \dot{\delta} \\ \ddot{\delta} \end{matrix} = \begin{matrix} A_{11} & A_{12} & B_1 & 0 \\ A_{21} & A_{22} & B_2 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1/\tau_\delta \end{matrix} \; \begin{matrix} r \\ V_y \\ \delta \\ \dot{\delta} \end{matrix} + \begin{matrix} 0 \\ 0 \\ 0 \\ 1/\,\tau_\delta \end{matrix} \; u \tag{6.3}$$

$$y = \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix} \; \begin{matrix} r \\ V_y \\ \delta \\ \dot{\delta} \end{matrix} \tag{6.4}$$

where

$$\begin{aligned} A_{11} &= \quad \frac{C_0}{mV_x} \\ A_{12} &= \quad V_x \quad \frac{C_1}{mV_x} \\ A_{21} &= \quad \frac{C_1}{I_z V_x} \\ A_{22} &= \quad \frac{C_2}{I_z V_x} \\ B_1 &= \frac{C_{\alpha f}}{m} \\ B_2 &= \frac{C_{\alpha f}\, a}{I_z} \end{aligned} \tag{6.5}$$

Table 6.1
Infiniti G35 Vehicle Parameters

Symbol	Description	Value
m	Mass	1,528.2 kg
a	Distance from CG to front tire	1.368m
b	Distance from CG to rear tire	1.482m
L	Wheelbase = $a + b$	2.85m
I_Z	Yaw moment of inertia	2,400 kg*m^2
$C_{\alpha f}$	Front tire cornering stiffness	91,674 N/rad
$C_{\alpha r}$	Rear tire cornering stiffness	152,788 N/rad
δ_{max}	Maximum steer angle	0.8 rad
$\dot{\delta}_{max}$	Maximum steer angle rate	0.85 rad/sec
τ_δ	Steering system time constant	0.1053 second

The classical techniques only require knowledge of the system output. When using state-space techniques, however, full state feedback is used; thus, the value of each state must be measured or estimated. All of the states used in this chapter can be measured or estimated using standard full-state estimation algorithms [2]. In the following sections, knowledge of all states will be assumed and no distinction will be made between actual state values and state estimates.

Simulation results are shown throughout the chapter. The parameters used in the simulations are for an Infiniti G35 sedan and are given in Table 6.1. Methods for identifying these parameters are discussed in Chapter 5.

6.3 Speed Controller

The vehicle model given above is an example of a multi-input, multioutput system. In order to simplify the design, however, speed and steering are controlled separately. A controller is designed to control the vehicle's forward velocity V_x. Then, for the purpose of steering control, the longitudinal velocity is considered constant.

A desired speed is selected for each segment of the vehicle's path. The speed controller uses this desired speed as well as the vehicle's current speed to set the throttle and brake positions. The model given in (3.30) is linearized to produce a simple first-order model described by a DC gain and time constant between input and velocity. A second-order model could be used to capture the dynamics of the DC motor used to actuate the throttle and brakes as well, but experience has shown that these models have time constants considerably faster than those of the vehicle dynamics and can be safely ignored. The model between the throttle input u_t and velocity is given by (6.6) and the model between braking input u_b and velocity is given by (6.7).

$$G_{P_t}(s)=\frac{V_x}{u_t}=\frac{K_t}{\tau_t s+1} \tag{6.6}$$

$$G_{Pb}(s)=\frac{V_x}{u_b}=\frac{K_b}{\tau_b s+1} \tag{6.7}$$

Although the same forms are used, the parameters are different in the two models due to the difference in the way throttle and brake affect torque at the wheels.

Two separate proportional-integral (PI) controllers are used to control the brakes and throttle. A feedforward term u_{tff} is also included for the throttle input. The feedforward input is given in (6.8) where DC_{pt} is the DC gain of the throttle plant given in (6.6).

$$u_{tff}=\frac{v_{des}}{DC_{P_t}} \tag{6.8}$$

Although a feedforward term is used, an integral term in the controller is still useful for accounting for uncertainty in the DC gain of the model and accounting for constant disturbances (e.g., road grade). The form of the feedback controller is the same for both the throttle and brake and is given in (6.9) where $V_{err} = V_{des} - V$.

$$G_{Cv}(s)=\frac{u}{V_{err}}=K_{P_v}\quad 1+\frac{K_{Iv}}{s}\div \tag{6.9}$$

Since applying both brake and throttle simultaneously is undesirable, control logic is needed to determine when the controllers are active. In order to prevent braking for speeds that are only slightly over the desired speed, the brake controller is only active when the speed error is negative and has a value that is 20% or more of the desired speed. If this condition is not met, the throttle controller is activated.

6.4 Vehicle Steering Control

All methods of vehicle control discussed in this chapter require the yaw rate of the vehicle to be controlled. Controlling the yaw rate of an Ackermann-steered vehicle requires control of the vehicle's steering angle. The left and right wheel speeds must be controlled to control the yaw rate for a differential drive vehicle. The relationship between wheel speeds and yaw rate is discussed in Section 3.6.1. Using state space techniques, the design of these controllers is incorporated into the designs of the heading and lateral position controllers that are discussed later in the chapter. When using classical design techniques, however, a cascaded control architecture is used, and separate, decoupled controllers control the steer angle δ and the yaw rate r of the vehicle.

6.4.1 Classical Steer Angle Controller

The complexity and type of controller used for the steering system depends on the type of vehicle and actuators used. The input to a typical steering actuator controls the rate of change of the steer angle, rather than the steer angle itself (e.g., a DC motor driving the steering shaft). As discussed previously, a first-order lag with time constant τ_δ and a DC gain of 1 is assumed between the input and steering rate. The steer angle can then be determined by integrating the steering angle rate. The resulting model between steering input and steer angle is given by:

$$G_{P\delta}(s) = \frac{\delta}{u} = \frac{1}{s(\tau_\delta s + 1)} \tag{6.10}$$

Nonlinearities in the steering mechanism such as deadband and saturations must be addressed separately from the baseline design, and for simplicity are not discussed here. Since the plant is type 1 (one open-loop integrator), a proportional-derivative (PD) controller $G_{C\delta}$ can be used to control the steer angle of the vehicle with zero steady-state error to a step input [1]. The form of the controller is given in (6.11).

$$G_{C\delta}(s) = K_{P\delta}(1 + K_{D\delta}s) \tag{6.11}$$

The closed-loop dynamics of the steering system can then be calculated and are given in (6.12).

$$G_{\delta CL}(s) = \frac{\frac{K_{P\delta}}{\tau_\delta}(1 + K_{D\delta}s)}{s^2 + s\left(\frac{1 + K_{D\delta}K_{P\delta}}{\tau_\delta}\right) + \frac{K_{P\delta}}{\tau_\delta}} \qquad (6.12)$$

A desired closed-loop characteristic is determined and standard pole placements techniques are used to calculate the proportional and derivative gains. Since additional controllers are to be added around this inner loop, a closed-width bandwidth is needed so that the bandwidth of the outer loop can be set to approximately 1 Hz and the inner-loop dynamics can still be ignored. A closed-loop bandwidth of approximately 20 Hz is chosen with a 0.707 damping ratio. Using root locus techniques results in gains of $K_{P\delta}$ = 787 and $K_{D\delta}$ = 0.01 when using the parameters given in Table 6.1.

6.4.2 Classical Yaw Rate Controller

Next, a yaw rate controller is developed using the same process. A second-order, bicycle model between the steering angle and yaw rate is used as an example:

$$G_{P_r}(s) = \frac{r}{\delta} = \frac{n_1 s + n_0}{s^2 + d_1 s + d_0} \qquad (6.13)$$

A PI controller is used to control the yaw rate. Since the plant is type 0, an integral term is needed in the controller to prevent steady-state error [1]. The form of the controller is given by (6.14).

$$G_{Cr}(s) = K_{P_r}\left(1 + \frac{K_{Ir}}{s}\right)G_{Cr}(s) \qquad (6.14)$$

The closed-loop system response can be sped up by including a feedforward control term. The feedforward term sets the predicted steady-state

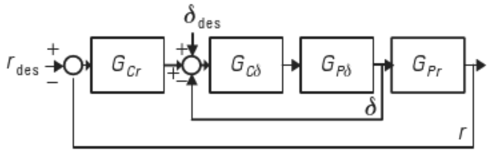


Figure 6.3 Closed-loop yaw rate block diagram.

steer angle so that only disturbances and errors in the model must be accounted for by the feedback controller. The desired steer angle is calculated using the DC gain of the plant given in (6.13). For low speeds, the DC gain can be approximated as V_X/L using the kinematic relationship given in (3.5).

$$\delta_{des} = \frac{r_{des}}{DC_r} \quad \frac{Lr_{des}}{V_x} \tag{6.15}$$

A block diagram of the closed-loop steering and yaw rate systems is shown in Figure 6.3. When designing the yaw rate controller, the closed-loop bandwidth is chosen to be 5–10 times slower than the bandwidth of the closed-loop steer angle. This means that the dynamics of the steering system can be ignored when determining the closed-loop yaw dynamics and only the DC gain DC_δ must be accounted for.

Again, the yaw rate gains can be determined using pole placement techniques. A closed-loop bandwidth of approximately 4 Hz with a damping ratio of 0.707 is chosen. The controller is designed using root locus techniques. For a forward velocity of 4 m/s, these closed-loop characteristics are achieved with gains of $K_{Pr} = 0.11$ and $K_{Ir} = 126.36$, resulting in a third pole located at $s = -39$.

Designing using root locus or other graphical techniques poses some practical challenges for the yaw rate system. The parameters of the bicycle model change with speed. A Bode plot of the model given in (6.13) for four speeds is shown in Figure 6.4. In order to maintain the same steering dynamics regardless of speed, the gains must also change with speed. One solution to this problem is to calculate the gains for a chosen set of speeds over the possible range and either linearly interpolate between speeds or generate a polynomial curve fit to calculate gains between the speeds.

An alternative (and simpler) approach is to calculate the gains using a desired characteristic equation CE. The form of a desired CE consisting of a

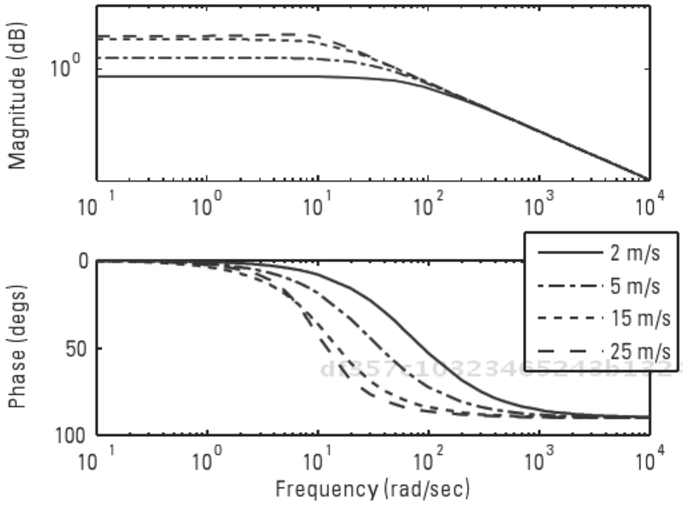


Figure 6.4 Bicycle model Bode plot for various speeds.

second-order system with damping ratio ζ, natural frequency ω_n, and a third, real pole p_3 is given in (6.16).

$$\begin{aligned} CE_{rDES}(s) &= \left(s^2 + 2\zeta\omega_n s + \omega_n^2\right)(s + p_3) \\ &= s^3 + (2\zeta\omega_n + p_3)s^2 + \left(\omega^2 + 2\zeta\omega_n p_3\right) + p_3\omega_n^2 \end{aligned} \tag{6.16}$$

The closed-loop characteristic equation for the vehicle's yaw rate with a PI controller is given by:

$$\begin{aligned} CE_{rCL}(s) = s^3 &+ \left(d_1 + DC_\delta K_{Pr} n_1\right)s^2 \\ &+ \left(DC_\delta K_{Pr} n_0 + DC_\delta n_1 K_{Ir} + d_0\right)s + DC_\delta n_0 K_{Ir} \end{aligned} \tag{6.17}$$

The gains and third pole location are calculated by equating coefficients in the two equations. This provides an equation for the gains in terms of the model parameters and desired characteristics allowing them to be recalculated as the parameters change. Since only two gains are used, only two poles can be placed. The third pole's location must be checked for stability once the gains are calculated.

6.5 Waypoint Control

A common method of defining a desired path for a vehicle is to define a list of evenly spaced locations or waypoints for the vehicle to drive through [3–5]. In the following section, a method for controlling a vehicle to follow a series of waypoints is presented. The method involves controlling the heading ψ of the vehicle to drive it towards the given waypoint. A model for determining the heading of the vehicle is presented, and controllers are designed using both classical and state-space methods to control the vehicle to a desired heading. Finally, a method for determining a desired heading from a given set of waypoints is presented.

6.5.1 Heading Model

Once a model between the vehicle input u and yaw rate r has been found, the vehicle's heading ψ can be found by integrating the yaw rate. For the classical, cascaded approach, an additional block representing an integrator is added to the model. The resulting block diagram is shown in Figure 6.5.

For the modern control approach, the state-space model given previously is updated to include the vehicle's heading with the addition of a fifth heading state, the derivative of which is yaw rate. The updated state-space model is given in (6.18) and (6.19).

$$\dot{x} = \begin{bmatrix} \dot{\psi} \\ \dot{r} \\ \dot{V}_y \\ \dot{\delta} \\ \ddot{\delta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & A_{11} & A_{12} & B_1 & 0 \\ 0 & A_{21} & A_{22} & B_2 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1/\tau_\delta \end{bmatrix} \begin{bmatrix} \psi \\ r \\ V_y \\ \delta \\ \dot{\delta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1/\tau_\delta \end{bmatrix} u \tag{6.18}$$

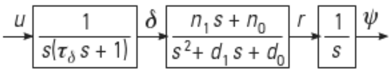


Figure 6.5 Heading model block diagram.

$$y = \begin{matrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{matrix} \begin{matrix} \psi \\ r \\ b \\ \delta \\ \dot{\delta} \end{matrix} \tag{6.19}$$

6.5.2 Heading Error Calculations

In order to control a vehicle to the desired path, a reference heading must be calculated. A list of waypoints consisting of positions and speeds are provided to the controller by a path planner. Before a reference heading can be determined, a single waypoint must be chosen to drive to. Choosing a waypoint requires knowledge of the vehicle's current position (e, n) and heading.

The waypoint chosen must meet two conditions, illustrated in Figure 6.6 [6]. First, it has to be in front of the vehicle. This is determined by drawing a line perpendicular to the heading of the vehicle and ignoring any waypoints behind the line. Waypoints that fail this condition are shown with thin, dashed Xs in Figure 6.6. Second, the chosen waypoint must be at least a given distance (referred to as the look-ahead distance,

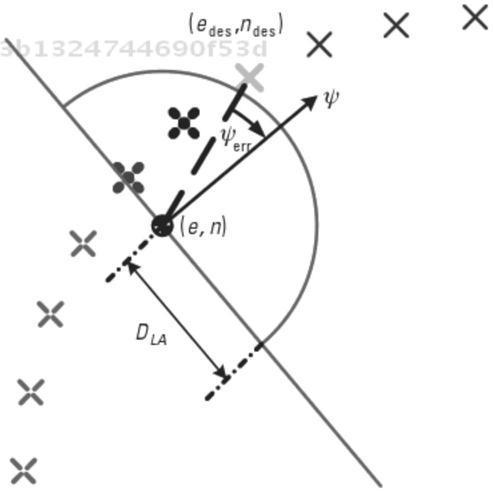


Figure 6.6 Waypoint selection.

D_{LA}) away from the vehicle. Waypoints that pass the first test, but fail this test are shown with thicker, dashed Xs. The closest waypoint that meets both of the given conditions (e_{des}, n_{des}) is chosen and is shown with a thick solid X.

The choice of the look-ahead distance substantially affects the nature of the reference path. Choosing a look-ahead distance that is too short can cause a waypoint that is almost directly beside the vehicle to be chosen, resulting in aggressive steering and driving perpendicular to the path, rather than smoothly merging onto it. Choosing a look-ahead distance that is too long causes the vehicle to choose the next waypoint long before the currently chosen waypoint has been reached, resulting in the vehicle cutting corners during turns. It is common to vary the look-ahead distance with speed to make the steering smoother by adding damping to the heading controller at higher speeds. This is similar to the method used by human drivers where at low speeds, focus is near the front of the vehicle; as speed increases, however, it becomes necessary to look further ahead. The look-ahead distance used in the simulations that follow is given in (6.34).

Once the waypoint is chosen, a reference heading and heading error is computed. The reference or desired heading ψ_{des} is the angle of a line drawn between the vehicle's position and the chosen waypoint. The desired heading is measured clockwise with respect to north and is given by:

$$\psi_{des} = \tan^{1} \frac{e_{des} \quad n_{des}}{e \quad n} \div \tag{6.20}$$

The heading error is defined as the angle between the vehicle heading and the reference heading as given in (6.21).

$$\psi_{err} = \psi \quad \psi_{des} \tag{6.21}$$

In the following section, a controller is developed to drive the vehicle toward the waypoint by driving this error to zero.

6.5.3 Heading Control

Once a model between the steering input and heading and a desired reference value have been obtained, a controller is needed to drive the vehicle's heading to the reference value. Both classical and modern methods are described.

6.5.3.1 Classical Method

When using the classical method, a third feedback loop is closed around the model between steering input and heading. As discussed previously, the transfer function between the yaw rate and heading consists of a pure integrator. A transfer function for the heading plant is given in (6.22).

$$G_{P\psi}(s) = \frac{\psi}{r} = \frac{1}{s} \tag{6.22}$$

Since the system is type 1, zero steady-state error can be achieved using a PD controller to control the vehicle's heading. The form of the controller is given by (6.23).

$$G_{C\psi}(s) = K_{P\psi}(1 + K_{D\psi}s) \tag{6.23}$$

A block diagram of the closed-loop system including the steering, yaw rate, and heading models is shown in Figure 6.7.

