

DESIGN AND PERFORMANCE OF A ROBUST
GPS/INS ATTITUDE SYSTEM FOR AUTOMOBILE
APPLICATIONS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF AERONAUTICS AND ASTRONAUTICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Santiago Alban
June 2004

© Copyright by Santiago Alban 2004
All Rights Reserved

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Dr. Stephen M. Rock
(Principal Adviser)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Dr. J. David Powell

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Dr. Per Enge

Approved for the University Committee on Graduate Studies.

Abstract

The increasing demand for navigation aids and sophisticated control systems in modern cars has led to a growing number of GPS applications in the automobile industry. In addition to position and velocity, precise attitude measurements are needed for many emerging control applications in cars, such as lane-keeping and collision avoidance systems. GPS can also be used to measure the attitude of a vehicle, and this approach offers a significant advantage over other methods in the tradeoff of accuracy vs. cost. However, few commercial GPS products exist with the capability of measuring attitude, and those that are available perform poorly in most urban environments and are prohibitively expensive for use in consumer vehicles.

This thesis presents the system design and performance analysis of an inexpensive and robust attitude system based on GPS and inertial sensors, suitable for use in automobiles. This system utilizes a variety of techniques and unique algorithms to improve GPS attitude robustness and availability under the high phase-noise and limited visibility conditions encountered during normal urban driving. Such methods include tight coupling with the inertial subsystem, and the ability to measure attitude with as few as one satellite. A prototype of the system was realized with inexpensive GPS and automotive grade inertial components, and demonstrated comparable accuracy and superior robustness to commercial systems that depend on multiple-satellite attitude solutions in urban driving environments.

A section of this thesis is also dedicated to GPS carrier-tracking loop design with Doppler aiding, in which the phase-lock loops for each channel are aided with information from an inertial navigation system. The resulting system benefits from higher bandwidth dynamic tracking of GPS signals, improved resistance to phase-noise, and enhanced robustness against cycle slips. These features vastly improve the quality and reliability of GPS carrier-phase measurements for attitude determination and other applications. Detailed analysis and simulation results are presented to illustrate the benefits of Doppler aiding at the signal tracking level and in the higher-level application of attitude determination. An experiment was also performed to collect real inertial navigation data from a moving car, and synchronized GPS signal samples from a commercial software receiver. Doppler aiding was implemented with this data, and the results from this experiment validate those from simulation. This experiment also demonstrated the viability of implementing Doppler aiding with automotive-grade inertial components.

Acknowledgements

The work presented in this thesis could not have been possible without the technical advice, funding, and support from many people.

Special appreciation is given to my advisor, Professor Steve Rock, for his continued support and great counseling throughout my graduate studies. In particular, I acknowledge the freedom with which I was allowed to develop this project in my own vision.

I also thank Professors Per Enge and J. David Powell for their great teaching of GPS, and for invaluable technical advice and constructive feedback. Their interest in my research has been a great source of confidence and motivation.

I am also very grateful to Professor Chris Gerdes for being the chair of my orals committee, and whose innovative research on vehicle control with GPS has given a greater purpose for my own work.

Special thanks go to Professor Claire Tomlin for her great teaching and for taking the time to be part of my orals committee.

The sponsors of my research deserve many thanks, as they have provided the financial support for this project. Primarily, I gratefully recognize the support of Bosch RTC, especially from Markus Lutz, Arne Dietrich, and their staff. From Daimler Chrysler, thanks go to Chris Wilson for initiating this project with our Department. I also extend my gratitude to The National Science Foundation for their generous three-year fellowship and to the Department of Aeronautics and Astronautics for their additional financial assistance.

The students and staff from the WAAS and LAAS laboratories, have been outstanding mentors and colleagues for me, and I have gained immeasurable benefit from their teachings and friendship. I especially acknowledge the help of Dennis Akos for his tireless efforts in helping to collect GPS samples, and of Demoz Gebre-Egziabher and Jenny Gautier, who have demonstrated continued interest in my work. Finally, appreciation also goes to Roger Hayward, who generously shared much of his work on inexpensive GPS attitude systems when I began this project.

Other parties that deserve recognition for their applications of my work include Seagull Technologies and the STAR laboratory in the EE Department.

The competence and amiability of the Aeronautics and Astronautics faculty and staff has always made this Department a great place to work. I especially recognize the invaluable efforts of Sally

Gressens, Sherann Ellsworth, Dana Parga, and Aldo Rossi for helping to make our Department what it is.

My few but distinguished friends outside of the department have also been contributors to my great experience at Stanford, especially Pradeep, Andy, Jon and Sharon.

Finally, I'd like to express my gratitude to my Mom and my brother for their loving support. They have always given me more acclaim than I deserve, and for that I dedicate this work to them.

Contents

Abstract	iv
Acknowledgements	v
1 Introduction	1
1.1 Overview of GPS and Carrier Phase Applications	1
1.2 GPS/INS Integration Techniques	4
1.3 GPS Applications in Automobiles	7
1.4 Prior Research	9
1.5 Thesis Contributions	10
1.6 Thesis Organization and Solution Overview	11
2 Attitude Determination with GPS	13
2.1 Differential Carrier-Phase Measurements	13
2.2 Linear Least-Squares Attitude Solution	17
2.2.1 Formulation	17
2.2.2 Line-Bias Estimation	20
2.2.3 Integer Ambiguity Determination	24
2.2.4 Linear Attitude Solution	27
2.3 Nonlinear Attitude Solution	30
2.4 ADOP	31
2.5 Phase-Delay Calibration	34
2.6 One-Satellite Attitude Solution	38
2.6.1 Formulation and Accuracy of One-Satellite Yaw	38
2.6.2 Formulation and Accuracy of One-Satellite Roll	44
2.6.3 Integer Determination for One-Satellite Attitude	46
2.6.4 One-Satellite Measurement Averaging	47
2.7 Baseline Configuration Design	49
2.7.1 Attitude Accuracy Performance Envelope	50
2.7.2 Attitude Robustness Performance Envelope	52

3	System Components of GPS/INS Attitude System Prototype	55
3.1	Baseline Configuration	55
3.2	GPS Receivers and Antennas	57
3.3	Inertial Sensors	59
3.4	Computer Hardware	61
3.4.1	Analog To Digital Converter	61
3.4.2	Serial-Port Expansion Card	61
3.4.3	Computer	62
3.5	Interface Software	63
3.5.1	Software Hierarchy	63
3.5.2	GPS Interface Software	65
3.5.3	IMU Interface Software	67
3.5.4	Graphical User Interface	68
3.6	Time Synchronization	69
4	System Design of a GPS/INS Automobile Attitude System with Tight Integration	73
4.1	Subsystem Integration	74
4.1.1	GPS-Attitude Module Selection	74
4.1.2	GPS/INS Integration Filters	75
4.2	Integrity Monitoring	89
4.2.1	dICP Fault Detection and Correction Measures	89
4.2.2	Baseline-Level Integrity Monitoring	91
4.2.3	Multiple-Baseline Integrity Monitoring and Recovery	96
4.2.4	Integrity Monitoring in the Primary GPS/INS Filter	98
5	GPS/INS Attitude System Performance	100
5.1	Accuracy	100
5.2	Robustness	107
5.3	Limitations of Tight Coupling	110
6	Inertial Aiding of GPS Tracking Loops	111
6.1	GPS Tracking Loops	112
6.1.1	The Phase Lock Loop	113
6.1.2	The Delay-Lock Loop	117
6.2	Doppler and Clock Error-Frequency Estimation	120
6.2.1	Doppler-Frequency Estimation	120
6.2.2	Clock Error-Frequency Estimation	121

6.3	Development and Design of Doppler-Aided Phase-Tracking Loops	124
6.3.1	Doppler-Aided PLL Model	124
6.3.2	Implementation Details	127
6.3.3	Frequency Error Models and Loop-Bandwidth Design	130
6.3.4	Oscillator Phase-Noise PSD Models	133
6.3.5	Doppler-Estimate Error Models	136
6.3.6	Satellite Clock Phase-Noise	144
6.3.7	Closed-Loop Bandwidth Selection	149
7	Performance of Doppler-Aided Phase-Lock Loops	152
7.1	Cycle-Slip Analysis	152
7.1.1	Closed-Loop Cycle-Slip Performance	153
7.1.2	Open-Loop Cycle-Slip Performance	156
7.2	Performance Results	158
7.2.1	Results from Simulation	159
7.2.2	Results with Real Data	164
8	Conclusions	169
8.1	Summary	169
8.2	Future Research	171
A	Coordinate Frames and Vector Notation	174
A.1	Reference Coordinate Frames	174
A.2	Vector Notation	177
B	Gauss-Markov Processes and Random Walks	179
B.1	Gauss-Markov Processes	179
B.2	Random Walks	180
C	Inertial-Sensor Error Modeling	182
D	Development of a PLL Estimator for Ultra-Tight Coupling	186
E	Accronyms and Symbols Guide	192
E.1	List of Acronyms	192
E.2	List of Symbols	193
	Bibliography	199

List of Tables

3.1	Characteristics of Different Types of Inertial Sensors	59
3.2	Bosch DRS-MM1 Specifications	60
4.1	Primary GPS/INS Filter Gains	83
5.1	Statistics of GPS Attitude Availability and Outages in Road Tests	110
6.1	Clock Phase-Noise PSD Model Coefficients	135
6.2	Minimum Loop Bandwidths (Hz) for Doppler-Aided PLL with 0.1rad Phase Jitter	151

List of Figures

1.1	Time Domain Representation of GPS Signal Components	2
1.2	GPS/INS System With Loose Integration	5
1.3	GPS/INS System With Tight Integration	6
1.4	GPS/INS System With Ultra-Tight Integration	7
1.5	Number of SVs Usable for GPS Attitude in an Urban Environment	9
1.6	Architecture of Automobile GPS/INS Attitude System with Doppler Aiding	12
2.1	Integrated Carrier-Phase Measurements	14
2.2	Vector Diagram for Attitude Determination	18
2.3	Definition of Baseline Orientation Angles	21
2.4	Attitude Search Based on Channel Line-Bias Variance	22
2.5	An Image of Line-Bias Estimation	23
2.6	Image of an Integer-Search Process	26
2.7	Derivation of Roll and Pitch Axes from Arbitrary Baselines	29
2.8	One-Dimensional Attitude for Visualization of ADOP	32
2.9	Phase-Delay Repeatability in One Channel	35
2.10	Example of a Phase-Delay Map	36
2.11	Attitude Measurements With and Without Phase-Delay Calibration	37
2.12	Vector Diagram for One-Sat Yaw	39
2.13	One-Satellite Yaw Accuracy and ADOP	41
2.14	One-Sat Yaw Sensitivity to Nonzero Baseline-Roll Angle	43
2.15	One-Satellite Roll Accuracy and ADOP	45
2.16	One-Satellite Roll Sensitivity to Errors in Yaw Estimate	46
2.17	Comparison of Mult.-Satellite and Avg. One-Satellite GPS Attitude Measure- ments	49
2.18	Attitude Accuracy Performance Envelope	52
2.19	Attitude Robustness Performance Envelope	54
3.1	Automobile Baseline Configurations With Three Antennas	56
3.2	Baseline Assembly on Test Vehicle	57
3.3	GPS Receiver Assembly for Attitude Determination	58
3.4	Inertial Sensors and IMU Assembly	61

3.5	Attitude System Prototype Equipment	62
3.6	Prototype Software Architecture	64
3.7	Representation of GPS Interface Data Structure	66
3.8	Representation of IMU Interface Data Structure	68
3.9	Screen Capture of Attitude System GUI	69
3.10	Time-Synchronization Diagrams	71
4.1	GPS/INS Attitude-System Prototype Subsystems	74
4.2	GPS Attitude Module Selection	75
4.3	Primary GPS/INS Filter Yaw Estimates	84
4.4	Primary GPS/INS Filter Yaw-Gyro Bias Estimates	84
4.5	Performance of dICP Filters	88
4.6	dICP Filter Updates and Integrity Monitoring	90
4.7	Noisy Channel Removal by Residual Inspection	92
4.8	Baseline-Level Integrity Monitoring for Multiple-Satellite Solution	93
4.9	Baseline-Level Integrity Monitoring for One-Satellite Solution	95
4.10	Integrity Monitoring with Multiple Baselines	97
4.11	Primary GPS/INS Filter Updates and Integrity Monitoring	99
5.1	Filter Models for Error Analysis	102
5.2	Innovation Measurements in Primary GPS/INS Filter	103
5.3	PSD Measurements and Models for Yaw Innovations	104
5.4	Attitude Measurements During a Road Test	105
5.5	Gyro-Bias Estimates During a Road Test	106
5.6	Performance of a Loosely-Coupled GPS/INS Attitude System using Novatel Beeline	108
5.7	Robustness Performance During a Road Test	109
6.1	Schematic of a GPS Receiver	113
6.2	Model of a Second Order PLL	115
6.3	Implementation of a PLL	116
6.4	Model of a First Order DLL	118
6.5	Model of a Rate-Aided DLL	119
6.6	Implementation of a Rate-Aided DLL	119
6.7	Measurements of Clock Error-Frequency with a TCXO and an OCXO	123
6.8	Model of a Frequency-Aided PLL	125
6.9	PLL Transfer Functions	126
6.10	Implementation of a Frequency-Aided PLL	127

6.11	GPS Signal Integrity Monitoring in Frequency-Aided PLL	128
6.12	GPS Signal Integrity Monitoring in Frequency-Aided PLL	129
6.13	Model Used for Error-Tracking Analysis of Frequency-Aided PLL	131
6.14	Phase-Noise PSDs of a TCXO and an OCXO	134
6.15	Measured and Modeled Vibration PSD of a Test Automobile in Motion	136
6.16	Time and Frequency-Domain Comparisons of Doppler-Estimate Error Model with Approximate Experimental Errors from Navigation Filter	142
6.17	PSD Models of Doppler-Estimate Errors with and without GPS Aiding of Navi- gation Filter with Automotive-Grade Sensors	144
6.18	Clock Error-Frequency and Frequency Residual Measurements	146
6.19	Cross-Correlation Results of PLL Frequency Residuals	148
6.20	Phase-Noise PSDs of Satellite-Specific Noise	149
6.21	Contributions to PLL Output Phase-Jitter	150
6.22	Phase Jitter vs PLL Noise-Bandwidth	151
7.1	Closed-Loop Cycle-Slip Performance of Traditional and Doppler-Aided PLLs . .	155
7.2	Model for Open-Loop Phase-Tracking Performance	157
7.3	Open-Loop Cycle-Slip Performance of a Doppler-Aided PLL	157
7.4	Simulated Trajectory with Dual Antennas	160
7.5	Software-Receiver Outputs from Simulation Test Case	161
7.6	Channel 1 Frequencies for Simulation Test-Case	162
7.7	Channel 1 Phase-Jitter for Simulation Test-Case	163
7.8	dICP Measurements and Cycle-Slip Aversion for Simulation Test Case, with and without GPS Navigation during Channel Outage	163
7.9	Early, Late, and Prompt Power from DLL Outputs for Simulation Test Case . . .	164
7.10	Real-Data Test Setup, Location, and Vehicle	165
7.11	Position and Velocity Measurements from GPS and Navigation Filter for Real- Data Test Case	166
7.12	Channel 4 Frequencies for Real-Data Test Case	167
7.13	Channel 4 Phase-Jitter for Rea-Data Test Case	168
7.14	Early, Late, and Prompt Power from DLL Outputs for Real-Data Test Case	168
A.1	Illustration of Various Coordinate Frames	176
B.1	Images of White Gaussian Driving Noise and Resulting Gauss-Markov Process and Random Walk	181
C.1	Illustration of Gyro Misalignments Relative to Platform	185

D.1	Performance of a PLL Implemented with an Estimator	189
D.2	Schematic of a Single-Channel PLL Estimator	191

Chapter 1

Introduction

Attitude sensors are an essential component of vehicle control systems and inertial navigation systems. Some traditional methods for measuring the attitude of a vehicle include magnetometers and compasses, tilt sensors, and high-grade gyroscopes. In the last decade, the use of the Global Positioning System (GPS) for attitude determination has become well known, as it offers advantages over other methods in the tradeoff of accuracy vs. cost. However, few GPS commercial products exist that provide attitude measurements, and none are available that meet the low cost and robustness requirements for the growing number of applications in the automobile industry. This thesis covers the design and performance analysis of a low cost and robust attitude system based on GPS and inertial sensors, specifically designed for the typical operating conditions of a car.

This introductory chapter reviews some of important background information pertaining to this thesis, including brief overviews of GPS and carrier-phase applications, techniques for integration of GPS with inertial sensors, and the use of GPS in automobiles. A discussion covering some relevant prior research is included, and a summary of thesis contributions delineates the main points of the original research presented in this document. The chapter concludes with a preview of the more detailed discussions that follow in the remaining chapters.

1.1 Overview of GPS and Carrier Phase Applications

GPS has become one of the most popular methods of navigation for users worldwide. The high accuracy of GPS and low cost of modern receivers have precipitated widespread use in a variety of terrestrial, marine, airborne, and space navigation applications. Primarily designed for military purposes by the U.S. Department of Defense, GPS has also been adopted eagerly by the civilian community through the use of its civil signal.

GPS consists of a network of over 24 satellites (including spares) designed to provide precise measurements of position, velocity and time. The GPS satellites, often referred to as SVs (for Space Vehicles), transmit on two frequencies in the L-band, designated as Link 1 (L1) at 1575.42

MHz, and Link 2 (L2) at 1227.60 MHz. The L1 band contains a civil signal and an encrypted military signal; the L2 band contains another encrypted signal for military use. For the purposes of this thesis, only the civil signal on L1 is of interest. The civil signal from a given satellite is composed of three parts:

Carrier: A sinusoidal carrier signal at the L1 frequency.

Code: The coarse/acquisition (C/A) code, unique to each SV. This code repeats every 1ms and consists of 1023 bits (called chips); the C/A code is used to differentiate signals coming from different SVs, and for ranging.

Navigation Data: Transmitted at 50 bits per second (bps), this data contains information about the satellite orbit and clock, and the time of transmission.

The product of the code and navigation bits is modulated onto the carrier by binary phase shift keying (BPSK), where a 180° shift in carrier phase represents a change in bit sign. Figure 1.1 illustrates the GPS civil signal components.

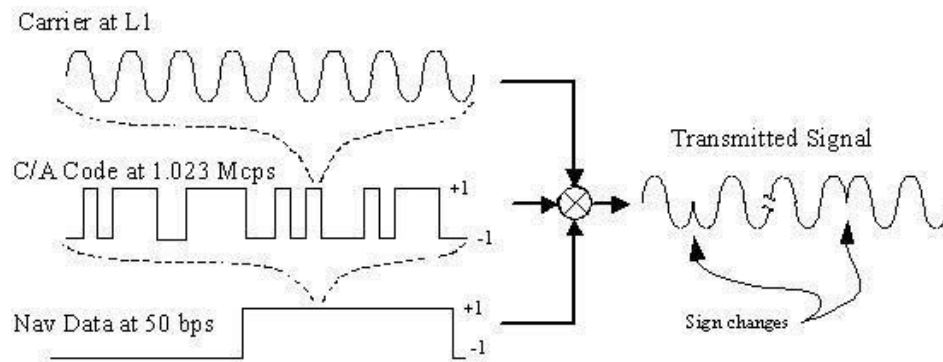


Figure 1.1: Time Domain Representation of GPS Signal Components

A GPS L1 receiver picks up the superposition of signals coming from all satellites in view. The resultant signal at the receiver antenna has a spectrum bandwidth of about 2.1MHz, centered at the L1 frequency. Through a process called frequency down-conversion, the RF (Radio Frequency) front end on the receiver translates the signal spectrum to a lower center frequency where it can be sampled. Typical receivers sample the down-converted signal with a 2-8 bit analog-to-digital converter (ADC), and with sampling frequencies of 4-20 MHz. Sometimes the sampling process uses deliberate aliasing to further translate the spectrum to its final intermediate frequency (IF), which will be the center frequency of the discrete signal spectrum.

Once the GPS signal is in discrete form, the GPS receiver performs acquisition and tracking of individual satellites through a bank of multiple channels (usually 6-12). Each channel is dedicated to one satellite, and contains tracking loops for the code and the carrier signal components. The carrier-tracking loop is typically a phase lock loop (PLL) that tracks the carrier frequency

and phase changes due to Doppler shift and local oscillator dynamics, and allows decoding of the navigation data. A delay-lock loop (DLL) is used to track the code needed to extract a given satellite signal from the composite GPS signal, and provides an accurate code phase for ranging. A raw range estimate from the receiver to a satellite is partially constructed from this code-phase, and is called a pseudorange (PR). The pseudoranges will contain an error due to the local clock bias, common to all the channels, which is resolved along with user position.

Typical GPS positioning is done by quadrilateration with four or more pseudorange measurements, and with no external corrections, is known as stand-alone GPS. Stand-alone GPS has a typical horizontal positioning accuracy of about 10m, and a vertical accuracy of about 15m. While adequate for many navigation purposes, this accuracy is not sufficient for applications such as aircraft precision approaches, which may require meter-level positioning accuracy or better, depending on visibility. This need gave rise to the differential GPS (DGPS) augmentation systems, such as Maritime GPS used by the U.S. Coast Guard, the Wide Area Augmentation System (WAAS) available to everyone over most of the conterminous U.S., and Local Area Augmentation System (LAAS) planned for aircraft navigation near airports. These systems broadcast pseudorange corrections to users sufficiently near a reference station, and correct errors due to the ionosphere and troposphere delay, satellite clock, and satellite ephemeris inaccuracies. With these corrections, the users of DGPS achieve positioning accuracies on the order of 0.5-3 meters, depending on their distance from a reference station.

Even better accuracy can be achieved by making use of the precise tracking of carrier phase in a GPS receiver, which allows for centimeter level positioning relative to a reference station within a few miles. This technique is known as carrier-phase positioning, and has been studied extensively for its unparalleled accuracy. Carrier-phase positioning is achieved by using the carrier phase instead of the code phase to formulate a range measurement. In this case, the range measurements refer to the inner product of the line of sight (LOS) unit vector to a satellite and a vector (called a baseline) between a reference antenna and a mobile antenna. Such a measurement can be obtained by taking the difference between integrated carrier phase measurements taken from the two antennas, and correcting for the whole number of cycles (integer ambiguities) that make up the range measurement. One of the biggest difficulties with carrier phase positioning involves determination of these integer ambiguities. This process can be numerically expensive and time-consuming, and susceptible to error. In terms of system robustness, integer ambiguity resolution is often the weakest link when repeated cycle slips occur and large phase noise is present. For this reason, much of the research on carrier phase positioning has focused on faster and more reliable methods of determining integer ambiguities [20, 48, 13].

A close relative of carrier-phase positioning is differential carrier-phase attitude determination. With multiple fixed GPS antennas on a vehicle, differential carrier-phase can also be used to

measure attitude. In contrast to carrier phase positioning, the magnitude of the baseline is fixed, and only its orientation defines the quantities of interest. Since the antennas are fixed relative to each other, another important simplification is the ability to use a common clock among multiple receivers. This advantage allows for the use of single differencing of carrier phase measurements, instead of the double differencing technique needed to remove the clock error in carrier phase positioning. Integer ambiguity searches are also part of the attitude determination process, but the use of shorter and fixed-length baselines simplifies the integer search process and improves its reliability.

1.2 GPS/INS Integration Techniques

The integration of GPS and inertial navigation systems (INS) systems is a well-studied field. The motivation for this type of sensor fusion is usually to overcome some of the shortcomings of GPS in terms of bandwidth, noise performance, and robustness.

GPS and INS have complementary qualities that make them ideal candidates for sensor fusion. The limitations of GPS include occasional high noise content, outages when satellite signals are blocked, vulnerability to interference, and low bandwidth. The strengths of GPS include its long-term stability and its capacity to function as a stand-alone navigation system. In contrast, inertial navigation systems are not subject to interference or outages, have high bandwidth and good short-term noise characteristics, but have long-term drift errors and require external information for initialization. A combined system of GPS and INS subsystems can exhibit the robustness, higher bandwidth and better noise characteristics of the inertial system with the long-term stability of GPS.

The level and complexity of GPS and INS coupling is dictated by several factors, including desired navigation accuracy, quality of the inertial measurement unit (IMU), and required robustness of the GPS receiver outputs. The levels of integration are usually classified as loose integration, tight integration, and ultra-tight or deep integration.

Loose integration is the simplest method of coupling, and is depicted in Figure 1.2. In this scheme, GPS and the inertial sensors generate navigation solutions independently (position, velocity and attitude (PVA)). The two independent navigation solutions are subsequently combined to form a blended (or filtered) GPS-inertial navigation solution. One of the benefits of loose integration is that the blended navigation solution typically has a higher bandwidth and better noise characteristics than the GPS solution alone.

Loose integration is best implemented with higher quality inertial sensors (navigation or tactical grade) if the GPS outages are long in duration. Lower quality inertial sensors can also

provide some immunity against momentary GPS outages, especially if their various biases were calibrated using GPS prior to the outage. In this case, the GPS-inertial loose integration is said to include feedback, whereby the difference between the GPS and inertial solutions is fed back to the inertial sensors to carry out the calibration. In general, lower quality inertial sensors (consumer or automotive grade) are suited for applications where GPS outages are infrequent and short in duration.

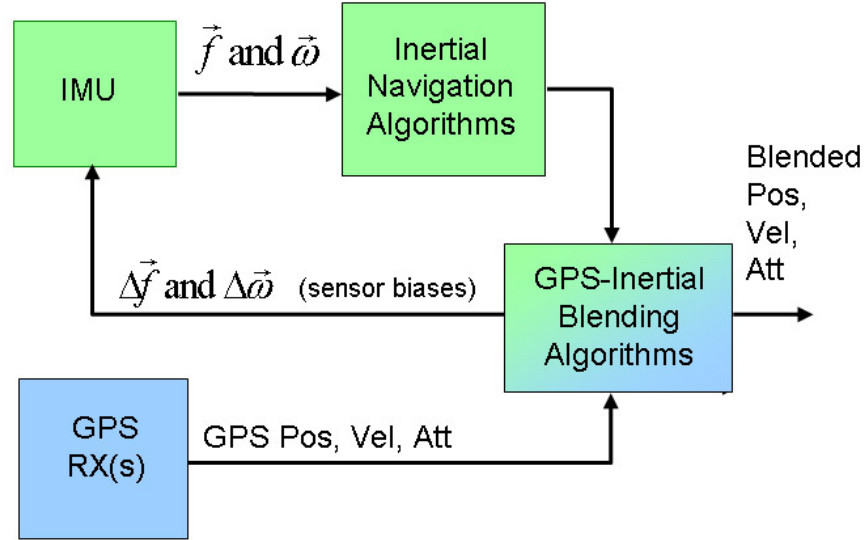


Figure 1.2: GPS/INS System With Loose Integration

A more complex level of coupling is tight integration, where GPS pseudoranges, Doppler, or carrier phase (CP) measurements are blended with the navigation solution generated by the inertial sensors. Figure 1.3 shows the general structure of a system with tight integration.

In addition to the benefits of loose coupling, a tightly integrated system can have a more accurate navigation solution. This benefit is obtained because the effect of some deterministic errors in channel-specific GPS observables (i.e. PR, Doppler, and phase measurements), such as signal-to-noise ratio, can be accounted for in the GPS/INS blending filter. Furthermore, the use individual channel GPS observables for INS calibration will tend to diffuse the effect of unpredictable GPS-measurement time-correlation on estimates of inertial sensor biases, as long as channels tend to be uncorrelated with each other. In contrast, the correlation mechanism of GPS PVA solutions used in loose coupling is generally unknown, and cannot be mitigated or accounted for in a navigation Kalman filter [22].

Tight integration also provides a means for implementing a more sensitive fault detection and isolation scheme that can be used to verify the quality of the GPS observables at the channel level [7, 16]. Applications which use carrier-phase output (attitude determination and CP positioning)

especially benefit from tight integration because integer ambiguities can be recovered and verified quickly from the navigation outputs, despite cycle slips and increased carrier-phase noise [2]. In attitude determination systems, this feature can be implemented with automotive-grade inertial sensors, whereas CP positioning applications would probably need the higher performance of tactical or navigation-grade sensors.

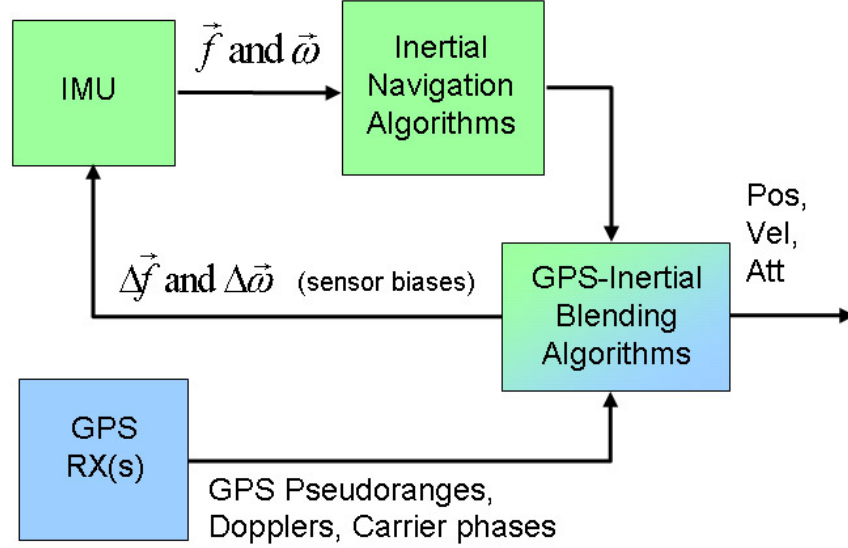


Figure 1.3: GPS/INS System With Tight Integration

The most complex and potentially the most beneficial level of GPS-inertial integration occurs at the GPS tracking-loop level, as shown in Figure 1.4. This level of coupling is called ultra-tight integration. This configuration is more complex than the other architectures discussed because it changes the structure of the traditional GPS tracking loops. In terms of performance, ultra-tight integration also offers the most benefits in terms of accuracy and robustness improvements to the GPS receiver and overall system.