If the closed-loop dynamics of the vehicle's heading are chosen to be sufficiently slower than the closed-loop yaw rate dynamics, the yaw rate dynamics can be ignored and only the DC gain (DC_r) of the yaw rate system must be included. The closed-loop dynamics of the vehicle's heading in terms of the controller gains are given in (6.24). For a velocity of 4 m/s, a closed-loop response with a bandwidth of approximately 0.9 Hz is desired so the 1, real pole is placed at 5.62. This is obtained by choosing the gains to be $K_{P\psi}$ = 12.8 and $K_{D\psi}$ = 0.1.

$$G_{\psi CL}(s) = \frac{DC_r K_{P\psi}(sK_{D\psi} + 1)}{s(1 + DC_r K_{p\psi} K_{D\psi}) + DC_r K_{P\psi}} \tag{6.24}$$

6.5.3.2 State Space Method

A state feedback controller can be developed by rewriting the state-space model previously given in (6.18) in terms of a heading error. This is done by

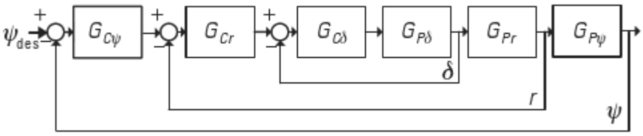


Figure 6.7 Heading closed-loop block diagram.

making the assumption that the dynamics of the heading error are equivalent to the dynamics of the vehicle's heading. This is true as long as the desired heading is constant. This assumption is not valid for the period immediately after a new waypoint is selected, which could possibly result in transient behavior as each new waypoint is selected, but is otherwise valid.

The dynamics of the heading error can be found by taking the derivative of (6.21) resulting in:

$$\dot{\psi}_{err} = \dot{\psi} \quad \dot{\psi}_{des} \tag{6.25}$$

Using the assumption that the desired heading is constant, (6.25) can be simplified to:

$$\dot{\psi}_{err} = \dot{\psi} \tag{6.26}$$

The state-space model can now be rewritten in terms of heading error:

$$\dot{x} = \begin{matrix} \dot{\psi}_{err} \\ \dot{r} \\ \dot{V}y \\ \dot{\delta} \\ \ddot{\delta} \end{matrix} = \begin{matrix} 0 & 1 & 0 & 0 & 0 \\ 0 & A_{11} & A_{12} & B_1 & 0 \\ 0 & A_{21} & A_{22} & B_2 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1/\tau_\delta \end{matrix} \begin{matrix} \psi \\ r \\ V_y \\ \delta \\ \dot{\delta} \end{matrix} + \begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 1/\tau_\delta \end{matrix} u \tag{6.27}$$

$$y = \begin{matrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{matrix} \begin{matrix} \psi_{err} \\ r \\ V_y \\ \delta \\ \dot{\delta} \end{matrix} \tag{6.28}$$

A full state feedback control law of the form

$$u = \quad Kx \tag{6.29}$$

where

$$x = [\psi_{err} \quad r \quad V_y \quad \delta \quad \dot{\delta}]^T \tag{6.30}$$

is used to drive the heading error calculated in Section 6.5.2 to 0. The reference or desired values for the remaining states are 0. The control gains K are calculated by solving the Riccati equation, resulting in a controller known as a linear quadratic regulator (LQR) [7]. The LQR control algorithm is an optimal control algorithm that optimizes the cost function given in (6.31) subject to the control feedback constraint given in (6.29).

$$J = \quad x^T Q_x x + u^T R_u u \tag{6.31}$$

The resulting controller minimizes a combination of the states weighted by the matrix Q_x and the input(s) weighted by the scalar (or matrix) R_u. Example weighting matrices are given in (6.32) and (6.33). These values are found experimentally by "hand tuning" until satisfactory performance is obtained [8].

$$\begin{aligned} Q_x &= diag[Q_\psi \quad Q_r \quad Q_b \quad Q_\delta \quad Q_{\dot{\delta}}] \\ &= diag[15 \quad 0 \quad 0 \quad 1 \quad 0] \end{aligned} \tag{6.32}$$

$$R_u = 0.1 \tag{6.33}$$

Since we are concerned with controlling the heading error, Q_ψ was set to 15 to penalize that state. Additionally, Q_δ was set to 1 to penalize the steer angle. Reducing the amount of steer angle adds more damping to the system. It is important to note that only the relative values of the LQR weighting matrices are important. The weighting matrices can be normalized with respect to any variable or state. If they are normalized to give a cost function that is unitless, then the units of each weighting value are the reciprocal of the square of the units of the weighted state.

6.5.4 Simulation Results

The vehicle is simulated following the path shown in Figure 6.8 consisting of waypoints spaced at 2-m intervals. These waypoints are shown with Xs in the figure. The path followed by the vehicle is shown with a solid line. The vehicle is started at a position 6-m north of the first waypoint. A constant speed of 4 m/s is used in the simulation. Two zoomed-in sections are shown on the plot showing the vehicle as it converges to the path and during a sharp turn. A plot of the heading error verses time is

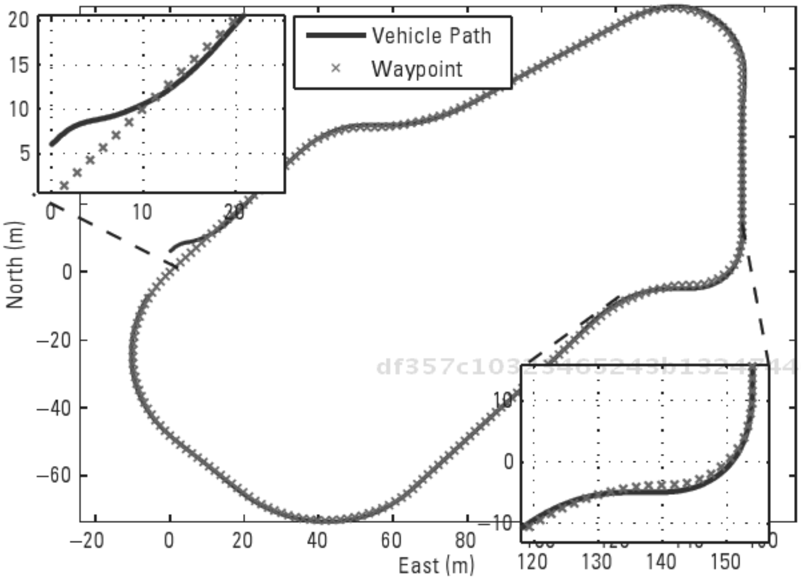


Figure 6.8 Waypoint following simulation path.

shown in Figure 6.9. The errors are low on the straighter sections of the path where the desired heading remains fairly constant, but increase in turns.

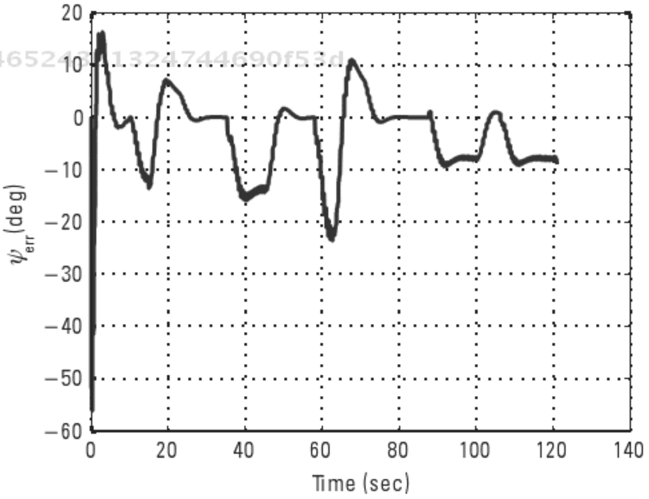


Figure 6.9 Waypoint following simulation heading error.

The look-ahead distance used in the simulation is given in (6.34), where D_{min} is a minimum look-ahead distance and D_{scale} is a scale factor that adjusts the look-ahead distance with speed [6]. For the previous simulation, both D_{min} and D_{scale} are set to 1.

$$D_{LA} = D_{scale}V_x + D_{min} \tag{6.34}$$

As discussed in Section 6.5.2, one of the drawbacks of waypoint control is the possibility of cutting corners at higher speeds or with longer look-ahead distances. An example of this is seen in the following simulation, where the vehicle speed is increased to 8 m/s and the look-ahead scale factor D_{scale} is increased to 1.5. The corner shown in the zoomed section in the bottom right of Figure 6.8 is shown in Figure 6.10. The result of the higher speed and look-ahead distance is that the vehicle chose a new waypoint significantly before reaching the previous waypoint, resulting in the corner being cut short. One possible solution to this problem is to include a lateral distance measurement from the path in the controller. This solution is explored in the following section.

6.6 Lateral Control

Another option for defining a desired path for a vehicle is a path defined by a continuous curve [9–11]. Paths of this type can be defined using many dif-

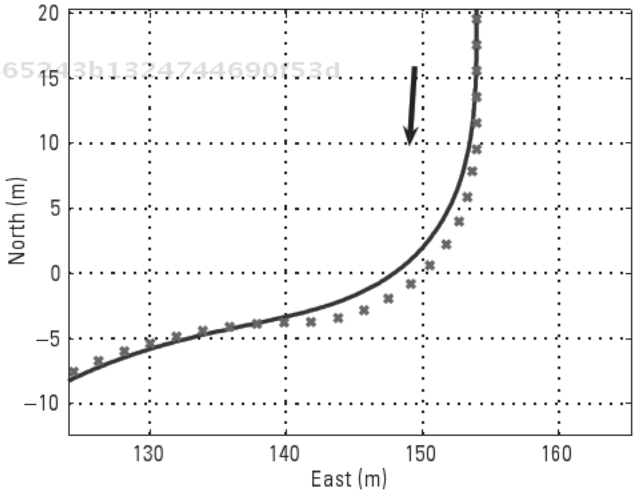


Figure 6.10 Effect of look-ahead distance on waypoint control.

ferent types of continuous functions [12]. A simple form of continuous path function consisting of line segments and circular arcs with a constant radius will be considered in this section. These paths are commonly referred to as Dubins's paths after L. E. Dubins who showed that the time-optimal (neglecting vehicle dynamics), continuously differentiable path between any two points in 2D space consists of lines and circular arcs of a minimum radius [13]. The disadvantage to this type of path is that a discontinuity in curvature exists at the intersection of line and arc segments [14]. As can be seen from (3.2) this equates to a discontinuity in the desired steering angle resulting in transient behavior at these intersections. Modifications involving clothoids or splines can be used to eliminate the problem, but result in suboptimal paths [14–16]. Also, paths defined using clothoids do not have a closed form expression and require numerical integration techniques to calculate lateral errors.

When following a path defined by a continuous curve, both the lateral position of the vehicle as well as its heading are controlled. The methods described are easily extensible to paths defined by other types of continuous curves. The control design is not dependent on the type of path. Only the error calculation methods need to be modified based on the type of path chosen. Lateral control is also useful in automated highway systems and other applications where no mathematical path is defined. Lateral errors from a lane or corridor can be calculated using sensors such as magnetometers [17], LIDAR [18], or vision systems [19] and used to control the vehicle's position.

6.6.1 Error Calculation

In the previous section, when controlling the vehicle along a series of waypoints, a reference heading was determined and used to steer the vehicle toward a waypoint. When paths are defined by a continuous function, a lateral error as well as a heading error are calculated and used to steer the vehicle along the path. Although desired paths are typically more difficult to define using continuous functions, the addition of the lateral error measurement allows this type of path to be followed more precisely than a path defined by waypoints.

Rather than choosing a waypoint to drive to, as was done in Section 6.5.2, a reference point on the path must be chosen. The lateral distance between this point and a control point on the vehicle is calculated to provide a lateral error y_{err} between the vehicle and the path. Controlling the vehicle's lateral error forces the control point on the vehicle to be as close as possible to the path.

It is desirable not only for the vehicle's control point to be close to the path, but also for the vehicle to be oriented so that it is pointed along the path. This is done by controlling the vehicle's heading (as was done in waypoint control) as well. A desired heading is found by calculating the angle of a line tangent to the curve at the reference point. The heading error can then be found as described previously in (6.21) by subtracting the desired heading from the vehicle's actual heading. Controlling the vehicle's heading results in the vehicle's orientation always being tangent to the path and therefore pointed in the direction of the next reference point.

Both the lateral and heading errors are illustrated in Figure 6.11. Methods for calculating these errors (as well as reference values for the remaining states) for both lines and circular arcs are given in the following sections.

The closest point on the path to the vehicle's control point is chosen as the reference point. A look-ahead distance similar to the one used in waypoint selection can be used in selecting the reference point to add additional damping to the system, but is not required. The need for using a look-ahead distance increases with speed as the system's open-loop damping decreases [6].

6.6.1.1 Error Calculation for Lines

The lateral error between the vehicle and a path segment given by a line is defined as the distance of a line segment drawn perpendicular to the vehicle's heading between the vehicle's control point and the path segment. This is the distance labeled y_{err2} in Figure 6.12. In order to simplify calculations, this distance is approximated by calculating the shortest distance between the

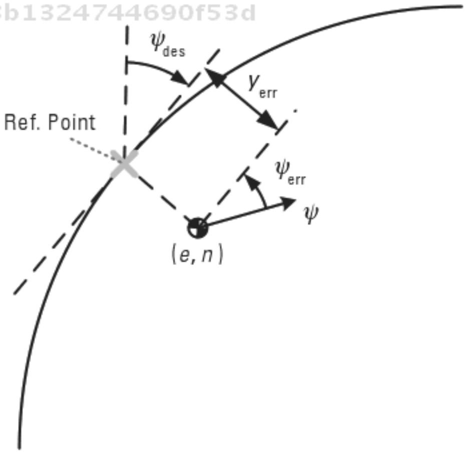


Figure 6.11 Lateral position error and heading error from a continuously defined path.

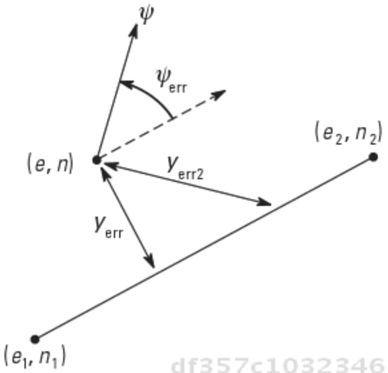


Figure 6.12 Error calculation for lines.

vehicle and the line. This approximated distance (y_{err}) is also shown in Figure 6.12. The two values are equal if the heading error is 0. Since the heading error is also being controlled, this is a valid assumption once the vehicle is on the path.

A simple expression for calculating the lateral error for a path consisting of a line segment between points (e_1, n_1) and (e_2, n_2) is given in (6.35) [9].

$$y_{err} = \frac{\left((n_1 \quad n_2)e + (e_2 \quad e_1)n + (e_1 n_2) \quad (e_2 n_1)\right)}{\sqrt{(e_2 \quad e_1)^2 + (n_2 + n_1)^2}} \tag{6.35}$$

The desired heading is found by calculating the angle with respect to north made by the line segment.

$$\psi_{des} = \tan^{\ 1} \ \frac{e_2 \quad e_1}{n_2 \quad n_1} \div \tag{6.36}$$

The heading error can then be found by subtracting the reference heading from the vehicle's actual heading.

$$\psi_{err} = \psi \quad \psi_{des} \tag{6.37}$$

When following a straight line in steady state, the vehicle should not be turning. Therefore the reference yaw rate, steering angle, and steering angle rate are all set to 0. A summary of the error equations is given in Table 6.2.

Table 6.2
Error Equations for Line Segments

Error	Equation
Lateral	$y_{err} = \frac{\left((n_1 \quad n_2)e + (e_2 \quad e_1)n + (e_1 n_2) \quad (e_2 n_1)\right)}{\sqrt{(e_2 \quad e_1)^2 + (n_2 + n_1)^2}}$
Heading	$\psi_{err} = \psi \quad \tan^{1} \; \frac{e_2 \quad e_1}{n_2 \quad n_1} \div$
Yaw rate	$r_{err} = r$
Lateral velocity	$V_{yerr} = V_y$
Steer angle	$\delta_{err} = \delta$
Steer angle rate	$\overset{\times}{\delta}_{err} = \overset{\times}{\delta}$

6.6.1.2 Error Calculation for Arcs

Lateral and heading error calculation for circular arcs is similar to that of line segments. Again, small heading errors are assumed and the lateral error is calculated by finding the shortest distance between the vehicle and the arc. This is done by finding the distance between the vehicle's position and the center of the arc and subtracting the radius of the arc [9]. The result is given in (6.38) and is illustrated in Figure 6.13.

$$y_{err} = \sqrt{(e \quad e_C)^2 + (n \quad n_C)^2} \quad R \tag{6.38}$$

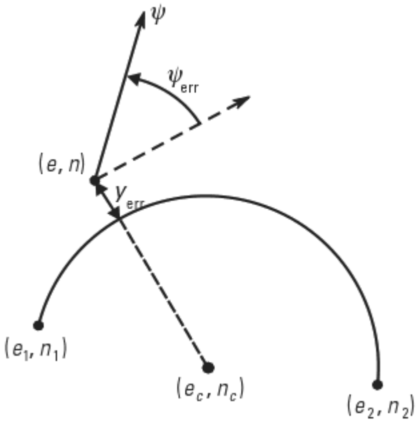


Figure 6.13 Error calculation for arcs.

The desired heading can be found by calculating the angle of a line tangent to the arc at the reference point. A simple method of calculating the desired heading is to find the angle of the line segment between the center of the arc and the vehicle's position. This line is perpendicular to the desired heading at the reference point. The desired heading in radians can then be found by adding $\pi/2$ to the angle of the line if the arc is being followed in a clockwise direction or subtracting $\pi/2$ from the angle if the arc is being followed in a counterclockwise direction. The equation for desired heading is given in (6.39) where $cw = 1$ for arcs followed in a clockwise direction and $cw = -1$ for arcs followed in a counterclockwise direction.

$$\psi_{des} = \tan^{1} \frac{e \quad e_c}{n \quad n_c} \div + \frac{\pi}{2} \quad cw \div \qquad (6.39)$$

Unlike line segments, a steady-state yaw rate (and therefore steer angle as well) is required to follow an arc. When driving on an arc of constant radius, the rate of change of the heading of the vehicle is equal to the angular velocity of the vehicle [11]. If input to the model was the yaw rate, this would be accounted for in a feedforward term in the controller. When the steering system dynamics are included, however, a reference yaw rate and steer angle are used. The reference yaw rate is given by:

$$r_{des} \quad \frac{V_x}{R} \qquad (6.40)$$

where R is the radius of the arc. A reference steer angle can then be determined by looking at the model between steer angle and yaw rate given in (3.2), which assuming small steer angles and no side slip is given by:

$$\delta_{des} = \frac{L}{R} \qquad (6.41)$$

where L is the vehicle wheelbase. When significant side slip is expected, the desired steer angle can be more accurately calculated by taking into account the DC gain of the bicycle model as shown in (6.15).