Ultra-tight integration can improve acquisition time [49] as well as the tracking performance of the phase-lock loop in terms of bandwidth and noise rejection, thus producing more accurate Doppler and phase measurements. The use of inertial sensors in ultra-tight integration allows reduction of the carrier-phase tracking-loop bandwidth by eliminating the need to track the vehicle platform dynamics. This integration scheme results in cleaner carrier-phase measurements and faster tracking of the carrier phase. Furthermore, estimation of the drift-rate of the GPS receiver clock permits uninterrupted tracking of a channel despite a brief line-of-sight blockage. This capability can prevent carrier-phase cycle slips, and therefore has potential applications in high performance navigation systems where robustness to cycle slips is of paramount importance [45, 54].

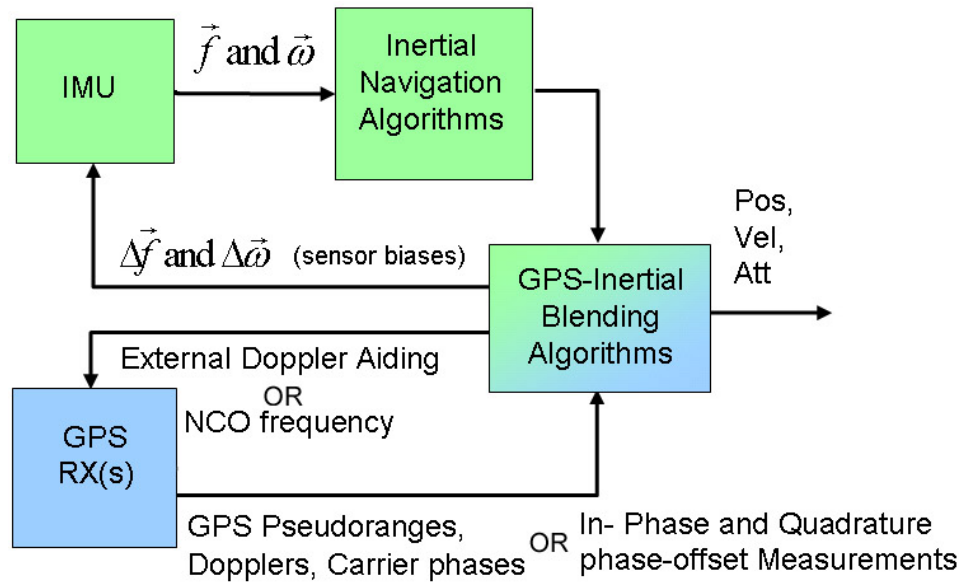


Figure 1.4: GPS/INS System With Ultra-Tight Integration

It should be noted that the term “tight integration” has been used in a broader context to encompass both the tight and ultra-tight architectures introduced here, as done in [56] and [61]. To clarify the terminology used in this thesis, ultra-tight coupling/integration refers to any coupled GPS/INS system where the GPS tracking loops are aided with data derived from inertial sensors. This nomenclature applies whether the tracking loops are implemented as part of a larger navigation filter, or with a federated architecture where the tracking loops are separate from the navigation filter.

1.3 GPS Applications in Automobiles

The advent of GPS receivers on cars was spurred by demand for accurate and inexpensive navigation aids that could be used to track vehicles and provide real-time driving directions. Vehicle tracking with GPS units is now vital to the operation of many businesses, as knowledge of the positions of a large deployment of vehicles is used for fleet management, and can greatly improve efficiency. In the consumer market, GPS navigation systems have become popular products (Garmin StreetPilot, Magellan 750M, etc). Custom navigation systems are also available in some rental cars (Hertz Neverlost Navigation System) and new car models, where seamless installation of the GPS system (including user interface and antenna) has been achieved.

Modern cars are also equipped with a variety of control systems that improve passenger safety and comfort. Braking control, stability control, and lateral control systems are an integral part of high-performance vehicles. Anti-lock braking and cruise control are examples of some common

existing control systems. More advanced systems for lane-keeping control and collision avoidance are being studied, and may appear in future consumer models [23]. Many of these systems utilize vehicle position and/or velocity measurements; stability control and lateral control in particular, also require measurements of the vehicles attitude, especially yaw and roll [58]. GPS is a suitable sensor for such control applications [1, 50], as it can provide all of those measurements with a pair of antennas. However, from a bandwidth standpoint, current GPS receivers do not have sufficiently fast update rates for vehicle control, and some form of integration with inertial sensors is often imperative to achieve a higher system bandwidth [23]. Coupling with an inertial system is also necessary for reliability and robustness of any safety-critical system on a car, as multipath and lack of visibility of satellites can hinder the performance of GPS in urban areas.

Attitude can be provided in the short term by integrating gyro outputs, but automotive-grade inertial sensors require initialization and frequent calibration updates from a GPS attitude system, or equivalent reference. Despite the significant accuracy and cost advantages of GPS over other methods of measuring attitude, few commercial GPS products have emerged with this capability. Some of the GPS attitude devices that are currently available are the Novatel Beeline (two-antenna system) [27], the Trimble TANS vector (four-antenna system), and the Furuno Satellite Compass (three-antenna system). These commercial systems are primarily designed for usage in aircraft [33], farm vehicles [8], and marine [26, 28] applications. With costs ranging from \$4k to over \$10k, these systems are prohibitively expensive for consumer automobile applications, and clearly do not target this market.

In addition to their high cost, existing commercial GPS attitude systems perform poorly in urban environments, where the number of visible satellites changes often and may be less than four (typical number required for acceptable accuracy). As an example of this problem, Figure 1.5 illustrates the number of channels that were usable for GPS attitude during a drive through a suburban neighborhood. With the likelihood that at least one of the few measurements available will contain significant phase- noise, traditional multiple-satellite GPS attitude solutions would have difficulty maintaining continuity and integrity during such a drive. Although an integrated INS system can provide navigation solutions during short GPS dropouts, the length of the potential outages shown in Figure 1.5 (tens of seconds) suggest that dead-reckoning with low-grade inertial components may not be sufficient because of the sporadic availability of a calibration reference.

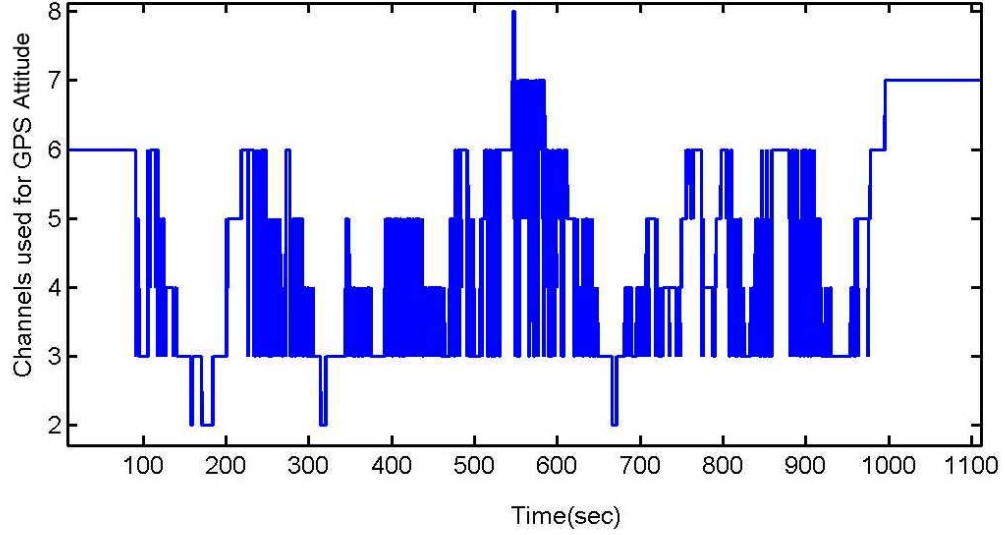


Figure 1.5: Number of SVs Usable for GPS Attitude in an Urban Environment

1.4 Prior Research

The use of GPS for attitude determination was conceived as early as the late 1970s, and saw its first practical implementations in the early 1990s [68]. The research of Cohen [13] produced one of the first comprehensive documents of GPS attitude methodology and capabilities, with a focus on aircraft and spacecraft applications. Hayward [43] and Dumaine [18] pioneered the prototyping of low-cost GPS attitude systems. Efforts to make carrier phase applications more practical and robust, both carrier-phase positioning and attitude determination, have concentrated on the integer-ambiguity search process, and have resulted in numerous publications that present algorithms to reduce the search space and execute more reliable and efficient searching [13, 14, 20, 21, 48]. However, prior research on GPS attitude systems has not addressed the difficulties introduced by large phase noise induced by multipath, frequent cycle slips, and limited satellite visibility, which can deter system robustness in urban areas.

Aiding of inertial navigation systems with some kind of RF-based measurements of position, velocity, or acceleration has been used for several decades. The use of GPS for this purpose is only the latest variant of this old idea. Research and applications of GPS/INS integration go back nearly to the inception of GPS itself, and publications on this subject are prolific. For conciseness, only notable prior research relevant to attitude determination is mentioned here.

Code-phase or carrier-phase from a single GPS antenna can be used to aid a six degree-of-freedom INS, which includes attitude outputs, for measuring pitch and roll angles. Used in conjunction with distance measuring equipment (DME) or instrument landing systems (ILS),

this technique is often used in aircraft for aiding their INS. Hayward [43] explored these ideas using low-cost inertial sensors and code-phase based GPS-derived acceleration and showed that automotive grade inertials could be used to achieve better than 1° accuracy. The yaw or heading angle, however, still used a two-antenna system to provide observability of the aircraft yaw angle, which may not be the same as the velocity heading obtained from a single GPS antenna. This idea is also applicable to a car, where yaw measurements can be obtained from a GPS attitude system or from single-antenna velocity heading when the car is not turning.

In regard to multiple-antenna aircraft attitude systems, Gebre and Hayward [33] implemented a loosely coupled GPS/INS attitude system, and demonstrated the benefits of such coupling. Tome [65], Fathy [24] and others have also published results with GPS/INS coupling for aircraft. For ground vehicles, Bevely [8] used loose coupling with a Trimble TANS Vector system for automatic control of a tractor. More recently, Ryu and Rossetter [58] used loose coupling with a Novatel Beeline to measure sideslip in a car. Most studies of GPS/INS integration have been done for specific applications, with specific components. A more general treatment of the topic was done by Gautier [31], where the full design space covering IMU types and methods of integration was investigated.

Ultra-tight GPS/INS integration is a popular subject in modern research, and is being developed by a few organizations, including Interstate Electronics Corporation [6], The Aerospace Corporation, and Draper Laboratories [40]. The approach taken in their implementations involves the use of a single large filter, or smaller multiple filters that use in-phase and quadrature samples from the GPS receiver channels as measurements for updating the filter states. Control of the replicated carrier and code generators comes from navigation filter outputs, which are propagated with IMU measurements to achieve high bandwidth carrier-phase tracking and anti-jam capability. This variant of ultra-tight integration is characterized by implementing the closed-loop signal tracking for all channels through the navigation filter itself, thus precluding the need to maintain separate code and carrier tracking loops. The disadvantage of this architecture is its complexity, as the designs of the navigation filter and the GPS receivers are coupled. Another important detail about the implementation of these primarily military applications, is that the IMU is of tactical or navigation grade, and too expensive to be considered for consumer automotive applications. Thus, the feasibility of implementing such a system with automotive-grade inertial components has not been assessed by previous research.

1.5 Thesis Contributions

This thesis focuses on a low-cost implementation of an automobile attitude determination system with GPS and INS, and methods for making it more robust in urban areas where GPS

outages, cycle slips, and large phase noise are commonplace. The contributions to this aim are as follows:

- Development of algorithms for single-satellite attitude determination, and quantification of the performance of these algorithms.
- Design and construction of an inexpensive prototype of a real-time GPS/INS attitude system that incorporates one-satellite attitude algorithms, tight integration at the carrier phase level, and multiple baseline redundancy.
- Development of controls-based techniques for the design of Doppler aided GPS tracking loops, utilizing both GPS clock and IMU quality parameters, and considering inexpensive designs for automotive applications.
- Quantification of advantages of Doppler aiding, including temporary cycle-slip immunity through open-loop tracking (important to automotive carrier phase applications)
- Demonstration of Doppler-aiding with real GPS signal samples taken in a car, and with automobile attitude system prototype (i.e, with automotive-grade IMU and inexpensive GPS equipment) to enable Doppler-frequency feedback from a navigation filter to individual channel PLLs.

1.6 Thesis Organization and Solution Overview

The remaining sections of this thesis will present the details of attitude determination with GPS for automotive applications, and robustness improvements through tight and ultra-tight GPS/INS integration.

Chapter 2 contains a detailed discussion of methods and algorithms utilized in GPS attitude determination with multiple satellites and with a single satellite. This chapter includes sections on the initialization steps for a GPS attitude system, line-bias estimation and integer-ambiguity resolution. Traditional operational methods of GPS attitude determination are also covered, with a focus on the linear least-squares solution.

Chapters 3 and 4 focus on the component and system design of an inexpensive attitude-system prototype for automobiles. The main points in these chapters include hardware descriptions, incorporation of multiple-satellite and one-satellite solutions into a single GPS subsystem, and robustness enhancement methods with and without tight integration. The performance of this system is evaluated in Chapter 5, which shows the results from various road tests.

Chapter 6 delves into the topic of ultra-tight GPS/INS integration through Doppler-aiding (with federated tracking loops), and the further benefits to GPS carrier-phase outputs that can be obtained from this technique. This discussion covers details such as PLL design, impact of local clock and IMU type, and selection of phase-lock loop bandwidth. This level of integration was not incorporated into the real-time attitude prototype discussed in Chapters 4 and 5, but was tested comprehensively in post processing with the use of a software GPS receiver, and utilizing real and simulated data. The results of this testing are covered in Chapter 7.

Chapter 8 contains some concluding remarks that review the main points of this thesis, and recommendations for future work.

A block diagram that summarizes the system architecture recommended in this thesis is presented in Figure 1.6. Note that the blocks are subdivided into two subsets, which are labeled “Real-Time System” and “Post-Process/Simulation”. While the entire diagram represents the eventual configuration of a real-time system, the tools available for the development of this research precluded the implementation of Doppler-aiding in real time. Thereby, the characteristics of this solution were demonstrated in two parts. The first part includes real-time performance analysis with all features and benefits of tight integration at the carrier phase level, and is relevant to Chapters 2-5. The second part that demonstrates the Doppler-aiding concepts discussed in Chapter 6 was implemented through post processing of both real and simulated data.

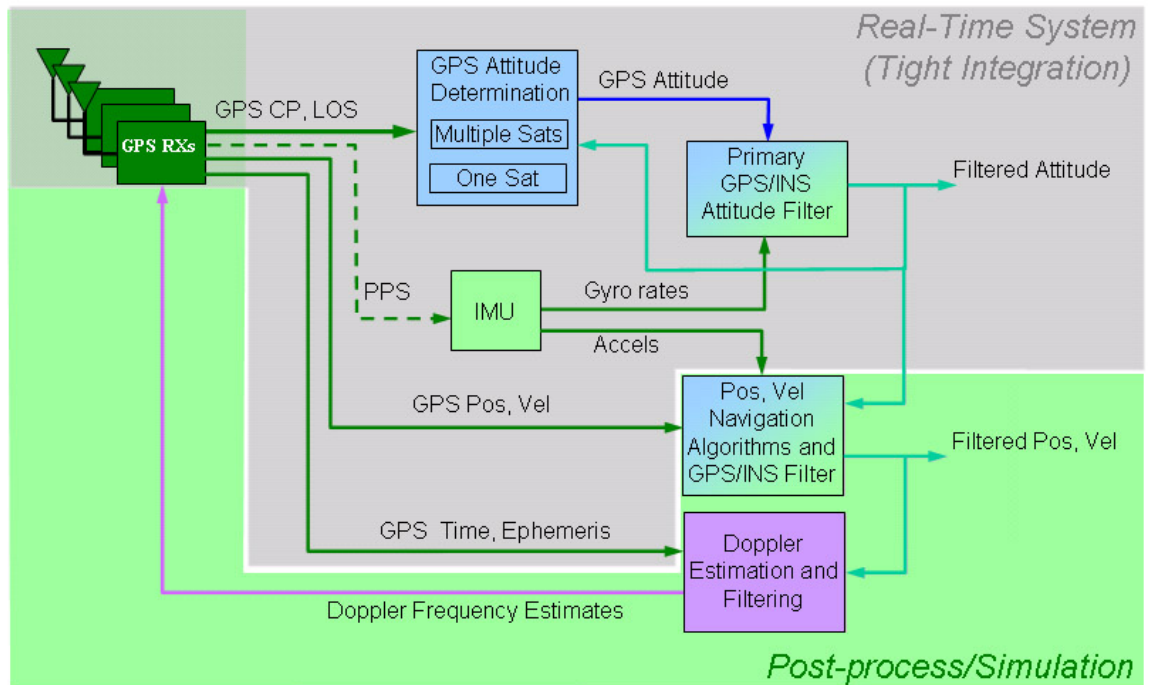


Figure 1.6: Architecture of Automobile GPS/INS Attitude System with Doppler Aiding

Chapter 2

Attitude Determination with GPS

This chapter begins with a review of differential carrier-phase and algorithms utilized in multiple-satellite/multiple antenna, attitude determination, with an emphasis on the linear least-squares solution. The review presents simple techniques for line bias and integer-ambiguity resolution, and describes the concept of attitude dilution of precision and the process of phase-delay calibration. This chapter also contains a detailed development and analysis of one-satellite attitude determination. This technique can be employed to provide GPS attitude availability when GPS satellite visibility is limited, and is particularly suitable for vehicles that experience small pitch and roll. The chapter concludes with a discussion on baseline configuration design, which describes the design tradeoff of accuracy versus robustness, and presents a quantitative analysis on the accuracy and robustness that can be expected from varying baseline lengths.

2.1 Differential Carrier-Phase Measurements

Carrier-phase measurements are obtained from the PLL in a GPS receiver. The specific details of how these measurements are generated will be discussed in Chapter 6. For the purposes of this chapter, it is assumed that the PLL in all channels is tracking the phase and frequency of the carrier signals. In other words, a component of the PLL called the numerically controlled oscillator (NCO) is generating a precisely synchronized replica of the down-converted carrier signal from a given satellite. The frequency of this replica is regulated by the PLL to adjust for Doppler frequency and local clock drift. The NCO can thus provide a measurement of the phase of the down-converted carrier signal at any given time, expressed as a fraction of a cycle. As long as the carrier is tracked continuously between carrier-phase measurements, the integrated carrier-phase (ICP) can also be generated. This quantity consists of the fractional phase measurement plus the whole number of cycles that have elapsed since initialization. The ICP measurement can be represented in terms of its components, as follows:

$$\varphi_A^k(t) = \varphi_{fA}^k(t) + m_A^k(t) \quad (2.1)$$

where

$\varphi_A^k(t)$ = ICP measurement for antenna A and SV#k at time t

$m_A^k(t)$ = Integrated cycles since initialization for antenna A and SV#k

$\varphi_{fA}^k(t)$ = Fractional phase measurement for antenna A and SV#k

Since the IF is constant, the ICP measurements provided by a receiver do not usually include number of cycles due to the IF component of the down-converted carrier. With this simplification, a change in the integrated number of cycles can be attributed only to relative motion between the satellite and the receiver (Doppler frequency) and local or satellite clock drift. To provide a visualization of how the ICP measurement behaves, Figure 2.1 illustrates a simple one-dimensional example of how the relative motion between a satellite and a receiver would be measured with the ICP.

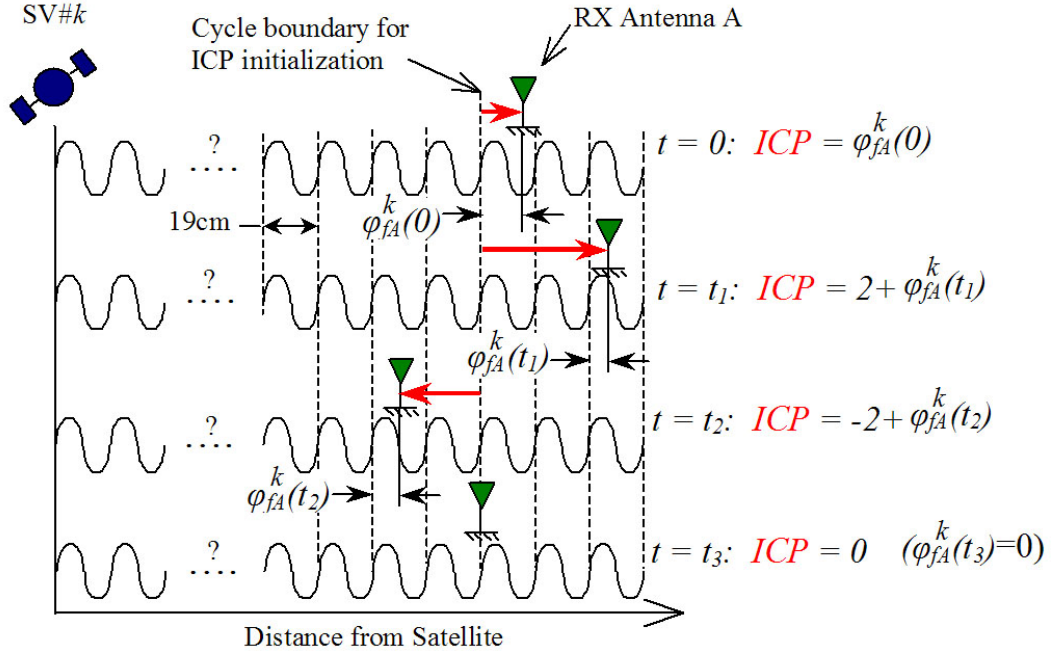


Figure 2.1: Integrated Carrier-Phase Measurements

The ICP at each time epoch is indicated with the thicker red arrows. Note that the ICP is made up of the integrated number of whole-number cycles since initialization, plus the current fractional phase measurement. Other important features pointed out in Figure 2.1 are that the ICP is proportional to the relative distance between the SV and receiver (RX) antenna, and that the absolute number of cycles between the satellite and the RX antenna is an unknown quantity.

Certain conditions like very high accelerations and high phase noise may cause the PLL to lose lock of its current cycle and begin tracking another cycle. This condition is known as a

cycle slip, and causes a discontinuity in the integrated number of whole-number cycles in the ICP. Cycle slips also occur occasionally during regular operating conditions, simply as a result of typical noise in the input signal to the NCO [62]. Cycle slips do not affect the fractional phase component, since the phase discriminator in the PLL always provides a measurement of the relative phase between the replicated and tracked carrier signals, whether or not the PLL has achieved phase lock.

The ICP measurement can also be written in terms of the physical quantities that influence its value [19]:

$$\varphi_A^k(t) = r_A^k(t) + T_{\varphi A}^k(t) + I_{\varphi A}^k(t) + n_A^k + f_{L1}(\delta t_A(t) + \delta t^k(t)) + l_A(t) + \varepsilon_{\varphi A}^k(t) \quad (2.2)$$

where

$$\begin{aligned} r_A^k(t) &= \text{Range between antenna } A \text{ and SV}\#k \text{ at time } t \\ T_{\varphi A}^k(t) &= \text{Troposphere phase-delay between antenna } A \text{ and SV}\#k \text{ at time } t \\ I_{\varphi A}^k(t) &= \text{Ionosphere phase-delay between antenna } A \text{ and SV}\#k \text{ at time } t \\ n_A^k &= \text{Integer ambiguity for antenna } A \text{ and SV}\#k \\ \delta t_A(t) &= \text{Clock error for antenna } A \text{ at time } t \\ \delta t^k(t) &= \text{Satellite clock error for SV}\#k \text{ at time } t \\ l_A(t) &= \text{Line propagation delay for antenna } A \text{ at time } t \\ \varepsilon_{\varphi A}^k(t) &= \text{Phase noise for antenna } A, \text{ in channel tracking SV}\#k \text{ at time } t \end{aligned}$$

Note that the integer ambiguity is not a function of time in this equation; its value is fixed after initialization and remains constant as long as there is no cycle slip. In reality, the phase-noise term may include phase-delay, which is a function of the incidence angle of the signal relative to the antenna; for now this effect is neglected, but will be revisited in Section 2.5.

The range term in Eq. 2.2 is the one that would be of interest for carrier phase applications, and all other terms can be considered nuisance parameters. The unobservability of the various individual terms in Eq. 2.2 makes the range very difficult to determine from a single ICP measurement. Due to this complication, ICP measurements from a single antenna at a single epoch have little use. However, the difference between concurrent ICP measurements taken from two nearby antennas tends to cancel out the atmospheric delays and other common mode errors, and makes relative ranging between antennas possible. The local clock error also cancels when the PLLs for the two antennas are driven from the same reference or local oscillator. This simplification can be assumed for attitude determination, but not for CP positioning. Similar cancellations

can be achieved through differencing of ICP measurements from the same antenna at different epochs, and these “delta phase” measurements can be used to measure user velocity [67] or relative position from an arbitrary starting point.

The difference between ICP measurements from two antennas is called differential carrier phase (abbreviated dICP in this document to emphasize difference of integrated carrier-phase measurements). For attitude determination, where multiple GPS receivers use the same reference oscillator, and where the distance between antennas A and B is typically no more than several meters, the following assumptions can be made:

$$T_{\varphi A}^k(t) \approx T_{\varphi B}^k(t) \quad (2.3a)$$

$$I_{\varphi A}^k(t) \approx I_{\varphi B}^k(t) \quad (2.3b)$$

$$\delta t_A(t) \approx \delta t_B(t) \quad (2.3c)$$

Utilizing these assumptions and Eq. 2.2, an expression for the dICP between antennas A and B is obtained:

$$\begin{aligned} \Delta\varphi_{AB}^k(t) &= \varphi_A^k(t) - \varphi_B^k(t) \\ &= r_{AB}^k(t) + n_{AB}^k + l_{AB}(t) + \varepsilon_{\varphi AB}^k(t) \end{aligned} \quad (2.4)$$

where

$$\begin{aligned} \Delta\varphi_{AB}^k(t) &= \text{dICP measurement between antennas } A \text{ and } B \\ r_{AB}^k(t) &= r_A^k(t) - r_B^k(t) \text{ (Delta-range term)} \\ n_{AB}^k &= n_A^k - n_B^k \text{ (Integer ambiguity term)} \\ l_{AB} &= l_A - l_B \text{ (Line-bias term)} \\ \varepsilon_{\varphi AB}^k(t) &= \varepsilon_{\varphi A}^k(t) - \varepsilon_{\varphi B}^k(t) \text{ (Phase-noise term)} \end{aligned}$$

The line bias term in Eq. 2.4 also accounts for differences in phase between frequency synthesizers in different RF front ends, which exhibits similar behavior to a true line-propagation delay that varies slowly in time.

The dICP measurement can also be written in terms of its integer and fractional components, analogous to Eq. 2.1:

$$\begin{aligned} \Delta\varphi_{AB}^k(t) &= \varphi_A^k(t) - \varphi_B^k(t) \\ &= \varphi_{fAB}^k(t) + m_{AB}^k(t) \end{aligned} \quad (2.5)$$

where

$$\begin{aligned}\varphi_{fAB}^k(t) &= \text{Fractional component of dICP measurement} \\ m_{AB}^k(t) &= \text{Integer component of dICP measurement}\end{aligned}$$

It is important to note that $m_{AB}^k(t)$ is not necessarily the same integer ambiguity as in Eq. 2.4; it is merely the whole-number portion of the raw dICP measurement, and it depends on the precise time at which the two PLLs initialize the ICP. It is possible to use only the fractional portion of the dICP by truncating out its integer component, but the carrier phase application would need to perform integer searches at every epoch. For most applications, this inconvenience is best avoided by keeping the whole-number components in the dICP measurements. Combining Eqs. 2.4 and 2.5 emphasizes this idea by illustrating how the integer component of the dICP measurement contributes to the tracking process of the terms in Eq. 2.4 that change with time (delta range, line bias), and allows the integer ambiguity to remain fixed as long as there is no cycle slip:

$$\begin{aligned}\Delta\varphi_{AB}^k(t) &= \varphi_{fAB}^k(t) + m_{AB}^k(t) \\ &= r_{AB}^k(t) + n_{AB}^k + l_{AB}(t) + \varepsilon_{\varphi AB}^k(t)\end{aligned}\tag{2.6}$$

2.2 Linear Least-Squares Attitude Solution

2.2.1 Formulation

The geometrical relationship between the orientation of a baseline, the LOS unit vector and the delta-range term in Eq. 2.5 is illustrated in Figure 2.2. Vector quantities are shown as the thicker, red arrows; the baseline vector points from antenna A to antenna B, and the LOS vector is points from the user (either antenna) to the SV. Figure 2.2 also shows the fractional and whole-number components of the delta range, and how they relate to the components of the measurement. To simplify the image, the line bias and phase-noise terms from Eq. 2.6 are assumed to be zero, and not included in the figure.

In mathematical terms, the delta-range (a scalar value) is the magnitude of the projection of the baseline vector onto the LOS unit vector, and can be written as the inner product of the two vectors:

$$r_{AB}^k(t) = -\vec{1}^k(t) \cdot {}^A\vec{B}^B(t)\tag{2.7}$$

This expression is valid regardless of the coordinate frame (also called a basis) in which the vectors are expressed. However, attitude is usually measured relative to a locally-level frame; some

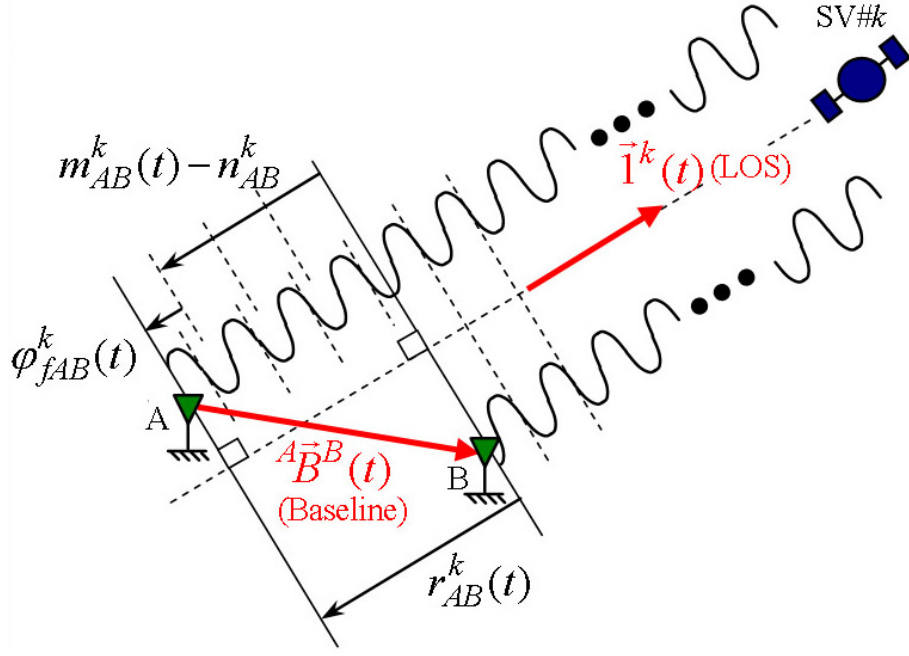


Figure 2.2: Vector Diagram for Attitude Determination

such frames include the local East-North-Up (ENU), North-West-Up or North-East-Down. True Earth-fixed coordinate frames can also be used, such as Earth-Centered-Earth-Fixed (ECEF). The locally-level frames are not truly Earth-fixed, as they would rotate with translation of the vehicle on the Earth's surface. However, for vehicles that move relatively slowly such as automobiles, the locally-level frames are nearly fixed (rotationally) relative to the Earth. To be consistent with the more intuitive GPS output formats, this thesis uses the ENU frame (denoted with subscript 'e') as the Earth-fixed frame for referencing vehicle attitude. This frame can be thought of as a locally-level frame centered on the vehicle, whose horizontal axes always point East and North. Expressed with an ENU basis, Eq. 2.7 becomes

$$r_{AB}^k(t) = -\underline{1}_e^{kT}(t) \underline{A}\underline{B}_e^B(t) \quad (2.8)$$

The vector and basis notation introduced in Eqs. 2.7 and 2.8 is used consistently throughout this document, and is summarized in Appendix A.

Substituting Eq. 2.8 into Eq. 2.6 yields an equation containing raw dICP measurements and unknown quantities:

$$\Delta\varphi_{AB}^k(t) = -\underline{1}_e^{kT}(t) \underline{A}\underline{B}_e^B(t) + n_{AB}^k + l_{AB}(t) + \varepsilon_{\varphi_{AB}}^k(t) \quad (2.9)$$

While all terms but the integer ambiguity are functions of time, the LOS vector and the line bias change very slowly when compared with the baseline, and are often assumed to remain constant

in the short term.

With N satellites in view, Eq. 2.9 represents a row of the matrix equation:

$$\underline{\Delta\Phi}_{AB}(t) = \underline{H}_e(t) {}^A\hat{\underline{B}}_e^B(t) + \underline{N}_{AB} + \underline{L}_{AB}(t) + \underline{E}_{\varphi AB}(t) \quad (2.10)$$

where

$$\underline{\Delta\Phi}_{AB}(t) = [\Delta\varphi_{AB}^1(t) \ \Delta\varphi_{AB}^2(t) \ \cdots \ \Delta\varphi_{AB}^N(t)]^T \quad (2.11a)$$

$$\underline{H}_e(t) = \begin{bmatrix} -\underline{1}_e^1(t) & -\underline{1}_e^2(t) & \cdots & -\underline{1}_e^N(t) \end{bmatrix}^T \quad (2.11b)$$

$$\underline{N}_{AB} = \begin{bmatrix} n_{AB}^1 & n_{AB}^2 & \cdots & n_{AB}^N \end{bmatrix}^T \quad (2.11c)$$

$$\underline{L}_{AB}(t) = l_{AB}(t) \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^T \quad (2.11d)$$

$$\underline{E}_{\varphi AB}(t) = \begin{bmatrix} \varepsilon_{\varphi AB}^1(t) & \varepsilon_{\varphi AB}^2(t) & \cdots & \varepsilon_{\varphi AB}^N(t) \end{bmatrix}^T \quad (2.11e)$$

With three or more dICP measurements (requires ≥ 3 common SVs tracked with both antennas), and if the line bias and integer ambiguities are known, the baseline vector components can be resolved with a least-squares solution:

$${}^A\hat{\underline{B}}_e^B(t) = \underline{H}_e^\dagger(t) (\underline{\Delta\Phi}_{AB}(t) - \underline{N}_{AB} - \underline{L}_{AB}(t)) \quad (2.12)$$

where

$$\underline{H}_e^\dagger(t) = \left(\underline{H}_e^T(t) \underline{H}_e(t) \right)^{-1} \underline{H}_e^T(t) \quad (2.13)$$

A variant of this equation includes a correction term to the original line-bias estimate, and can be computed with four or more dICP measurements:

$$\begin{bmatrix} {}^A\hat{\underline{B}}_e^B(t) \\ \delta \hat{l}_{AB}(t) \end{bmatrix} = \underline{H}_1^\dagger(t) (\underline{\Delta\Phi}_{AB}(t) - \underline{N}_{AB} - \underline{L}_{AB_0}) \quad (2.14)$$

where

$$\begin{aligned} \underline{H}_1(t) &= \begin{bmatrix} -\underline{1}_e^1(t) & -\underline{1}_e^2(t) & \cdots & -\underline{1}_e^N(t) \\ 1 & 1 & \cdots & 1 \end{bmatrix}^T \\ \delta \hat{l}_{AB}(t) &= \text{Line-bias correction term} \end{aligned} \quad (2.15)$$

Note that in this equation, the line-bias correction term is a function of time, allowing the line bias matrix to remain constant after initialization. Including this correction allows tracking of slow time variations in the line bias, which occur as a result of temperature changes in the cables

and RF front ends.

Since the baseline-vector solutions in Eqs. 2.12 and 2.14 are expressed in an ENU frame and the baseline is fixed on a vehicle, simple geometry yields two attitude degrees of freedom from a single baseline; that is, rotations about two axes orthogonal to the baseline. This topic will be revisited in Section 2.2.4.

As suggested by Eq. 2.9 the total line-bias is indistinguishable from the integer ambiguity in the raw dICP measurement. However, the combined quantity can be split into an integer part and a fractional part. The integer part can then become part of the \underline{N}_{AB} vector, and the fractional part can be attributed to line bias. This notation is advantageous, for it allows for simpler integer-ambiguity searches in which true integers (not real numbers) are sought. Furthermore, a cycle slip will cause a change in the integer ambiguity but not in the line bias; since line bias changes slowly with time, repeated computation of its value after cycle slips would be overly redundant. For these reasons, the algorithms described in the following sections treat line-bias estimation and integer determination separately.

2.2.2 Line-Bias Estimation

Line-bias estimation is the first step to obtaining an attitude solution, as future integer searches (as will be described in the next section) will require an estimate of line bias to search for whole-number integer ambiguities. Furthermore, this initial line bias estimate will serve as a constant in subsequent attitude solutions, as shown in Eq. 2.14. The method described in this section uses the fact that line bias is a common term in all channels, as represented in Eq. 2.11d. This fact will be exploited in a search through the possible baseline-orientation space, which will yield an attitude estimate based on the point in the search-space where the line biases from all channels agree with minimum variance.

The orientation of a baseline is determined by two angles, which define the rotation matrix between a baseline-fixed frame and the ENU frame. Note that by using a body basis aligned with the baseline, there is no need to use the three angles (roll,pitch,yaw) that define the *vehicle* orientation. Figure 2.3 illustrates the definitions of the baseline orientation angles, where ψ_{AB} and ϕ_{AB} are the baseline's yaw and roll orientation angles, respectively. As shown, the baseline-fixed frame is defined such that the y and z components of the baseline vector are zero when expressed in this basis.

Using Eq. 2.9 and dropping the integer ambiguity results in an expression for a line-bias guess (symbol ‘ \sim ’ denotes a guess) in one channel, as a function of a guess for the baseline orientation:

$$\tilde{l}_{AB}^k(t_i) = \text{fraction} \left(\Delta\varphi_{AB}^k(t_i) - \underline{1}_e^{k^T}(t_i)^A \underline{\tilde{B}}_e^B \right) \pm 1 \quad (2.16)$$

where

$$fraction(x) = \begin{cases} x - floor(x) & x \geq 0 \quad (floor = \text{round down operation}) \\ x - ceil(x) & x < 0 \quad (ceil = \text{round up operation}) \end{cases} \quad (2.17)$$

$${}^A\tilde{\underline{B}}_e^B = R_{AB} \begin{bmatrix} \cos \tilde{\psi}_{AB} \cos \tilde{\phi}_{AB} \\ \sin \tilde{\psi}_{AB} \cos \tilde{\phi}_{AB} \\ -\sin \tilde{\phi}_{AB} \end{bmatrix} \quad (2.18)$$

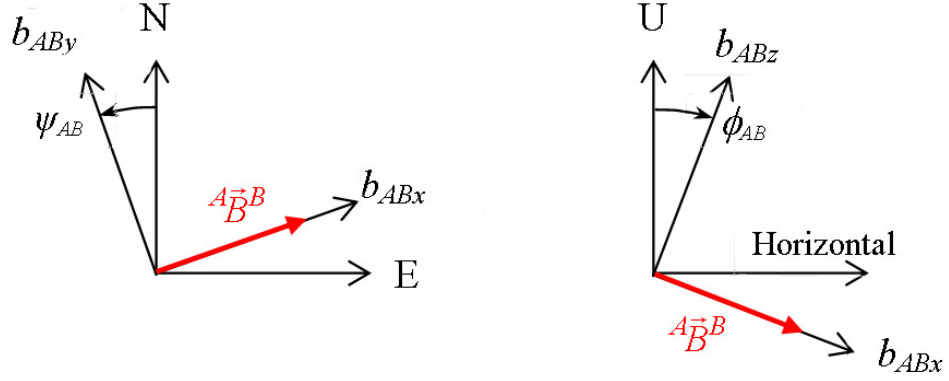


Figure 2.3: Definition of Baseline Orientation Angles

The ± 1 term in Eq. 2.16 adjusts for cycle rollover, such that all channels have the same sign on the line bias guess, and R_{AB} in Eq. 2.18 is the length of baseline AB .

To facilitate expression of the channel line-biases variance, the following $N \times 1$ matrices are defined, for N channels:

$$\tilde{\underline{l}}_{AB}(t_i) = \begin{bmatrix} \tilde{l}_{AB}^1(t_i) & \tilde{l}_{AB}^2(t_i) & \cdots & \tilde{l}_{AB}^N(t_i) \end{bmatrix}^T \quad (2.19)$$

$$\tilde{\underline{l}}_{AB}(t_i) = \tilde{\underline{l}}_{AB}(t_i) \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^T \quad (2.20)$$

where

$$\tilde{l}_{AB}(t_i) = \frac{1}{N} \sum_{k=1}^N \tilde{l}_{AB}^k(t_i) \quad (2.21)$$

The sample variance of the channel biases as a function of the guess for baseline orientation can now be computed as:

$$\sigma_{lAB}^2(t_i, \tilde{\psi}_{AB}, \tilde{\phi}_{AB}) = \left(\tilde{\underline{l}}_{AB}(t_i) - \tilde{\underline{l}}_{AB}(t_i) \right)^T \left(\tilde{\underline{l}}_{AB}(t_i) - \tilde{\underline{l}}_{AB}(t_i) \right) \quad (2.22)$$

Estimates of the baseline's yaw and roll angles can be obtained through an iterative search

for the values that minimize Eq. 2.22 (globally) in the search space:

$$(\hat{\psi}_{AB}(t_i), \hat{\phi}_{AB}(t_i)) = \min_{\tilde{\psi}_{AB}, \tilde{\phi}_{AB}} \{ \sigma_{LAB}^2(t_i, \tilde{\psi}_{AB}, \tilde{\phi}_{AB}) \} \quad (2.23)$$

Note, the symbol ‘ \sim ’ denotes an estimate henceforth.

Iterations in yaw and roll should only span plausible values. In automobile applications for example, the baseline roll is not likely to exceed $\pm 5^\circ$ in most situations. Furthermore, if the baselines are short (on the order of 0.5m), iterations in the roll dimension can usually be ignored under the assumption that the baseline is level ($\phi_{AB} = 0$).

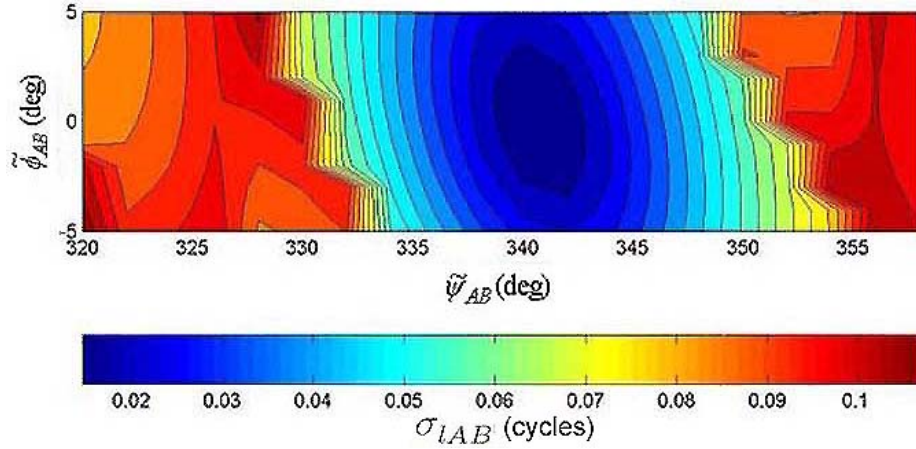


Figure 2.4: Attitude Search Based on Channel Line-Bias Variance

Figure 2.4 shows the results of this search process for a 50cm baseline at a single epoch. As shown clearly in the contour plot, $\hat{\psi}_{AB}(t_i) \approx 340^\circ$ and $\hat{\phi}_{AB}(t_i) \approx 0^\circ$ would be the baseline orientation estimate in this case, as the channel line-bias variance is minimized at these values. This plot was generated with a coarse attitude search with increments of 2° in yaw and 1° in roll. A subsequent finer resolution search (with 0.5° angle increments) in the neighborhood of the coarse estimate yielded attitude estimates of $\hat{\psi}_{AB}(t_i) \approx 341^\circ$ and $\hat{\phi}_{AB}(t_i) \approx -0.5^\circ$. With long baselines ($> 1\text{m}$) iteration in roll is more necessary and a smaller increment in the coarse search may be needed to locate the minimum, as the width of the depression in the contour plot becomes smaller with increasing baseline length.

Once this rough estimate of the baseline orientation is determined, the baseline vector estimate can be obtained with Eq. 2.18, and the channel line-bias estimates with Eq. 2.16:

$${}^A\hat{\underline{B}}_e^B = R_{AB} \begin{bmatrix} \cos \hat{\psi}_{AB} \cos \hat{\phi}_{AB} \\ \sin \hat{\psi}_{AB} \cos \hat{\phi}_{AB} \\ -\sin \hat{\phi}_{AB} \end{bmatrix} \quad (2.24)$$

$$\hat{l}_{AB}^k(t_i) = \text{fraction} \left(\Delta\varphi_{AB}^k(t_i) - \underline{1}_e^T(t_i)^A \hat{\underline{B}}_e^B \right) \pm 1 \quad (2.25)$$

In Eq. 2.24, the magnitude of the baseline is assumed to be a known quantity. Finally, the common line-bias is estimated from the average of the individual channel line-biases, which should agree to within about 0.1 cycles:

$$\hat{l}_{AB}(t_i) = \frac{1}{N} \sum_{k=1}^N \hat{l}_{AB}^k(t_i) \quad (2.26)$$

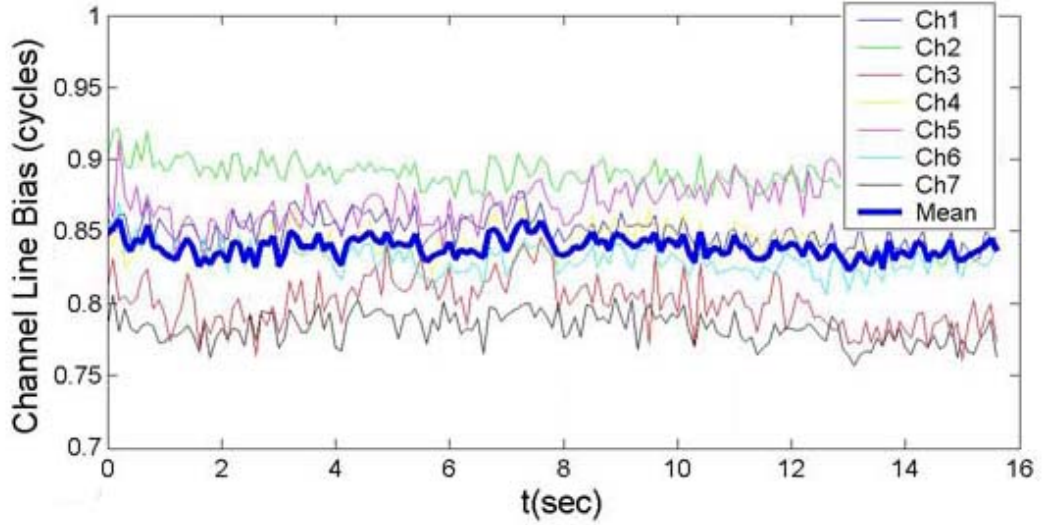


Figure 2.5: An Image of Line-Bias Estimation

Figure 2.5 illustrates the results of this process, for the same data set used to obtain Figure 2.4. As shown, the estimated line bias for this baseline is about 0.85 cycles, and this value can be estimated with sufficient accuracy at a single epoch or as an average over multiple epochs. From a single epoch, the initial line-bias matrix estimate can now be expressed as:

$$\hat{\underline{L}}_{AB_0} \approx \hat{l}_{AB}(t_0) \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^T \quad (2.27)$$

To correct any errors in this initial estimate and to maintain accuracy over long periods, it is necessary to track line-bias variations in time, as shown in Eq. 2.14. While line-bias corrections are not necessary at every epoch, they should be computed whenever more than four dICP measurements are available. As long as temperature transients in the lines and RF front ends have reached steady state after startup, these corrections tend to remain small (on the order of a few centimeters). The time-varying component of the line-bias matrix is best obtained as the low-pass filtered value or running average of several consecutive line-bias correction terms, and

can be expressed as follows:

$$\underline{\delta\hat{L}}_{AB}(t_i) = \underline{\delta\hat{L}}_{AB}(t_i) \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}^T \quad (2.28a)$$

where

$$\underline{\delta\hat{L}}_{AB}(t_i) = \text{mean}(\delta\hat{L}_{AB}(t_i), \delta\hat{L}_{AB}(t_{i-1}), \dots, \delta\hat{L}_{AB}(t_{i-n})) \quad (2.28b)$$

At any given time, the total line-bias estimate used when computing attitude can be expressed as the sum of the initial estimate, plus the time-varying component:

$$\underline{\hat{L}}_{AB}(t_i) = \underline{\hat{L}}_{AB_0} + \underline{\delta\hat{L}}_{AB}(t_i) \quad (2.29)$$

2.2.3 Integer Ambiguity Determination

As explained in Section 2.1, the nature of dICP measurements between different receivers makes integer ambiguities (as defined in Eq. 2.6) essentially random. This difficulty makes it impractical to use an N -dimensional integer search space, since the value of an integer ambiguity would be limited only by the size of the register that accumulates the integrated number of cycles of the ICP measurement. Thus, this section discusses a method for integer searches by iterating in discrete steps through the baseline *attitude* space, similar to the technique used in the previous section to estimate the line bias. Other integer-determination methods are discussed briefly at the end of this section.

Making use of Eq. 2.10 and assuming that a line-bias estimate is available, a guess for the integer ambiguities as a function of a guess for baseline yaw and roll is:

$$\underline{\tilde{N}}_{AB} = \text{round} \left(\underline{\Delta\Phi}_{AB}(t_i) - \underline{\hat{L}}_{AB}(t_i) - \underline{H}_e(t_i) \underline{\tilde{B}}_e^{AB} \right) \quad (2.30)$$

From this integer guess, a more precise baseline guess not limited to the resolution of the angular increments of the integer search can be computed by using Eq. 2.12:

$$\underline{\tilde{B}}_e^{AB} = \underline{H}_e^\dagger(t_i) (\underline{\Delta\Phi}_{AB}(t_i) - \underline{\tilde{N}}_{AB} - \underline{\hat{L}}_{AB}(t_i)) \quad (2.31)$$

To test the validity of the integer ambiguity guess in Eq. 2.30, the baseline-length error and the residual of the baseline guess can be used, and are defined as:

$$\delta B_{AB}(\underline{\tilde{N}}_{AB}) = \left| \underline{\tilde{B}}_e^{AB} \right| - R_{AB} \quad (\text{Baseline Length Error}) \quad (2.32)$$

$$\begin{aligned}
e_{AB}(\tilde{N}_{AB}) &= |\underline{E}_{\varphi AB}| \\
&= \left| \underline{\Delta\Phi}_{AB}(t_j) - \underline{H}_e(t_j) \hat{\underline{B}}_e^{AB} - \tilde{N}_{AB} - \hat{\underline{L}}_{AB}(t_i) \right| \quad (\text{Residual})
\end{aligned} \tag{2.33}$$

These two quantities tend to be minimized when \tilde{N}_{AB} matches the correct set of integers, and can thus be used to differentiate correct integers from incorrect ones. The residual test is the more discriminating test, as the baseline-length error test can often have minimum values for incorrect sets of integers.

To simplify the decision process, the residual and baseline-length error tests can be used together in a single cost function, which the integer search process seeks to minimize. For example, a cost function that weighs the residual test three times more than the baseline error test would be:

$$J_N(\tilde{N}_{AB}) = \delta B_{AB}(\tilde{N}_{AB}) + 3 e_{AB}(\tilde{N}_{AB}) \tag{2.34}$$

The integers that minimize this cost function in the search space would be the best guess for the correct integers:

$$\hat{\tilde{N}}_{AB} = \min_{\tilde{N}_{AB}(\hat{\phi}_{AB}, \hat{\psi}_{AB})} (J_N(\tilde{N}_{AB})) \tag{2.35}$$

Figure 2.6 shows the results of an integer search process for the same data used to generate the line bias estimation plots in the last section. As shown, the search space covers the 360° yaw space in 10° increments, and found equivalent cost-function minimums (corresponding to the same set of integers) at $\hat{\psi}_{AB} = 0^\circ, 350^\circ, 340^\circ$, and 330° . As expected, these results are consistent with those of Figure 2.4, which shows that the baseline yaw angle should be about 340°.

Since the 50cm baseline was known to have small tilt ($< 5^\circ$), this integer search did not iterate in the roll degree-of-freedom. However, baselines that expect high roll angle may need to iterate in both of their rotational degrees of freedom, depending on the baseline length and magnitude of the roll angle. Longer baselines may also require smaller angle increments to cover all possible integer combinations, thus increasing the size of the integer search-space. Furthermore, the distinction between correct and incorrect integers (based on the minimum of the cost function) becomes less pronounced with increasing baseline length, thus compromising integrity in environments with high phase noise [15].

Because of phase noise from such phenomena as multipath, signal attenuation, and RF interference, finding the minimum of a cost function does not always correspond to the correct set of integers. For this reason, further verification is necessary to assess the validity of the integers determined per Eq. 2.35. With no external aiding, the baseline-length error and the residual values can be compared with previously established maxima for these errors, and the integers rejected if the measured errors exceed either of the maximum acceptable values. This technique

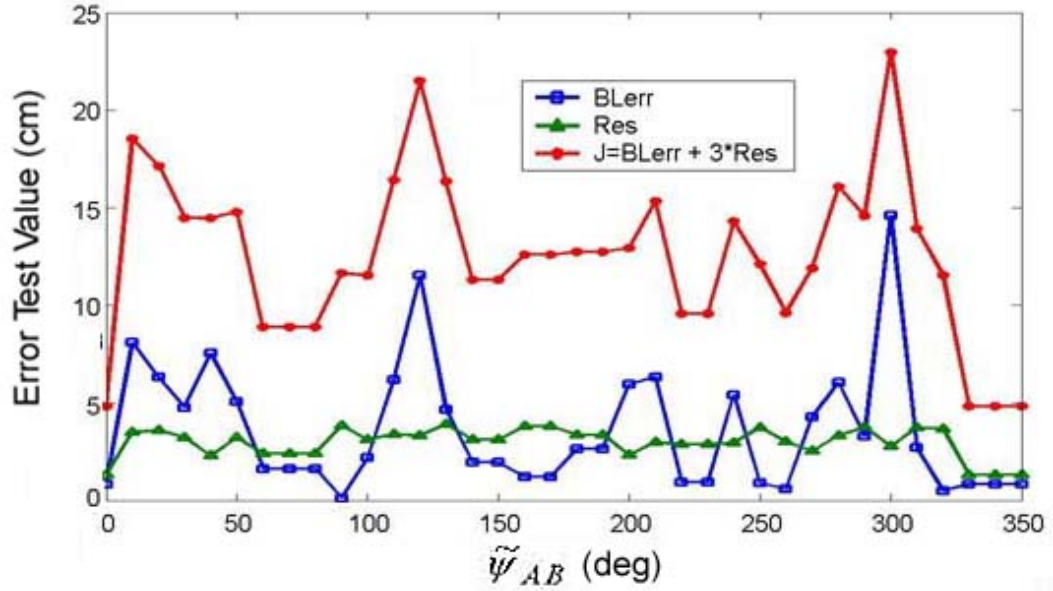


Figure 2.6: Image of an Integer-Search Process

is expressed as follows:

$$\hat{N}_{AB} = \begin{cases} \hat{N}_{AB} & \text{if } (\delta B_{AB} < \delta B_{\max}) \text{ AND } (e_{AB} < e_{\max}) \\ \text{Unknown} & \text{Else} \end{cases} \quad (2.36)$$

The maximum acceptable error values can be established empirically from test data, as values can change depending on the baseline lengths and the amount of phase noise present. Typical values for both thresholds are on the order of 1-2cm.

The integer-determination method described in this section can be described as an “instantaneous” method, as data from only the current epoch is necessary to execute the integer search. This feature is desirable in real-time systems that need to be initialized without motion. Other instantaneous methods for attitude determination utilize the length of the baseline to reduce the size of the search-space, but introduce a sign ambiguity on the vertical baseline component [21].

Non-instantaneous methods can also be used to determine integer ambiguities for attitude determination, and may be necessary for long baselines of several meters. The technique described in [13] uses the antenna motion over several epochs to resolve the integer ambiguities at the first epoch, and thus requires that there are no cycle slips during the integer determination process. While the use of more data increases confidence in the integer solution through geometric diversity and averaging, most attitude applications are not suited to provide sufficient motion at any time for proper initialization.

Once the GPS attitude system has been initialized as described above, subsequent integer

searches can utilize information from channels for which the integer ambiguity is known or from an external estimate of attitude. The former case is applicable if at least three channels are available with known integers, as a baseline-orientation estimate could be computed from these channels. This knowledge would restrict the size of the search-space to the neighborhood of the original attitude estimate, and it may preclude the need to iterate at all depending on the baseline length and confidence in the attitude solution provided by the known channels. If an external estimate of the baseline orientation is available, as would be the case with a coupled inertial system, the same simplification can be made for all the channels. This technique can be a very powerful robustness improvement to a GPS attitude system, as integer searches through blind iteration can be avoided entirely as long as the inertial system does not operate without GPS calibration for an extended period. The advantage of this scheme is that erroneous integer combinations which could potentially result in low values of the cost function and large attitude errors are not considered. Without such aiding, phase noise in one or more channels may cause the iterative integer-search process to converge to the wrong integers, thus causing a failure in integrity.

Since the integer-determination process with an external attitude estimate is an important part of the robustness enhancements discussed in Chapter 4, it will be also be documented here to be used as a reference later. The process of determining the integers for an arbitrary number of channels can be adapted easily from Eq. 2.30 by replacing the baseline guess by an external estimate:

$$\hat{\underline{N}}_{AB} = \text{round} \left(\underline{\Delta\Phi}_{AB}(t_i) - \hat{\underline{L}}_{AB}(t_i) - \underline{\underline{H}}_e(t_i)^A \underline{\underline{B}}_{ext}^B \right) \quad (2.37)$$

The integers determined in this fashion should still be subject to passing tests in Eq. 2.36.