Once the desired steer angle is reached, it is constant for the duration of the arc and so the desired steering rate is 0. A summary of the error equations is given in Table 6.3.

Table 6.3
Error Equations for Arc Segments

Error	Equation
Lateral error	$y_{err} = \sqrt{(e \quad e_C)^2 + (n \quad n_C)^2} \quad R$
Heading error	$\psi_{err} = \psi \quad \tan^{1} \frac{e \quad e_C}{n \quad n_C} \div + \frac{\pi}{2} \quad cw \div$
Yaw rate error	$r_{err} = r \quad \frac{V_X}{R}$
Lateral velocity error	$V_{yerr} = V_y$
Steer angle error	$\delta_{err} = \delta \quad \frac{L}{R}$
Steer angle rate error	$\dot{\delta}_{err} = \dot{\ }_{y}$

6.6.2 Lateral Position Model

Once a heading and lateral error are determined, a model is needed to relate these errors to the steering input. A model for the vehicle's heading was given previously in Section 6.5.1 when discussing waypoint control. A model of the vehicle's lateral position is now needed. The dynamics of the lateral error from a line or curve of the vehicle are given by [8]:

$$\dot{y}_{err} = V_X \sin(\psi_{err}) + (V_Y \quad L_{CP}\dot{\psi})\cos(\psi_{err}) \tag{6.42}$$

where L_{CP} is the distance from the control point to the center of gravity of the vehicle. In general the longitudinal velocity V_X of the vehicle is much greater than the lateral velocity at the CG V_Y and so the lateral velocities are neglected. This allows the model to be simplified to:

$$y_{err} = V_X \sin(\psi_{err}) \quad L_{CP}\dot{\psi}\cos(\psi_{err}) \tag{6.43}$$

Other researchers, including [20], have used a kinematic model, which also assumes no lateral velocities due to sideslip. For a kinematic model of the vehicle, the point at the center of the rear axle will have no lateral velocity. Because the kinematic model assumes no tire side slip, the velocity vector must be along the direction of the tire. There is assumed to be no lateral velocity at the tires and therefore no lateral velocity of the vehicle between the rear tires.

It is important to note that the dynamics of the system change by moving the control point (thereby changing L_{CP}) [21]. The system becomes a nonminimum phase (contains a zero in the right half plane) for a control point located behind the CG assuming the lateral velocity at the CG is 0. For this work, the control point is assumed to be at the CG of the vehicle. Setting the control point to be at the vehicle's CG ($L_{CP} = 0$) and assuming small heading errors, the model can be simplified to:

$$\dot{y}_{err} = V_X \psi_{err} \tag{6.44}$$

The resulting block diagram between input and lateral position is shown in Figure 6.14.

The state space model developed previously for heading control and given in (6.27) can be augmented to include the lateral position dynamics through the inclusion of a lateral error state. The resulting model is given in (6.45) and (6.46).

$$\dot{x} = \begin{matrix} \dot{y}_{err} \\ \dot{\psi}_{err} \\ \dot{r} \\ \dot{V}_y \\ \dot{\delta} \\ \ddot{\delta} \end{matrix} = \begin{matrix} 0 & V_X & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & A_{11} & A_{12} & B_1 & 0 \\ 0 & 0 & A_{21} & A_{22} & B_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1/\tau_\delta \end{matrix} \begin{matrix} y_{err} \\ \psi_{err} \\ r \\ V_y \\ \delta \\ \dot{\delta} \end{matrix} + \begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1/\tau_\delta \end{matrix} u \tag{6.45}$$

$$y = \begin{matrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{matrix} \begin{matrix} y_{err} \\ \psi_{err} \\ r \\ V_y \\ \delta \\ \dot{\delta} \end{matrix} \tag{6.46}$$

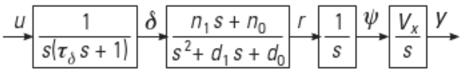

u
$\frac{1}{s(\tau_\delta s + 1)}$
δ
$\frac{n_1 s + n_0}{s^2 + d_1 s + d_0}$
r
$\frac{1}{s}$
ψ
$\frac{V_x}{s}$
y

Figure 6.14 Lateral position model block diagram.

6.6.3 Lateral Position Control

6.6.3.1 Classical Method

When using the classical method, a third feedback loop is closed around the model between steering input and lateral position. To simplify the design, the model between yaw rate and heading used for heading control is combined with the lateral position model given in (6.44) to form a model between the yaw rate and lateral position. The transfer function for the lateral model is given in (6.47).

$$G_{P_y}(s) = \frac{y_{err}}{r} = \frac{V_X}{s^2} \tag{6.47}$$

A PD controller is used to control the vehicle's lateral position. Additional damping is added to the system through the derivative term to account for the two poles at the origin in the model. The form of the controller is given by (6.48).

$$G_{C_y}(s) = K_{P_y}(1 + K_{D_y}s) \tag{6.48}$$

A block diagram of the closed loop system including the steering, yaw rate, and lateral position models is shown in Figure 6.15. When determining the lateral position closed-loop dynamics, the inner-loop yaw rate dynamics can be neglected since they are many times faster than the lateral position dynamics. Only the DC gain of the inner-loop transfer function DC_r is included in the closed-loop dynamics. The closed-loop lateral position dynamics, neglecting the yaw rate dynamics, are given in (6.49). For a forward velocity of 4 m/s, setting $K_{P_y} = 8.385$ and $K_{D_y} = 0.24$ results in a closed-loop system with a bandwidth approximately 0.9 Hz and a damping ratio of 0.707.

$$G_{yCL} = \frac{DC_r V_X K_{P_y}(1 + K_{D_y}s)}{s^2 + DC_r V_X K_{P_y} K_{D_y} s + DC_r V_X K_{P_y}} \tag{6.49}$$

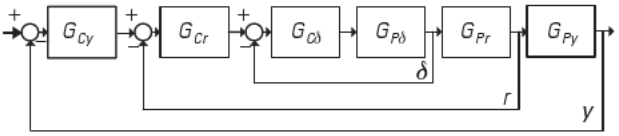


Figure 6.15 Lateral position closed-loop block diagram.

6.6.3.2 State-Space Method

A state feedback control law of the form

$$u = \quad Kx \tag{6.50}$$

where

$$x = [y_{err} \quad \psi_{err} \quad r \quad V_y \quad \delta \quad \dot{\delta}]^T \tag{6.51}$$

is used to drive the lateral position error to 0. Since full-state feedback is used, it is assumed that a measurement or estimate of all states is available. The control gains K are calculated to form a linear quadratic regulator as discussed in Section 6.5.3.2. Example weighting matrices are given in (6.52) and (6.53).

$$\begin{aligned} Q_x &= diag[Q_y \quad Q_\psi \quad Q_r \quad Q_b \quad Q_\delta \quad Q_{\dot{\delta}}] \\ &= diag[1 \quad 0 \quad 0 \quad 5 \quad 0] \end{aligned} \tag{6.52}$$

$$R_u = 0.1 \tag{6.53}$$

Since we are concerned with controlling the lateral error, Q_y was set to 1 to penalize that state. Additionally, Q_δ was again set to 5 to penalize steer angle, which adds additional damping to the system. The selected control weighting values given above were found experimentally by hand tuning until satisfactory performance was obtained. Using the parameters given in Table 6.1 and forward velocities between of 2 m/s and 8 m/s, the above weighting matrices result in the gains given in Table 6.4.

The open-loop and closed-loop poles of the system are shown in Figure 6.16 and Figure 6.17, respectively. Note that the bandwidth of the system

Table 6.4
Lateral Error Gains

Velocity	Gains					
2 m/s	K = [3.16	17.85	0.267	0.019	18.28	1.76]
4 m/s	K = [3.16	22.51	0.663	0.046	28.17	2.61]
6 m/s	K = [3.16	26.45	1.138	0.078	37.24	3.34]
8 m/s	K = [3.16	30.06	1.670	0.113	45.71	3.98]

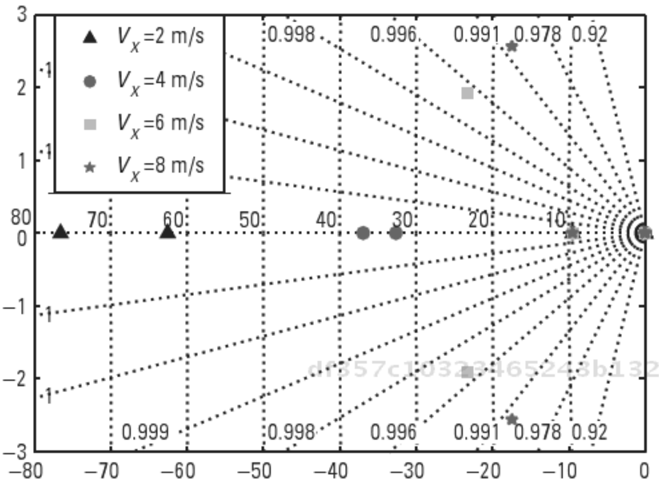


Figure 6.16 Lateral position model open-loop poles.

varies with velocity. It is also interesting to notice that only the integrator poles are altered from the open-loop system to the compensated closed-loop system.

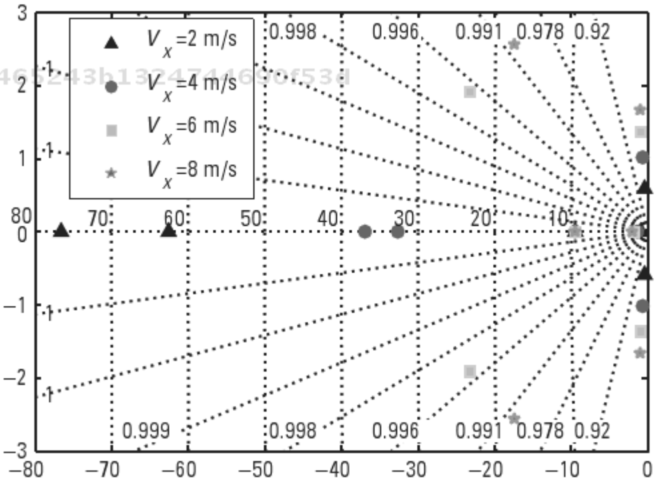


Figure 6.17 Lateral position model closed-loop poles.

6.6.4 Simulation Results

The vehicle is simulated with forward velocities of 2 m/s and 8 m/s along a line heading due north using the state-space controller developed in Section 6.6.3.2. There is an initial offset of 2m from the path. The resulting lateral error response is shown in Figure 6.18. The settling time decreases at higher speeds, but the damping remains constant as is predicted from the closed-loop pole locations.

The same straight line path is simulated again using two different sets of gains at a forward velocity of 2 m/s. The lateral error response is shown in Figure 6.19. For the first run, the Q_δ entry in the LQR weighting matrix is set to 5. In the second run it is set to 0. The additional damping added by penalizing steer angle is seen in the response.

6.7 Implement/Trailer Control

Previously, controllers were developed for autonomous path following of a point on a vehicle. In some applications (precision agriculture [8, 21], factory automation [22], UXO/mine detection [9], and so forth), however, it is advantageous to control a point on a trailer or implement towed behind

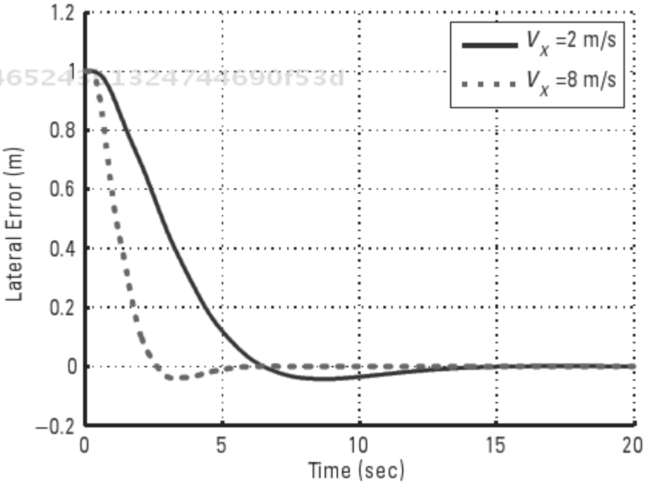


Figure 6.18 Simulated line tracking at 2 m/s and 8 m/s.

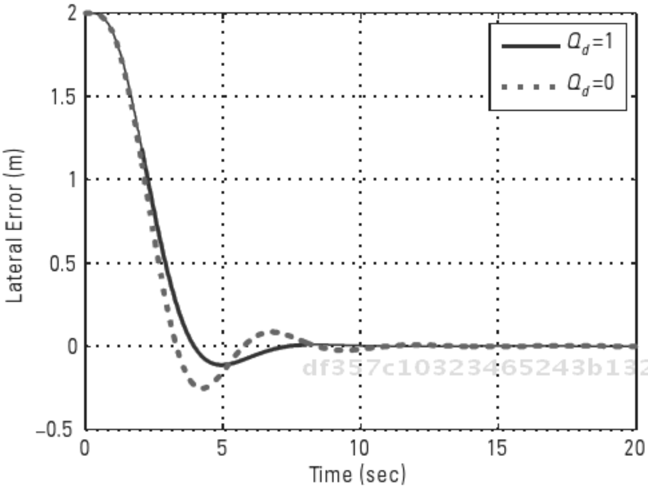


Figure 6.19 Effect of weighting matrices on response.

a vehicle. In the following section, a method is presented to place a control point on a trailer or implement towed behind a vehicle on a path. A model of the towed trailer is given and a state feedback controller is developed to control the trailer.

6.7.1 Trailer Model

A kinematic model is used to describe a trailer or implement towed behind the vehicle [8, 23]. A schematic of a vehicle pulling a trailer is shown in Chapter 3 in Figure 3.13. The trailer's point of zero lateral velocity (ZLV), as well as its control point (cp), is shown in the figure. The control point defines the point on the trailer that should follow the specified path. This point can be chosen by the designer and is often chosen to be the location of the GPS antenna on the trailer if one is used to determine the trailer's position.

A kinematic model that relates the vehicle velocity V_X and yaw rate r to the angle between the vehicle and the trailer referred to as the hitch angle φ is given by (3.63) in Chapter 3. The position of the trailer's control point can be related to the position of the vehicle using the geometry of the system and this hitch angle. The relationships between the positions, orientations, and velocities of the robot and trailer are given by [9]:

$$e_t = e \quad L_H \sin(\psi) \quad L_C \sin(\psi + \varphi) \tag{6.54}$$

$$n_t = n \quad L_H \cos(\psi) \quad L_C \cos(\psi + \varphi) \tag{6.55}$$

$$\psi_t = \psi + \varphi \tag{6.56}$$

$$V_t = V_X \cos(\varphi) \quad L_C r \sin(\varphi) \tag{6.57}$$

where (e_t, n_t) is the position of the trailer's control point, ψ_t is the heading of the trailer, V_t is the velocity of the trailer, and L_C is the distance from the hitch point to the control point. These equations can also be used to calculate the position of the trailer from the position of the vehicle and the hitch angle if a measurement of the hitch angle is available rather than a direct measurement of the trailer's position [24].

The lateral dynamics of the trailer can be obtained by assuming the vehicle and trailer are tracking a line oriented due north and taking the derivative of the east position of the trailer given in (6.54). The resulting lateral dynamics are given by:

$$\dot{y}_t = \dot{e}_t = \dot{e} \quad L_H \dot{\psi} \cos(\psi) \quad L_C (\dot{\psi} + \dot{\varphi}) \cos(\psi + \varphi) \tag{6.58}$$

Assuming small hitch angles and heading errors, this can be simplified to:

$$\dot{y}_t = V_X \psi_{err} \quad L_H \dot{\psi}_{err} \quad L_C (\dot{\psi}_{err} + \dot{\varphi}) \tag{6.59}$$

The hitch dynamics given in (3.60) can also be linearized assuming small angles resulting in:

$$\dot{\varphi} = \quad r \;\; 1 + \frac{L_H}{L_T} \div \;\; \frac{V_X}{L_T} \varphi \tag{6.60}$$

The dynamics of the hitch angle (6.60) and implement lateral position (6.59) are placed in state-space form and added to the vehicle model given previously in (6.27). The resulting state-space model is given in (6.61) and (6.62).

$$\dot{x} = \begin{array}{c} \dot{y}_{terr} \\ \dot{\varphi} \\ \dot{\psi}_{err} \\ \dot{r} \\ \dot{V}_y \\ \dot{\delta} \\ \ddot{\delta} \end{array} = \begin{array}{cccccccc} 0 & \frac{V_X L_C}{L_T} & V_X & \frac{L_H L_C}{L_T} & L_H & 0 & 0 & 0 \\ 0 & \frac{V_X}{L_T} & 0 & 1+\frac{L_H}{L_T}\div & 0 & 0 & 0 & \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & \\ 0 & 0 & 0 & A_{11} & A_{12} & B_1 & 0 & \\ 0 & 0 & 0 & A_{21} & A_{22} & B_2 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/\tau_\delta & \end{array} \begin{array}{c} y_{terr} \\ \varphi \\ \psi_{err} \\ r \\ V_y \\ \delta \\ \dot{\delta} \end{array} + \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{\tau_\delta} \end{array} u \tag{6.61}$$

$$y = \begin{array}{ccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \begin{array}{c} y_{terr} \\ \varphi \\ \psi_{err} \\ r \\ V_y \\ \delta \\ \dot{\delta} \end{array} \tag{6.62}$$

6.7.2 Error Calculation

Calculating the lateral error for trailer control uses the same methods given in Section 6.6.1 for vehicle lateral control with the exception that the position of the trailer is substituted for the position of the vehicle. The equations for the lateral error from a line and arc segment are given in (6.63) and (6.64), respectively, where (e_t, n_t) is the position of the trailer and R_t is the radius of the path followed by the trailer.

$$Y_{err} = \frac{\left((n_1 \quad n_2)\quad e_t + (e_2 \quad e_1)\quad n_t + (e_1 \quad n_2)\quad (e_2 \quad n_1)\right)}{\sqrt{(e_2 \quad e_1)^2 + (n_2 + n_1)^2}} \tag{6.63}$$

$$y_{err} = \sqrt{(e_t \quad e_C)^2 + (n_t \quad n_C)^2} \quad R_t \tag{6.64}$$

The desired heading, yaw rate, and steer angle can also be calculated using the methods given in Section 6.6.1, with a few minor modifications. The equations presented previously calculate desired values for the vehicle when it is following a given path. Now, however, the desired values for the vehicle are needed when the trailer is following a given path. When following line segments, the path followed by the vehicle and trailer are the same and the equations given previously hold. When the trailer tracks an arc of a given radius, however, the path followed by the vehicle and the trailer are not the same. The vehicle must follow a larger arc. For a desired trailer path with a radius R_t, the radius of the arc followed by the vehicle is given by:

$$R = \sqrt{{L_T}^2 \quad {L_H}^2 + R_t^2} \tag{6.65}$$

This larger radius must be taken into account when calculating the desired yaw rate and steer angle of the vehicle. Once the radius of the vehicle's path is known, the desired vehicle yaw rate and steer angle can be calculated using (6.66) and (6.67), respectively.

$$r_{des} = \frac{V_X}{R} \tag{6.66}$$

$$\delta_{des} \quad \frac{L}{R} \tag{6.67}$$

A desired hitch angle φ_{des} is found by calculating the steady-state value of the hitch angle for both line and arc path segments. For a line segment, the trailer follows directly behind the vehicle and the desired hitch angle is 0. When following an arc, a constant angle exists between the vehicle and trailer and is given by:

$$\varphi_{des} = \tan^{\;1} \quad \begin{matrix} \dfrac{L_H R_t R + L_T^3 + L_T R_t^2 \quad L_H^2 L_T}{R\left(L_T^2 + R_t^2\right)} \\ \\ \dfrac{L_H L_T R + R_T L_T^2 + R_t^3 \quad R_t L_H^2}{R\left(L_T^2 + R_t^2\right)} \end{matrix} \begin{matrix} \div \\ \div \\ \div \\ \div \\ \div \\ \div \end{matrix} \tag{6.68}$$

The desired heading of the trailer can also be calculated using the methods given in Section 6.6.1. The desired trailer heading ψ_{tdes} is found using (6.36) for line segments and (6.39) for arc segments, again substituting the position of the trailer for that of the robot. The trailer model given previously, however, is in terms of the vehicle's heading. The desired vehicle heading can be found in terms of the desired trailer heading by rearranging the relation given in (6.56) resulting in:

$$\psi_{des} = \psi_{tdes} \quad \varphi_{des} \tag{6.69}$$

A summary of the error equations for lines is given in Table 6.5 and for arcs is given in Table 6.6.