2.2.4 Linear Attitude Solution

With the line bias estimate and resolved integer matrix, the three components of the baseline-vector estimate can be computed readily from Eq. 2.12:

$$\begin{aligned} {}^A\hat{\underline{B}}_e^B(t_i) &= \begin{bmatrix} \hat{x}_{eAB} & \hat{y}_{eAB} & \hat{z}_{eAB} \end{bmatrix}^T \\ &= \underline{\underline{H}}_e^\dagger(t_i) (\underline{\Delta\Phi}_{AB}(t_i) - \hat{\underline{N}}_{AB} - \hat{\underline{L}}_{AB}(t_i)) \end{aligned} \quad (2.38)$$

Finally, the two-axis orientation of the baseline is determined with simple geometry:

$$\hat{\psi}_{AB}(t_i) = \arctan 2 \left[\frac{\hat{y}_{eAB}(t_i)}{\hat{x}_{eAB}(t_i)} \right] \quad (2.39)$$

$$\hat{\phi}_{AB}(t_i) = \arcsin \left[\frac{-\hat{z}_{eAB}(t_i)}{R_{AB}} \right] \quad (2.40)$$

In Eq. 2.39, “arctan2” refers to the four-quadrant arctangent of the horizontal components of the baseline.

If the baseline-fixed basis depicted in Figure 2.3 lines up with the body axes of a vehicle, then the baseline yaw and roll angles determined with Eqs. 2.39 and 2.40 would correspond to the yaw and roll of the vehicle itself. However, due to mounting constraints and to provide observability over three axes of rotation, baselines often do not line up with the body axes of a vehicle. In this case, the baseline vectors computed per Eq. 2.38 require some manipulation before vehicle Euler angles can be extracted.

The following development presents one possible method for transforming three arbitrary baselines into vectors parallel to the vehicle roll and pitch axes, such that vehicle attitude can be computed easily. It is assumed that the antennas are coplanar with the body x - y plane (spanned by the pitch and roll axes), as shown in Figure 2.7. The three antennas define three baselines (one redundant) that may not line up with the body axes.

The baseline-defined basis for the longest baseline (basis b_{AB} for ${}^A\vec{B}^B$) is chosen as an intermediate basis, whose axes line up with two perpendicular vectors derived from the three baselines. The construction of these orthogonal vectors is illustrated in Figure 2.7a, and are computed as follows:

$${}^A\vec{B}^D = {}^A\vec{B}^B + {}^A\vec{B}^C - {}^B\vec{B}^C \quad (2.41)$$

$${}^A\vec{B}^F = {}^A\vec{B}^C + {}^B\vec{B}^C - \frac{({}^A\vec{B}^C + {}^B\vec{B}^C) \cdot {}^A\vec{B}^B}{R_{AB}^2} {}^A\vec{B}^B \quad (2.42)$$

As shown in Figure 2.7, the b_{AB} basis is offset from the body basis (b) by angle γ_{AB} , so a rotation by the negative of this angle about the b_z axis would transform ${}^A\vec{B}^D$ and ${}^A\vec{B}^F$ into vectors parallel to the body axes, which are shown in Figure 2.7b as ${}^A\vec{B}^{D*}$ and ${}^A\vec{B}^{F*}$. Assuming that Eqs. 2.41 and 2.42 are implemented in the ENU basis (as the result of Eq. 2.38 is in ENU basis), this transformation is achieved through the following quaternion rotations:

$${}^A\underline{B}_e^{D*} = {}^A\underline{B}_e^D + 2 \left(q_4 \underline{q}_e^X {}^A\underline{B}_e^D + \underline{q}_e^X \underline{q}_e^X {}^A\underline{B}_e^D \right) \quad (2.43)$$

$${}^A\underline{B}_e^{F*} = {}^A\underline{B}_e^F + 2 \left(q_4 \underline{q}_e^X {}^A\underline{B}_e^F + \underline{q}_e^X \underline{q}_e^X {}^A\underline{B}_e^F \right) \quad (2.44)$$

where

$$\begin{aligned} \underline{q}_e &= \begin{bmatrix} q_{e1} & q_{e2} & q_{e3} \end{bmatrix}^T \\ &= \frac{{}^A\underline{B}_e^{B^X} {}^A\underline{B}_e^{C^X}}{|{}^A\underline{B}_e^{C^X} {}^A\underline{B}_e^{C^X}|} \sin \left(\frac{-\gamma_{AB}}{2} \right) \end{aligned} \quad (2.45)$$

$$q_4 = \cos\left(\frac{-\gamma_{AB}}{2}\right) \quad (2.46)$$

The ‘X’ superscript in these equations is the skew-symmetric matrix used as the cross-product operator as defined in Appendix A. Using these new vectors for attitude measurements does not affect sensitivity in roll and pitch, as the vectors ${}^A\vec{B}^{D*}$ and ${}^A\vec{B}^{F*}$ span the same plane as the original three baselines.

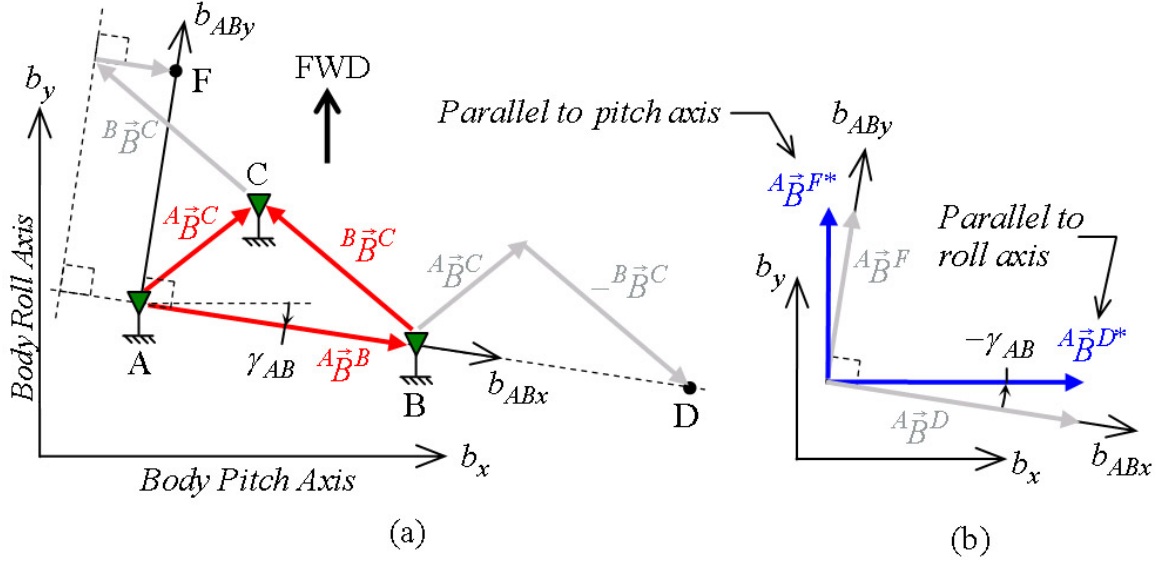


Figure 2.7: Derivation of Roll and Pitch Axes from Arbitrary Baselines

Finally, the vehicle’s three-axis attitude estimate can be computed from estimates of the rotated vectors. Using a yaw-pitch-roll sequence (from ENU to body basis), the vehicle’s three Euler angle estimates are:

$$\hat{\psi}(t_i) = \arctan 2 \left[\frac{\hat{y}_{eAD^*}(t_i)}{\hat{x}_{eAD^*}(t_i)} \right] \quad (2.47)$$

$$\hat{\theta}(t_i) = \arcsin \left[\frac{\hat{z}_{eAF^*}(t_i)}{R_{AF^*}} \right] \quad (2.48)$$

$$\hat{\phi}(t_i) = \arcsin \left[\frac{-\hat{z}_{eAD^*}(t_i)}{R_{AD^*} \cos(\hat{\theta}(t_i))} \right] \quad (2.49)$$

where

$$\begin{aligned} {}^A\hat{\underline{B}}_e^{D*} &= \begin{bmatrix} \hat{x}_{eAD^*} & \hat{y}_{eAD^*} & \hat{z}_{eAD^*} \end{bmatrix}^T \\ {}^A\hat{\underline{B}}_e^{F*} &= \begin{bmatrix} \hat{x}_{eAF^*} & \hat{y}_{eAF^*} & \hat{z}_{eAF^*} \end{bmatrix}^T \end{aligned}$$

2.3 Nonlinear Attitude Solution

As shown in Figure 2.3, only two unknown angles define the orientation of a baseline. However, Eq. 2.38 requires at least three channels to solve for the three Cartesian coordinates of the vector, as it does not utilize the baseline length constraint that fixes one vector component as a function of the other two. Thus, while the linear solution is simple and relatively inexpensive (numerically) to compute, it requires one more channel than unknowns in the attitude equation, and results in less accuracy than what is achievable from multiple channels. A set of equations that contains only two unknown quantities to solve for the baseline orientation is inherently nonlinear and more difficult to solve, but typically has a more accurate solution. For this reason, the nonlinear attitude solution may be preferred in some applications at the expense of slightly higher implementation complexity and computation time. The best solution can also be a combination of the linear and nonlinear methods, in which the linear solution is used as an accurate first guess to solve the nonlinear equation numerically, thus reducing the number of iterations needed to solve for the nonlinear attitude solution.

The nonlinear attitude equation is traditionally written to solve for Euler angles directly [13, 31]. To rewrite Eq. 2.10 as a nonlinear equation, the baseline vector is first expressed as a function of its orientation by using the rotation matrix from the baseline-fixed basis to the ENU basis:

$${}^A\mathbf{B}_e^B(t) = \mathbf{C}_{e=b_{AB}}(t) {}^A\mathbf{B}_{b_{AB}}^B \quad (2.50)$$

where

$${}^A\mathbf{B}_{b_{AB}}^B = \begin{bmatrix} R_{AB} & 0 & 0 \end{bmatrix}^T \quad (2.51)$$

$$\mathbf{C}_{e=b_{AB}}(t) = \begin{bmatrix} \mathbf{c}\psi_{AB}\mathbf{c}\phi_{AB} & -\mathbf{s}\psi_{AB} & \mathbf{c}\psi_{AB}\mathbf{s}\phi_{AB} \\ \mathbf{s}\psi_{AB}\mathbf{c}\phi_{AB} & \mathbf{c}\psi_{AB} & \mathbf{s}\psi_{AB}\mathbf{s}\phi_{AB} \\ -\mathbf{s}\phi_{AB} & 0 & \mathbf{c}\phi_{AB} \end{bmatrix} \quad (2.52)$$

Note, ‘s’ and ‘c’ are abbreviations for sine and cosine of angles in Eq. 2.52. Substituting Eq. 2.50 into Eq. 2.10 yields:

$$\underline{\Delta\Phi}_{AB}(t) = \underline{\mathbf{H}}_e(t) \mathbf{C}_{e=b_{AB}}(t) {}^A\mathbf{B}_{b_{AB}}^B + \underline{N}_{AB} + \underline{L}_{AB}(t) + \underline{E}_{\varphi AB}(t) \quad (2.53)$$

This matrix equation is now a nonlinear function of two unknown angles that define the baseline orientation. Using the previously obtained integer ambiguities and line-bias estimate, a cost function is formulated from Eq. 2.53 to quantify the value of the residual as a function of a guess

for baseline orientation:

$$J_{AB}(\tilde{\psi}_{AB}, \tilde{\phi}_{AB}) = \left\| \underline{\Delta\Phi}_{AB}(t_i) - \underline{\hat{N}}_{AB} - \underline{\hat{L}}_{AB}(t_i) - \underline{H}_e(t_i) \underline{C}_{e=b_{AB}}(\tilde{\psi}_{AB}, \tilde{\phi}_{AB}) {}^A\underline{B}_{b_{AB}}^B \right\|^2 \quad (2.54)$$

Finally, a search for the minimum of this cost function in the neighborhood of an initial guess for the baseline orientation results in the nonlinear attitude estimate:

$$(\hat{\psi}_{AB}, \hat{\phi}_{AB}) = \min_{(\tilde{\psi}_{AB}, \tilde{\phi}_{AB})} J_{AB}(\tilde{\psi}_{AB}, \tilde{\phi}_{AB}) \quad (2.55)$$

2.4 ADOP

The attitude dilution of precision (ADOP) is analogous to the better-known DOP quantities related to GPS positioning. In positioning, the DOPs are a measure of how the current configuration of the satellite constellation relative to the user affects positioning accuracy; the HDOP refers to horizontal positioning, and the VDOP to vertical positioning. Likewise, the ADOP is a measure of how the configuration of the satellite constellation relative to the user affects the accuracy of attitude measurements. Several different definitions of ADOP exist [36], and this section will focus on the definition presented in [70].

To give a simple and intuitive introduction to ADOP, this section begins with the one-dimensional example shown in Figure 2.8, in which all vectors shown lie in the plane of the page. The unknown attitude is the angle θ , which is given by the relationship

$$\begin{aligned} \theta &= El^k - \beta_{AB}^k \\ &= El^k - \cos^{-1} \left(\frac{r_{AB}^k}{R_{AB}} \right) \end{aligned} \quad (2.56)$$

The partial derivative of the attitude estimate with respect to the measurement (r_{AB}^k) is:

$$\frac{\partial \theta}{\partial r_{AB}^k} = \frac{-1}{R_{AB} \sin \beta_{AB}^k} \quad (2.57)$$

This equation can be interpreted as the sensitivity of the attitude estimate to errors in the measurement; that is, for a small change in the measurement, the change in the attitude is:

$$\delta \theta = \frac{-1}{R_{AB} \sin \beta_{AB}^k} \delta r_{AB}^k \quad (2.58)$$

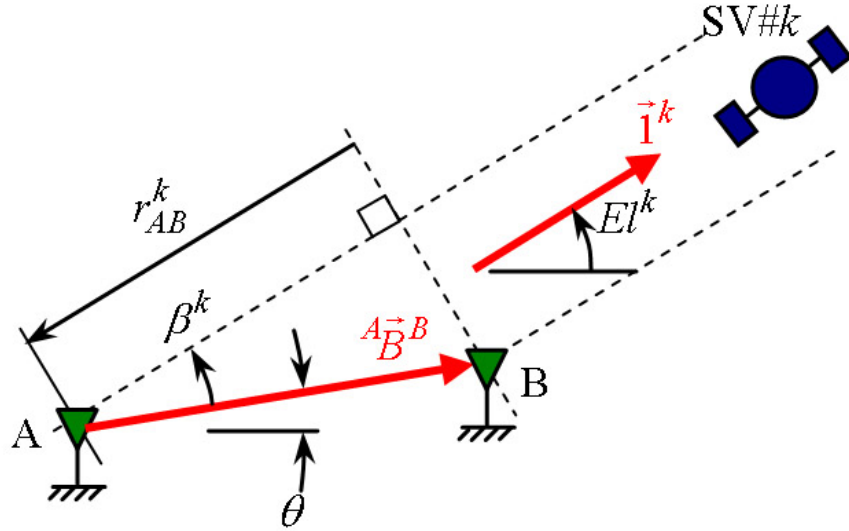


Figure 2.8: One-Dimensional Attitude for Visualization of ADOP

Or in terms of the variances,

$$\text{var}(\delta\theta) = \frac{1}{R_{AB}^2 \sin^2 \beta_{AB}^k} \text{var}(\delta r_{AB}^k) \quad (2.59)$$

As shown in this equation, the error sensitivity of the attitude measurement is inversely proportional to the baseline length and to the sine of the angle between the LOS and the baseline. When this angle is zero (i.e., when the SV and the baseline are collinear) the error sensitivity has a singularity and the attitude error diverges. The ADOP for this one-dimensional case is simply the relationship between the variances in Eq. 2.59. Written in terms of the LOS and baseline attitude, the ADOP is:

$$\theta DOP = \frac{1}{R_{AB}^2 \sin^2 (El^k - \theta)} \quad (2.60)$$

This simple example demonstrates how the ADOP depends on the baseline length and the relative orientation of the baseline to the LOS vectors.

A more general expression for the ADOP of a vehicle with multiple baselines and channels would need to relate the variance of carrier phase measurements to the variance of three vehicle Euler angles. To this end, the baseline vector in Eq. 2.10 can be expressed as a function of two rotation matrices, one from the estimated body frame to ENU (function of Euler angle estimates) and one from true to estimated body frames (function of small Euler angle perturbations):

$${}^A \underline{B}^B = {}_{e \hat{=} b} \underline{C}(\hat{\theta}, \hat{\phi}, \hat{\psi}) {}_{\hat{b} = b} \underline{C}(\delta\theta, \delta\phi, \delta\psi) {}^A \underline{B}_b^B \quad (2.61)$$

where

$${}_{e\hat{b}}\underline{C}(\hat{\theta}, \hat{\phi}, \hat{\psi}) = \begin{bmatrix} c\hat{\psi}c\hat{\phi} - s\hat{\psi}s\hat{\theta}s\hat{\phi} & -s\hat{\psi}c\hat{\theta} & c\hat{\psi}s\hat{\phi} + s\hat{\psi}s\hat{\theta}s\hat{\phi} \\ s\hat{\psi}c\hat{\phi} + c\hat{\psi}s\hat{\theta}s\hat{\phi} & c\hat{\psi}c\hat{\theta} & s\hat{\psi}s\hat{\phi} - c\hat{\psi}s\hat{\theta}s\hat{\phi} \\ -c\hat{\theta}s\hat{\phi} & s\hat{\theta} & c\hat{\theta}c\hat{\phi} \end{bmatrix} \quad (2.62)$$

$$\begin{aligned} {}_{\hat{b}}\underline{C}(\delta\theta, \delta\phi, \delta\psi) &\approx \begin{bmatrix} 1 & -\delta\psi & \delta\phi \\ \delta\psi & 1 & -\delta\theta \\ -\delta\phi & \delta\theta & 1 \end{bmatrix} \\ &\approx \underline{I}_{3 \times 3} + \underline{\delta A}^X \end{aligned} \quad (2.63a)$$

$$\underline{\delta A} = \begin{bmatrix} \delta\theta & \delta\phi & \delta\psi \end{bmatrix}^T \quad (2.63b)$$

Substitution of Eqs. 2.61 and 2.63a into Eq. 2.10, and rewriting the dICP noise as a perturbation to the true value yields:

$$\underline{\Delta\Phi}_{AB} + \delta\underline{\Delta\Phi}_{AB} = \underline{H}_e {}_{e\hat{b}}\underline{C}(\underline{I}_{3 \times 3} + \underline{\delta A}^X) {}^A\underline{B}_b^B + \underline{N}_{AB} + \underline{L}_{AB} \quad (2.64)$$

Using the fact that the true and estimated body frames are approximately equal, the nonlinear attitude equation can be extracted from this equation, leaving a relationship between the two perturbation terms:

$$\begin{aligned} \delta\underline{\Delta\Phi}_{AB} &= \underline{H}_e {}_{e\hat{b}}\underline{C} \underline{\delta A}^X {}^A\underline{B}_b^B \\ &= -\underline{H}_e {}_{e\hat{b}}\underline{C} {}^A\underline{B}_b^{AX} \underline{\delta A}_b \end{aligned} \quad (2.65)$$

This equation can be expanded to include an arbitrary number of baselines. In fact, to make the three Euler angles observable, two or more non-collinear baselines must be included. Using two of the baselines in Figure 2.7 (the third is redundant), the relationship between perturbations is:

$$\begin{aligned} \delta\underline{\Delta\Phi} = \begin{bmatrix} \delta\underline{\Delta\Phi}_{AB} \\ \delta\underline{\Delta\Phi}_{AC} \end{bmatrix} &= \begin{bmatrix} -\underline{H}_e {}_{e\hat{b}}\underline{C} {}^A\underline{B}_b^{BX} \\ -\underline{H}_e {}_{e\hat{b}}\underline{C} {}^A\underline{B}_b^{CX} \end{bmatrix} \underline{\delta A}_b \\ &= \underline{G} \underline{\delta A}_b \end{aligned} \quad (2.66)$$

Assuming that dICP measurements are uncorrelated, the covariance of the attitude perturbation vector can now be evaluated as:

$$\begin{aligned} \text{cov}(\underline{\delta A}_b) &= \left(\underline{G}^T \underline{G} \right)^{-1} \text{var}(\delta\underline{\Delta\Phi}) \\ &= \underline{F} \text{var}(\delta\underline{\Delta\Phi}) \end{aligned} \quad (2.67)$$

In this implementation, the matrix F is the 3x3 ADOP matrix, whose diagonal entries correspond to the variance of individual Euler angles. The corresponding ADOPs are:

$$\theta DOP = \underline{\underline{F}}(1,1) \text{ (Pitch DOP)} \quad (2.68a)$$

$$\phi DOP = \underline{\underline{F}}(2,2) \text{ (Roll DOP)} \quad (2.68b)$$

$$\psi DOP = \underline{\underline{F}}(3,3) \text{ (Yaw DOP)} \quad (2.68c)$$

If using the line-bias correction term, its DOP could also be obtained with a slightly different development that would result in a 4x4 ADOP matrix. This variation would use the geometry matrix in Eq. 2.15 instead of that in Eq. 2.11b, and have a fourth term in the attitude perturbation vector in Eq. 2.63b.

2.5 Phase-Delay Calibration

The discussions regarding phase measurements up to this point have assumed that the phase center of an antenna is at its physical geometric centroid, which is typically used in practice to define the endpoints of a baseline. The phase center refers to the point in space on which phase measurements appear to be centered. Phase delay is an error that is incurred through the use of imperfect GPS antennas, in which the phase center does not exactly coincide with the physical center of the antenna. The phase center is not even a constant for a given antenna, but varies depending on the incident angle of the signal with respect to the antenna [43, 63, 64]. These errors can be as large as 2cm even for survey-grade antennas [4], which would result in differential phase-delay errors of up to 4cm for an attitude system. Clearly, phase delays of this magnitude can be a primary source of error in systems with short baselines. Fortunately, these errors can be measured in a calibration process and the antenna phase map stored as a lookup table, such that the phase delay can be corrected in real time. Of course, the quantity of interest for an attitude system is the differential phase-delay between two antennas, but for brevity it is referred to as phase delay.

The error term in Eq. 2.6 can be expanded to include the phase-delay term and a broadband noise term:

$$\varepsilon_{\varphi AB}^k(\underline{1}_b^k, t) = \eta_{\varphi AB}(\underline{1}_b^k) + v_{\varphi AB}(t) \quad (2.69)$$

where

$$\begin{aligned} \eta_{\varphi AB}(\underline{1}_b^k) &= \text{Differential Phase-Delay between antennas A and B} \\ v_{\varphi AB}(t) &= \text{Broadband differential phase-noise} \end{aligned}$$

As indicated, the phase-delay term is a function of the LOS vector in the vehicle body basis (or another antenna-fixed basis), as the phase delay will depend only the orientation of the satellite relative to the antenna. This dependency is illustrated in Figure 2.9, which shows measured phase delay for a stationary baseline, and for one channel. The two data sets shown were taken two orbit periods apart, to ensure that the satellites had the same LOS vectors relative to the baseline. The repeatability of the phase delay illustrated in this figure was similar in all the other channels and with longer periods of time between data sets, suggesting that calibration would probably mitigate most of the error. The antennas used for this data collection were vehicle-tracking patch antennas made by Micropulse (32000 series).

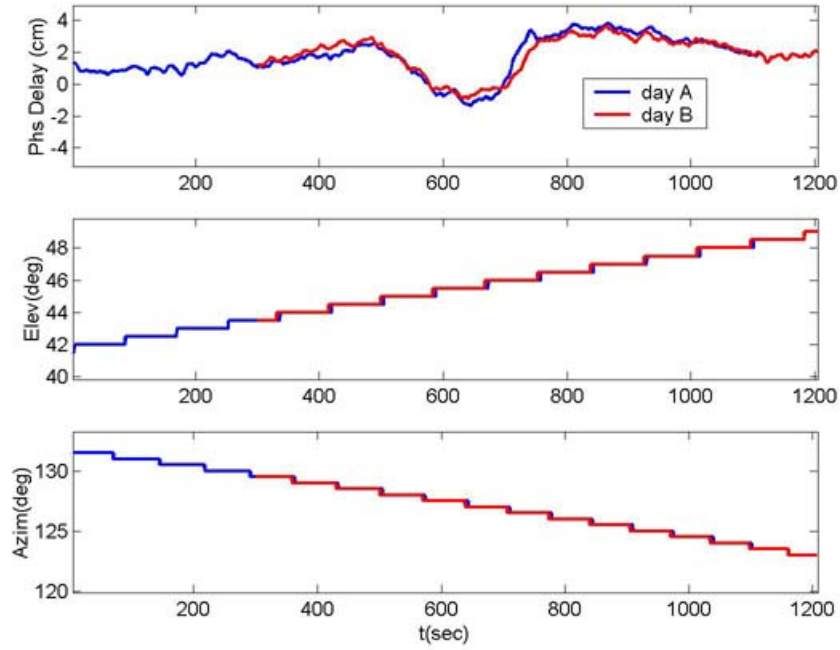


Figure 2.9: Phase-Delay Repeatability in One Channel

As shown in Figure 2.9, the phase delay can vary several centimeters with only a few degrees of change in the LOS. Thus, a fairly accurate attitude reference is necessary to measure the phase delay. If such an attitude reference is available, the phase delay can be measured as follows:

$$\hat{\eta}_{\varphi AB}(\underline{1}_b^k) = \underline{1}_e^{k^T}(t_i) \underline{C}_{e=b_{ref}}(t_i) \underline{A}_b^B - \Delta\varphi_{AB}^k(t_i) + n_{AB}^k + \hat{l}_{AB}(t_i) \quad (2.70a)$$

where

$$\underline{C}_{e=b_{ref}} = fcn(\theta_{ref}, \phi_{ref}, \psi_{ref}) \quad (2.70b)$$

If an attitude reference can be used that is independent of the GPS measurements themselves, then simply rotating the baseline can provide the geometric diversity needed to cover the entire

phase-map. If such a reference cannot be provided, as is usually the case for lower budget applications, then the yaw reference angle can be estimated by averaging measurements from a static baseline. The pitch and roll reference angles can be zero if the baseline is leveled, or measured with an inclinometer. If the baseline is forced to remain static for measuring the reference angles accurately, then the geometric diversity needed to cover the phase map must be provided by satellite motion, and by concatenating data taken with various different baseline orientations. This technique is obviously much more tedious as the LOS vectors change slowly over time, but the collection of more data allows time-averaging of measurements and improves the accuracy of the phase map.

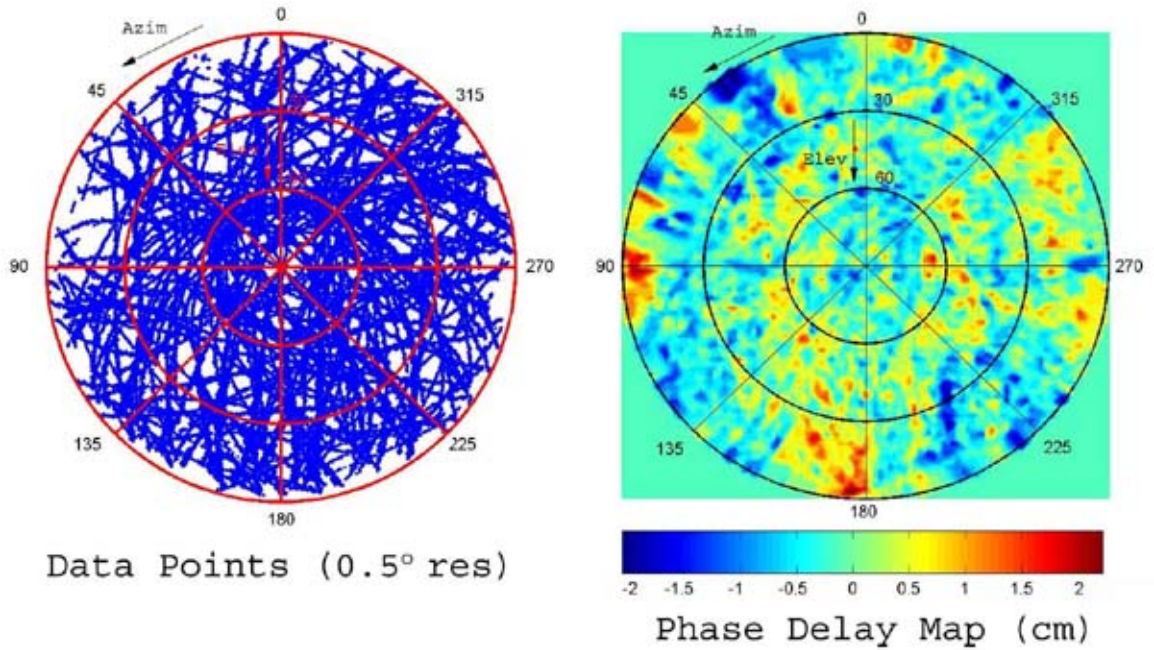


Figure 2.10: Example of a Phase-Delay Map

The static-baseline method described above was used to generate Figure 2.10, in which the left image shows the azimuth and elevation of the satellites (in the vehicle body frame) during multiple data-collection periods, and the right image is a contour plot of the phase-delay map. The data shown in the phase map has been smoothed and interpolated from the raw measurements computed per Eq. 2.70a.