6.7.3 Trailer Control

Once the errors from the path are determined, a controller is needed to drive the errors to 0. A full-state feedback control law is used to control the trailer's lateral position (as opposed to the vehicle's lateral position controlled in the previous section) [8]. The form of the controller is given by:

$$u = \quad Kx \tag{6.70}$$

Table 6.5
Error Equations for Line Segments for Trailer Control

Error	Equation
Lateral	$y_{err} = \dfrac{\left((n_1 \quad n_2)e_t + (e_2 \quad e_1)n_t + (e_1 n_2) \quad (e_2 n_1)\right)}{\sqrt{(e_2 \quad e_1)^2 + (n_2 + n_1)^2}}$
Heading	$\psi_{err} = \psi \quad \tan^{1} \dfrac{e_2 \quad e_1}{n_2 \quad n_1} \div$
Hitch angle	$\varphi_{err} = \varphi$
Yaw rate	$r_{err} = r$
Lateral velocity	$V_{yerr} = V_y$
Steer angle	$\delta_{err} = \delta$
Steer angle rate	$\dot{\delta}_{err} = \dot{\delta}$

Table 6.6
Error Equations for Arc Segments for Trailer Control

Error	Equation
Lateral	$y_{err} = \sqrt{(e_t \quad e_C)^2 + (n_t \quad n_C)^2} \quad R_t$
Heading	$\psi_{err} = \psi \quad \tan^{\ 1} \frac{e \quad e_C}{n \quad n_C} \div + \ \frac{\pi}{2} \quad cw \div$
Vehicle radius	$R = \sqrt{L_T{}^2 \quad L_H{}^2 + R_t^2}$
Hitch angle	$\varphi_{err} = \varphi \quad \tan^{\ 1}(M,N)$ where: $M = \quad \frac{L_H R_t R + L_I^3 + L_T R_t^2 \quad L_H^2 L_T}{R\left(L_T^2 + R_t^2\right)}$ $N = \frac{L_H L_T R + R_T L_T^2 + R_t^3 \quad R_t L_H^2}{R\left(L_T^2 + R_t^2\right)}$
Yaw rate	$r_{err} = r \quad \frac{V_x}{R}$
Lateral velocity	$V_{yerr} = V_y$
Steer angle	$\delta_{err} = \delta \quad \frac{L}{R}$
Steer angle rate	$\dot{\delta}_{err} = \dot{\delta}$

where

$$x = [y_{terr} \quad \varphi \quad \psi_{err} \quad r \quad b \quad \delta \quad \dot{\delta}]^T \tag{6.71}$$

The control gains *K* are again calculated to form an LQR as discussed in Section 6.5.3.2. Example weighting matrices are given in (6.72) and (6.73).

$$\begin{aligned} Q_x &= diag[Q_{yt} \quad Q_\varphi \quad Q_\psi \quad Q_r \quad Q_b \quad Q_\delta \quad Q_{\dot{\delta}}] \\ &= diag[15 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1] \end{aligned} \tag{6.72}$$

$$R_u = 0.1 \tag{6.73}$$

Table 6.7
Trailer Parameters

Parameter	Value
L_H	1.6m
L_T	3.5m
L_C	3.6m

Since we are concerned with controlling the trailer's lateral error, Q_{yt} was set to 15 to penalize that state. Additionally, Q_{δ} and Q_{δ}×were set to 1 to penalize steer angle and add additional damping to the system. The selected control weighting values were found experimentally by hand tuning until satisfactory performance was obtained.

6.7.4 Simulation Results

The vehicle pulling a trailer is simulated following a line segment with an initial offset of 2 meters and a forward velocity of 2 m/s. The parameters given

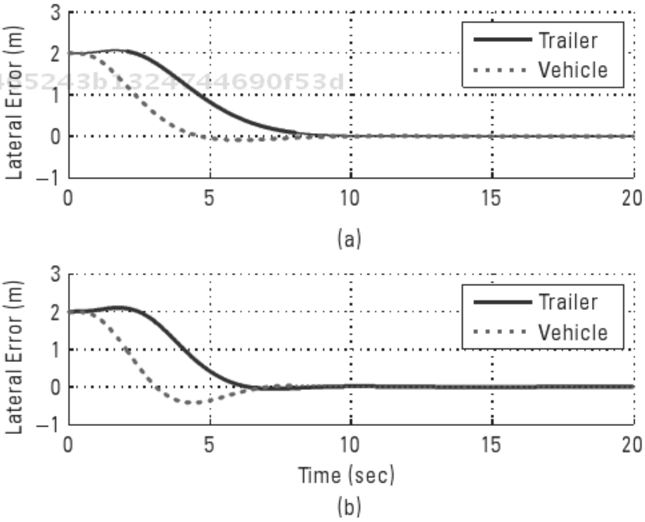


Figure 6.20 (a, b) Vehicle versus trailer controller comparison.

previously in Table 6.1 are used for the vehicle in the simulation. The trailer parameters are given in Table 6.7.

In Figure 6.18(a) the control algorithm given in Section 6.6.3 is used to control the position of the vehicle, while Figure 6.20(b) has the feedback wrapped around the position of the trailer using the algorithm given in Section 6.7.3. As seen in the figure, by controlling the trailer, the vehicle overshoots quite a bit in order to bring the trailer lateral error to zero almost 50% faster [and in a very similar fashion to the way the controller brings the lateral error of the tractor to zero in Figure 6.20(a)].

A simulation of the trailer along an arc segment is shown in Figure 6.21. The same parameters are used as the previous simulation with the vehicle's velocity increased to 4 m/s. The vehicle and trailer start tracking a straight line near (0m, 0m) and then follows a 15-meter radius arc centered at (40m, 15m). Both the trailer and vehicle positions for the simulated experiment are shown in the figure, with the feedback loop closed around the trailer's position. Note that the implement does not start tracking the arc at the tangent of the arc. Rather, the implement was tracking a line and then was asked to track the circle once it reached the edge of the arc. In order for the trailer to track the 15-m radius arc, the vehicle follows an arc of radius 15.4m. This discrepancy shows the importance of being able to control the trailer position on curved trajectories.

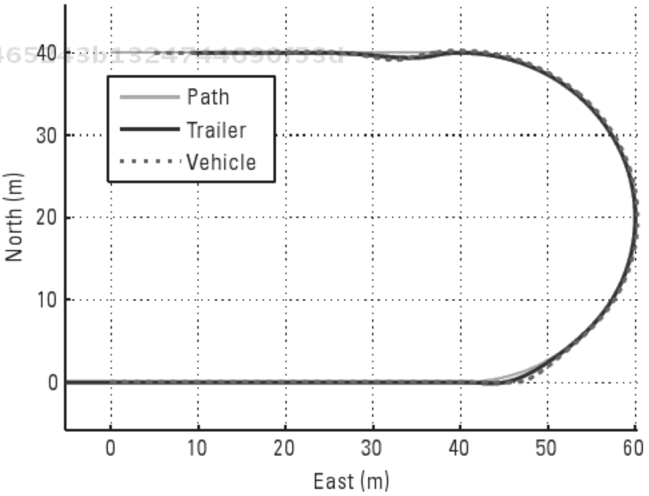


Figure 6.21 Trailer control along a line and arc.

References

[1] Dorf, R., and R. Bishop, *Modern Control Systems*, Upper Saddle River, NJ: Prentice-Hall, 2005.

[2] Brogan, W. L., *Modern Control Theory,* 3rd ed., Upper Saddle River, NJ: Prentice-Hall, 1991.

[3] Buehler, M., K. Iagnemma, and S. Singh, (eds.), *The 2005 DARPA Grand Challenge: The Great Robot Race*, Berlin: Springer, 2007.

[4] Massey, J. P., "Control and Waypoint Navigation of an Autonomous Ground Vehicle," Master's thesis, Texas A&M University, May 2006.

[5] Singh, S., et al., *A System for Fast Navigation of Autonomous Vehicles*, Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-91-20, September 1991.

[6] Travis, W., R. Daily, and D. M. Bevly, "SciAutonics-Auburn Engineering's Low-Cost, High-Speed ATV for the 2005 DARPA Grand Challenge," *Journal of Field Robotics*, Vol. 23, No. 8, 2006, pp. 579–597.

[7] Stengal, R. F., *Optimal Control and Estimation*, New York: Dover Publications, 1994.

[8] Bevly, D. M., "High Speed, Dead Reckoning, and Towed Implement Control for Automatically Steered Farm Tractors Using GPS," Ph.D. dissertation, Stanford University, August 2001.

[9] Hodo, D. W., "Development of an Autonomous Mobile Robot-Trailer System for UXO Detection," Master's thesis, Auburn University, August 2007.

[10] Hedrick, J. K., M. Tomizuka, and P. Varaiya, "Control Issues in Automated Highway Systems," *IEEE Control Systems Magazine*, Vol. 14, No. 6, December 1994, pp. 21–32.

[11] Rajamani, R., *Vehicle Dynamics and Control*, New York: Springer, 2006.

[12] Laumond, J. P., (ed.), "Robot Motion Planning and Control," Lecture Notes in Control and Information Sciences 229, Springer, 1998.

[13] Dubins, L. E., "On Curves of Minimal Length with a Constraint on Average Curvature and with Prescribed Initial and Terminal Positions and Tangents," *American Journal of Mathematics*, Vol. 79, No. 3, July 1957, pp. 297–516.

[14] Nelson, W., "Continuous-Curvature Paths for Autonomous Vehicles," *IEEE International Conference on Robotics and Automation*, Vol. 3, May 14–19, 1989, pp. 1260–1264.

[15] Fleury, S., et al., "Primitives for Smoothing Mobile Robot Trajectories," *IEEE Transactions on Robotics and Automation*, Vol. 1, No. 3, June 1995, pp. 441–448.

[16] Fraichard, T., and A. Scheuer, "From Reeds and Shepp's to Continuous Curvature Paths," *IEEE Transactions on Robotics and Automation*, Vol. 20, No. 6, December 2004, pp. 1025–1035.

[17] Yang, Y., and J. A. Farrell, "Magnetometer and Differential Carrier Phase GPS-Aided INS for Advanced Vehicle Control," *IEEE Transactions on Robotics and Automation*, Vol. 19, No. 2, April 2003, pp. 269–282.

[18] Lu, G., and M. Tomizuka, "Lidar Sensing for Vehicle Lateral Guidance: Algorithm and Experimental Study," *IEEE/ASME Transactions on Mechatronics*, Vol. 11, No. 6, December 2006, pp. 653–660.

[19] Redmill, K. A., "A Simple Vision System for Lane Keeping," *Proc. IEEE Conference on Intelligent Transportation System ITSC 97*, November 9–12, 1997, pp. 212–217.

[20] Cordesses, L., et al., "CP-DGPS Based Combine Harvester Control Without Orientation Sensor," *Proceedings of the 1999 ION-GPS Meeting*, Nashville, TN, September 1999.

[21] Bell, T., "Precision Robotic Control of Agricultural Vehicles on Realistic Farm Trajectories," Ph.D. dissertation, Stanford University, December 1999.

[22] Gonzalez-Cantos, A., J. I. Maza, and A. Ollero, "Design of a Stable Backing Up Fuzzy Control of Autonomous Articulated Vehicles for Factory Automation," *Proc. 8th IEEE International Conference on Emerging Technologies and Factory Automation*, October 15–18, 2001, pp. 447–451.

[23] Lamiraux, F., S. Sekhavat, and J.-P. Laumond, "Motion Planning and Control for Hilare Pulling a Trailer," *IEEE Transactions on Robotics and Automation*, Vol. 15, No. 4, August 1999, pp. 640–652.

[24] Hodo, D. W., et al., "Effects of Sensor Placement and Errors on Path Following Control of a Mobile Robot-Trailer System," *Proc. American Control Conference ACC '07*, 2007, pp. 2165–2170.

7

Pseudolites for Vehicle Navigation

Stewart Cobb

As discussed in Section 1.1.2, navigation with GNSS satellite signals requires that several (generally four) satellite signals be visible to the navigating receiver simultaneously, and the accuracy and reliability of GNSS navigation increases with each additional signal. Thus, GNSS navigation works well in outdoor environments with a clear view of much of the sky.

Unfortunately, ground vehicles are often required to operate in environments where much of the sky may be obscured by buildings, foliage, or terrain. Pseudolites (Section 1.2) have proved useful in such cases. This chapter examines several potential and actual applications of pseudolites, and several existing pseudolite systems, which have been deployed for such applications.

7.1 Pseudolite Applications

The most fruitful application of pseudolites to date has been in open-pit mining. Pseudolites have also been proposed to assist navigation in and around construction sites and through "urban canyons" formed by tall buildings in metropolitan areas. Some researchers have conducted experiments using groups of pseudolites indoors to enable GNSS-like navigation without access to any actual GNSS satellite signals. This section will discuss these actual and potential pseudolite applications.

7.1.1 Open-Pit Mining

Miners have become accustomed to using GNSS technology [1, 2]. Geologists use GNSS to plot the locations of surface samples and core samples, and then use the data to build a 3D map of the subsurface ore body. Mining engineers use the 3D map to plan their excavations efficiently. Surveyors use GNSS to lay out the excavation plan on the real-world topography. Blasting engineers define patterns of boreholes to be filled with explosives in order to fracture the proper amount of ore and rock, and drillers then use GNSS to place the holes precisely in the desired locations.

After the explosives are detonated, excavator shovels move in to load the broken rock and ore into giant haul trucks. One haul truck may take the valuable ore to a crusher for further processing; another haul truck takes the ordinary rock to a dump pile. Each shovel-full of ore or rock must be loaded into the correct haul truck and sent to the correct destination, or the mine will lose money. Unfortunately, ore and rock often look the same. Rather than trying to discriminate between them by eye, the excavator operators use displays driven by GNSS and the 3D ore body map to decide whether a given shovel-full contains ore or rock and to place it in the correct haul truck.

GNSS receivers mounted on the haul trucks allow a dispatcher to see the location of each truck in the mine and to direct the fleet of trucks most efficiently. Some mines are even beginning to experiment with robotic haul trucks, guided by GNSS and other sensors without a human driver.

As we can see, GNSS technology has become essential to the efficient operation of open-pit mines. Unfortunately, many ore bodies extend deep beneath the surface. As the mine pit becomes deeper, following the ore body, the walls of the pit appear higher from the mine floor. The higher walls block more of the sky and more of the available GNSS satellite signals. Since the 1990s, some mines have been shutting down operations during certain times of day because too few GNSS signals were available. These shutdowns can cost tens of thousands of dollars per hour in lost profit, but continuing to operate without good position data would cost even more [2].

Pseudolites provide an obvious solution to this problem. Pseudolites emplaced around the rim of the pit and at other appropriate locations can substitute for the GNSS satellite signals blocked by the walls of the pit, enabling mine operations to continue uninterrupted (Figure 7.1).

Beginning in 2000, Phelps-Dodge (now part of Freeport-MacMoran) installed a system of IN400 pseudolites from IntegriNautics (now Novariant) in their open-pit copper mine near Morenci, Arizona [3, 4]. This was the

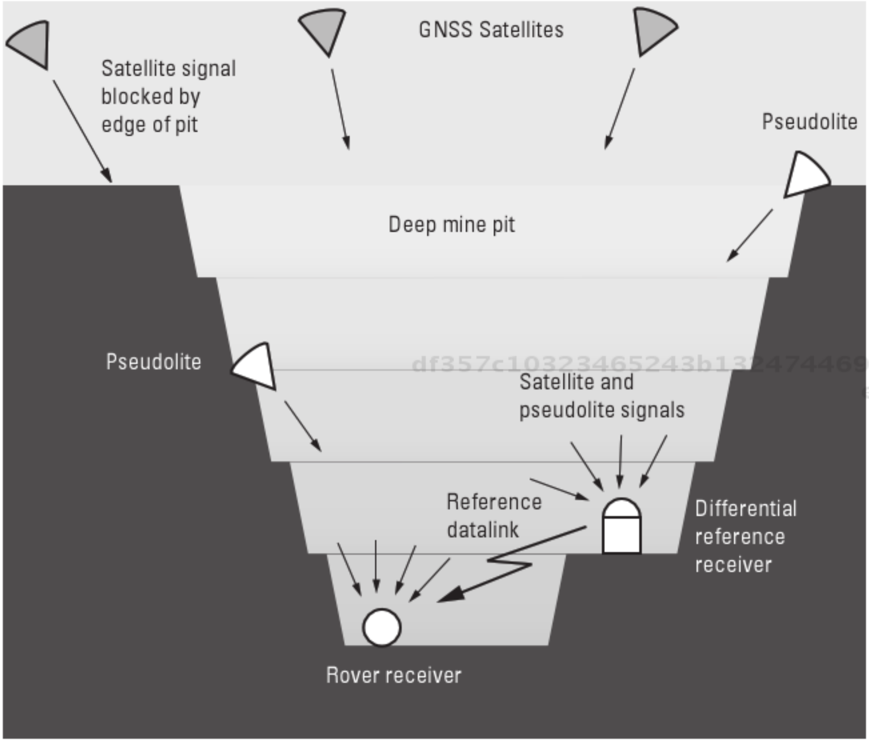

GNSS Satellites
Satellite signal blocked by edge of pit
Pseudolite
Deep mine pit
Pseudolite
Satellite and pseudolite signals
Reference datalink
Differential reference receiver
Rover receiver

Figure 7.1 Pseudolites replace GPS signals in an open-pit mine.

first operational pseudolite system installed in a mine. The IN400 system is described in Section 7.2.1.