In concurrence with Figure 2.9 and results obtained by other authors [42], the phase map shown in this figure indicates that phase-delay can vary significantly over small spans of azimuth and elevation. This result suggests that phase-delay effects would resemble uncorrelated noise for a rotating baseline, and could be mitigated by integration with an INS and proper filtering. For a static or slowly moving baseline (which does occur in a car), the phase-delay would cause more distinct time-correlated errors that could only be corrected through calibration. Such a case

is shown in Figure 2.11, for a three-antenna attitude system with 50cm baselines (equilateral triangle configuration). The plots were generated with the same platform attitude ($\theta \approx 0^\circ$, $\phi \approx 0^\circ$, $\theta \approx 35^\circ$) and exactly two orbit periods apart to ensure the same satellite geometry relative to the baselines.

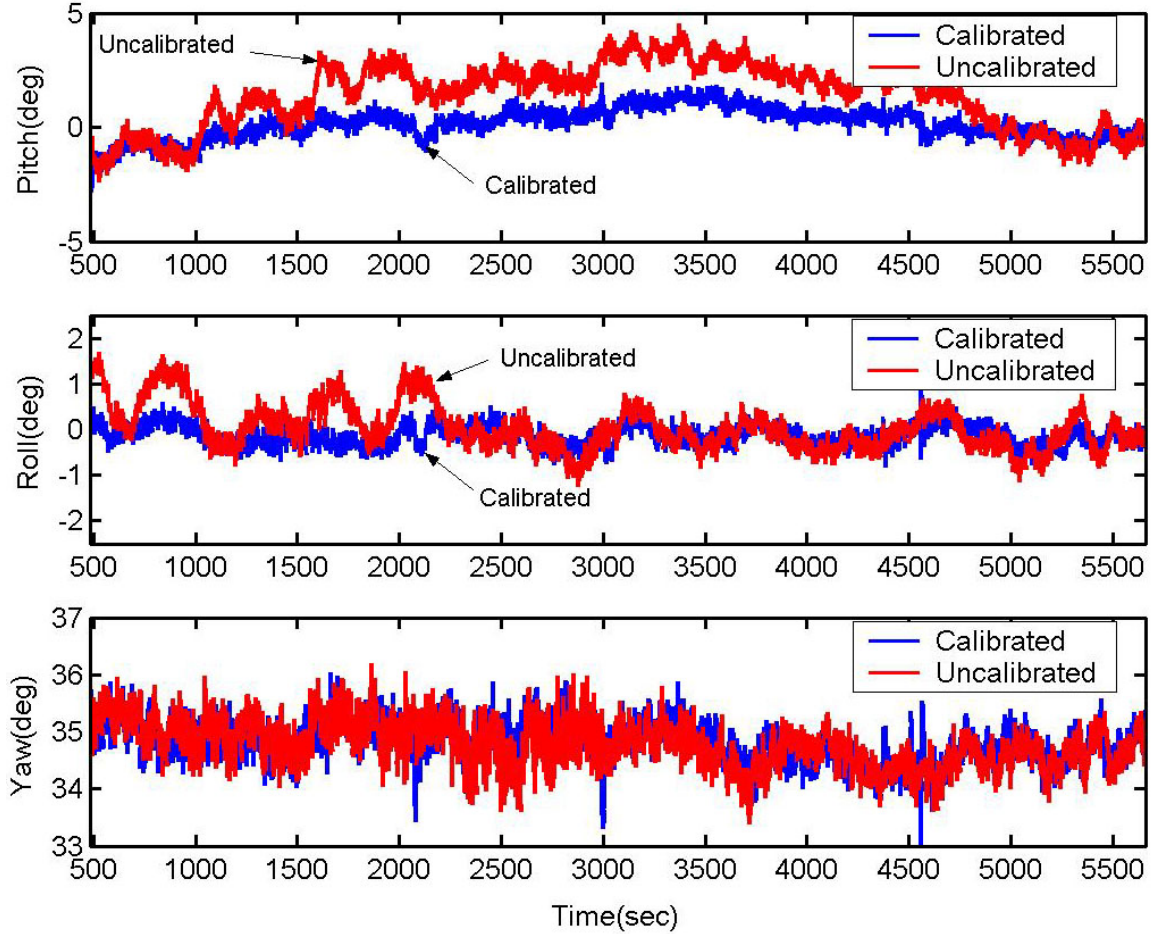


Figure 2.11: Attitude Measurements With and Without Phase-Delay Calibration

The benefit of calibration on the pitch and roll measurements is clear, but the effect on the yaw measurement is negligible. This result is primarily due to the method used for measuring the reference attitude during calibration. As described previously, the reference yaw angle was based on filtered measurements of the GPS yaw measurements, while the reference roll and pitch angles were set to zero with the aid of an inclinometer. While this method does not require external attitude estimates during calibration, it has the disadvantage of not fully estimating the phase delay for low elevation signals, which are not well observed with roll and pitch; this shortcoming is revealed by the remaining correlation of the yaw measurement noise between the two data sets.

2.6 One-Satellite Attitude Solution

This final section of Chapter 2 will discuss the details of one-satellite attitude determination with GPS. This technique, introduced in [2], refers to Euler-angle estimation based on a dICP measurement from a single baseline channel. One-satellite attitude determination is particularly useful in environments where GPS satellite visibility is limited and signal multipath may be severe, as multiple satellite solutions are often not realizable with too few uncorrupted dICP measurements. As discussed in Chapter 1, such is the case for automobiles in urban environments. When multiple measurements are available, one-satellite attitude estimates can be combined into a weighted average to form a composite estimate. In this implementation, individual dICP measurements are not dependent on each other to produce Euler-angle estimates, and can thus result in a more robust, but less accurate, estimate than the multiple-satellite solution. As will be discussed in Chapter 4, a combination of both solutions can be used to exploit their respective strengths in an attitude system for automobiles.

The primary goal of one-satellite attitude is to enhance the yaw determination capability of a GPS attitude system. Yaw is the most critical vehicle-orientation angle for lateral control applications and for inertial navigation with strap-down sensors. While knowing roll and pitch is also important for inertial navigation, these angles are often less ambiguous than yaw in a vehicle with small tilt, and can often be measured by other means, including inclinometers (in a non-accelerating vehicle) or accelerometers combined with GPS velocity measurements. Since the use of a single channel allows solving for only one unknown, the one-satellite attitude solution must assume a value for one of the orientation angles of a baseline, as defined in Figure 2.3. If small vehicle tilt can be assumed, then zero baseline roll is a reasonable assumption in solving for one-satellite yaw. Of course, having an estimate of the baseline roll can result in a more accurate one-satellite yaw measurement. Likewise, if an estimate of yaw is available, then the dICP measurement can also be used to compute a one-satellite baseline roll estimate. The discussion to follow will cover formulation of both one-satellite yaw and roll baseline-orientation angles, error equations, and implementation issues.

2.6.1 Formulation and Accuracy of One-Satellite Yaw

The one-satellite yaw is most conveniently derived relative to the LOS of the satellite being used. For this purpose, a basis is defined whose y-axis is lined up with the horizontal projection of the LOS vector. This basis is depicted in Figure 2.12 with its axes labeled s_x^k and s_y^k and is referred to as the s^k basis. The one-satellite yaw for baseline AB and SV# k is labeled ψ_{AB}^k and represents the rotation of the baseline-fixed basis with respect to the s^k basis.

With these definitions, and if the LOS vector is defined in the ENU frame as

$$\underline{1}_e^k = \begin{bmatrix} C_x^k & C_y^k & C_z^k \end{bmatrix}^T \quad (2.71)$$

then it is defined in the s^k basis as

$$\underline{1}_{s^k}^k = \begin{bmatrix} 0 & \sqrt{C_x^{k2} + C_y^{k2}} & C_z^k \end{bmatrix}^T \quad (2.72)$$

Assuming that the baseline roll angle is zero, the baseline vector can also be expressed in the s^k basis as

$$\underline{A\vec{B}}_{s^k}^B = \begin{bmatrix} R_{AB} \cos \psi_{AB}^k & R_{AB} \sin \psi_{AB}^k & 0 \end{bmatrix}^T \quad (2.73)$$

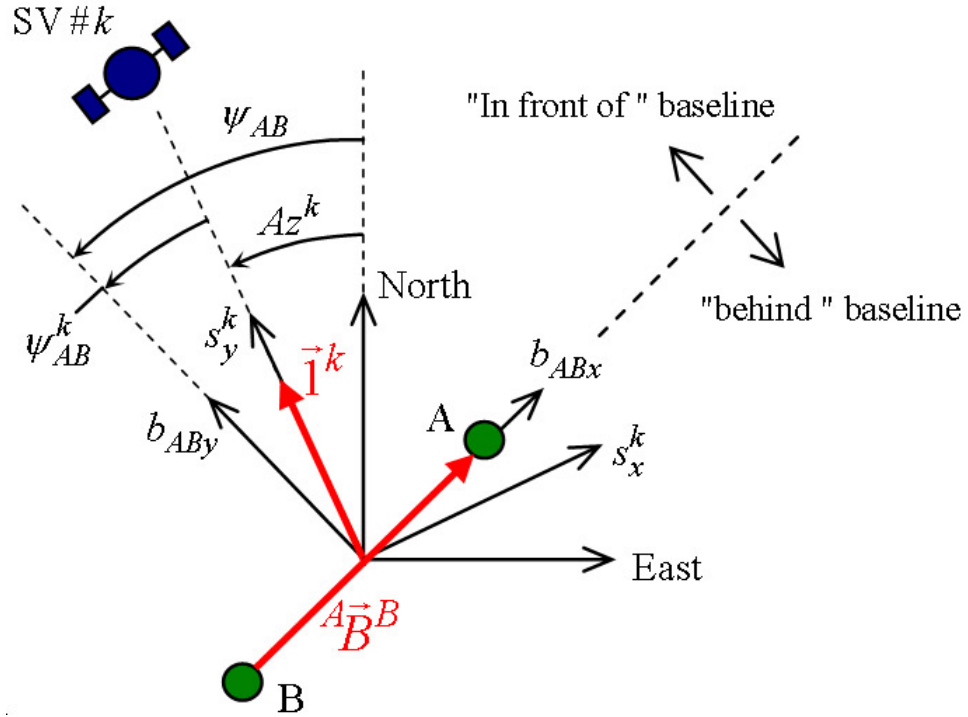


Figure 2.12: Vector Diagram for One-Sat Yaw

Now, substituting Eqs. 2.72 and 2.73 into Eq. 2.7 and the result into Eq. 2.6 results in an equation with only the one-satellite yaw as an unknown:

$$\Delta \varphi_{AB}^k(t) = R_{AB} \sqrt{C_x^{k2}(t) + C_y^{k2}(t)} \sin \psi_{AB}^k(t) + n_{AB}^k + l_{AB}(t) \quad (2.74)$$

This equation can be solved for the one-satellite yaw using previously determined values for the

integer ambiguity and line bias estimate:

$$\hat{\psi}_{AB}^k(t) = \arcsin \left(\frac{\Delta\varphi_{AB}^k(t) - \hat{n}_{AB}^k + \hat{l}_{AB}(t)}{R_{AB} \sqrt{C_x^{k2}(t) + C_y^{k2}(t)}} \right) \quad (2.75)$$

Since the possible values for yaw span 360° , Eq. 2.75 has a dual solution. Specifically, ψ_{AB}^k and $\pi - \psi_{AB}^k$ have the same sine, so it must be determined whether the satellite is in front of, or behind the baseline to obtain a unique solution. This information requires a very rough estimate of the baseline orientation and can be obtained in a number of ways, including an accompanying INS system or GPS velocity heading.

As expected from the concept of ADOP, the accuracy of the one-satellite yaw will depend on the direction of the LOS relative to the baseline. This dependency is partially indicated in Eq. 2.75, as the equation has a singularity when the satellite is directly overhead ($C_x^k = C_y^k = 0$) where the dICP measurement would be independent of the baseline yaw angle. As pointed out in Section 2.4, the equation is also ill conditioned when the satellite is collinear with the baseline. A more thorough analysis of the one-satellite ADOP is necessary to identify all configurations for which the one-satellite yaw is not well observed. As with the multiple-satellite ADOP, this analysis is done by applying a perturbation to the measurement and the solution, and extracting a relationship between the two perturbations. In this case, the perturbations are applied to $\Delta\varphi_{AB}^k$ and ψ_{AB}^k in Eq. 2.74:

$$\Delta\varphi_{AB}^k + \delta\Delta\varphi_{AB}^k = R_{AB} \sqrt{C_x^{k2} + C_y^{k2}} \sin(\psi_{AB}^k + \delta\psi_{AB}^k) + n_{AB}^k + l_{AB} \quad (2.76)$$

Applying a trigonometric transformation and assuming that the perturbation to the yaw angle is small, the sine term can be expressed as

$$\begin{aligned} \sin(\psi_{AB}^k + \delta\psi_{AB}^k) &= \sin\psi_{AB}^k \cos\delta\psi_{AB}^k + \cos\psi_{AB}^k \sin\delta\psi_{AB}^k \\ &\approx \sin\psi_{AB}^k + \delta\psi_{AB}^k \cos\psi_{AB}^k \end{aligned} \quad (2.77)$$

Substituting Eq. 2.77 into Eq. 2.76 and extracting the relationship between perturbations yields:

$$\begin{aligned} \delta\psi_{AB}^k &= \frac{\delta\Delta\varphi_{AB}^k}{R_{AB} \cos\psi_{AB}^k \sqrt{C_x^{k2} + C_y^{k2}}} \\ &= \frac{\delta\Delta\varphi_{AB}^k}{R_{AB} \cos\psi_{AB}^k \cos El^k} \end{aligned} \quad (2.78)$$

The relationship between the variances of the perturbations is the one-satellite yaw ADOP:

$$\text{One-Sat Yaw ADOP} = \frac{1}{R_{AB}^2 (\cos \psi_{AB}^k \cos El^k)^2} \quad (2.79)$$

This equation indicates that the accuracy of the one-satellite yaw breaks down near $\psi_{AB}^k = \pm 90^\circ$ and for high elevation satellites.

To illustrate the impact of the ADOP on the yaw estimate accuracy, Figure 2.13 shows plots of one-satellite yaw measured with a 50cm baseline, and the one-satellite ADOP normalized with unity baseline length. The multiple-satellite yaw estimate is also plotted for reference to better visualize the error in the one-satellite yaw. As shown, the yaw estimate from this one satellite is usable until the normalized ADOP is greater than about 10, after which accuracy degrades significantly. Note, the ADOP as described in Eq. 2.79 is being computed with the one-satellite yaw estimate, so its value also gets noisy when the yaw estimate gets noisy.

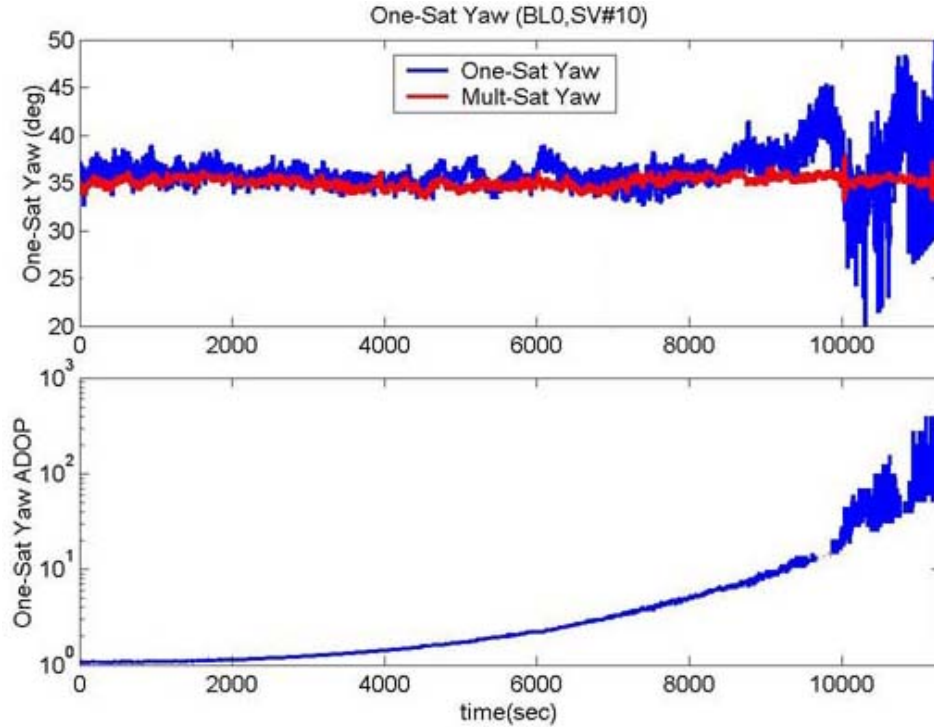


Figure 2.13: One-Satellite Yaw Accuracy and ADOP

The other factor that affects the accuracy of the one-satellite yaw is the magnitude of the baseline roll angle, as it was assumed to be zero. A deviation of this assumption introduces error into the yaw estimate, and the error increases with increasing elevation angle. This behavior can be anticipated by visualizing how high elevation satellites have a larger component of the dICP measurement associated with the baseline roll angle, and thus contribute more to the observability of roll than yaw. The opposite is true for low elevation satellites, which have a higher

component of their dICP measurement associated with yaw, and are thus better for measuring baseline yaw than roll.

The sensitivity of the one-satellite yaw to nonzero baseline-roll can be analyzed by modifying Eq. 2.74 to include effects of both baseline orientation angles. This variation is done simply by using an expression for the baseline vector with nonzero roll angle:

$$\underline{B}_{s^k}^{AB} = R_{AB} \begin{bmatrix} \cos \psi_{AB}^k \cos \phi_{AB} & \sin \psi_{AB}^k \cos \phi_{AB} & -\sin \phi_{AB} \end{bmatrix}^T \quad (2.80)$$

The use of this equation in the same process that results in Eq. 2.74 gives an expression that relates the dICP measurement to the baseline roll and yaw. Omitting the notation to indicate functions of time, this relationship is

$$\Delta \varphi_{AB}^k = R_{AB} \sqrt{C_x^{k2} + C_y^{k2}} \sin \psi_{AB}^k \cos \phi_{AB} - R_{AB} C_z^k \sin \phi_{AB} + n_{AB}^k + l_{AB} \quad (2.81)$$

Once again, perturbations are applied (analogous to Eq. 2.76) to the dICP measurement and the baseline orientation angles. Assuming a small roll angle, a relationship between perturbations in the dICP measurement and orientation angles is obtained:

$$\delta \psi_{AB}^k = \frac{1}{R_{AB} \cos \psi_{AB}^k \cos El^k} \delta \Delta \varphi_{AB}^k + \frac{\tan El^k}{\cos \psi_{AB}^k} \delta \phi_{AB} \quad (2.82)$$

The second term on the right-hand side describes the sensitivity of the one-satellite yaw estimate to nonzero baseline roll angles. The numerator is related to the elevation of the satellite, and confirms that low elevation satellites are better for measuring baseline yaw. As in the ADOP expression (Eq. 2.79), the denominator indicates that the yaw estimate error sensitivity is unbounded when $\psi_{AB}^k = \pm 90^\circ$. To illustrate the magnitude of errors due to nonzero baseline-roll, Figure 2.14 shows the sensitivity of the one-satellite yaw to nonzero roll angle for a fixed baseline-yaw ($\psi_{AB}^k = 0^\circ$) with varying SV elevation, and for a fixed SV elevation ($El^k = 45^\circ$) with varying baseline yaw. Note, the axes in this figure represent absolute values.

In automobile applications, using the one-satellite yaw as discussed thus far may have limited benefits in situations for which it is intended, as most of the satellites that are likely to be visible in an urban canyon tend to be at high elevations. However, the error in the one-satellite yaw due to nonzero baseline roll can be mitigated if an estimate of the baseline roll is available. If such is the case, then the effect of the roll angle can be accounted for, and the error of the one-satellite yaw angle would increase with the error in the roll estimate, rather than with the roll magnitude. Figure 2.14 would still be applicable in this case, but the x-axis would represent the magnitude of the *error* in the baseline roll estimate.

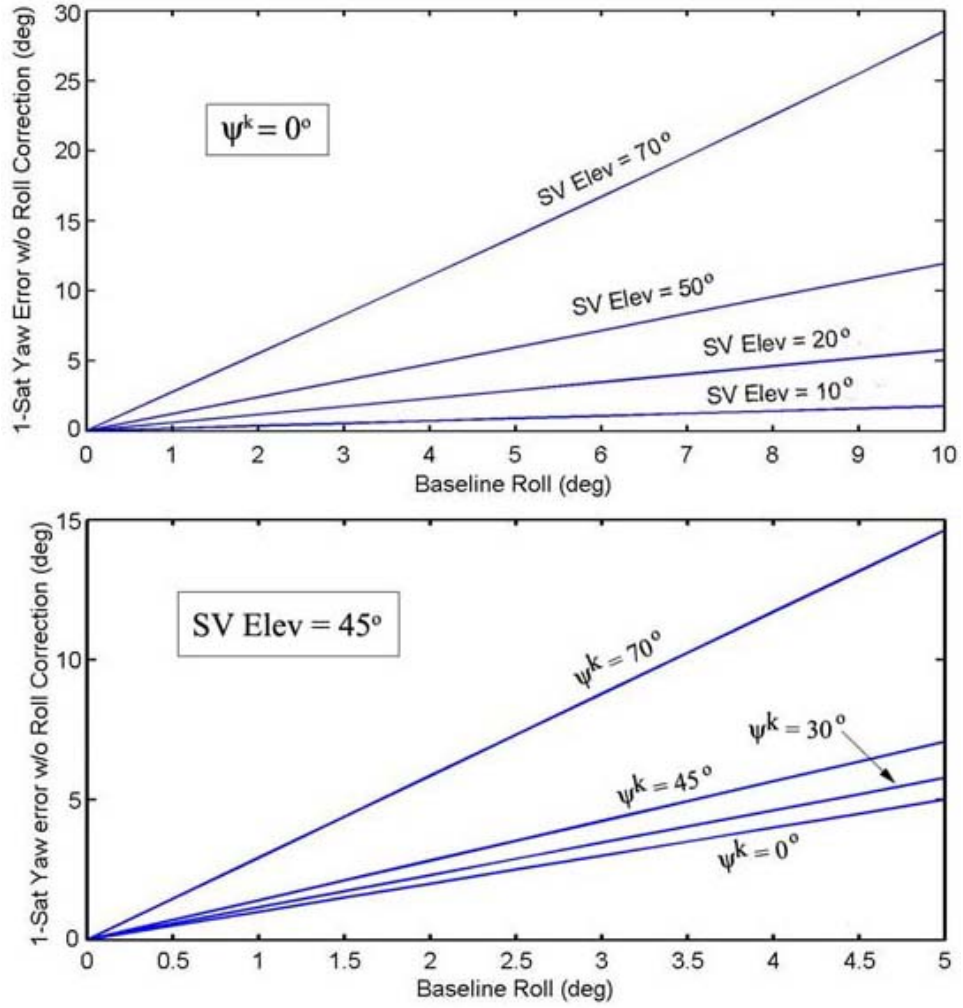


Figure 2.14: One-Sat Yaw Sensitivity to Nonzero Baseline-Roll Angle

The need to provide a baseline-roll estimate to the one-satellite yaw algorithm to improve its accuracy implies that this method would be best implemented with a secondary set of sensors that could provide such estimates. As will be discussed in Chapter 4, this external input to the one-satellite yaw algorithm can be realized with the filter output of an integrated GPS/INS attitude system. For now it can be assumed that a baseline-roll estimate is given, and a similar formulation to that of Eq. 2.78 (except using Eq. 2.80 instead of Eq. 2.73) results in an equation that uses external baseline-roll estimates to provide more accurate yaw measurements. This modification can be obtained by rearranging the terms in Eq. 2.81 and assuming small baseline roll. Denoting the roll angle as an external estimate, the modified expression for one-satellite yaw estimates is:

$$\hat{\psi}_{AB}^k(t) = \arcsin \left(\frac{(\Delta\varphi_{AB}^k(t) - \hat{n}_{AB}^k - \hat{l}_{AB}(t)) + R_{AB}C_z^k(t) \sin \hat{\phi}_{AB_{ext}}(t)}{R_{AB} \sqrt{C_x^k(t) + C_y^k(t)} \cos \hat{\phi}_{AB_{ext}}(t)} \right) \quad (2.83)$$

The terms that include the external roll estimate in this equation give an insight into the qualitative

behavior of the error that would be induced by an inaccurate estimate. More specifically, a positive error in the external roll estimate would result in a positive error in the one-satellite yaw if the satellite was in front of the baseline, and in a negative error if the satellite was behind the baseline (where the one-satellite yaw would be $\pi - \hat{\psi}_{AB}^k$ radians). This characteristic can actually be beneficial, since the errors with opposite sign from a satellite in front of the baseline and one behind would tend to cancel each other out when averaged.

2.6.2 Formulation and Accuracy of One-Satellite Roll

The previous section concluded with a discussion on improving the accuracy of one-satellite yaw estimates by using external estimates of the baseline roll angle. It was also mentioned that these estimates could be obtained from a coupled INS system, which would include gyroscopes to measure vehicle pitch and roll rates. When using automotive-grade gyros, calibration with GPS attitude measurements must be executed as often as possible, as the integrated noise from the gyros results in a random walk that can degrade attitude estimates up to a few degrees per minute. For this reason, it would be advantageous to be able to also compute baseline-roll estimates from a single satellite, to maximize the availability of GPS roll measurements during periods of poor GPS-satellite visibility.

In developing an expression for one-satellite baseline roll, no value of yaw can be assumed. Thus, an external estimate of the baseline yaw angle must be used to leave only one unknown to be solved from a single equation. Once again, this estimate could be provided by the coupled INS system that will be discussed in Chapter 4. The one-satellite baseline-roll angle can be expressed from Eq. 2.81 simply by rearranging terms, assuming a small baseline-roll angle, and using the yaw angle as a given quantity. The result of this process is the following expression for the one-satellite baseline-roll estimate:

$$\hat{\phi}_{AB}^k(t) \approx \frac{R_{AB} \sqrt{C_x^k(t) + C_y^k(t)} \sin \hat{\psi}_{AB_{ext}}^k(t) - (\Delta \varphi_{AB}^k(t) - \hat{n}_{AB}^k - \hat{l}_{AB}(t))}{R_{AB} C_z^k(t)} \quad (2.84)$$

One of the properties of the one-satellite baseline-roll ADOP is immediately obvious. It is clear that this equation has a singularity for satellites with zero elevation ($C_z^k = 0$), confirming the fact that high elevation satellites are best for measuring baseline roll. Another important characteristic is that the value of the one-satellite yaw will also affect the precision of the one-satellite roll estimate. The useful equation derived previously that relates perturbations in the measurement to perturbations in the baseline orientation angles (Eq. 2.82) can be rearranged to emphasize the sensitivity of one-satellite baseline-roll estimates to dICP measurements and

external yaw estimates:

$$\delta\phi_{AB}^k = \frac{1}{R_{AB} \sin El^k} \delta\Delta\varphi_{AB}^k + \frac{\cos\psi_{AB}^k}{\tan El^k} \delta\psi_{AB}^k \quad (2.85)$$

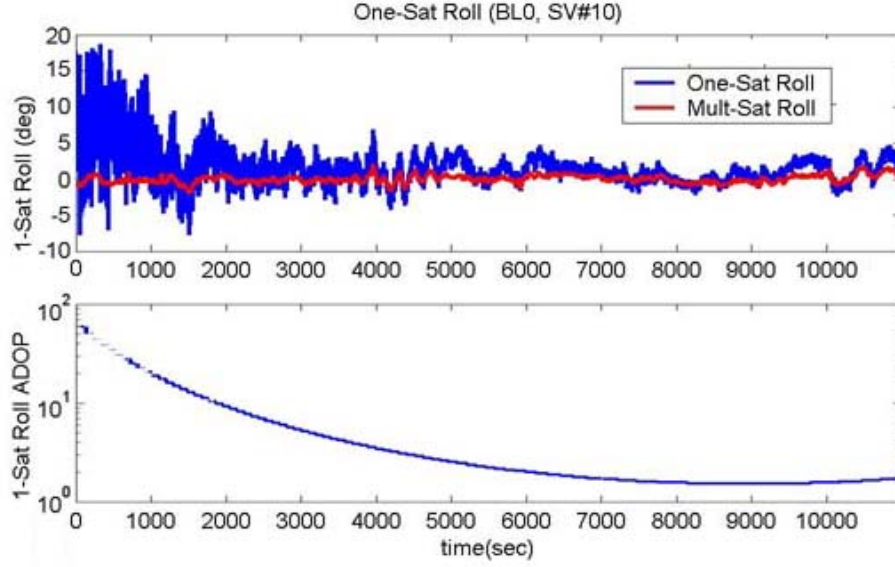


Figure 2.15: One-Satellite Roll Accuracy and ADOP

The one-satellite baseline-roll ADOP is evident from the first term on the right-hand side of Eq. 2.85, which relates errors in the dICP measurements to errors in the roll estimate:

$$\text{One-Sat Roll ADOP} = \frac{1}{R_{AB}^2 (\sin El^k)^2} \quad (2.86)$$

The second term relates error in the one-satellite baseline yaw estimate to errors in the one-satellite roll, and indicates that the best satellites for measuring one-satellite roll are at high elevations and for higher values of ψ_{AB}^k . Naturally, this relationship is the inverse of the one-satellite yaw sensitivity to errors in baseline-roll estimates (See Eq. 2.82). The effect of the one-satellite roll ADOP on the accuracy of the roll estimate is shown in Figure 2.15. The multiple-satellite baseline-roll angle is shown for comparison, and the ADOP is normalized to unity baseline length.