Lessons learned from this installation contributed to the design of the follow-on Novariant Terralite XPS system. The Terralit XPS system was again first installed at Morenci, and has been in daily operational use since 2005 [4–6]. Since then, the Terralite XPS system has been installed in over a dozen mine sites and is becoming a standard in the industry. The Terralite XPS system is described in Section 7.2.2.

The Australian company Locata has also designed a system of pseudolites, known as LocataLites. Locata has performed several experiments with LocataLites in mines, but they have not yet reported that LocataLites have ever been installed for operational use in a mine. LocataLites are described in Section 7.2.3.

7.1.2 Construction Sites

GNSS technology has become pervasive in construction. Architects create site plans based on GNSS mapping data. Surveyors use GNSS to lay out the construction site. Operators of earthmoving equipment use GNSS-driven real-time displays of the site plan inside their cab to determine where, when, and how much to dig. In some cases, the excavator shovel or bulldozer blade is automatically guided by a GNSS system following the site plan.

All this GNSS technology works well in an open field with a clear view of the sky. Much construction work, however, takes place in built-up areas. If GNSS satellite signals are blocked by other buildings or vegetation in one direction, the PDOP (see Section 1.1.2) will be degraded, and the navigation accuracy with it. If satellite signals are blocked in too many directions, navigation will become impossible. Again, pseudolites have been proposed as a solution to these problems. A pseudolite placed on or near each obstruction can restore the usefulness of GNSS navigation as if the construction were taking place in an open field.

While pseudolites have been proposed for construction sites, there have been no published reports yet of even experiments in this area. Any of the pseudolite systems suitable for open-pit mining (see Section 7.1.1) should be suitable for construction sites as well.

7.1.3 Urban Navigation

In many urban areas, streets are surrounded by tall buildings, to the point that they have been called “urban canyons.” The buildings can block so many satellite signals that GNSS navigation becomes difficult or impossible at street level. Other techniques discussed in this book (inertial aiding, dead-reckoning, map-matching, and so forth) can be integrated into a navigation or guidance system, so that it is not wholly dependent on GNSS signals. Another approach, however, is to replace the blocked satellite signals with pseudolite signals. A pseudolite emplaced near the roof of a tall building, broadcasting down into the street below, can substitute for a satellite signal blocked by the building.

In 2001 and 2002, the government of Japan funded several experiments to assess the feasibility of pseudolite-aided GNSS navigation in urban areas [7]. University researchers produced a prototype assistive device, which used GNSS and pseudolite signals, digitized maps, voice recognition, and voice synthesis to help blind pedestrians navigate city streets. These experiments used IntegriNautics (now Novariant) IN400 pseudolites, and matching IN2400 receivers to augment GNSS navigation in the streets of downtown Tokyo (Figure 7.2).

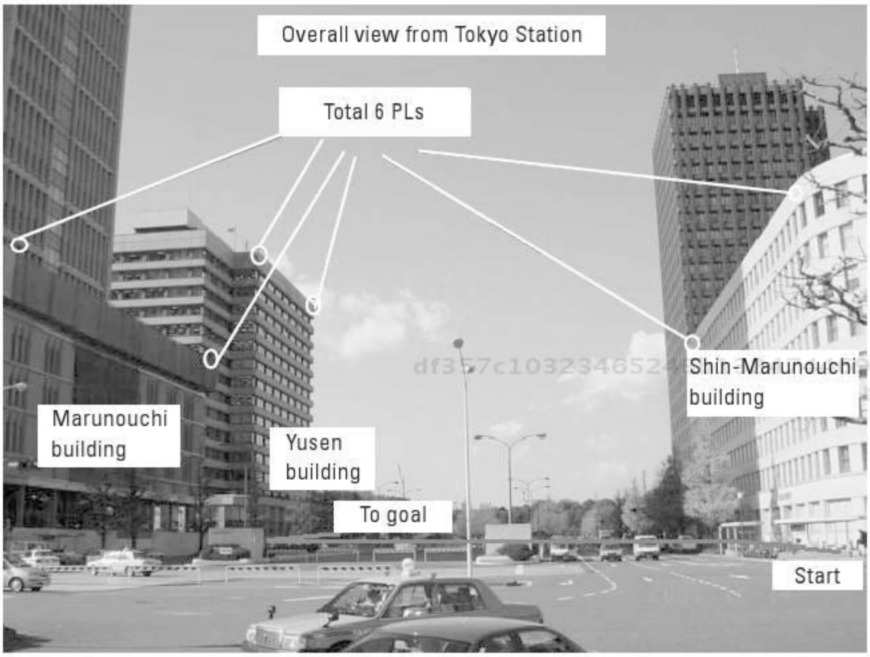


Figure 7.2 Pseudolites installed for an experiment on a Tokyo street (GNSS Technologies Inc.).

One can imagine a future in which pseudolites are widely distributed to help GNSS-equipped vehicles navigate through urban areas. If this future is to become real, three things must occur. A single standard for pseudolite signals must be defined and become widely accepted. The cost of equipping a street with pseudolites must become suitably small, perhaps comparable to the cost of equipping the same street with streetlights. Finally, the cost of adding pseudolite capability to a GNSS receiver must become negligible compared to the cost of the receiver itself. Given proper system design and economies of scale, these three goals may yet be achieved, and pseudolites may become an expected part of the urban infrastructure.

7.1.4 Indoor Applications

It is easy to imagine an "indoor GPS" system that uses pseudolites inside a building to replace the GPS satellite signals blocked by the building's roof. This would be difficult in practice to accomplish [8]. It might work in a large open space such as a gymnasium or concert hall, but it will not work well in a typical building containing corridors and rooms with low ceilings.

The problems are PDOP, signal blockage, and multipath. GPS navigation requires line of sight reception of signals from as many different directions as possible. Pseudolites placed on the ceiling and walls of a large open space can meet that requirement. A room with a low ceiling, however, presents problems. Good VDOP requires at least one pseudolite to be above (or perhaps below) the user, and it is difficult to achieve this for all locations in the room without scattering a very large number of pseudolites across the ceiling. One might neglect VDOP by simply assuming that a receiver familiar with the room was at a known height from the floor. With that assumption, the horizontal position could be calculated (with good HDOP) using signals from pseudolites placed in the corners of the room. Unless the room is empty, however, the line of sight signals from those pseudolites will likely be blocked by other objects in the room. The receiver may track pseudolite signals reflected around obstacles and compute a position, which is incorrect due to multipath error, or it may fail to compute a position at all due to an insufficient number of signals. Both cases are problematic for reliable navigation.

In a long, narrow room such as a hallway or tunnel, these problems are even worse. Unless a very large number of pseudolites are distributed along the ceiling and walls of the tunnel, a receiver will be unable to calculate either its height or its lateral position (across the width of the tunnel) with good DOP. The position along the length of the tunnel can be calculated, but only if the tunnel is sufficiently empty that the pseudolite signals are not obstructed. For these reasons, a conceptual "indoor GPS" system using pseudolites is generally impractical.

Despite these difficulties, it is possible to create a pseudolite system, which can give a GPS receiver a general sense of its location indoors. The idea is to use the pseudolite as the source of a data signal rather than a navigation signal [9]. Each indoor pseudolite can broadcast a preprogrammed signal describing its own location. A properly programmed GPS receiver that hears one of these pseudolites can report the pseudolite's location as its own. Such a system would not allow precise navigation, but it could provide location information sufficient for many uses, such as directing emergency workers to a specific incident. The placement of such data-signal pseudolites could be adjusted to be as sparse or dense as the particular application requires.

Unlike pseudolites for navigation, such data-transmitter pseudolites do not require precise clocks or synchronization systems. They can, therefore, be made quite small and inexpensive. The Japanese company GNSS Technologies is working to commercialize this technology, which they call Indoor Messaging System (IMES) [10].

7.2 Pseudolite Systems

At least three pseudolite systems were, are, or soon will be commercially available. These are the IntegriNautics IN400/IN2400 system, the Locata LocataLite system, and the Novariant (formerly IntegriNautics) Terralite XPS system. These three systems will be discussed in this section.

Other pseudolites have been sold for research purposes, but did not find uses beyond research. Examples include the IntegriNautics IN200, IN300, and IN500, which were briefly available during the 1990s. Some researchers have constructed their own pseudolites, including Chris Bartone's group at Ohio University in the United States and Changdon Kee's group at Seoul National University in Korea.

The original pseudolites, used in the 1970s in the original tests of the GPS concept, were built for the U.S. Air Force by Stanford Telecom. These were large complex devices, containing integral atomic clocks. They deserve a mention as the first pseudolites ever built, but they are long obsolete.

7.2.1 IntegriNautics IN400

The first pseudolite system actually used for ground vehicle control was the IntegriNautics (now Novariant) IN400 system. One such system was installed in an open-pit copper mine near Morenci, Arizona, operated by Phelps-Dodge (now part of Freeport-McMoran). The system consisted of several IN400 pseudolite transmitters, one IN2400 fixed reference receiver, and several IN2400 rover receivers. Figure 7.3 shows an overview of the entire system. Figure 7.4 shows one IN400 installed in the mine in 2001.

The IN400 pseudolite transmitted a signal on the same L1 frequency used by the GPS satellite system. The pseudolite signal itself closely resembled a GPS satellite signal, except that there was no attempt to synchronize the pseudolite's internal timebase to any other source of timing. The pseudolite's internal timebase was constrained by design so that the transmitted signal frequency was within about 2 parts per million of the nominal GPS frequency, which is within the expected Doppler frequency range of GPS satellite signals.

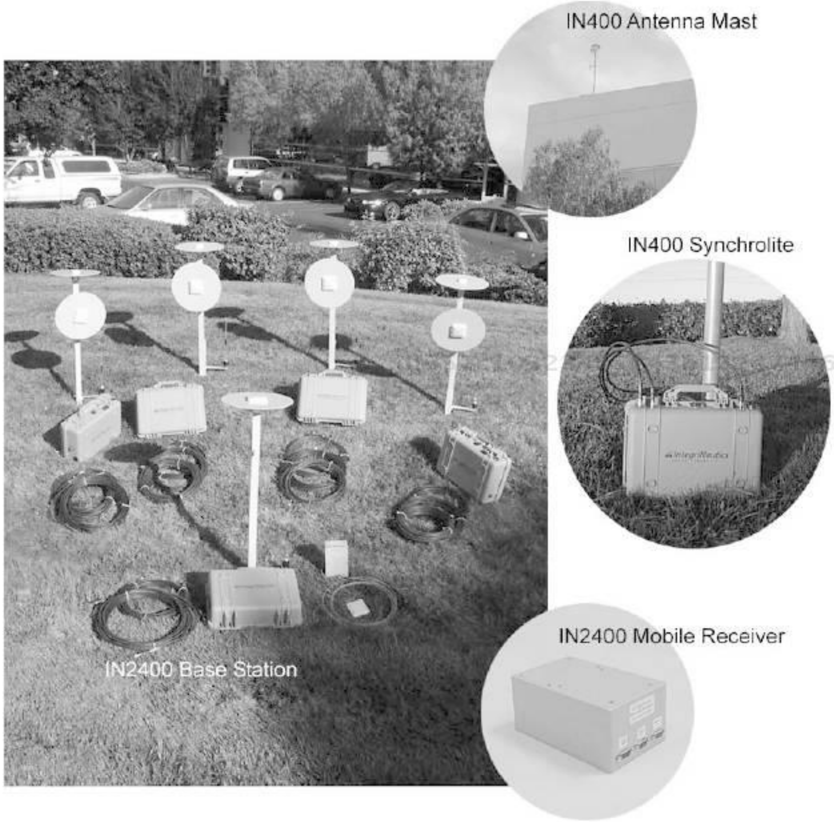


Figure 7.3 IntegriNautics IN400 pseudolite system (Novariant, Inc.).

To mitigate the near-far problem, each IN400 pseudolite transmitted its signal in short pulses. The timing of the transmitted pulses was controlled so that the pulses from different pseudolites did not overlap. To achieve this, each IN400 pseudolite synchronized its pulse generator to an accurate pulse-per-second generated by a GPS receiver contained within the pseudolite. The microsecond-level synchronization required for the pulse pattern was far easier to achieve than the nanosecond-level synchronization which would have been required to use the transmitted signal directly for precise positioning.

Because the signals from the pseudolites were not accurate enough to use directly for positioning, the overall system operated differentially. A centrally located base station received all available signals from the pseudolites and the visible GPS satellites. At particular instants determined by GPS sys-

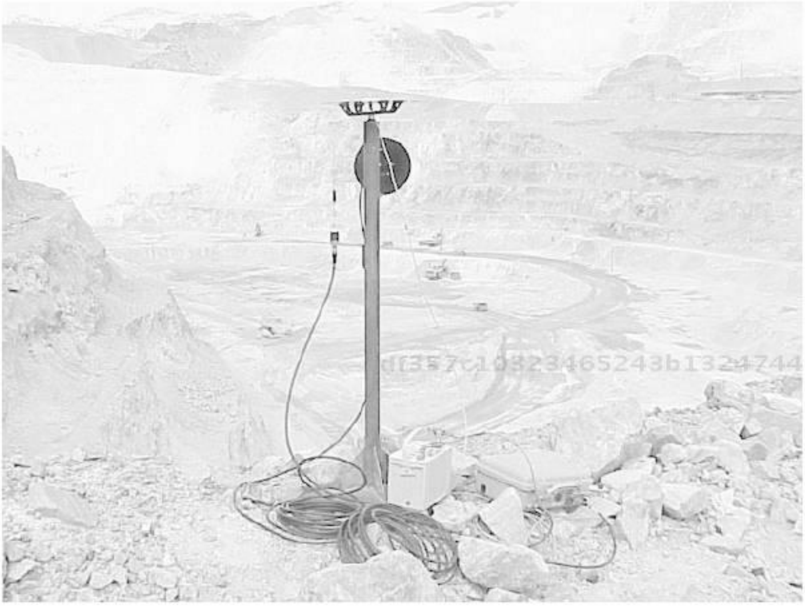


Figure 7.4 IntegriNautics IN400 Pseudolite in open-pit copper mine, Morenci, Arizona (Novariant, Inc.).

tem time, the base station simultaneously measured the timing of each signal it saw, and then broadcast this set of timing measurements over a secondary data link. At the same instant, the mobile (rover) receivers measured the timing of the signals they could hear, and then compared those measurements to the base station's measurements that they received over the datalink. The rover receivers used the results of these comparisons to compute their positions relative to the well-known location of the base station.

7.2.2 Novariant Terralite XPS System

As GPS systems have become more widespread, it has become more difficult to obtain permission to install pseudolites like the IN400, which transmit on the same frequencies as the GPS satellites. It is now essentially impossible to obtain a government license for a ground-based transmitter on any GPS frequency. This limitation, coupled with advances in technology, inspired Novariant to invent the Terralite XPS system [4].

Like the earlier IN400 system, the Terralite XPS system consists of a fixed ground constellation and any number of MX100 mobile rover receivers. The ground constellation consists of several (often six) TX100 Terralite transmitters (Figure 7.5) and one fixed IX100 Reference Station. The most obvious difference is that the Terralite XPS transmitters emit their signals in a frequency range known as the X-band, between 9.5 and 10 GHz. This band of frequencies is reserved under international treaties for radiolocation. This term has historically been used to mean "radar," but pseudolite systems are also radiolocation systems according to international frequency regulations. It is thus comparatively easy to obtain licenses to install Terralite XPS systems anywhere in the world.

The change to X-band also allows the Terralite XPS system to be much more precise than its pseudolite predecessors. The code chip rate in this band can be as high as 250 MBps, versus only 1 to 10 MBps in the GPS bands. The receiver's ability to measure position changes is increased by a corresponding amount. The carrier frequency is about 6 times higher than the GPS L1 frequency, so systems that use carrier phase measurements to determine position changes are more accurate as well. The major drawback to this frequency change is that radio electronics have been more expensive in the new band.

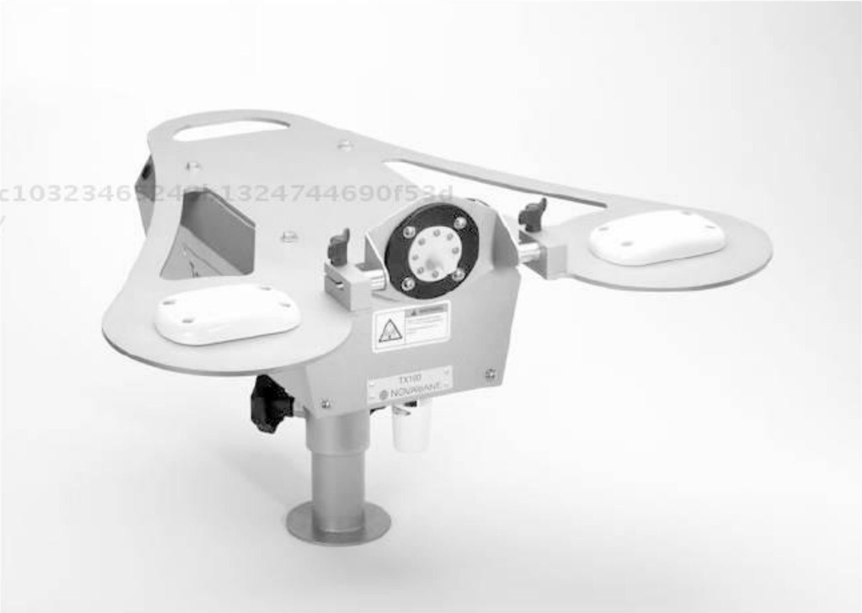


Figure 7.5 Novariant TX100 Terralite XPS transmitter (Novariant, Inc.).