To finalize the comparison between errors in one-satellite yaw and roll, Figure 2.16 shows the sensitivity of the one-satellite roll to errors in baseline-yaw estimates, for a fixed baseline yaw ($\psi_{AB}^k = 0^\circ$) with varying SV elevation, and for a fixed SV elevation ($El^k = 45^\circ$) with varying baseline yaw. While similar to Figure 2.14, the important difference between these two figures is that satellites that are poor for measuring one-satellite yaw are good for measuring one-satellite

roll (and vice versa), so all satellites in view can usually be used to measure at least one baseline orientation angle.

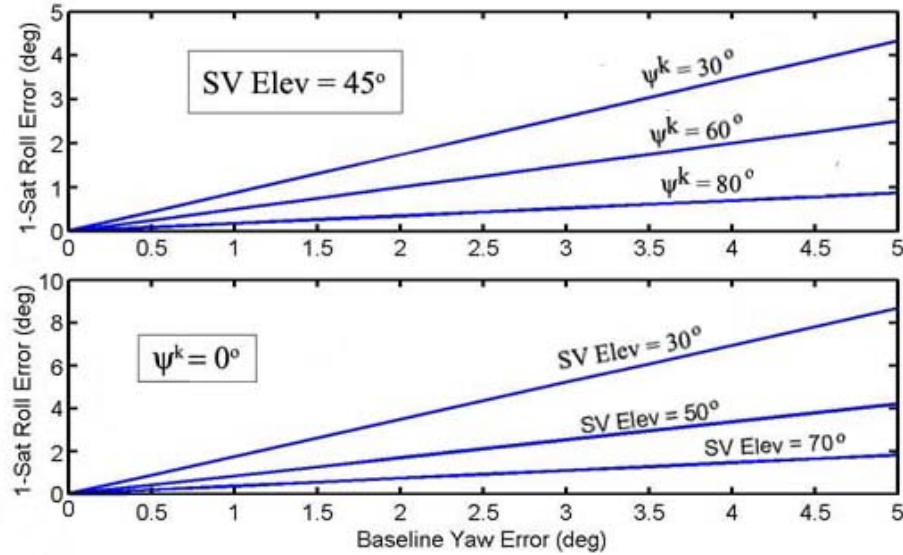


Figure 2.16: One-Satellite Roll Sensitivity to Errors in Yaw Estimate

The interdependence between one-satellite roll and yaw that has been discussed in the last two sections may lead to the supposition that an iterative process between Eqs. 2.84 and 2.83 could be used to converge to a solution for both baseline orientation angles from a single dICP measurement. While this process may converge to *possible* values of baseline roll and yaw, the idea is erroneous because a single measurement cannot be used to find a unique solution for two unknowns.

2.6.3 Integer Determination for One-Satellite Attitude

The nature of the solution for one-satellite attitude requires a different strategy for integer searches. The essential difference is that for one-satellite attitude, integer searches must be conducted without dependence on other channel measurements, to allow for a single satellite in view to be useful and to remain decoupled from other channels which may be experiencing phase noise. This difficulty precludes the use of baseline-length and a compound residual measurement as error checks on integers, so a baseline orientation estimate must come from an external source to resolve the integer ambiguity. This necessity comes from the fact that the residuals are always zero if computed with a one-measurement iterative scheme that covers the space of possible yaw angles, analogous to that described for the multiple-satellite case. Without this type of residual test, a cost function cannot be defined to discriminate correct integers from incorrect ones. Consequently, the ability to achieve single-channel integer-ambiguity determination depends on

having an external attitude estimate, and reveals yet another benefit of having a coupled INS or some other attitude reference.

As done in the last section, it will be assumed that an external baseline-yaw estimate is available from a coupled INS subsystem. Assuming small baseline roll, the one-satellite integer can be computed as follows:

$$\hat{n}_{AB}^k = \text{round} \left(\Delta\varphi_{AB}^k - \hat{l}_{AB} + R_{AB} \sqrt{C_x^{k2} + C_y^{k2}} \sin(\hat{\psi}_{AB_{ext}}^k) \right) \quad (2.87)$$

For short baselines, this integer does not change in the neighborhood of the original baseline-yaw estimate because of the rounding operation. Thereby, the integer-determination process may consist of checking only one integer, even with an error of a few degrees in the external yaw estimate. The residual of the equation can be computed by using the external estimate, and is equivalent to comparing the resulting one-satellite yaw with the external estimate:

$$res_{AB}^k = \Delta\varphi_{AB}^k - \hat{l}_{AB} - \hat{n}_{AB}^k - R_{AB} \sqrt{C_x^{k2} + C_y^{k2}} \sin(\hat{\psi}_{AB_{ext}}^k) \quad (2.88)$$

Of course, nonzero baseline-roll angles will have a component in this residual, but if an external roll-angle estimate is also available, it can be used to refine the equation in a similar way in which Eq. 2.83 was obtained. Therefore, the use of a maximum residual value for accepting an integer must consider the expected accuracy of the external baseline yaw and roll estimates, the baseline length, and the satellite position.

2.6.4 One-Satellite Measurement Averaging

The one-satellite attitude solutions are not only useful when a single satellite is in view, but can also be valuable when multiple measurements are available. In many situations where phase noise is present in one or more channels, the multiple-satellite solution (linear or nonlinear) experiences difficulty maintaining integrity, as all channels are dependent on each other to obtain an attitude solution and solve for integer ambiguities. In contrast, the one-satellite solutions can provide an alternative attitude estimate that is obtained from all available channels independently of each other. With this scheme, a simple weighted average of the one-satellite attitude estimates can be used, in which noisy channels do not affect the attitude estimates and integer-ambiguity searches from good channels. While less accurate than the multiple satellite solution, the improved robustness of this strategy makes it a very useful option when few satellites are in view, and/or several channels experience significant phase-noise.

Recall that the nonlinear attitude solution for a baseline only contains two unknowns, and can theoretically be resolved with two measurements. For the special case where two measurements

are available, this option is an alternative to using the average of the two one-satellite solutions. However, the nonlinear solution requires iteration, and is a more complex implementation than averaging the two one-satellite attitude estimates. If the nonlinear attitude solution is used with as few as two satellites, the accuracy of the solution will be largely dependent on the position of the two satellites relative to the baseline; as such, the ADOP of each satellite with respect to each degree of freedom may have to be considered when defining the iteration limits of the nonlinear solution. For simplicity, the attitude system that will be discussed in Chapters 4-5 uses the weighted average of one-satellite solutions for each degree of freedom rather than the nonlinear attitude solution in such cases.

As discussed previously, the ADOPs have a large impact on the accuracy of the one-satellite attitude estimates. The significant difference in accuracy that can be achieved with different ADOPs implies that when averaging attitude estimates from more than one channel, each channel should be weighed according to its expected accuracy. Since the one-satellite ADOPs are related to the error variance of the baseline orientation angles, a logical choice for a weighting function may be the square root of the inverse of the ADOP expressions. With this weighing function, the error contribution from each channel to the composite attitude estimate would be roughly equal for a given baseline, assuming small errors in the external attitude estimates (see Eqs. 2.83 and 2.84). One can also choose to account for the sensitivity of the one-satellite attitude estimate to the external baseline-orientation estimate, which is not dependent on baseline length. In this case, it is useful to define the ratio of the variances of external attitude estimates to the variance of dICP measurements:

$$r_{\phi_{AB}} = \frac{\text{var}(\delta \hat{\phi}_{AB_{ext}})}{\text{var}(\delta \Delta \varphi_{AB})} \quad (2.89)$$

$$r_{\psi_{AB}} = \frac{\text{var}(\delta \hat{\psi}_{AB_{ext}})}{\text{var}(\delta \Delta \varphi_{AB})} \quad (2.90)$$

These ratios may not remain constant over time, but values can be chosen based on typical dICP measurements and from the covariance matrix of the navigation filter which provides attitude estimates. Using these relationships, the one-satellite ADOPs could be rewritten from Eqs. 2.82 and 2.85 to include the sensitivity to errors in the external attitude estimates. Taking the square root of the inverse of these ADOP expressions would result in the following weighing functions:

$$W_{\phi_{AB}}^k = \sqrt{\frac{R_{AB}^2 \sin^2 El^k}{1 + R_{AB}^2 r_{\psi_{AB}} \cos^2 El^k \cos^2 \psi_{AB}^k}} \quad (2.91)$$

$$W_{\psi_{AB}}^k = \sqrt{\frac{R_{AB}^2 \cos^2 El^k \cos^2 \psi_{AB}^k}{1 + R_{AB}^2 r_{\phi_{AB}} \sin^2 El^k}} \quad (2.92)$$

The baseline-orientation estimates would be computed from N available channels as follows:

$$\hat{\psi}_{AB} = \frac{1}{\sum_{k=1}^N W_{\psi_{AB}}^k} \sum_{k=1}^N W_{\psi_{AB}}^k (\hat{\psi}_{AB}^k + A z^k) \quad (2.93)$$

$$\hat{\phi}_{AB} = \frac{1}{\sum_{k=1}^N W_{\phi_{AB}}^k} \sum_{k=1}^N W_{\phi_{AB}}^k \hat{\phi}_{AB}^k \quad (2.94)$$

Once this process is completed for multiple baselines on a vehicle, the vehicle attitude can be determined as described in Section 2.2.4. With many satellites in view, the average one-satellite attitude is comparable in accuracy and noise statistics with the multiple-satellite solution, as shown in Figure 2.17. These measurements were taken with a set of three 50cm baselines in an equilateral triangle configuration, and were generated without phase-delay calibration to illustrate how even the phase-delay errors affect the two solutions similarly.

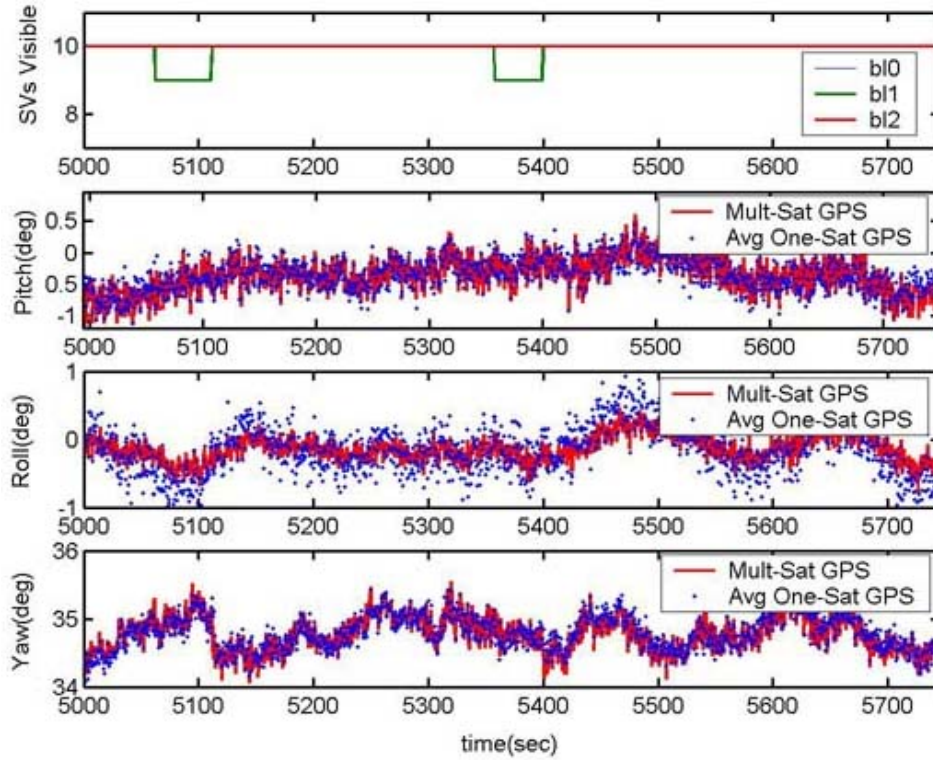


Figure 2.17: Comparison of Mult.-Satellite and Avg. One-Satellite GPS Attitude Measurements

2.7 Baseline Configuration Design

The baseline positions and lengths in an attitude system must be chosen to suit a particular application. The measurement accuracy needed in each attitude degree-of-freedom should be

the primary concern in choosing the baseline lengths. As shown in the various expressions for ADOP in this chapter, the pointing accuracy of the system is proportional to baseline length. It was also shown that phase-delay calibration could be implemented to improve the accuracy of a short-baseline system, but it would require a tedious process to obtain accurate measurements of the entire phase-delay map. The decision whether or not to use phase-delay calibration could be influenced by the use of a coupled inertial subsystem in a highly dynamic platform, as the use of filtering will mitigate the uncalibrated phase-delay errors if they have correlation times on the order of a few seconds or less.

Unfortunately, the accuracy criterion is not the only one to consider when choosing baseline lengths. It was mentioned briefly in Section 2.2.3 that increasing the baseline length arbitrarily could jeopardize the robustness of integer searches in the presence of phase noise. For this reason, the intended operating environment of the attitude system should also be considered before increasing baseline lengths beyond that which is needed to meet accuracy specifications. In the case of automobiles, it should be expected that strong multipath signals could introduce significant phase errors. Thus, an automobile attitude system should be as robust as possible to avoid erroneous integer solutions, if an integer search happens to coincide with momentary high phase-noise on a channel. In other words, the baselines should be of the minimum length allowed by accuracy requirements.

The subsections to follow present a simple design guideline for choosing baseline lengths based on required attitude accuracy and integer-determination robustness. Since these guidelines are developed with the use of various simplifying assumptions, they should be used as a starting point for choosing baseline lengths; based on empirical results, these lengths should then be adjusted to meet specifications more closely.

2.7.1 Attitude Accuracy Performance Envelope

The attitude performance envelope refers to the theoretical maximum accuracy of a GPS attitude system. The concept was introduced by Cohen [13], where a relationship between pointing accuracy, tracking-loop bandwidth, signal-to-noise ratio, and baseline length is given. Various simplifications are implied in the simple closed form for this relationship, including perfect synchronization between RF front ends, applicability to a one-dimensional problem (one angle to solve with one equation), and a LOS vector orthogonal to the baseline vector. The process by which this expression is obtained will be reviewed here, as it is not specified in [13] and the result is slightly different.

For a commonly used Costas phase-discriminator, the variance of carrier-phase measurement noise is given in [44] as the sum of the clock-error variances (local clock and satellite clock) and

the variance induced by thermal (or interference) noise. Remaining consistent with the notation used in Eq. 2.2, this relationship is:

$$\text{var}(\delta\varphi_A^k) = f_{L1}^2 \text{var}(\delta t_A + \delta t^k) + \text{var}(\varepsilon_{\varphi A}^k) (\text{cycles})^2 \quad (2.95)$$

The variance of the thermal noise component is also given in [44]:

$$\text{var}(\varepsilon_{\varphi A}^k) = \frac{B_{n\varphi}}{(S/N_0)^k} \left[1 + \frac{1}{2T_{av}(S/N_0)^k} \right] (\text{rad})^2 \quad (2.96)$$

where

$$\begin{aligned} B_{n\varphi} &= \text{Noise-equivalent bandwidth of PLL (Hz)} \\ (S/N_0)^k &= \text{Signal-to-noise ratio for SV\#k} \\ T_{av} &= \text{Pre-detection integration time (1-20ms)} \end{aligned}$$

Typical values of $B_{n\varphi}$ are in the neighborhood of 15Hz [19], and S/N_0 is typically between 35dB-Hz (ratio of 3160) for low-elevation satellites and 45dB-Hz (ratio of 31600) for high-elevation satellites. Thus, depending on the values used, the second term between the brackets in Eq. 2.96 may be ignored, as it is in [13]. However, in cases of high loop bandwidth, low SNR, and small averaging time, this term may be larger than unity and should not be ignored.

The dICP measurement is the difference of two measurements which have equivalent (but uncorrelated) noise terms. With the use of a common reference oscillator, the single-difference of CP measurements from two antennas cancels out the majority of the phase errors coming from the satellite and local clock, as long as they are tracked by the PLLs. However, if different frequency synthesizers are used in the two front ends, there may be some notable remaining noise in the dICP measurement that can be attributed to clock jitter. This case is applicable to the attitude system prototype discussed later in this thesis. Thus, the variance of the dICP measurement between two antennas can be expressed as two times the expression in Eq. 2.96 to account for the addition of two random variables, plus an additive term to account for the use of different frequency synthesizers in the RF front ends.

To relate these measurement errors to errors in attitude, the problem is simplified by reducing it to a planar scenario, as described in Section 2.4 to introduce the concept of ADOP. Furthermore, the denominator of Eq. 2.59 is maximized by assuming that the baseline is perpendicular to the LOS. Noting that the perturbation in the delta-range measurement in Eq. 2.59 is equivalent to an error in the dICP measurement, and combining this equation with Eq. 2.96 yields an

expression for the attitude performance envelope of baseline AB :

$$\text{var}(\delta\theta_{AB}) = \text{var}\left(\varepsilon_{\varphi AB}^k\right) \left(\frac{\lambda_{L1}}{2\pi R_{AB}}\right)^2 (\text{rad})^2 \quad (2.97)$$

where

$$\text{var}\left(\varepsilon_{\varphi AB}^k\right) = \frac{2B_{n\varphi}}{(S/N_0)^k} \left[1 + \frac{1}{2T_{av}(S/N_0)^k}\right] + \sigma_{clk}^2 (\text{rad})^2 \quad (2.98)$$

λ_{L1} is the wavelength of the L1 carrier (19cm) and σ_{clk}^2 is the additive term to account for the use of separate frequency synthesizers. Figure 2.18 shows various plots of this equation, indicating the effects from varying bandwidth, SNR, and baseline length. The predetection averaging time in this figure is fixed to 1ms (minimum for a PLL based on length of PRN code). σ_{clk}^2 is chosen as 0.02 rad^2 , which is a conservative estimate equal to twice the value of the phase-noise specification in [62], for the design of a second-order PLL based only on local oscillator characteristics. Note, effects from multipath and uncalibrated phase-delay are not considered in this analysis.

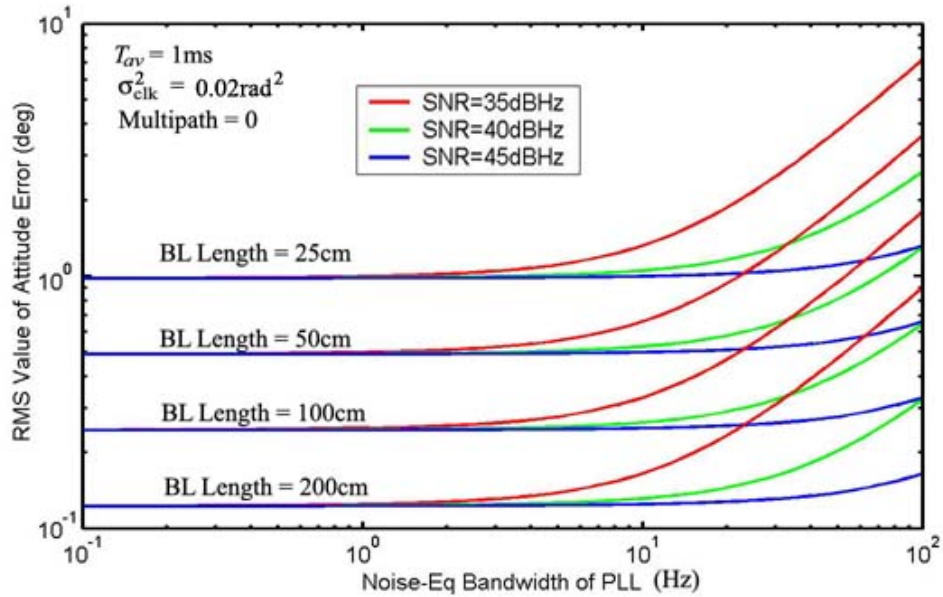


Figure 2.18: Attitude Accuracy Performance Envelope

2.7.2 Attitude Robustness Performance Envelope

Quantification of the robustness of an attitude system also requires some simplification of the problem. In this case, a planar problem is also considered by analyzing the robustness of the one-satellite integer ambiguity resolution specified in Eq. 2.87, with $\sqrt{C_x^2 + C_y^2} = 1$. The robustness of this equation is quantified as the probability that it will round to the wrong integer, which is equivalent to the probability that the argument to the rounding operation has more than

half a cycle of error:

$$P(\text{wrong integer}) = P\left(\left|\left(\Delta\varphi_{AB}^k - \hat{l}_{AB} - R_{AB} \cos \hat{\beta}_{AB_{ext}}^k\right) - n_{AB}^k\right| > \frac{1}{2} \text{ cycle}\right) \quad (2.99)$$

where

$$\begin{aligned} \hat{\beta}_{AB_{ext}}^k &= \text{External estimate of angle between baseline and LOS vectors} \\ &= \frac{\pi}{2} - \hat{\psi}_{AB_{ext}}^k \quad (\text{for a level baseline and zero elevation satellite}) \end{aligned} \quad (2.100)$$

This error can be introduced into the dICP measurement by thermal noise, RF interference, multipath, or it may come from inaccuracies in the external attitude estimate. Error in the line-bias estimate is also possible, but will be neglected in this analysis. The contribution from thermal noise is the same as in Eq. 2.97. The effect from multipath should also be included, as it can contribute up 1/4 of a cycle of phase distortion, assuming that the reflected ray amplitude is equal to or less than that of the direct ray [19].

To simplify Eq. 2.99, the true integer ambiguity cancels out all terms but the noise in the dICP measurement, and the external attitude estimate is written as its true value plus a perturbation ($\beta_{AB_{ext}}^k + \delta\beta_{AB_{ext}}^k$). The effect from maximum multipath can be included simply by changing the right-hand side of the equation to 1/4 cycle. After implementing these simplifications and assuming a small value for the external attitude estimate perturbation, the robustness equation becomes:

$$P(\text{wrong integer}) = P\left(\left|\varepsilon_{\varphi AB}^k + \delta\beta_{AB_{ext}}^k R_{AB} \sin \beta_{AB}^k\right| > \frac{1}{4} \text{ cycle}\right) \quad (2.101)$$

Equation 2.101 indicates that the probability of a getting a wrong integer during an integer-search process will depend on the amount of phase noise in the channel and the accuracy of the external attitude estimate. More importantly, the latter is scaled by the length of the baseline and the sine of the angle between the baseline and LOS vectors. Qualitatively, this description implies that the probability of rounding to the wrong integer will increase with increasing baseline length, unless the baseline and LOS vectors are collinear. Figure 2.19 shows values of this probability for a PLL with fixed bandwidth of 15Hz and a channel with 35dB-Hz SNR (for a low-elevation satellite), and assumes Gaussian distributions for $\varepsilon_{\varphi AB}^k$ and $\delta\beta_{AB_{ext}}^k$. It should be noted that this figure assumes the maximum possible multipath in the channel, which makes it applicable to situations where significant multipath is expected. Otherwise, values obtained from this graph for channels free of multipath would be very conservative.

Equation 2.101 can also be used to assess the reliability of integer searches with multiple satellites. In this case, the robustness of the integer search would also be analyzed one channel at a time, and the accuracy of the external attitude estimate would be analogous to the GPS attitude

accuracy achievable with the other channels, including effects of phase-noise and ADOP.

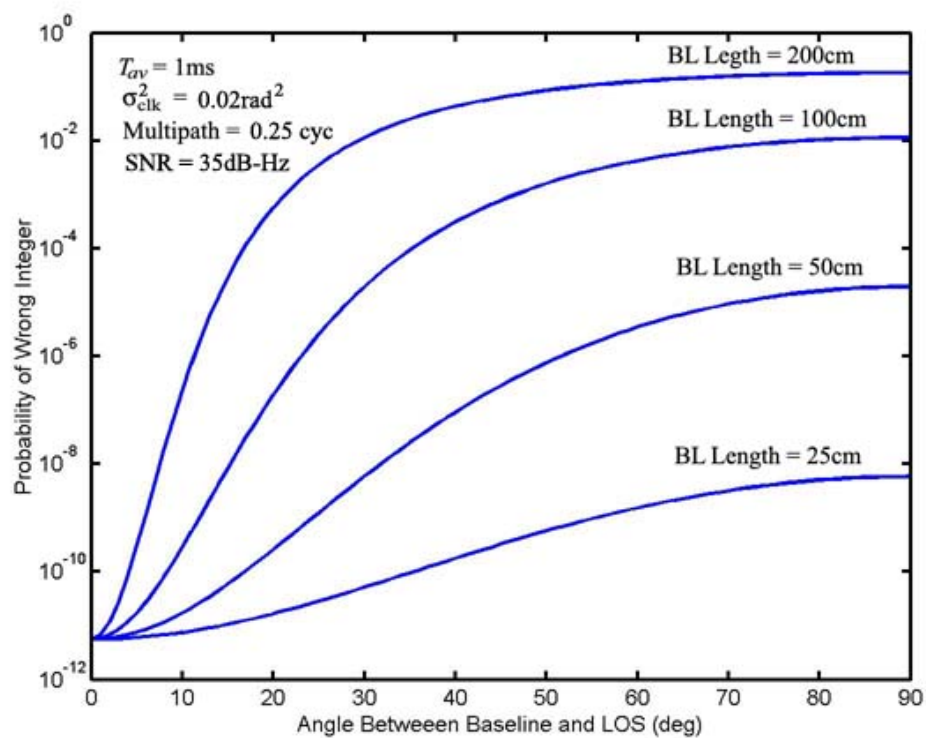


Figure 2.19: Attitude Robustness Performance Envelope

Chapter 3

System Components of GPS/INS Attitude System Prototype

This chapter begins a comprehensive description of an attitude system prototype for automobiles, based on the GPS attitude algorithms discussed Chapter 2 and tight integration with inertial sensors. As a precursor to the more high level system details covered in Chapter 4, the focus of this chapter involves the hardware and software devices that enable a realization of this system. The discussion begins with an assessment of adequate baseline configurations for a car and continues with details of inexpensive component selection, including GPS receivers, inertial sensors, and computer hardware. Cost was an important criterion for hardware selection, as this prototype is intended to be representative of the type of inexpensive equipment that could be used in large-scale production. A section on interface software covers the low level data structures that were implemented to handle the delayed serial-port data from the GPS receivers and nearly real-time data from the inertial sensors. Proper handling of this information as it is input into the application is a key low-level function that allows use of time-synchronized GPS and inertial sensor data in the GPS/INS filters. As time synchronization of sensor data is a fundamental feature of the system and not a necessarily a trivial detail, the last section of this chapter is dedicated to the method that was used in this prototype to apply GPS time tags to inertial sensor data.

3.1 Baseline Configuration

The configuration of the baselines on an automobile is the first system-design parameter to consider. The baseline lengths must be chosen to give sufficient accuracy for automobile control applications, and still allow the system to operate with sufficient robustness (with low probability of obtaining wrong integers). It is expected that the low satellite visibility and high multipath in urban environments will strain the system robustness, so a reasonable starting point would be to choose the minimum baseline lengths that meet accuracy specifications based on the accuracy performance envelope discussed in Section 2.7.1. The accuracy specification was chosen as

1σ of 1° , based on the achievable heading control accuracy of a tractor for precision farming [53]. Since this accuracy is desired regardless of channel SNR, Figure 2.18 suggests that a 50cm baseline is adequate for this application, with PLL bandwidths up to about 20Hz. The robustness of a 50cm baseline is also reasonable based on Figure 2.19, which shows that the probability of rounding off to a wrong integer is about 2×10^{-5} for a low elevation satellite with significant multipath.

Mounting constraints of GPS antennas on a vehicle can also be a design criterion for baseline configuration. The most suitable location for GPS antennas on a car is on the roof of the vehicle, probably below a plastic surface flush with the contour of the metal frame. The width of a typical consumer vehicle is on the order of 1.0-1.3 meters and the length of the roof is usually over 1 meter, which allows for ample space for mounting multiple 50cm baselines.

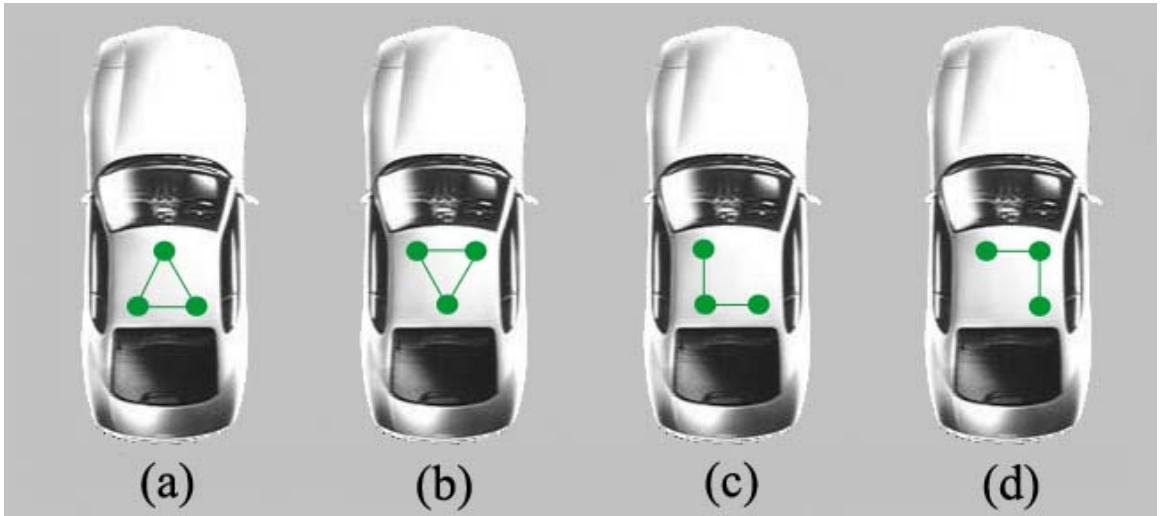


Figure 3.1: Automobile Baseline Configurations With Three Antennas

The last consideration when placing baselines on a car addresses the issue of one-satellite ADOP. As explained in section 2.6.1, satellites in front of and behind the baseline are best for measuring yaw. A car in an urban canyon is more likely to see satellites in front of and behind the car than to either side, so the primary baseline for measuring yaw should be placed perpendicular to the length of the car. This way, the few satellites that may be visible in an urban canyon are more likely to have good one-satellite yaw ADOP. This baseline will also measure the vehicle's roll angle directly. A third antenna can be added to enable vehicle pitch measurements and to provide baseline redundancy. In addition, the use of multiple, nonparallel baselines will improve GPS one-satellite attitude availability, as satellites that have poor ADOP for one baseline may have good ADOP for another baseline. Figure 3.1 shows several baseline configurations that are possible. The two equilateral-triangle configurations (a and b) are equivalent and the two right-triangle configurations (c and d) are equivalent. Note, all line-segments drawn represent

baselines of the same length. The symmetry of the equilateral configurations can make some aspects of the software easier to implement, but the right-triangle configurations would result in slightly more accurate pitch and yaw measurements. This difference is due to the fact that configurations c and d have longer baseline components parallel to the roll axis of the vehicle and have a longer diagonal baseline.