However, direct broadcast satellite television systems using similar frequencies are now in mass production, which has made X-band radio parts newly affordable.

7.2.3 Locata LocataLites

Locata Corporation, based in Canberra, Australia, has designed a considerably different navigation system based on pseudolite-style transceivers, which they call LocataLites. The Locata system has evolved through at least three generations of LocataLite designs and, as of 2009, was not yet available for general sale outside the research community. Many details of the Locata system remain proprietary, but it is possible to describe some features of the system from published papers [11].

LocataLites do not use GNSS satellite signals, except that the positions of some (not all) LocataLites must be determined at installation using a GNSS survey. Each LocataLite transmits navigation signals and receives signals from other LocataLites. This two-way communication allows them to synchronize their internal clocks with the system as a whole, and allows LocataLites that were not initially surveyed to determine their positions within the system.

The signal format transmitted by LocataLites is not publicly documented, but the signals are transmitted in the 2.4-GHz band. This is an "unlicensed" radio band worldwide, meaning that no special license is required to transmit radio signals in this band, provided that the transmitting equipment itself meets certain requirements. LocataLites meet these requirements, as do many other modern devices, including microwave ovens, Bluetooth headsets, and Wi-Fi wireless internet connections. The potential for interference among these devices is obvious, and in fact Locata has redesigned their signal format at least once to minimize interference from Wi-Fi connections [12]. The transmitted signals carry both timing information and data, just as GNSS satellite signals do.

Destructive interference between direct and reflected (multipath) signals can cause the received signal strength at a particular location to be low or even unusable. This is known in the literature as "signal fading." Because the LocataLites do not move, the locations of severe fading of their signals also will not move. To combat fading, the current generation of LocataLites transmits signals on at least two frequencies within the 2.4-GHz band, and each LocataLite transmits through two antennas separated by about 2 meters. This improves the probability that at least one signal from one antenna will be usable at every point within the coverage volume of each LocataLite.

At system startup, none of the LocataLites know their clock offset or frequency errors, and many of them may not know their locations. As they take turns transmitting and listening to each other, the LocataLites synchronize their clocks and determine their relative distances from one another. Eventually the system converges, so that all the LocataLite clocks are synchronized and all the LocataLites have determined their locations relative to the few surveyed LocataLites. At this point, the system becomes operational and can be used for navigation.

Each Locata rover contains a receiver for GNSS satellite signals and another receiver for LocataLite signals. The receivers are tied to a common time base, so that GNSS and LocataLite signals can be used together in any combination to determine the rover's position.

References

[1] Coyle, R., and S. Holmes, "Smart Mining Systems," *Proceedings of Australian Mining Technology Conference*, October 2007.

[2] Flinn, J., "Practical Aspects of GPS Implementation at the Morenci Copper Mine," *Proceedings of Institute of Navigation GPS Conference,* Nashville, TN, 1999.

[3] Bartlett, M., et al., "GPS Pseudolites for Mining Operations," *Proceedings of MINExpo*, Las Vegas, NV, 2004.

[4] Zimmerman, K. R., et al., "A New GPS Augmentation Solution: Terralite™ XPS System for Mining Applications and Initial Experience," *Proceedings of Institute of Navigation GNSS Conference*, Long Beach, CA, 2005.

[5] Zimmerman, K. R., "Terralites for Mining Operations," *Proceedings of APCOM*, Tuscon, AZ, 2005.

[6] Carter, R. A., "Hybrid GPS System Solves Problems at Morenci," *Engineering and Mining Journal*, May 2006, pp. 46–52.

[7] Petrovski, I., et al., "Pseudolite Implementation for Social Infrastructure and Seamless Indoor/Outdoor Positioning," *Proceedings of ION GPS Conference*, Portland, OR, 2002.

[8] Zimmerman, K. R., "Experiments in the Use of the Global Positioning System for Space Vehicle Rendezvous," Ph.D. dissertation, Stanford University, 1996.

[9] Cobb, H. S., "GPS Pseudolites: Theory, Design, and Applications," Ph.D. dissertation, Stanford University, 1997.

[10] Manandhar, D., et al., "Development of Ultimate Seamless Positioning Method Based on QZSS IMES," *Proceedings of International Symposium on GPS/GNSS*, Odaiba, Tokyo, Japan, 2008.

[11] "Locata Corporation Technology Primer," Version 1.3, February 2005.

[12] Khan, F. A., et al., "Locata-Based Positioning in the Presence of WiFi Interference: Test Results," *Proceedings of Institute of Navigation GNSS Conference*, Savannah, GA, 2008.

Appendix
Estimation Methods

William Travis

A.1 Introduction

A brief overview of the estimation methods used in this book is presented in this appendix. A description of the underlying system model assumed for all derivations is provided, followed by techniques to discretize continuous systems. The least squares method is derived first and used as the foundation for all other derivations. Weighted and recursive least squares derivations show how to use statistical information to determine the minimum variance estimate. The derivation continues by presenting the Kalman filter, which is able to incorporate dynamic information about a system to produce an optimal state estimate. Finally, nonlinearities are handled in an extended Kalman filter.

The reader is encouraged to study these topics in greater detail if a fundamental understanding of estimation is desired. Excellent works dedicated to estimation methods are given in [1–5].

A.2 System Model

The derivations in this appendix operate on the assumption that a continuous system can be represented with a dynamics and measurement model in the following forms:

$$\dot{x}(t) = f(x(t), u(t), t) + L(t)\omega(t) \tag{A.1}$$

$$z(t) = h(x(t), t) + \upsilon(t) \tag{A.2}$$

where the rate of change of x is a function of the state vector x and input vector u at time t, plus an additive zero mean, white noise vector ω, and the measurement vector is some function of x at time t, plus an additive zero mean, white noise vector υ. The matrix L is the process noise gain matrix that relates the process noise to the state vector.

The covariance matrices of ω and υ are defined next. Note the assumption of no correlation between the process and measurement noise.

$$E\{\omega\omega^T\} = Q \tag{A.3}$$

$$E\{\upsilon\upsilon^T\} = R \tag{A.4}$$

$$E\{\omega\upsilon^T\} = 0 \tag{A.5}$$

In practice, one may encounter situations where process and measurement noises are correlated. Estimation algorithms can be derived to account for such a situation if necessary.

The nonlinear expressions in (A.1) and (A.2) can be linearized about some operating region and written in a continuous state space format.

$$\dot{x}(t) = F(t)x(t) + G(t)u(t) + L(t)\omega(t) \tag{A.6}$$

$$z(t) = H(t)x(t) + \upsilon(t) \tag{A.7}$$

The matrices F, G, and H are defined as follows:

$$F(t) = \frac{f(x(t), u(t), t)}{x(t)}\Big|_{x(t),u(t)} \tag{A.8}$$

$$G(t) = \frac{f(x(t), u(t), t)}{u(t)}\Big|_{x(t),u(t)} \tag{A.9}$$

$$H(t) = \frac{f(x(t), t)}{x(t)}\Big|_{x(t)} \tag{A.10}$$

A discrete state space representation is shown here.

$$x_k = \quad x_{k\ \ 1} + \quad u_k + \omega_k \tag{A.11}$$

$$z_k = H_k x_k \tag{A.12}$$

The continuous process gain matrix, L, is absorbed into the discrete process noise, ω_k, during the discretization process. If a direct discrete time model is defined, a discrete process gain matrix, , can be defined.

$$x_k = \quad x_{k\ \ 1} + \quad u_k + \quad \omega_k \tag{A.13}$$

A.3 Discretization

Navigation system dynamics are continuous time processes, but implementation usually occurs in a digital system. Discretization of the continuous time model can make the implementation process more convenient. The discretization process occurs under a zero-order hold assumption, that is, the input remains constant between discrete time intervals,

$$u(t) = u(t_k) = u_k \quad t_k \quad t < t_{k+1} \tag{A.14}$$

where the time interval is defined as follows:

$$t = t_{k+1} \quad t_k \tag{A.15}$$

The state transition matrix, , maps the state vector from some time, t, to a future time, τ, during which time the matrix F is assumed to be constant.

$$(\tau, t) = e^{F(t\ \ \tau)} \tag{A.16}$$

$$x(\tau) = \quad (\tau, t)x(t) \tag{A.17}$$

The time intervals can be reduced if variation in F invalidates the assumption stated above. Multiple state transition matrices can be multiplied together to produce the matrix capable of mapping the state vector over the desired time interval.

$$(\tau, t) = \quad (\tau, \tau\ ) \quad (\tau\ , t) \tag{A.18}$$

A state transition matrix can be produced that maps the state vector over discrete time intervals by incorporating (A.15) into (A.16).

$$(t_k, t_{k\quad 1}) = \quad {}_k = e^{F(t_k \quad t_{k\quad 1})} = e^{F \quad t} \tag{A.19}$$

Using the matrix defined above, a state vector $x_{k\quad 1}$ can be mapped to x_k.

$$x_k = \quad {}_k x_{k\quad 1} \tag{A.20}$$

The other matrices can be discretized through integration. The discrete input matrix is calculated as follows:

$${}_k = \ {}^{t_k}_{t_{k\quad 1}} \quad (t_k, \tau) G \, d\tau \tag{A.21}$$

The discrete process noise and process noise covariance are similarly defined.

$$\omega_k = \ {}^{t_k}_{t_{k\quad 1}} \quad (t_k, \tau) L\omega \, d\tau \tag{A.22}$$

$$Q_k = \ {}^{t_k}_{t_{k\quad 1}} \quad (t_k, \tau) LQL^T \quad (t_k, \tau)^T \ d\tau \tag{A.23}$$

Calculation of the discrete state space matrices using the formal definitions can be tedious, especially for complex systems. A method presented in [6] determines the state transition matrix and discrete process covariance matrix for time invariant or slowly time varying systems with a constant update interval.

$$A = \begin{array}{cc} F & LQL^T \\ 0 & F^T \end{array} \tag{A.24}$$

$$B = e^{A \quad t} = \begin{array}{cc} B & {}^1Q_k \\ 0 & {}^T \end{array} \tag{A.25}$$

If the sample rate is sufficiently small, the state matrices can be easily approximated.

$${}_k \quad I + F \quad t \tag{A.26}$$

$${}_k \quad G \quad t \tag{A.27}$$

$$Q_k \quad LQL^T \quad t \tag{A.28}$$

A.4 Least Squares

The goal of least squares is to determine an unbiased, minimum variance estimate of the state vector x using available measurements. The estimated state vector is denoted by $\hat{x}$. The error between the estimated state and true state, or state residual, can be defined as:

$$\varepsilon_x = x \quad \hat{x} \tag{A.29}$$

Ideally, this is a zero mean vector with some covariance given by P.

$$E[\varepsilon_x] = 0 \tag{A.30}$$

$$E[\varepsilon_x \varepsilon_x^T] = P \tag{A.31}$$

A quadratic cost function is used to define the optimal state estimate. The cost function incorporates the square of the state residual.

$$J(x) = \frac{1}{2}\varepsilon_x^T \varepsilon_x \tag{A.32}$$

Taking the square of the state residual guarantees a local minimum. The cost function, as defined, will fluctuate only with the magnitude of the state residual and therefore is independent of the sign of the state residual. This effect is depicted in Figure A.1. One could define the optimal state estimate by defining the cost function as the absolute value of the state residual. However, a discontinuity in the derivative of the absolute value operation at zero makes the derivation of the optimal state estimate difficult.

This state residual is often impractical to calculate, especially in real-time systems, as the true state is rarely known precisely. However, a measurement prediction, $\hat{z}$, can be formed given the state estimate.

$$\hat{z} = H\hat{x} \tag{A.33}$$

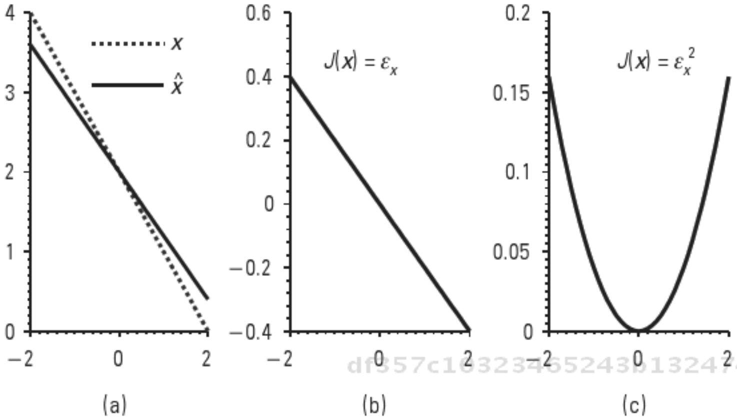


Figure A.1 (a) A true and estimated state. (b) A cost function is defined as the state residual. Notice how the error changes signs, becoming more negative as the independent variable increases. (c) A cost function is defined as the square of the state residual, where only the magnitude of the error fluctuates with the independent variable. A minimum is created, occurring at (0, 0).

The error between the true and predicted measurements can be determined. Note that this error, known as the measurement residual, contains not only errors in the estimated state, but also errors in the measurement.

$$\varepsilon_z = z \quad \hat{z} \tag{A.34}$$

The cost function can be redefined to use the readily available information in the measurement residual.

$$\begin{aligned} J(z) &= \frac{1}{2}\varepsilon_z^T \varepsilon_z = \frac{1}{2}(z \quad H\hat{x})^T (z \quad H\hat{x}) \\ &= \frac{1}{2(z^T z \quad z^T H\hat{x} \quad \hat{x}^T H^T z + \hat{x}^T H^T H\hat{x})} \end{aligned} \tag{A.35}$$

The optimal estimate of the state vector will be one that minimizes the magnitude of the error and therefore will minimize the cost function. A necessary condition for the existence of the minimum is the gradient of the cost function, or Jacobian, must equal zero when evaluated at the minimum.

$${}_{\hat{x}}J(z) = H^T H\hat{x} \quad H^T z = 0 \tag{A.36}$$

The normal equation is derived from the above expression:

$$H^T H\hat{x} = H^T z \tag{A.37}$$

The second derivative of the cost function is known as the Hessian.

$$^{2}_{\hat{x}}J(z) = H^T H \tag{A.38}$$

The resulting matrix will be an $n \quad n$ symmetric matrix. A sufficient condition for the existence of a minimum is that the Hessian must be positive definite. Positive definiteness is obtained when the rank of H is n, which means a minimum of n linearly independent observation equations exist. If the sufficient condition is satisfied, the least squares estimate of the state vector is defined as follows:

$$\hat{x}_k = (H_k^T H_k)\ ^{1} H_k^T z_k \tag{A.39}$$

Example A.1

A gyroscope is placed on a rate table for calibration. The rate table is extremely accurate, but the gyroscope output is corrupted with a constant scale factor and constant bias. Use least squares to determine the scale factor and bias.

A model of the gyroscope output must be constructed first.

$$z = ay + b + \upsilon \tag{A.40}$$

A scale factor, a, scales the true rotation rate given by the rate table, y, which is also biased by some amount, b, and corrupted by white noise, υ. With the understanding that multiple measurements are recorded, the model can be rewritten in matrix format.

$$z = \begin{matrix} ay_1 + b + \upsilon_1 \\ ay_2 + b + \upsilon_2 \\ \vdots \\ ay_m + b + \upsilon_m \end{matrix} = \begin{matrix} y_1 & 1 \\ y_2 & 1 \\ \vdots & \\ y_m & 1 \end{matrix} \begin{matrix} a \\ b \end{matrix} + \begin{matrix} \upsilon_1 \\ \upsilon_2 \\ \vdots \\ \upsilon_m \end{matrix} = Hx + \upsilon \tag{A.41}$$

Note that linear independence among the observations is achieved not by the number of measurements, but by the variation in the actual rotational rates observed. Measuring a single rotational rate does not provide enough information to estimate both the scale factor and bias terms, as their effects are indistinguishable. In the case of m measurements of the same rate, H can be written as follows:

$$H = \begin{matrix} y_1 & 1 \\ y_1 & 1 \\ \vdots & \\ y_1 & 1 \end{matrix}_{mx1} \tag{A.42}$$

where resulting Hessian is symmetric, but not positive definite as the rank is equal to one.

$$H^T H = \begin{matrix} my_1^2 & my_1 \\ my_1 & m \end{matrix} \tag{A.43}$$

Operating the table at two or more distinct rates increases the rank of H^TH to two. The following Hessian assumes m measurements with distinct rotational rates, and its rank is two.

$$H^T H = \begin{matrix} y_1^2 + y_2^2 + \cdots + y_m^2 & y_1 + y_2 + \cdots + y_m \\ y_1 + y_2 + \cdots + y_m & m \end{matrix} \tag{A.44}$$

If the sufficient condition is met, the least squares estimate of the scale factor and bias can be calculated with (A.39), where $\hat{x} = [\hat{a} \quad \hat{b}]^T$.

A.5 Weighted Least Squares

More often than not, available measurements are not equal in terms of quality. The regular least squares method does not account for statistical differences among the measurements. A simple addition of a weighting matrix to the cost function adds the capability to weight some measurements over others to improve the overall accuracy of the state estimate. Weighting is proportional to the anticipated quality of the measurement; the weight should approach infinity as the quality increases, and it should approach zero as the quality decreases.

Given a series of measurements at time k, a weight is assigned to each.

$$\begin{aligned} z_1 &= H_1 x_k + \upsilon_1 \qquad w_1 \\ z_2 &= H_2 x_k + \upsilon_2 \qquad w_2 \\ &\vdots \\ z_m &= H_m x_k + \upsilon_m \qquad w_m \end{aligned} \tag{A.45}$$

The weights are placed into a diagonal weighting matrix, W, where each element is in the row corresponding to its measurement.

$$W = \begin{matrix} w_1 & \cdots & 0 \\ & \ddots & \\ 0 & \cdots & w_m \end{matrix} \tag{A.46}$$

The cost function in (A.35) is rewritten using the weighting matrix.

$$\begin{aligned} J(z) &= \frac{1}{2}\varepsilon_z^T W \varepsilon_z = \frac{1}{2}(z \quad H\hat{x})^T W (z \quad H\hat{x}) \\ &= \frac{1}{2}(z^T Wz \quad z^T WH\hat{x} \quad \hat{x}^T H^T Wz + \hat{x}^T H^T WH\hat{x}) \end{aligned} \tag{A.47}$$

The Jacobian, normal equations, and Hessian are updated accordingly.

$$_{\hat{x}} J(z) = H^T WH\hat{x} \quad H^T Wz = 0 \tag{A.48}$$

$$H^T WH\hat{x} = H^T Wz \tag{A.49}$$

$$^2_{\hat{x}} J(z) = H^T WH \tag{A.50}$$

The weighted least squares solution is expressed as follows:

$$\hat{x}_k = (H_k^T W_k H_k) \ ^1 H_k^T W_K z_k \tag{A.51}$$

If statistical information of the measurements exists, the inverse of the covariance of the measurements can be used as a weighting matrix. The weighted least squares estimator produces the best linear unbiased minimum variance estimate, in a statistical sense, when using the inverse of the measurement covariance as a weighting factor.