Due to easier construction and vehicle mounting, the configuration shown in Figure 3.1a was chosen (with 50cm baselines) for the automobile attitude system prototype. The baselines were assembled on an aluminum structure that was designed to mount onto a removable roof rack. The assembly is shown in Figure 3.2, mounted on a test vehicle.

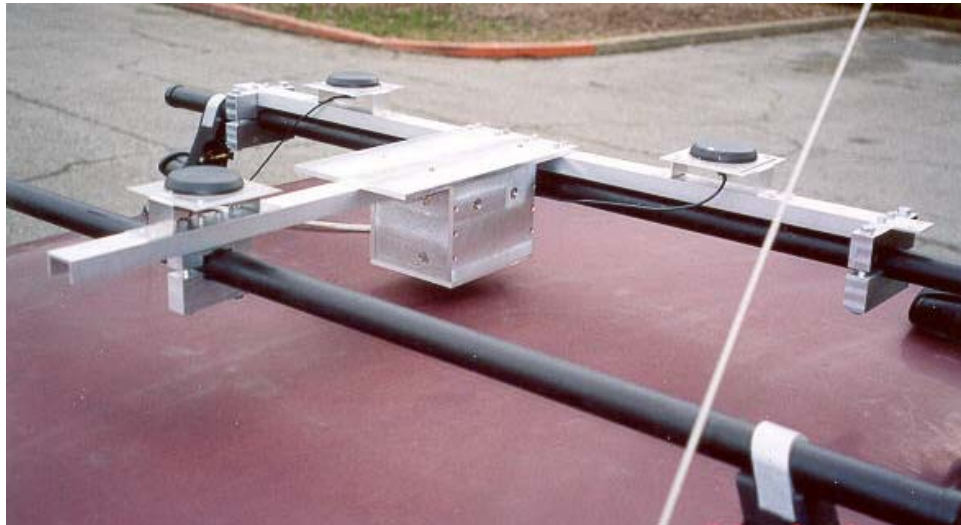


Figure 3.2: Baseline Assembly on Test Vehicle

3.2 GPS Receivers and Antennas

The GPS receivers used in the prototype were Canadian Marconi (now Novatel) Allstar OEM boards. These boards were chosen based on their low cost, versatility, and ease of interaction with a custom computer program. Furthermore, these boards have been used by researchers in several implementations of low-cost attitude systems [18, 43, 51, 71].

As of November 2003, these Novatel Allstar boards cost about \$500 for 1Hz PVT outputs and 10Hz carrier phase outputs. The boards use the Zarlink GP2010 (RF Front End and ADC) and GP2021 (12-channel correlator) chipsets, which are available individually for about \$10 and \$18, respectively, if purchased in large quantities [72, 73]. Including the passive components, clock, and a dedicated microprocessor to interact with the correlator chip, the total parts cost for a single receiver would probably not exceed \$60, making it a relatively low-cost component in a consumer vehicle.

The only modification performed to use these receivers in an attitude system was using a single reference oscillator for all receivers in the system. For only three receivers, this alteration involves removal of the clock from two of the boards and replacing it with parallel connections to the clock output in the unmodified board. The oscillator used in these receivers is a Rakon 4080 TCXO, which can provide sufficient current to drive the clock inputs to roughly four boards. To minimize noise in the clock lines, the parallel clock connections should be implemented with a coaxial cable or a tightly-wound twisted pair that connects ground and clock inputs in all receivers, with wires kept as short as possible.

A more professional implementation of a multiple-receiver common-clock setup may utilize a common *local* oscillator, which provides intermediate frequencies for down conversion of the GPS signal and ADC triggers [13]. The local oscillator in an RF front end derives these frequencies from the reference oscillator. The use of a common local oscillator provides more stable phase-offsets between receivers as a function of temperature changes, which is perceived in the software as line-bias drift. However, this technique is more complicated than simply using a common reference oscillator, in that it requires customized design of the RF front ends, and cannot be implemented with inexpensive “off-the-shelf” components. For this reason, the prototype system discussed here does not have this deeper level of receiver integration, but a commercial product based on customized chips for attitude determination most likely would.

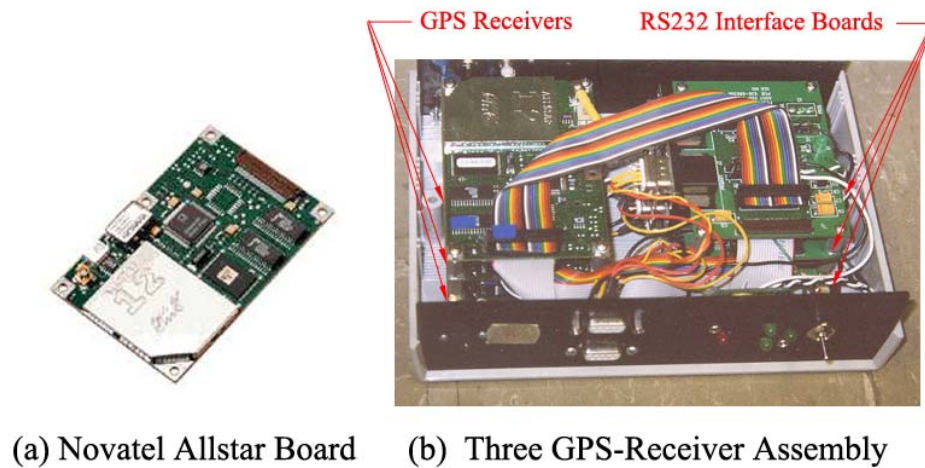


Figure 3.3: GPS Receiver Assembly for Attitude Determination

The Allstar boards use a serial communications protocol that is compatible with a common computer serial port, but the line voltage levels need external conversion to RS232 specifications. This conversion can be implemented easily with a very simple circuit based on any number of inexpensive IC chips available for that purpose, including the popular line of Maxim RS232 line drivers/receivers (MAX231, MAX232, etc).

Figure 3.3 shows images of an Allstar board, and an assembly of three boards used in the attitude-system prototype.

GPS antennas are available in a wide range of quality and cost. The antennas that would most likely be used in an inexpensive automobile system are simple patch antennas, which are also used in the prototype. The antennas used were 26dB active (26dB of gain), vehicle-tracking Micropulse antennas (32000 series), but any other model of patch antenna with comparable specifications is expected to perform similarly.

3.3 Inertial Sensors

Inertial sensors typically refer to gyroscopes for measuring angular rates, and accelerometers for measuring specific force (gravity plus platform accelerations). A complete IMU contains three sets of each, such that three-axis attitude and three-dimensional acceleration can be measured. Many different kinds of gyroscopes and accelerometers can be used in an INS, and are usually classified as one of four types depending on their long-term stability: navigation, tactical, automotive, and consumer.

A sensor's long-term stability and noise can be quantified with the Allan variance of the device's output. The Allan variance is defined as the variance of a number of averaged subsets of data of equal length. The time constant (τ) specifies the length of the averaged data subsets, and the number of subsets used to compute the variance ranges from as few as two to the number of samples in the complete data set. The Allan variance is a very useful method for characterizing a random process, as both long-term stability and broadband noise characteristics of a signal can be ascertained from a graph of Allan variance vs τ .

Table 3.1: Characteristics of Different Types of Inertial Sensors

Sensor Quality	Gyroscopes				Accelerometers			
	Stability		Noise std (deg/sec)	Cost per axis US\$	Stability		Noise std (g)	Cost per axis US\$
	τ (sec)	σ_{av} (deg/hr)			τ (sec)	σ_{av} (g)		
Navigation	3600	0.003	1.3×10^{-4}	10k+	3600	25×10^{-6}	5×10^{-6}	10k+
Tactical	100	0.35	0.0017	1k+	60	50×10^{-6}	50×10^{-5}	1k+
Automotive	100	180	0.05	100+	100	120×10^{-3}	10×10^{-3}	100+
Consumer	100	360	0.05	40+	100	240×10^{-3}	10×10^{-3}	5+

The different grades of inertial sensors are summarized in Table 3.1, which includes measures of Allan variance (σ_{av}^2) at the specified time constant, wide-band noise, and cost range. These

details were obtained from [31] and [32], and a more detailed analysis on Allan variance and the stability of the various types of inertial sensors can be found in these citations.

Modern cars are usually equipped with inexpensive automotive-grade inertial sensors, which are used in various control and safety subsystems. These sensors do not have the necessary long-term stability for prolonged dead-reckoning navigation, and are therefore not used by themselves for this purpose. Integrated with GPS, however, automotive-grade sensors can be used for navigation, as sensor biases can be calibrated whenever GPS is available, and these biases can be assumed to remain constant during short GPS outages (on the order of 100 seconds). Considering the significant cost difference between automotive and tactical or navigation sensors, a combination of GPS with inexpensive inertial sensors is a much more practical alternative for consumer automobile applications than high-grade inertial navigation systems.

While a full INS navigation system needs accelerometers and gyroscopes, an integrated *attitude* system needs only gyroscopes as long as a secondary subsystem (GPS in this case) can provide three-axis attitude measurements. With GPS velocity, accelerometers can also be used to provide roll and pitch information to a navigation filter, though this technique is not implemented in this attitude system prototype. Nevertheless, accelerometers are included in the IMU package and their outputs are stored for post-processing navigation outputs.

Table 3.2: Bosch DRS-MM1 Specifications

	Gyroscope	Accelerometer
Measurement Range	$\pm 100^\circ/\text{sec}$	$\pm 1.8g$
Scale Factor	$18\text{mV}/^\circ/\text{sec}$	$1000\text{mV}/g$
Scale Factor Error	$\leq 5\%$	$\leq 5\%$
Offset Error	$\leq 5^\circ/\text{sec}$	$\leq 0.06g$
Max Offset Error Rate	$\pm 0.2^\circ/\text{sec} / \text{min}$	$\pm 0.03g/\text{min}$
Dynamic Response	$\geq 30\text{Hz}$	$\geq 30\text{Hz}$
Electrical Noise	$\leq 5\text{mV rms}$	$\leq 5\text{mV rms}$
root Allan Variance	$\sigma_{av} = 36\text{deg/hr } \tau = 100\text{sec}$	$\sigma_{av} = 2 \times 10^{-4}g \tau = 100\text{sec}$

The inertial sensors used in the attitude system prototype are Bosch DRS-MM1 Yaw-Rate Sensors. Some of the important specifications for these sensors are included in Table 3.2. More detailed information can be found in [57]. These MEMS devices consist of one accelerometer and one gyroscope, and are packaged in a rugged enclosure for use in automobiles. The three-axis IMU was constructed from three of these units, mounted orthogonally to each other inside an aluminum box that mounts near the center of the GPS antenna triangle. The image in Figure 3.2 shows the IMU box hanging below the aluminum frame. Figure 3.4 shows an image of an individual sensor and more detail of the IMU assembly.

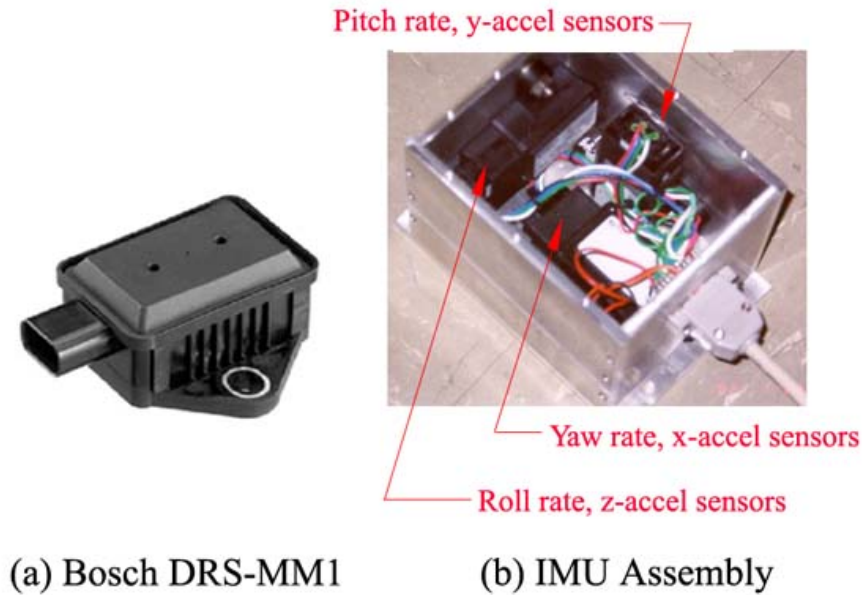


Figure 3.4: Inertial Sensors and IMU Assembly

3.4 Computer Hardware

3.4.1 Analog To Digital Converter

The outputs of the inertial sensors are analog, and they must be sampled with an ADC that runs in real time and in parallel with the GPS attitude-determination software on the same computer. The ADC selected for this purpose was a Quatech DAQP-16 PCMCIA device, which has eight differential ADC channels that are sampled with 16 bits and eight digital I/O lines for control or monitoring (one of which was utilized in the timing circuit discussed in Section 3.6). This device also has the important feature of having optional external triggering for the sampling clock, which allows precise time-synchronization with a pulse-per-second signal from a GPS receiver. The ADC can be initialized and operated with user-defined functions written in C, allowing the INS software and GPS attitude-determination software to run in the same program. The card is also hardware-interrupt driven, which precludes the need to run a separate software thread for the IMU or perform time-consuming polling for the next IMU output. Once the ADC is initialized, the samples are triggered from an external source synchronized to GPS time and processed with an interrupt-service routine which uses very little CPU time.

3.4.2 Serial-Port Expansion Card

Each Allstar GPS receiver communicates with a computer through a standard serial port. To accommodate concurrent use of the three of these GPS receivers, a minimum of three serial ports

are needed, which is more than most computers have. Therefore, a serial-port expansion card was necessary to provide enough serial ports for assembly of the attitude-system prototype. For this purpose, a Quatech QSP-100 PCMCIA card was selected, which provides four additional serial ports.

3.4.3 Computer

The computer used in the prototype system is a Dell Inspiron 5000 laptop with a 500MHz processor. A laptop was used to facilitate portability of the testing equipment. The only other requirements for the computer were that it have two PCMCIA slots for the ADC and serial-port cards, and sufficient CPU speed to handle all aspects of the attitude system; that is, process GPS attitude determination code at 10Hz (including full integer ambiguity searches), service the ADC interrupt routine at 100Hz, read data from three serial ports at 19200 baud without losing packets due to buffer overflows, and display a graphical user interface for monitoring program status and vehicle attitude. Furthermore, this performance was necessary without the use of a real-time operating system (such as QNX), as development and testing of the software was done with a Windows 2000 operating system. Through testing, it was found that this computer was sufficient to run all the processes simultaneously, though some optimization of the graphics and serial-port access was necessary.



Figure 3.5: Attitude System Prototype Equipment

The decision to use Windows was made on the facts that inexpensive programming resources are well developed for Windows (including graphics programming), and that most GPS and computer-hardware manufacturers provide drivers and interface software for Windows, allowing for a variety of choices when selecting the previously-mentioned hardware. The disadvantage of

using Windows is that applications do not have exclusive use of the CPU, and running processes are constantly interrupted (if for a short time) by other background processes. This behavior makes the implementation of a true real-time system difficult, as the time delay for measurements to be available by the software is unpredictable. However, for the sole purpose of prototype development where no other subsystem needs to use attitude measurements in real time, time delays associated with CPU interruptions are negligible. For applications that go beyond this prototype development, the applicable portable software for this system has also been used successfully with QNX and Linux operating systems.

Figure 3.5 shows a photograph of all the equipment used to construct the attitude system prototype, and illustrates its portability. Note, the picture does not include the DC to AC power inverter and adapters needed to supply power to the computer, GPS receivers, and inertial sensors.

3.5 Interface Software

This section gives an overview of some important aspects of the software that are related to interfacing with hardware and with a user. The software hierarchy is the first topic covered, and specifies the overall architecture of the attitude system application. This discussion includes the subdivision of functions, which are organized in such a way that exportable code can be easily extracted and used in other systems which may utilize different hardware and have a different user interface. The section on GPS interface software details the use of special data structures used to receive and match GPS data from different serial buffers and inertial data from the ADC. The objective of the serial-port interface software is to provide a means for storing and matching GPS data as it is read from multiple serial ports, allowing for time-lags caused by operating-system interruptions and accumulation of data in serial buffers. The IMU interface software ensures that IMU data is properly time-tagged, used to obtain a real-time attitude estimate, and stored until it has been used by the GPS/INS filter. The user-interface section contains a short description of the graphical interface used in the development and testing of this attitude-system prototype.

3.5.1 Software Hierarchy

The attitude-determination software and all of the hardware interface functions are programmed in C and are executed from a single custom Windows application. However, the software is written in such a way that system-specific code is separate from exportable code, and exchange of data between these sections takes place through a single global data structure.

System-specific code includes control software for the ADC, and functions that utilize the Windows Application Programmers Interface (API), such as control of serial ports and the graphical user interface (GUI). Operating-system (OS) specific and ADC-specific software are grouped together, since any ADC used will require drivers written for a specific OS. Exportable code is software that can be copied without modification to a project compiled under another operating system and that uses different hardware. Functions under this category include the GPS attitude-determination and filtering (GPS/INS integration) functions. A smaller subset of the software can be classified as semi-exportable, as it can be copied to a project that uses the same GPS receivers but on a different operating system. The functions used to decode the various serial messages from the GPS receivers are included in this category.

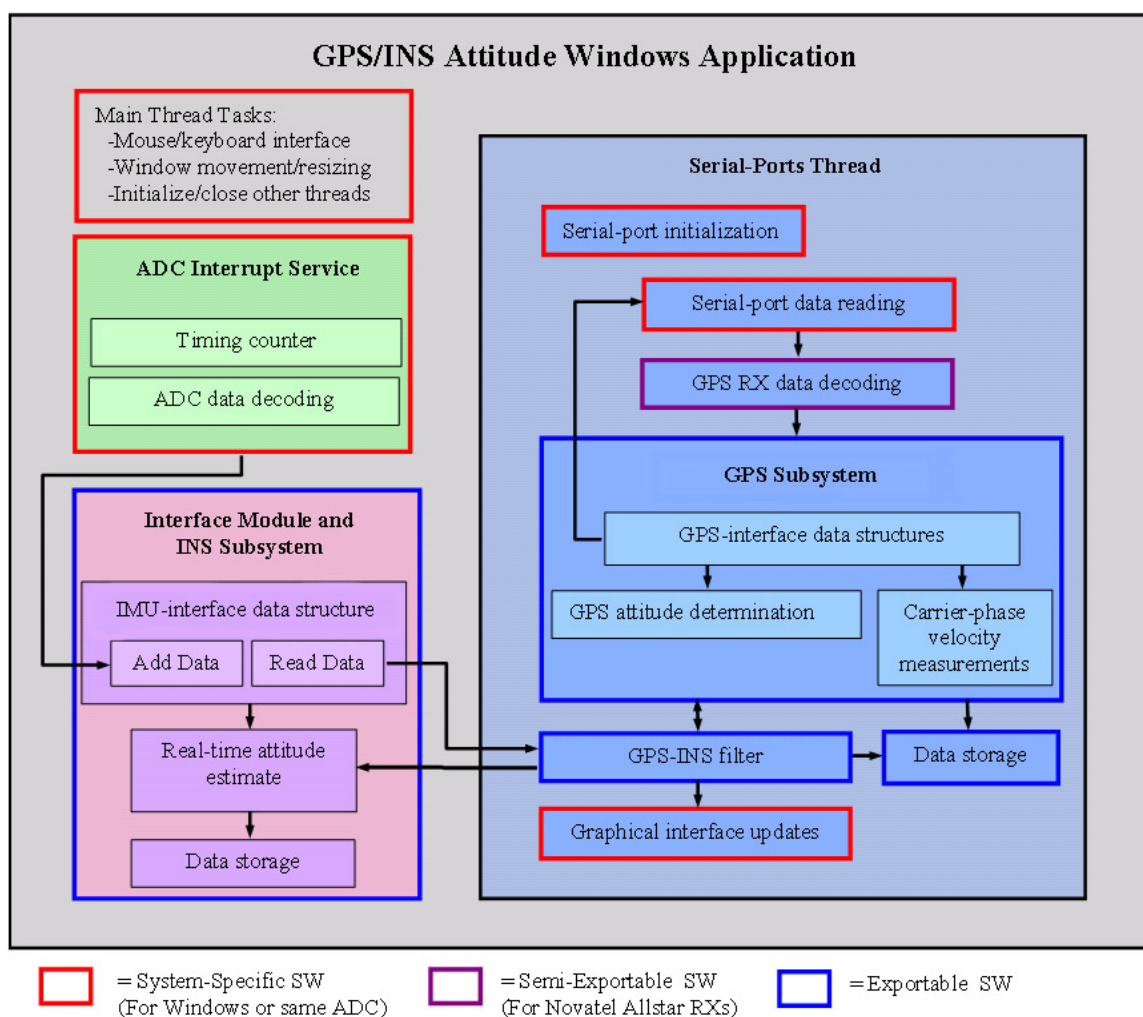


Figure 3.6: Prototype Software Architecture

Figure 3.6 shows a schematic of the major software blocks of the prototype system, where each software block is classified as exportable, semi-exportable, or system-specific. The communication between exportable and non-exportable code is bridged by a global data structure,

such that adaptation of exportable code in another system would only involve assigning values to the global-structure fields from another system-specific code. As shown in this figure, interface software serves as a medium for communication between exportable code and system-specific code.

3.5.2 GPS Interface Software

The serial ports which receive data packets from the GPS receivers have to be accessed continuously to flush out the data in their buffers. The serial data accumulates in the buffer when the port is not being read, and data is retrieved in a first-in-first-out (FIFO) order when the port is read. Consequently, the port has to be read fast enough so that incoming-data rate does not exceed outgoing-data rate in a time period long enough to overflow the buffer. The consequences of a buffer overflow vary, depending on which message is lost. In most cases, loss of a data packet results only in a missed GPS update for the GPS/INS integration filter, resulting in a longer propagation step based on inertial measurements. Integration with an INS is designed to overcome such momentary losses in GPS, so this scenario is relatively innocuous. However, a more serious integrity problem can occur in some implementations. Some consumer GPS receivers which use inexpensive oscillators, including the Allstar boards used in this prototype, often correct their internal estimates of GPS time, and modify their ICP measurements when they do so. In the case of the Allstar receivers, the occurrence of a time correction is accompanied by a quarter-cycle discontinuous jump in phase measurement; the occurrence of this event is reported as part of the raw-measurement data-packet which contains ICP measurements. If the message is received, the discontinuity in the phase measurement can be corrected, and the correction remains applicable until the next clock correction. If a message which coincides with a time correction is missed, the ICP is not corrected properly, and a quarter-cycle error is introduced into all dICP measurements. This situation is equivalent to suddenly invalidating the current line-bias estimate, and could possibly result in a condition which can be difficult to rectify if wrong integer ambiguities are determined as a result of the error. For this reason, the handling of serial port data is of critical importance for maintaining the robustness of this attitude system, and similar systems that utilize serial ports for communication with inexpensive GPS receivers.

The sequence in which the different ports are accessed can be implemented in different ways. One possible method is to use a multi-threaded program with one thread dedicated to each receiver, as described in [2]. This method reads the ports as fast as possible, but complicates the data interchange between threads. An easier technique can be employed, in which a single thread iterates to read from the multiple ports, and reads from whichever receiver has the oldest timestamp on its latest carrier-phase message. With this strategy, if a receiver sends more packets than the others (such as ephemeris or user-position packets which are only needed from one

receiver), the IO buffer for its port would temporarily accumulate more data, but the port would be read more often when it falls behind the other two receivers. The same process would apply if one receiver falls behind due to a CPU interruption while its port was being read. This method was found to work very well and is easier to implement than the multithreaded alternative, with only a slightly longer average delay (less than 0.1 sec) in retrieving carrier-phase packets from the serial ports. If used in a real-time system, this delay would only mean that the GPS/INS integration filter would run slightly behind real time, and inertial measurements would have to be propagated from the last filter update to obtain the real-time navigation solution.

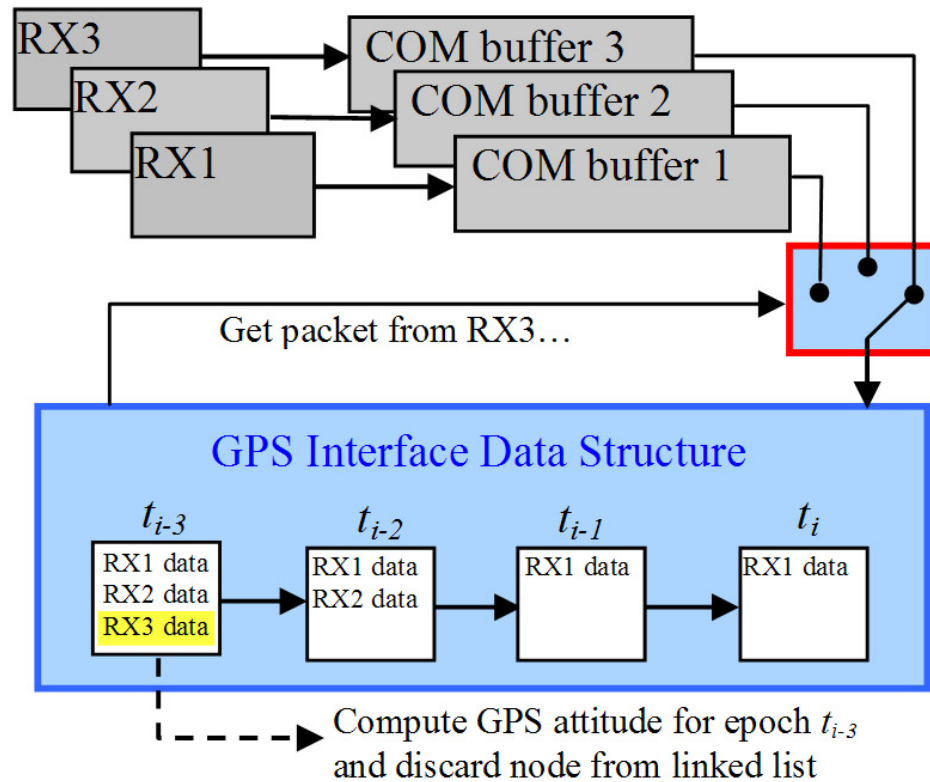


Figure 3.7: Representation of GPS Interface Data Structure

The implication of using a random sequence for serial-port reading is that data packets are also obtained in a random order, and carrier-phase measurements from different receivers must be matched according to their time tag (which is part of the message itself). Thereby, if a receiver is not the last to send a packet at a particular epoch, the data must be buffered until the last receiver sends an update for this epoch. Data buffering was implemented with the use of a linked-list structure, where each node of the list contains data from a single epoch. When data from a new epoch is reported by any receiver, a node is added to the front of the list. When a second receiver sends data for the same epoch, the node is found in the list and the data is inserted. The node will remain in the list in chronological order, until data from all three receivers has been read for that

epoch, at which time the data is processed into an attitude estimate and the node is discarded.

Figure 3.7 represents this process pictorially for a scenario in which RX1 is two epochs ahead of RX2 and three epochs ahead of RX3. Since the latest time-tag for RX3 is older than that for the other receivers, the software reads a CP data packet for RX3, places the data in the appropriate linked-list node, and GPS attitude determination can be computed at epoch t_{i-3} . The same linked-list strategy can be used with any data that is combined from the three receivers to form a composite measurement, such as GPS velocity and position data which can be averaged from various receivers to give better estimates of vehicle velocity and position.

3.5.3 IMU Interface Software

The function of the IMU interface software is to buffer inertial data as it comes in from the ADC, and supply appropriate measurements to the GPS/INS filter when they are invoked after a GPS attitude update. Since the GPS/INS filter will have a time delay and the IMU data is obtained practically in real time, the IMU interface software needs to identify the appropriate data to send to the filter by matching GPS time-tags. Furthermore, inertial data that have been sampled recently but whose time-tags are newer than the GPS time in the filter update need to be kept until they are used by the filter, and can be used immediately to give an attitude estimate that is as current as the last IMU sample. Since these samples are based on hardware interrupts, they have a time-delay of less than a millisecond and can be used to give a nearly real-time attitude estimate.