The measurement covariance, R, is defined as the expectation of the square of the measurement noise, assuming all measurement noise is zero mean.

$$R = \mathrm{cov}(\upsilon) = E[\upsilon\upsilon^T] \tag{A.52}$$

The weighting matrix can be defined as the inverse of the measurement covariance matrix.

$$W = R^{\ 1} \tag{A.53}$$

The linear, unbiased, minimum variance estimator is typically expressed as follows:

$$\hat{x}_k = \left(H_k^T R_k^{\ 1} H_k\right)^{\ 1} H_k^T R_k^{\ 1} z_k \tag{A.54}$$

Often, it is useful to express the estimator in terms of an estimate of the covariance of the estimation error, P, which is defined in (A.31).

$$\begin{aligned} P_k &= \left(H_k^T R_k^{\ 1} H_k\right)^{\ 1} \\ \hat{x}_k &= P_k H_k^T R_k^{\ 1} z_k \end{aligned} \tag{A.55}$$

The term $(H^T R^{\ 1} H)^{\ 1}$ maps the known statistical information of the measurement vector to the state vector. Therefore, the matrix P should be a symmetric, positive definite matrix that contains statistical information of the estimated state vector (i.e., $P = E[\varepsilon_x\, \varepsilon_x^T]$). Note that the expression in (A.55) can be used in lieu of (A.51), but caution is advised when analyzing because P can be arbitrary due to the fact that elements in W might not be accurately selected.

Example A.2

A localization system consists of four radio beacons at known locations and a receiver. The receiver uses the signal broadcast from a beacon to determine a range measurement to that beacon. Therefore, the receiver in the described system shown in Figure A.2 will provide four range measurements to four known locations, measured in units of meters. The beacons are located at the

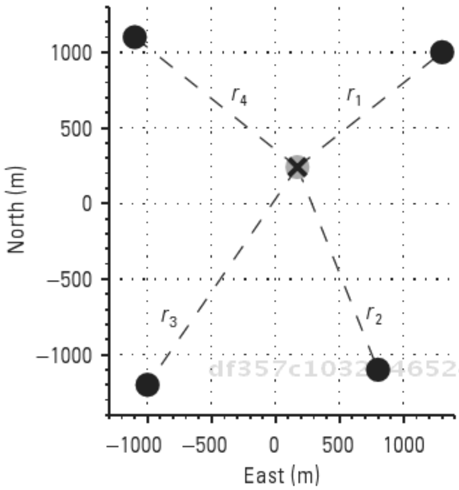


Figure A.2 A user wants to determine his position in a local, two-dimensional coordinate system. The black dots are the location of the radio beacons, and the dot marked with an X is the user's actual location. The user has a receiver that measures the range from it to the radio beacons.

following points in a local, two-dimensional frame: (1,300, 1,000); (800, 1,100); (1,000, 1,200); and (1,100, 1,100). The range measurements are corrupted by white noise, but the amplitude of the noise is dependent upon the distance between the beacon and receiver. The noise standard deviation was experimentally determined as $\sigma_r = \mu r$, where r is the true range and μ is a scaling factor. The measurements have no correlation with one another. A user wishes to determine his static position within a local, two-dimensional reference frame. Use the minimum variance estimator to determine his optimal position.

A range measurement can be modeled as the norm of the difference in position between the radio beacon and the user location, where r_u^b is the range from the user to beacon, e_u and n_u are the respective east and north components of the user position, and e^b and n^b are the respective east and north components of the beacon position.

$$r_u^b = f(e_u, n_u) = \sqrt{(e_u \quad e^b)^2 + (n_u \quad n^b)^2} \tag{A.56}$$

An estimate of the user position can be used to calculate a range estimate, since the user position is unknown.

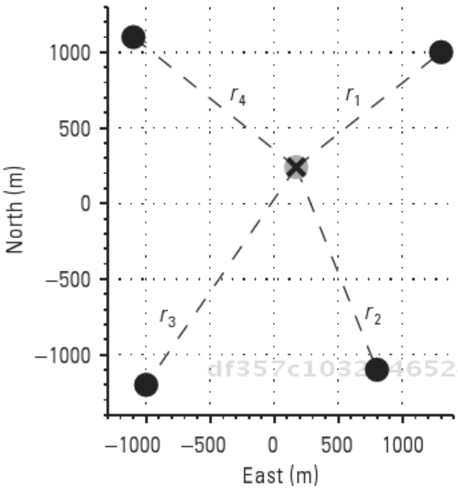


Figure A.2 A user wants to determine his position in a local, two-dimensional coordinate system. The black dots are the location of the radio beacons, and the dot marked with an X is the user's actual location. The user has a receiver that measures the range from it to the radio beacons.

following points in a local, two-dimensional frame: (1,300, 1,000); (800, 1,100); (1,000, 1,200); and (1,100, 1,100). The range measurements are corrupted by white noise, but the amplitude of the noise is dependent upon the distance between the beacon and receiver. The noise standard deviation was experimentally determined as $\sigma_r = \mu r$, where r is the true range and μ is a scaling factor. The measurements have no correlation with one another. A user wishes to determine his static position within a local, two-dimensional reference frame. Use the minimum variance estimator to determine his optimal position.

A range measurement can be modeled as the norm of the difference in position between the radio beacon and the user location, where r_u^b is the range from the user to beacon, e_u and n_u are the respective east and north components of the user position, and e^b and n^b are the respective east and north components of the beacon position.

$$r_u^b = f(e_u, n_u) = \sqrt{(e_u \quad e^b)^2 + (n_u \quad n^b)^2} \tag{A.56}$$

An estimate of the user position can be used to calculate a range estimate, since the user position is unknown.

$$\hat{r}_u^b = f\left(\hat{e}_u \quad \hat{n}_u\right) = \sqrt{\left(\hat{e}_u \quad e^b\right)^2 + \left(\hat{n}_u \quad n^b\right)^2} \qquad \text{(A.57)}$$

The measurement residual, $\varepsilon_r = r - \hat{r}$, which is the basis of the cost function, is formed using (A.56) and (A.57). However, these expressions are nonlinear at this point and therefore cannot be placed into a state space format. Ideally, the measurement residual is zero. The following expression can be formed assuming a zero measurement residual.

$$f(e_u, n_u) = f(\hat{e}_u, \hat{n}_u) \qquad \text{(A.58)}$$

This equation can be linearized with a Taylor series expansion where the zero- and first-order terms are retained to develop an expression for position error as a function of the range measurements and beacon positions. Expanding (A.58) leads to the following:

$$f(e_u, n_u) = f(\hat{e}_u, \hat{n}_u) + \frac{f(\hat{e}_u, \hat{n}_u)}{\hat{e}_u}\bigg|_{\hat{e}_u, \hat{e}_u} \div (e_u \quad \hat{e}_u) + \frac{f(\hat{e}_u, \hat{n}_u)}{\hat{n}_u}\bigg|_{\hat{e}_u, \hat{e}_n} \div (n_u \quad \hat{n}_u) \qquad \text{(A.59)}$$

where $f(e_u, n_u)$ is the measured range; $f(\hat{e}_u, \hat{n}_u)$ is the position estimate calculated with the right side of (A.57); $\frac{f(\hat{e}_u, \hat{n}_u)}{\hat{e}_u}$ and $\frac{f(\hat{e}_u, \hat{n}_u)}{\hat{n}_u}$ are the components of the unit vector pointing from the receiver to radio beacon; and $(e_u - \hat{e}_u)$ and $(n_u - \hat{n}_u)$, which can be expressed as $\quad e$ and $\quad n$, are the errors in the estimated state. Placing this information into (A.59) yields the following:

$$r_u^b = \sqrt{\left(\hat{e}_u \quad e^b\right)^2 + \left(\hat{n}_u \quad n^b\right)^2} + u_e \quad e + u_n \quad n \qquad \text{(A.60)}$$

where u_e and u_n are the unit vectors described above.

$$u_e = \frac{e^b \quad e_u}{\sqrt{\left(\hat{e}_u \quad e^b\right)^2 + \left(\hat{n}_u \quad n^b\right)^2}}$$

$$u_n = \frac{n^b \quad n_u}{\sqrt{\left(\hat{e}_u \quad e^b\right)^2 + \left(\hat{n}_u \quad n^b\right)^2}} \qquad \text{(A.61)}$$

The only remaining unknown terms in (A.60), with initial approximations of $\hat{e}_u$ and $\hat{n}_u$, are the errors in the estimated state. The equation can be rewritten and placed into a $z = Hx$ form, where $m = 4$ for this example.

$$r_u^b \quad \sqrt{(\hat{e}_u \quad e^b)^2 + (\hat{n}_u \quad n^b)^2} = +u_e \quad e + u_n \quad n \tag{A.62}$$

$$z = \begin{matrix} r_{u1}^b \quad \sqrt{\left(\hat{e}_u \quad e_1^b\right)^2 + \left(\hat{n}_u \quad n_1^b\right)^2} \\ \vdots \\ r_{um}^b \quad \sqrt{\left(\hat{e}_u \quad e_m^b\right)^2 + \left(\hat{n}_u \quad n_m^b\right)^2} \end{matrix} \tag{A.63}$$

$$H = \begin{matrix} u_{e1} & u_{n1} \\ \vdots & \vdots \\ u_{em} & u_{nm} \end{matrix} \tag{A.64}$$

$$x = [\quad e \qquad n]^T \tag{A.65}$$

An attempt to estimate the position can be made at this point. However, the statistical quality information of the measurements would not be utilized. The covariance matrix is formed with the estimated range since the noise is dependent upon distance.

$$R = \mu \quad \begin{matrix} \left(\mu \hat{r}_{u1}^b\right)^2 & \cdots & 0 \\ & \ddots & \\ 0 & \cdots & \left(\mu \hat{r}_{um}^b\right)^2 \end{matrix} \tag{A.66}$$

After each iteration, the position estimate should be updated by adding the error term to the original linearization point.

$$\begin{aligned} \hat{e}_u &= \hat{e}_u + \quad e \\ \hat{n}_u &= \hat{n}_u + \quad n \end{aligned} \tag{A.67}$$

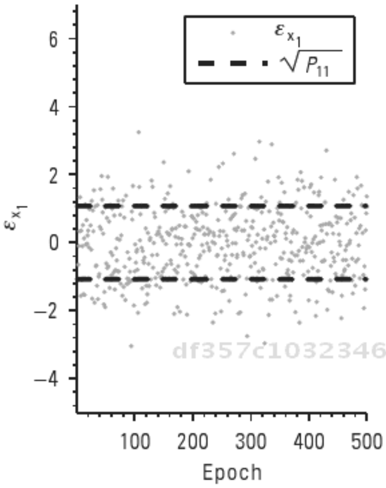


Figure A.3 The state residual for east position is plotted. The black dashed lines represent the estimated standard deviation of the residual.

Figures A.3 and A.4 were generated by simulating this example with an initial position estimate of (0, 0) and $\mu = 0.001$. Figure A.3 displays the state residual for the east position and the estimated standard deviation. The estimated standard deviation was determined by taking the square root P_{11}.

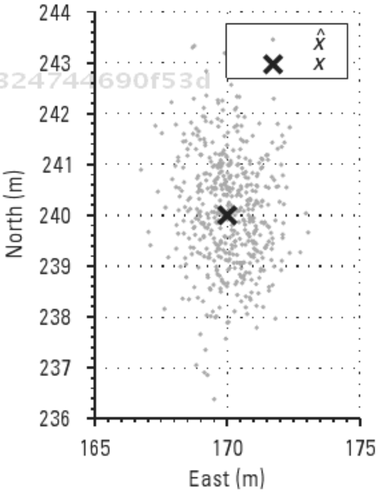


Figure A.4 A plot of estimated north and east position centered around the true position, which is denoted by X.

Figure A.4 shows the estimated position compared to the true position. The estimated points are bounded by a circle around the true position. This example presented an overdetermined solution with near optimal geometry, so different beacon locations and/or a different quantity of beacons would alter the shape of the bounding region.

A.6 Recursive Weighted Least Squares

The previously derived estimation formulas given in (A.39) and (A.55) are batch processing algorithms in the sense that all available data at some time k is used to compute an estimate. Estimates can be determined with a single batch of data where all data prior to and including time k. Sequential batch processing would use only information available at time k. One possible hindrance to sequential batch processing is the estimated state vector does not rely upon previous measurements; its value is determined solely from the observations at k. Real-time applications present the challenge of noncoincident information availability and data storage limitations, so an ideal estimation algorithm would process measurements sequentially to update the state vector. The recursive least squares algorithm processes available measurements at each instance in time to computes an estimator gain. This gain is multiplied by the measurement residual to determine the adjustment to be applied to the state vector. The result is equivalent to batch processing all past and current data.

Given two measurement vectors, sequential in time from $k\ \ 1$ and k:

$$\begin{aligned} z_{k\ 1} &= H_{k\ 1}x + \upsilon_{k\ 1} \\ z_k &= H_k x + \upsilon_k \end{aligned} \tag{A.68}$$

where the least squares estimate of the state at time k is given by

$$\hat{x}_{k\ 1} = \left(H_{k\ 1}^{T} R_{k\ 1}^{\ 1} H_{k\ 1}\right)^{\ 1} H_{k\ 1}^{T} R_{k\ 1}^{\ 1} Z_{k\ 1} \tag{A.69}$$

The sequential measurements can be grouped to determine the state estimate at time k using measurements from both epochs.

$$\bar{z} = \bar{H}x + \bar{\upsilon} \tag{A.70}$$

$$\bar{z} = \begin{matrix} z_{k\ 1} \\ z_k \end{matrix} \quad \bar{H} = \begin{matrix} H_{k\ 1} \\ H_k \end{matrix} \quad \bar{\upsilon} = \begin{matrix} \upsilon_{k\ 1} \\ \upsilon_k \end{matrix} \quad \bar{R} = \begin{matrix} R_{k\ 1} & 0 \\ 0 & R_k \end{matrix} \tag{A.71}$$

The estimate $\hat{x}_{k+1}$ is computed using the grouped measurements.

$$\hat{x}_k = \left(\bar{H}^T \bar{R}^{\ \ 1}\bar{H}\right)^{\ 1} \bar{H}^T \bar{R}^{\ \ 1}\bar{z} \qquad \text{(A.72)}$$

This equation can be expanded as follows:

$$\hat{x}_k = \left(H_{k\ \ 1}^T R_{k\ \ 1}^{\ 1} H_{k\ \ 1} + H_k^T R_k^{\ 1} H_k\right)^{\ 1}\left(H_{k\ \ 1}^T R_{k\ \ 1}^{\ 1} z_{k\ \ 1} + H_k^T R_k^{\ 1} z_k\right) \qquad \text{(A.73)}$$

Equation (A.55) presented

$$P_{k\ \ 1} = \left(H_{k\ \ 1}^T R_{k\ \ 1}^{\ 1} H_{k\ \ 1}\right)^{\ 1} \qquad \text{(A.74)}$$

so P_k can be similarly defined.

$$P_{k+1} = \left(H_{k\ \ 1}^T R_{k\ \ 1}^{\ 1} H_{k\ \ 1} + H_k^T R_k^{\ 1} H_k\right)^{\ 1} = \left(P_{k\ \ 1}^{\ 1} + H_k^T R_k^{\ 1} H_k\right)^{\ 1} \qquad \text{(A.75)}$$

Using the matrix inversion lemma, P_k can be redefined.

$$P_k = P_{k\ \ 1} \quad P_{k\ \ 1} H_k^T \left(H_k P_{k\ \ 1} H_k^T + R_k\right)^{\ 1} H_k P_{k\ \ 1} \qquad \text{(A.76)}$$

Equation (A.76) can be substituted into (A.73) to determine a recursive expression for the state estimate update.

$$\begin{aligned}
\hat{x}_k = &\quad P_{k\ \ 1} \quad P_{k\ \ 1} H_k^T \left(H_k P_{k\ \ 1} H_k^T\right)^{\ 1} H_k P_{k\ \ 1} \quad * \left(H_{k\ \ 1}^T R_{k\ \ 1}^{\ 1} z_{k\ \ 1} + H_k^T R_k^{\ 1} z_k\right) \\
= &\ P_{k\ \ 1} H_{k\ \ 1}^T R_{k\ \ 1}^{\ 1} z_{k\ \ 1} \quad P_{k\ \ 1} H_k^T \left(H_k P_{k\ \ 1} H_k^T + R_k\right)^{\ 1} H_k P_{k\ \ 1} H_{k\ \ 1}^T R_{k\ \ 1}^{\ 1} z_{k\ \ 1} \\
&+ P_{k\ \ 1} H_k^T R_k^{\ 1} z_k \quad P_{k\ \ 1} H_k^T \left(H_k P_{k\ \ 1} H_k^T + R_k\right)^{\ 1} H_k P_{k\ \ 1} H_k^T R_k^{\ 1} z_k \\
= &\ \hat{x}_{k\ \ 1} \quad P_{k\ \ 1} H_k^T \left(H_k P_{k\ \ 1} H_k^T + R_k\right)^{\ 1} H_k \hat{x}_{k\ \ 1} \\
&+ P_{k\ \ 1} H_k^T \quad I \quad P_{k\ \ 1} H_k^T \left(H_k P_{k\ \ 1} H_k^T + R_k\right)^{\ 1} H_k P_{k\ \ 1} H_k^T \quad R_k^{\ 1} z_k
\end{aligned} \qquad \text{(A.77)}$$

$$I = \left(H_k P_{k\ 1} H_k^T + R_k\right)^{\ 1} \left(H_k P_{k\ 1} H_k^T + R_k\right) \tag{A.78}$$

$$\hat{x}_k = \hat{x}_{k\ 1} + P_{k\ 1} H_k^T \left(H_k P_{k\ 1} H_k^T + R_k\right)^{\ 1} * \left(z_k \quad H_k \hat{x}_{k\ 1}\right) \tag{A.79}$$

The estimator gain, K, can be pulled from the recursive equation to update $\hat{x}$.

$$K_k = P_{k\ 1} H_k^T \left(H_k P_{k\ 1} H_k^T + R_k\right)^{\ 1} \tag{A.80}$$

The estimator gain is a function of the prior estimated state covariance and the current measurement and measurement covariance matrices. The gain is multiplied by the residual between the current measurement and an estimate of the current measurement based off of the previous state estimate. The resulting vector is an adjustment added to the previous state estimate to determine the current state estimate. The recursive least squares algorithm can be summarized as follows:

$$\begin{gathered} K_k = P_{k\ 1} H_k^T \left(H_k P_{k\ 1} H_k^T + R_k\right)^{\ 1} \\ \hat{x}_k = \hat{x}_{k\ 1} + K_k \left(z_k \quad H_k \hat{x}_{k\ 1}\right) \\ P_k = \left(P_{k\ 1}^{\ 1} + H_k^T R_k^{\ 1} H_k\right)^{\ 1} \end{gathered} \tag{A.81}$$

Notice that the estimated state error covariance is a function of the measurement matrix and measurement covariance matrix, and the gain is reliant upon the estimated state covariance matrix, but neither have any dependence on the actual value of the estimated state. Because of this relationship, or lack thereof, an accurate state estimate does not guarantee a correspondingly accurate covariance, and an acceptable covariance value does not indicate a correct estimate. Fortunately (or unfortunately), the fault of these occurrences can usually be assigned to the user, and not the algorithm, in the form of modeling error.

The fact that modeling error would corrupt an estimator's output is quite obvious, but it is a common occurrence nonetheless. The model may contain an invalid assumption of the signal. For instance, (A.56) contains the assumption that the beacons and receiver operate on a common, perfect clock. In reality this can never be the case, and therefore any clock biases and delays will cause the output to deviate from its true value. This error would manifest itself in the state estimate, and the residuals would become biased.

It is important for the user to realize that recursive least squares should not be used to estimate dynamic states. This would be a form of a modeling error, as the user improperly assumes estimated signals will be static. The gain matrix approaches zero as time increases, which diminishes the influence of the measurement residual on the state update. Any changes to the state would effectively be de-weighted and heavily filtered. The following section discusses the Kalman filter, which uses knowledge of a system's response as a function of time to propagate the state and state covariance forward in time.

A.7 Kalman Filter

If a system's model and initial conditions are known exactly, the states can be propagated forward in time, possibly based on some external input, with an infinite degree of certainty. The need for an estimator would be nonexistent. Consider the case where an airplane departs Atlanta, Georgia, traveling towards Dallas, Texas. The flight distance is known exactly, and the fuel consumption model is perfect. To save money, the airline puts the exact amount of fuel necessary to get the plane from Atlanta to Dallas, which was determined by the model.

Most people would not board that plane with this knowledge. What if the plane's route is diverted around a storm, a headwind is encountered, or the cargo is unusually heavy? Reality dictates uncertainty in the model and initial states. Modeling inaccuracies will usually exist due to assumptions made when deriving the model, effects experienced during operation that were unaccounted for when deriving the original model, or unknown system components that affect the behavior. Variation can exist among initial states even in highly controlled environments. Propagation of a system model will quickly produce inaccurate estimates, but external measurements of the states can be used to correct any drift in the estimates that may occur.

The previous section discussed the use of stochastic information to combine weighted measurements and produce a minimum variance estimate. The state vector could not be defined exactly, but an estimate of the state covariance provided a quality indicator of the state estimate accuracy.

The Kalman filter incorporates dynamic and stochastic information with measurements of the states to produce an optimal, nonbiased, minimum variance estimate. Estimates and their covariance are propagated from time $k\quad 1$ to k using a mathematical model of the system's dynamic and any known input. Measurements of the states are recorded, and the estimator gain, K, is optimally determined using knowledge of the measurement quality and information about process noise entering the system. Once the gain is determined, the states and their covariance are updated with the measurements.

Propagating the state forward in time uses the state transition matrix, $_{k,\,k\quad 1}$, and a deterministic input, u, if one exists. The following equation propagates the state estimates.

$$\hat{x}_k = \quad \hat{x}_{k\quad 1} + \quad u_{k\quad 1} \tag{A.82}$$

Note that the subscripts on the system matrices have been dropped for simplicity.

Covariance propagation starts by defining the covariance at successive instances in time.

$$P_{k\quad 1} = E\quad \varepsilon_{x_{k\quad 1}}\ \varepsilon^T_{x_{k\quad 1}} \tag{A.83}$$

$$P_k = E\quad \varepsilon_{x_k}\,\varepsilon^T_{x_k} \tag{A.84}$$

Noting that the state transition matrix can also be used to propagate the error,

$$\varepsilon_{x_k} = \quad \varepsilon_{x_{k\quad 1}} + \quad \omega_{k\quad 1} \tag{A.85}$$

Equation (A.82) can be rewritten as follows:

$$P_k = E\quad\Big(\quad \varepsilon_{x_{k\quad 1}} + \quad \omega_{k\quad 1}\Big)\Big(\quad \varepsilon_{x_{k\quad 1}} + \quad \omega_{k\quad 1}\Big)^T \tag{A.86}$$

Expanding the expression inside the expected value operator and then evaluating the expected value of the terms yields the discrete expression to propagate the estimated state error covariance.

$$
\begin{aligned}
P_k = E & \quad \varepsilon_{x_k\ 1}\varepsilon^T_{x_k\ 1} \quad ^T + \quad \varepsilon_{x_k\ 1}\omega^T_{k\ 1} \quad ^T \\
& + \quad \omega_{k\ 1}\varepsilon^T_{x_k\ 1} \quad ^T + \quad \omega_{k\ 1}\omega^T_{k\ 1} \quad ^T \\
= & \quad E\ \ \varepsilon_{x_k\ 1}\varepsilon^T_{x_k\ 1} \quad ^T + \quad E\ \ \varepsilon_{x_k\ 1}\omega^T_{k\ 1} \quad ^T \\
& + \quad E\ \ \omega_{k\ 1}\varepsilon^T_{x_k\ 1} \quad ^T + \quad E\ \ \omega_{k\ 1}\omega^T_{k\ 1} \quad ^T \\
= & \quad P_{k\ 1} \quad ^T + \quad Q_{k\ 1} \quad ^T
\end{aligned}
\tag{A.87}
$$

Next, the optimal Kalman gain is calculated. It is a function of the process and measurement covariance matrices, and therefore the optimality is conditioned upon the accuracy of the contents of these matrices.

$$K_k = P_k\ H_k^T\left(H_k P_k\ H_k^T + R_k\right)^{\ 1} \tag{A.88}$$

The measurement update is the final step of the iterative process. The state is updated by adding the previous estimate to the gain multiplied by the measurement residual.

$$\hat{x}_k^+ = \hat{x}_k + K_k\left(z_k \quad H_k\hat{x}_{\bar{k}}\right) \tag{A.89}$$

$$P_k^+ = \left(I \quad K_k H_k\right)P_k \tag{A.90}$$

The measurement residual, $z_k - H_k\hat{x}_{\bar{k}}$, is the error between the actual measurements and predicted measurements. Therefore, it should consist of zero mean, white noise vectors if the filter is functioning correctly. If any bias exists in the residual, the filter is improperly tuned or the model is inadequate. The residual can be monitored online to adapt the covariance matrices or reject potentially bad or corrupt measurements.

Note that the estimate of error covariance is independent of the state vector. This offers the designer the opportunity to predict the estimator performance through simulation. Various models, different tunings, or a mixture of sensors can be simulated to determine the best combination that achieves the desired estimate accuracy. For some systems, the gain and covariance can be computed offline to reduce the computational load at runtime.

The Kalman filter algorithm is summarized here.

$$\begin{gathered}
\text{Time Update}\\
\hat{x}_k = \quad {}_k\hat{x}_{k\;\;1} + \quad {}_k u_k\\
P_k = \quad {}_k P_{k\;\;1} \quad {}_k^{T} + \quad {}_k Q_k \quad {}_k^{T}\\
\text{Gain Calculation}\\
K_k = P_k H_k^T \left(H_k P_k H_k^T + R_k\right)^{\;\;1}\\
\text{Measurement Update}\\
\hat{x}_k = \hat{x}_k + K_k(z_k \quad H_k\hat{x}_k)\\
P_k = \left(P_k^{\;\;1} + H_k^T R_k^{\;\;1} H_k\right)^{\;\;1}
\end{gathered} \tag{A.91}$$

The dynamics model given by and is used to propagate the state from one instance in time to another. The model also provides a method of predicting the states' motion, so the Kalman filter can be used to estimate dynamic states. Also, notice the addition of the process noise covariance to the propagated state error covariance in the time update. This term prevents the gain from approaching a steady state value of zero, so the effectiveness of the measurement update is retained as time approaches infinity.

A.8 Extended Kalman Filter

All estimators to this point have been derived with the assumption of a linear system and measurements. In practice, most systems are inherently nonlinear, and therefore the estimators must be altered to accommodate the nonlinearities. The following discussion will involve the Kalman filter; application of linearization techniques to the other least squares estimators should be obvious afterwards. Specifically, the focus will be on the extended Kalman filter. Other methods are available for handling nonlinearities in systems and measurements such as the unscented filter and particle filter, but the extended Kalman filter provides acceptable performance with the least computational burden for ground vehicle dynamics with measurement rates from 1 Hz to 500 Hz.

The extended Kalman filter is capable of handling nonlinearities in the system. Linearization of the nonlinear dynamics and measurement models provides the necessary information to compute the Kalman gain and update the estimates. However, there is a sacrifice. The minimum variance guarantee does not exist for the extended Kalman filter as the minimization process can focus on a local, rather than global, minimum variance. This is more likely to occur when initial condition errors are large, which corrupts the information contained in the linearized state matrices.

The extended Kalman filter operates in principle like the linear Kalman filter. The differences are that the nonlinear expressions must be linearized each iteration with the current estimate and the time propagation and residual calculation are performed with the nonlinear equations. The components of an extended Kalman filter can be mixed with those in the linear Kalman filter depending on where the nonlinear relationships exist. The nonlinear expression must be integrated to propagate the states of a system with a nonlinear dynamics model and linear measurement model, but (A.10) can be used to compute the measurement prediction. Alternatively, (A.8) and (A.9) can be used to propagate the estimates of a system with a linear dynamics model, but (A.10) cannot be used to calculate the state measurement update.

The nonlinear state vector propagation requires numerical integration, so an appropriate routine, such as Euler, trapezoidal, or Runge-Kutta integration, must be selected by the system designer. The propagation is the evaluation of the differential equation describing the system's dynamics.

$$\hat{x}_k^- = \hat{x}_{k-1}^+ + \int_{t_{k-1}}^{t_k} f(\hat{x}(\tau), u(\tau), \tau)\, d\tau \qquad \text{(A.92)}$$

The covariance is propagated accordingly.

$$P_k^- = P_{k-1}^+ + \int_{t_{k-1}}^{t_k} F(\tau)P(\tau) + P(\tau)F^T(\tau) + L(\tau)Q(\tau)L^T(\tau)\, d\tau \qquad \text{(A.93)}$$

The matrix F is obtained by evaluating (A.8) with the most recent estimate, and Q is the continuous process noise covariance.

The gain computation is unchanged from the linear version of the Kalman filter. The measurement model is linearized with (A.10) at the most recent estimate, if necessary.

$$K_k = P_k\ H_k^T\,(H_k P_k\ H_k^T + R_k)^{\ 1} \tag{A.94}$$

The state update utilizes the nonlinear measurement model to form the predicted measurement vector (i.e., $\hat{z}_k = h(\hat{x}_k, t_k)$).

$$\hat{x}_k^+ = \hat{x}_k + K_k(z_k \quad h(x_k, t_k)) \tag{A.95}$$

The covariance update equation remains unaltered, but uses the linearized measurement matrix.

$$P_k^+ = (I \quad K_k H_k) P_k \tag{A.96}$$

The extended Kalman filter algorithm is listed here.

Linearize System

$$F(t) = \left.\frac{f(x(t),u(t),t)}{x(t)}\right|_{\hat{x}(t),u(t)}$$

$$G(t) = \left.\frac{f(x(t),u(t),t)}{u(t)}\right|_{\hat{x}(t),u(t)}$$

$$H(t) = \left.\frac{f(x(t),t)}{x(t)}\right|_{\hat{x}(t)}$$

Discretize System (A.97)

$$F(t) \qquad {}_k \quad G(t) \qquad {}_k \quad L(t) \qquad {}_k$$

$$Q(t) \quad Q_k \quad H(t) \quad H_k$$

Time Update

$$\hat{x}_k = \hat{x}_{k\ 1}^+ + \ {}_{t_{k\ 1}}^{t_k} f(\hat{x}(\tau) + u(\tau), \tau)\, d\tau$$

$$P_k = P_{k\ 1}^+ + \ {}_{t_{k\ 1}}^{t_k} \begin{array}{l} F(\tau)P(\tau) + P(\tau)F^T(\tau) \\ +L(\tau)Q(\tau)L^T(\tau)\, d\tau \end{array}$$

Gain Calculation

$$K_k = P_k H_k^T\,(H_k P_k H_k^T + R_k)^{\ 1}$$

Measurement Update

$$\hat{x}_k^+ = \hat{x}_k + K_k(z_k \quad h(x_k, t_k))$$

$$P_k^+ = (I \quad K_k H_k) P_k$$

A.9 Initialization

Initialization of a Kalman filter is a matter of providing an educated guess of the initial conditions with some associated uncertainty.

$$\hat{x}_0 = \begin{matrix} \hat{x}_{1_0} \\ \hat{x}_{2_0} \\ \vdots \\ \hat{x}_{n_0} \end{matrix} \quad P_0^+ = \begin{matrix} \sigma_{x_1}^2 & \cdots & \rho\sigma_{x_1}^2\sigma_{x_n}^2 \\ \vdots & \ddots & \vdots \\ \rho\sigma_{x_n}^2\sigma_{x_1}^2 & \cdots & \sigma_{x_n}^2 \end{matrix} \tag{A.98}$$

Inputs might range from being mostly arbitrary to very precise. The accuracy of the initial conditions can affect filter performance, especially when using an extended Kalman filter. In some applications, measurements are available before the Kalman filter is started. Measurements of the appropriate states can be used to initialize the state vector. This is often useful as the operator usually has good knowledge about the accuracy of the measurements, and therefore this accuracy can be used to initialize the covariance matrix.

References

[1] Gelb, A., *Applied Optimal Estimation*, Cambridge, MA: MIT Press, 1974.

[2] Stengel, R., *Optimal Control and Estimation*, Mineola, NY: Dover Publications, 1994.

[3] Crassidis, J. L., and J. L. Junkins, *Optimal Estimation of Dynamic Systems*, Boca Raton, FL: CRC Press, 2000.

[4] Bryson, A. E., and Y. Ho, *Applied Optimal Control*, New York: Hemisphere Publishing Corporation, 1970.

[5] Maybeck, P. S., *Stochastic Models, Estimation, and Control*, Springfield, VA: Navtech GPS, 1979.

[6] van Loan, C., "Computing Integrals Involving the Matrix Exponential," *IEEE Transactions on Automatic Control*, Vol. AC-14, 1978, pp. 396–404.

About the Authors

David M. Bevly is an associate professor in the Department of Mechanical Engineering at Auburn University. He directs the university's GPS and Vehicle Dynamics Laboratory (GAVLAB), which focuses on the control and navigation of vehicles using GPS in conjunction with other sensors, such as Inertial Navigation System (INS) sensors. The GAVLAB's research thrusts are focused towards vehicle dynamics and transportation systems, including heavy trucks, passenger cars, off-road vehicles, and autonomous and unmanned vehicles. Dr. Bevly received a B.S. from Texas A&M University in 1995, an M.S. from the Massachusetts Institute of Technology in 1997, and a Ph.D. from Stanford University in 2001 in mechanical engineering.

Stewart Cobb is a founder of Novariant Corporation, where he designs GNSS receivers and pseudolites for precise control of air and ground vehicles. He earned an S.B., an M.S., and a Ph.D. in aeronautics and astronautics from the Massachusetts Institute of Technology and Stanford University and an M.S. in systems management from the University of Southern California. His doctoral research showed how pseudolites could help conventional GPS to reach the navigation performance levels necessary to land airliners on autopilot.

Robert Daily is a systems analyst for Cobham Analytic Solutions. He received a B.S, an M.S., and a Ph.D. in mechanical engineering from Auburn University. His doctoral research investigated path planning and control of ground vehicles using artificial potential fields modeled on fluid flow. Past projects include designing and implementing the vehicle controller for the SciAutonics-Auburn Engineering Grand Challenge entry vehicles.

David Hodo earned a B.E.E. and an M.S. in electrical engineering from Auburn University in 2005 and 2007, respectively. He is currently pursuing a Ph.D. in the same field at Auburn. His masters' research focused on the navigation and control of an unmanned ground vehicle towing a trailer. His current research focuses on navigation in environments with limited or no GPS.

Lowell S. Brown earned a B.S. in mechanical engineering from Auburn University in 2008. He is currently pursuing an M.S. in mechanical engineering while working as a research assistant in the GPS and Vehicle Dynamics Laboratory at Auburn University. His research interests include vehicle roll and rollover, parameter estimation, and the online application of rollover prediction and parameter estimation.

William Travis earned a B.S, an M.S., and a Ph.D. in mechanical engineering from Auburn University. As a graduate student, he worked in the GPS and Vehicle Dynamics Lab (GAVLAB) under the advisement of Dr. David Bevly. He focused on vehicle dynamics, vehicle parameter estimation, GPS, navigation systems, and unmanned systems. His doctoral work focused on developing GPS carrier based relative positioning methods for path duplication in automated ground vehicle convoys.

Dustin Edwards earned a B.S. and an M.S. in mechanical engineering from Auburn University. While at Auburn, his research consisted of modeling and parameter identification of ground vehicles. Currently, he works in the aerospace field developing models and simulations of missile systems.

Benjamin Clark is a graduate research assistant in Auburn University's GPS and Vehicle Dynamics Laboratory. At Auburn, he has studied navigation since 2004. His research background includes guidance for mobile robots, GPS error source detection and exclusion, and GPS/INS integration techniques. He has worked with various types of land vehicles ranging from custom robots to military convoys with an emphasis on positioning improvement. In 2004 he received a B.S. in mathematics from Freed-Hardeman University. In 2006 and 2008 he earned a B.S. and an M.S. in mechanical engineering from Auburn University.

John Allen received a B.S. in mechanical engineering from Auburn University. Currently, he is working on an M.S. in mechanical engineering at Auburn. His research interests include sensor fusion for land vehicle navigation and the use of local based measurements to aid GPS in environments with limited satellite visibility.

Index