The requirements of the IMU interface software are also well served by a linked list. In this case, each node in the list contains IMU data (time tag, accelerometer and gyro outputs) for a single ADC sampling epoch, and for simplified processing, also contains integrated IMU data up to and including that epoch. Unlike the GPS linked-list which may sometimes miss epochs during outages, this list is updated without exception every 0.01 seconds immediately after each IMU sample, at which time a new node is placed at the head of the list. The continuity of this process is what guarantees real-time measurement availability at a high bandwidth. When the GPS/INS filter needs the data, the time-tag of the filter update is matched to that of a node in the list, and the integrated inertial data from the previous node is sent to the filter to generate a propagated-state. Subsequently, used nodes are discarded and remaining nodes will continue to be used to propagate inertial measurements into the real-time attitude estimates. Figure 3.8 depicts an example of this process, in which the time-tags in the list nodes are marked with two subscripts: i (first subscript) represents an index for the GPS time update, and j (second subscript) is an index for the IMU samples between GPS updates. Thus, for 10Hz GPS updates and 100Hz IMU samples, the j index would range from 0-9 in between successive GPS samples,

as there are 10 IMU samples in between GPS updates (assuming no outage). In this example, the GPS/INS filter receives a GPS attitude update at time $t_{i,0}$ and n IMU samples are used to propagate the updated state into a real-time estimate.

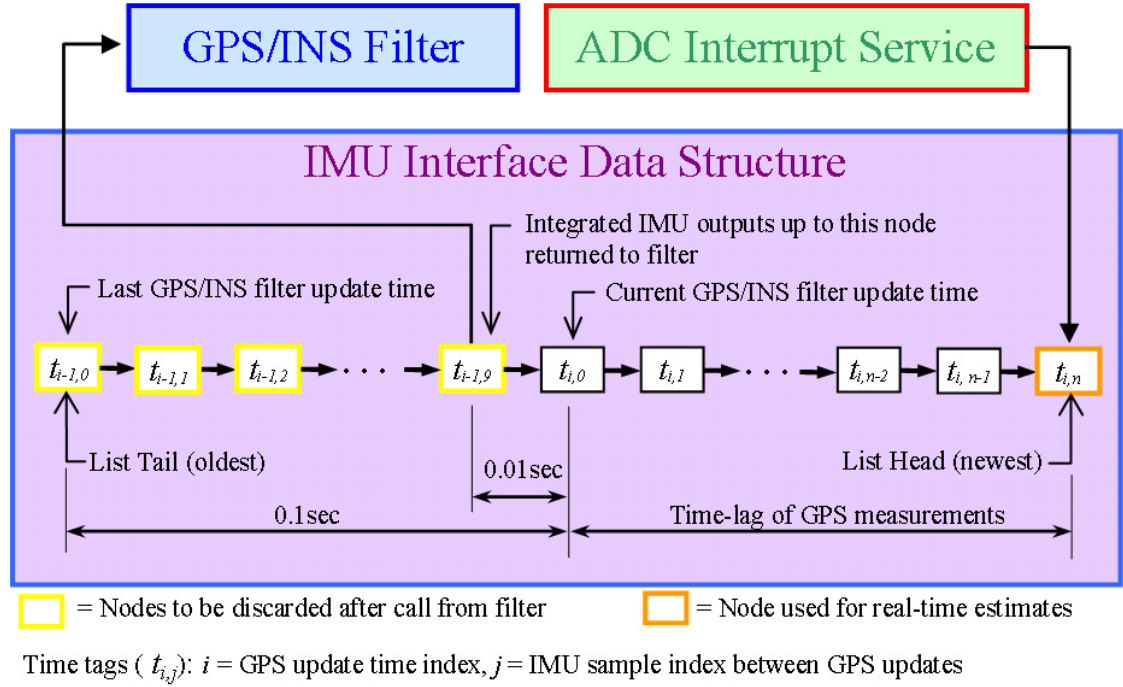


Figure 3.8: Representation of IMU Interface Data Structure

Note that two separate threads access the IMU interface linked list; the ADC interrupt routine adds nodes to the front of the list when new data is sampled, and the serial-ports thread reads data from the list and discards used nodes. Although all linked-list operations are handled from a single module, care must be taken to prevent data access conflicts. This problem can occur when a section of data is being read or written by one thread, and another thread interrupts in the middle of the update to access the same data. For this reason, access to the interface module must be controlled to ensure that a thread completes its operations before another thread is allowed access to the list. This important feature is common to multithreaded programs and is usually implemented with the use semaphores (or critical sections in Windows API), which act a software “traffic lights” to allow only one thread at time to access shared data [55].

3.5.4 Graphical User Interface

For the development process of this attitude-system prototype it was useful to have a GUI to display as much information as possible in real time. For this purpose, a custom Windows application was written to display various system variables, such as one-satellite yaw estimates, inertial-sensor measurements and bias estimates, lengths of linked lists, etc... In addition, the

window shows numerical and graphical representations of the GPS and INS attitude estimates which can be easily monitored visually. Figure 3.9 shows a screen-capture of the GUI window in operation. In this instance, the GPS information window shows all the satellites in view, their one-satellite yaw and averaging weight for three baselines, and a few other parameters pertaining to the size of the dICP linked list and time-lag of the three receivers. This information is displayed when the “GPS Info” button is selected. To display other useful items, The “Messages” button can be selected to view scrolling status messages from the software, and the “INS Info” button is used to view inertial measurements, bias estimates, calibration readiness status, and other variables and flags relevant to the inertial subsystem. The application also allows the user to save data and debugging files with a unique suffix to identify a particular data set later.

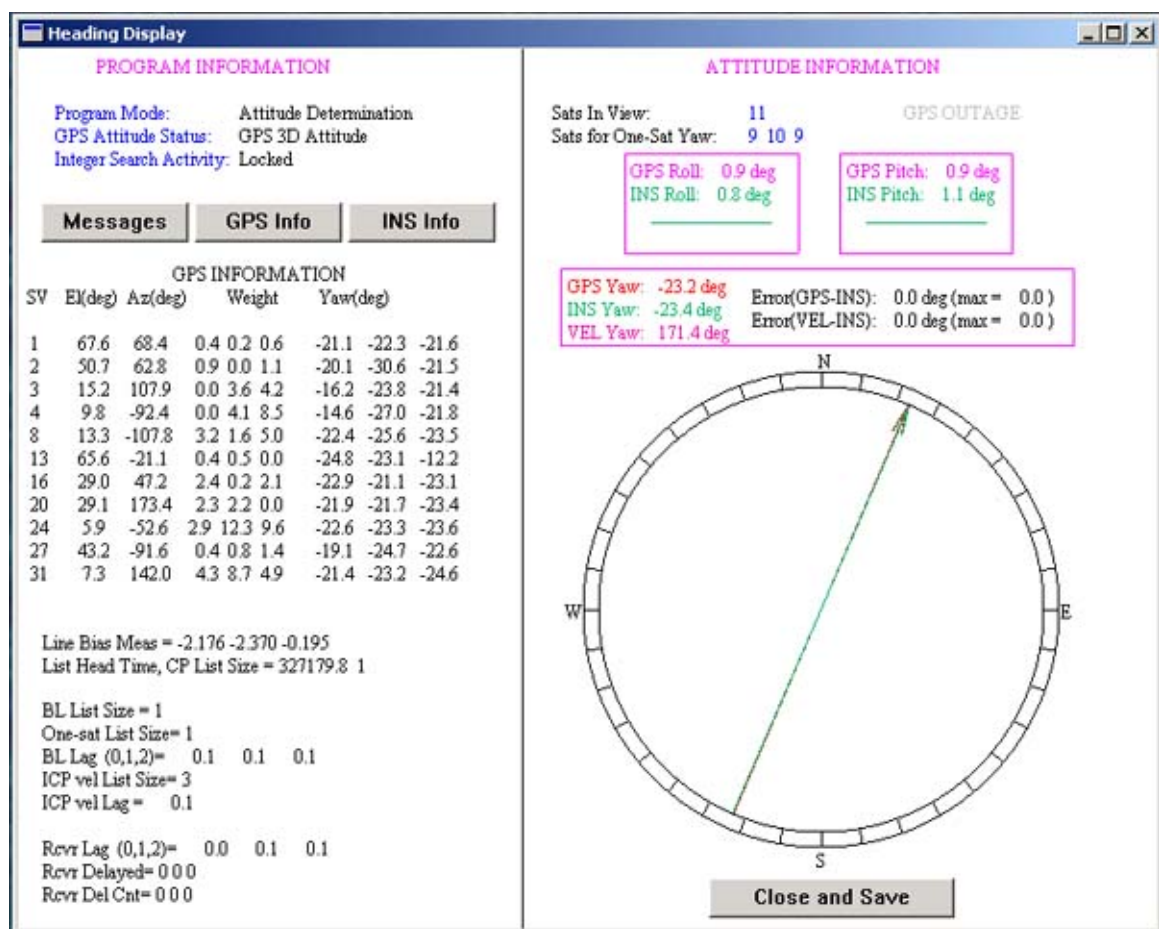


Figure 3.9: Screen Capture of Attitude System GUI

3.6 Time Synchronization

The necessity to synchronize IMU measurements to GPS time was emphasized in Section 3.5, where the appropriate IMU measurements to send to the GPS/INS filter were identified

by their time tag. Commercially-available IMUs do not generally have GPS time-tags on their measurements, but the inertial system in this prototype was built with hardware that made this important characteristic possible. This section describes the hardware and software components that were used to achieve precise time synchronization of GPS and inertial measurements.

The two features of the hardware in this prototype that make time synchronization possible are the pulse-per-second (PPS) signals from the GPS receivers and the ability to trigger the ADC from an external source. The PPS signal is a 1Hz digital signal available in most GPS receivers, which consists of a pulse train whose rising or falling edges are lined up in real-time with the one-second transitions of GPS time, also commonly referred to as Universal Time Coordinated (UTC). The rising edges also correspond to the time when GPS PVT navigation outputs are valid. The PPS signal is based on the receivers internal estimate of GPS time, which is resolved along with code-based user position. The software time estimate in a receiver is accurate to a few nanoseconds, but the rising edges of the PPS signal are usually several hundred nanoseconds from the UTC second transitions. The pulse train typically has a low duty cycle, meaning that the signal stays high (+5V) or low (0V) for a relatively short fraction of its period. For the Allstar boards, the rising edges are specified to be within $1\mu\text{s}$ of UTC, and the pulse-width (dwell-time with signal high) is $1.01\pm0.01\text{ms}$ [9].

To synchronize a signal faster than 1Hz to UTC, the PPS signal can be used as a reference from which to generate a faster periodic signal. Traditionally, synchronization is achieved through a phase-lock loop or similar control scheme in which the phase or edge of the signal to be synchronized is compared to that of a reference signal (in this case the PPS signal), and the frequency (or phase) of the generated signal is adjusted to achieve synchronization. The technique which was implemented in this prototype is somewhat different and possibly less accurate, but was much simpler to implement with inexpensive hardware. In this approach, synchronization was not achieved with a control system, but rather with a repetitive scheme which consists of polling for the rising edge of the PPS signal, outputting a pulse train of 100 pulses spaced 0.01 seconds apart (initiated at rising edge of PPS), and repeating indefinitely. Thus, the 100Hz signal realigns itself with UTC every second, with an accuracy limited by the PPS signal itself and the speed with which its rising edge can be detected by polling. The accuracy of the next 99 pulses will be dependent on a local oscillator, so each pulse in the sequence may have a slightly larger misalignment than the last, with the 100th pulse having the largest error. However, this open-loop drift will incur little error in timing ($<1\text{ms}$) during the one second before realignment, even utilizing an inexpensive ceramic oscillator. The resulting 100Hz output signal is relatively seamless in the transition between the end of a 100-pulse sequence and the beginning of the next, and is more than sufficient for the timing needs of a filter which processes dynamic signals no faster than several tens of hertz.

Given that there is virtually no distinction between the pulses of the 100Hz signal, the PPS signal still needs to be used to identify the first pulse of a 100-pulse sequence to initialize a time-stamp counter. Detection of the PPS signal can be implemented with a simple delay on the falling edge of the PPS pulse, and detecting this longer pulse with a digital input on the ADC card. Once this first pulse is identified, the counter can keep track of subsequent pulses and is used to form the fractional part of the time-stamp (in seconds) on each IMU sample. The whole-number part of the time-stamp (in seconds of week) can be determined easily from the last time-tag on the 10Hz carrier-phase message from a GPS receiver, which is never delayed for more than a few tenths of a second.

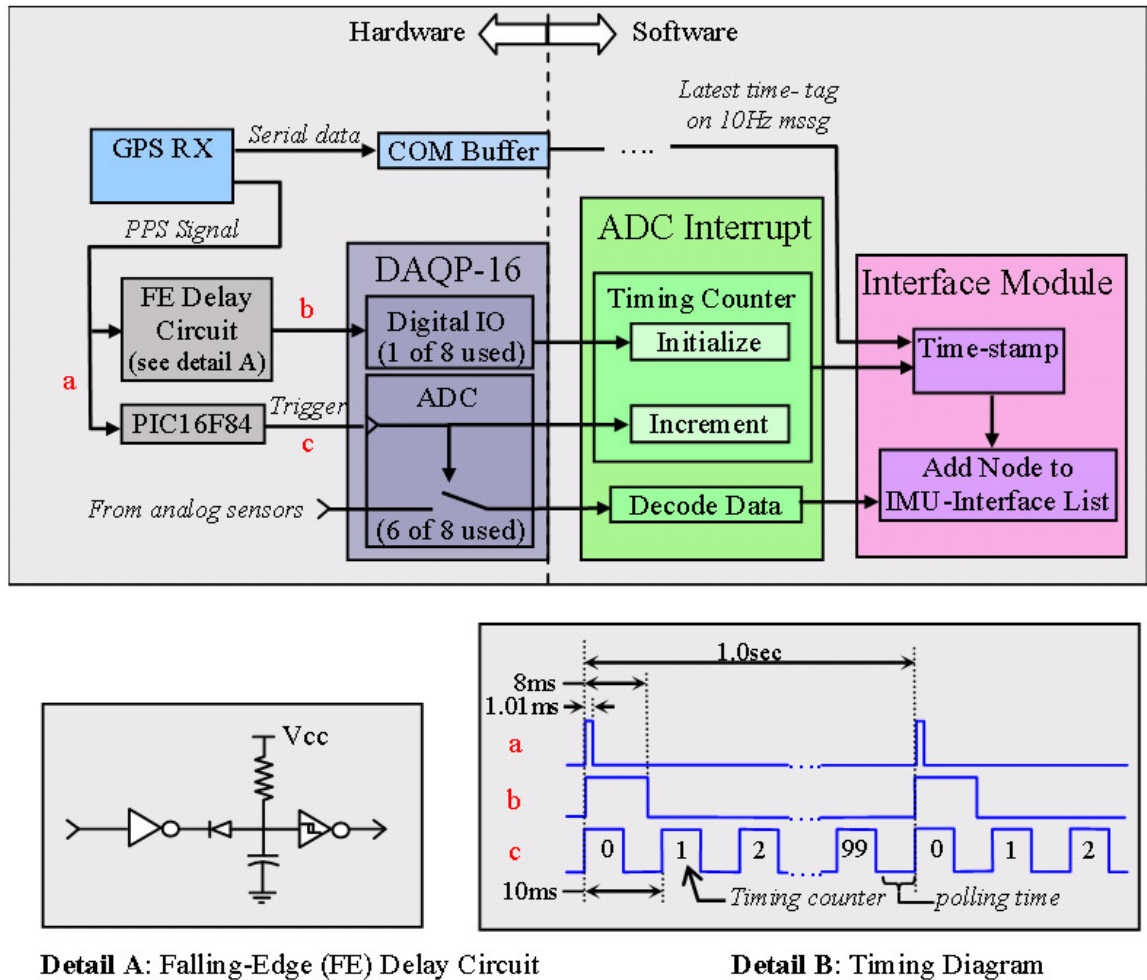


Figure 3.10: Time-Synchronization Diagrams

The hardware needed to implement this timing strategy is nothing more than an inexpensive microprocessor with at least one input pin and one output pin, a compatible ceramic oscillator, and a few passive components. The total cost of this hardware in the prototype was less than \$10, though the digital input on the ADC card is not considered as part of this cost. The specific parts used were a PIC16F84 microprocessor and a 10MHz ceramic oscillator. The PIC was

programmed in assembly language to do the polling for the PPS rising edge on one input pin (detectable within $2\mu\text{s}$), and output the 100Hz trigger signal on one output pin. This microprocessor has many features that are not necessary for this application, and a lower cost alternative could be used.

Figure 3.10 shows a block diagram of this timing scheme, and includes representations of the hardware and software components. A digital waveform diagram also shows the hardware signals at various points of interest in the circuit, and depicts the timing relationships between the PPS signal, the PPS signal with elongated pulse-width, and the 100Hz signal used as the ADC trigger.

Chapter 4

System Design of a GPS/INS Automobile Attitude System with Tight Integration

Chapter 2 presented a detailed description of GPS attitude determination, including techniques for line-bias estimation and integer-ambiguity resolution, traditional methods for determining attitude with multiple satellites, and alternative methods for estimating attitude with as few as one satellite. The development of one-satellite attitude estimation techniques used the simplification of having small vehicle tilt, and thus made it particularly suitable for automobile applications, where vehicle tilt is typically less than 5° in any direction. It was also shown that multiple-satellite and one-satellite solutions have advantages and disadvantages with respect to one another. Specifically, multiple-satellite solutions have better accuracy, but one-satellite solutions can function with fewer satellites and provide better GPS measurement availability in urban environments. Another contrast is that multiple-satellite solutions are inherently self-contained at any single epoch, while one-satellite solutions benefit greatly from having external estimates of attitude from a coupled INS subsystem. The complementary strengths of both GPS attitude-determination methods suggests that the best solution for an automobile system may be a combination of both, with an inertial subsystem to provide the external estimates needed by the one-satellite methods and allow for dead-reckoning navigation during GPS outages.

In Chapter 3, the low-level details of realization of an inexpensive GPS/INS attitude system were covered, including the baseline configuration on a vehicle, hardware components, software data structures needed to interface with typical hardware units, and synchronization of measurements. The next logical step in the development of such a system is to specify the details of subsystem integration.

This chapter focuses on the system-level design and performance of an attitude system prototype that incorporates both of the techniques for GPS attitude determination, and integration with an inertial subsystem. The filtering techniques employed in this system for integration of GPS and INS are specified. In addition to system integration, a detailed discussion on integrity monitoring is included which presents an overview of fault detection at various levels, from ensuring single-channel dICP continuity to the use of multiple-baseline redundancy.

4.1 Subsystem Integration

As mentioned in the introduction to this chapter, two of the unique features of this attitude-system prototype are the usage of multiple-satellite and one-satellite attitude algorithms together as a single GPS subsystem, and tight integration with an inertial subsystem. This section describes some of the important details pertaining to the union of both GPS attitude modules, and the filtering techniques that are utilized to integrate the GPS and INS subsystems. Figure 4.1 shows a diagram of the subsystems that are discussed in this section and the next, and the information paths between them. This figure will serve as a reference in the following subsections which will explain in more detail how the GPS subsystem output is chosen from that of two independent GPS modules, and the integration filters that merge the GPS and INS subsystems.

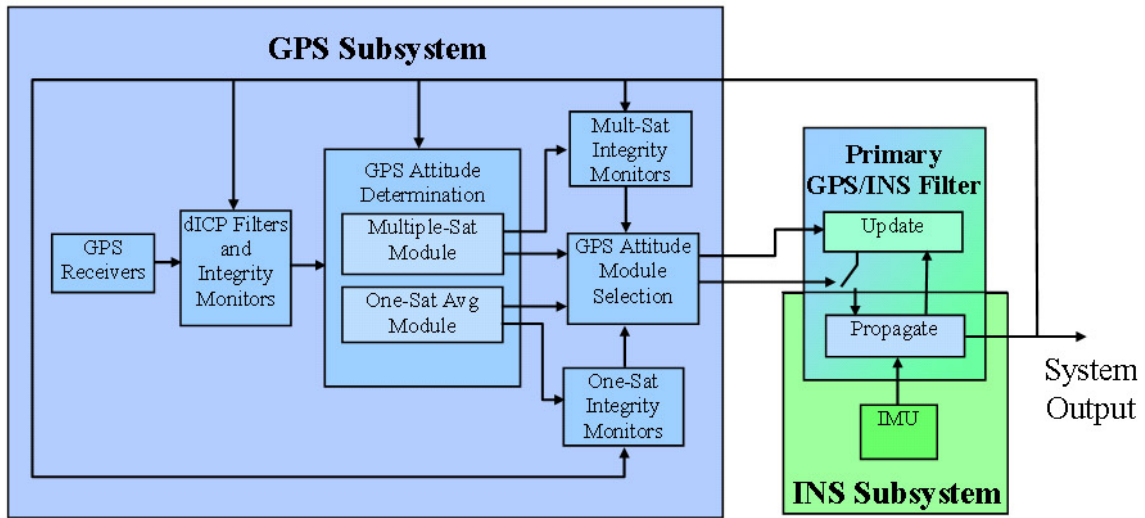


Figure 4.1: GPS/INS Attitude-System Prototype Subsystems

4.1.1 GPS-Attitude Module Selection

The GPS attitude-determination block in Figure 4.1 contains the multiple-satellite solution module (per Eq. 2.38) and the average one-satellite solution module (per Eqs. 2.93 and 2.94). As specified in Chapter 2, these two methods take the same information from the GPS receivers and produce independent measurements of attitude. Besides attitude measurements, the outputs of these modules include various system variables that are used to assess the integrity of their respective attitude solutions. The integrity monitors shown in Figure 4.1 combine this information into a single flag for each module, which indicates whether the applicable attitude output is considered usable or not. The algorithms for integrity monitoring are the subject of Section 4.2, and it will be assumed in this section that the binary value of their output flags is known.

Using only these two flags, the decision of which set of attitude measurements to use as the

overall GPS subsystem output is relatively simple. If the multiple-satellite solution is usable, it should be used as the default because of its better accuracy. If for any reason the multiple-satellite solution is not considered usable, then the average one-satellite solution should be used. If neither solution is usable, then the GPS/INS filter should not receive a measurement update and the propagation step (dead-reckoning from the last state update) would be lengthened until a GPS solution is available again. The ability to transition to and from dead-reckoning is represented in Figure 4.1 by the switch symbol inside the GPS/INS filter, in the path from the update to the propagation step.

Figure 4.2 shows a diagram that summarizes this description symbolically as the contents of the GPS-module selection block. The input flags to this block are the output flags from the integrity monitors, and are labeled *mult_sat_att_ok* and *one_sat_att_ok*; the output flag from this block is equivalent to the union of *mult_sat_att_ok* and *one_sat_att_ok* (equivalent to an OR operation), and indicates whether GPS attitude-measurements should be used in the update step in the filter, or whether the filter should revert to dead-reckoning.

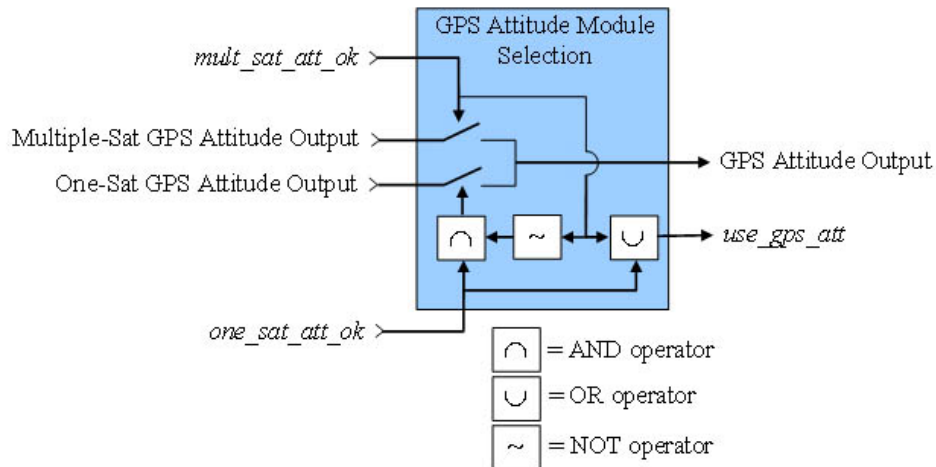


Figure 4.2: GPS Attitude Module Selection

4.1.2 GPS/INS Integration Filters

As indicated in Figure 4.1, integration of GPS and INS is accomplished at various stages in the attitude system, including inside the GPS subsystem. The goals of this integration are to calibrate the inertial sensors and to enhance the robustness of the GPS attitude algorithms by providing a reference for fault-detection at the channel level. Inertial-sensor calibration can be implemented with either a loose-coupling or a tight-coupling filter. Without the feedback path from the system output to the GPS subsystem, the block diagram in Figure 4.1 would represent a loosely coupled system, where the GPS attitude measurements are not influenced by the inertial

sensors. In a literal interpretation of a tight-coupling filter, the inertial sensors are calibrated with a nonlinear filter whose state vector would include at least estimated dICP quantities and sensor biases. In contrast to loose coupling, the GPS attitude measurements would be obtained from filtered estimates of dICP that are influenced by the inertial sensors. In terms of inertial sensor calibration, tight coupling offers a slight benefit over loose coupling due to the fact that the individual channel dICP measurements that drive the calibration process are less correlated than the full GPS attitude solution. This advantage alone is rarely sufficient motivation to incur the added complexity of tight integration. The more important improvement when implementing tight integration in a GPS carrier-phase application benefits the GPS subsystem, and involves dICP filtering and fault-detection at the channel level. This feature can be used to smooth the dICP measurements during periods identified as having high phase-error in a channel and to detect cycle slips.

The filtering architecture in an implementation of tight coupling can have different forms. The use of a single centralized (Kalman) filter is arguably the most complex alternative, as the relationships between all inputs, measurements, and estimated states in the system are related through large gain and covariance matrices in ways that can be nonintuitive. This interrelation is theoretically optimal for the implemented system models, but poses difficulty for performing fault detection on individual states and subsystems, and for maintaining stability under highly variable satellite visibility conditions. Implementation of tight integration can also be achieved with a *federated* filtering approach, which is a simpler alternative to using a centralized filter. This technique involves the use of multiple smaller filters instead of one larger, more complex filter. A federated approach can be more stable and is simpler to construct, but the theoretical optimality of a centralized filter may be renounced for sub optimality [12]. In applications where the conditions for optimality of a single filter are compromised regardless, federated filtering may be a better alternative if it offers a more straightforward and more robust configuration. For example, a GPS/INS attitude system on an automobile will often experience periods of unpredictable high phase-noise on dICP measurements, which will invalidate the current “optimal” filter gains that were chosen based on different values of measurement variance. In this situation, the filter would no longer be considered optimal, because previously specified measurement variances that are used to compute filter gains are not applicable to current measurements. Since the use of a single integration filter in this application offers limited advantage in terms of optimality and is more difficult to implement, a federated approach is a more favorable alternative.

Another reason to use a federated architecture is to simplify the option to bypass filtering (but not integrity monitoring) of dICP measurements. Under good signal conditions, the error of a dICP measurement is on the order of 5mm (RMS); the error of an integrated automotive rate-gyro multiplied by the distance between the IMU and antenna (considering both bias-drift

and wide-band noise) may easily exceed that of a raw dICP measurement after a sufficiently long integration time. In this case, the result of blending a propagated estimate of a dICP value with a more accurate dICP measurement would defeat the original purpose for filtering, and be detrimental to system accuracy. Using filtered dICP values for attitude estimation is more useful when the dICP measurement is corrupted, and integration times of the IMU are short. Considering both of these circumstances, filtering at the dICP level for the benefit of the GPS subsystem should only be done on noisy channels. When it is not done on any channel, the filtering characteristics of the system resemble those of a loosely-coupled system but the critical channel-level integrity monitoring of a tightly-coupled system remains.

As shown in Figure 4.1, a federated GPS/INS filtering architecture was adopted for this system, and is divided into two blocks. The primary tasks of the block labeled “Primary GPS/INS filter” are calibration of the inertial sensors with GPS-based attitude measurements and to provide synchronized attitude estimates as system output and to the GPS subsystem. The block labeled “dICP filtering and Integrity Monitors” performs dICP filtering for fault-detection (namely cycle slips and high noise conditions) and to use in attitude determination when needed. The branches of the feedback path from the GPS/INS filter to the GPS attitude-determination modules and integrity monitors represent the aspects of tight integration that are always present regardless of the dICP filtering activity. These aspects of the integration do not require filtering, and are covered in Subsections 2.2.3 and 2.6.3 for integer-ambiguity searches and in Section 4.2 for integrity monitoring. The focus of this section is the filtering architectures that are employed for dICP integrity monitoring, and for inertial sensor calibration and blended attitude estimation.

Primary GPS/INS Filter

The primary GPS/INS filter is responsible for inertial sensor calibration and blended attitude estimation. Under normal system operation with good dICP measurements, the dICP filters are only used for channel-level fault detection and the primary GPS/INS filter is the only system component that performs literal blending of GPS and inertial measurements. The function of the INS is also fundamentally coupled to the function of this filter, as the inertial sensor biases are part of the estimated state. In this system, the INS is comprised of the IMU and the algorithms that use its measurements to produce attitude estimates. These algorithms are implemented in the propagation step of the filter, and consist of integrating gyro rates and biases from the last state-update to the current time. State updates are the other core function of this filter, and occur whenever GPS attitude measurements are available. The state updates give corrected inertial-sensor bias estimates and a blended attitude estimates from the GPS and the propagated attitude solutions.

The state vector of the filter contains all the quantities (called state elements) that are estimated. This filter needs to estimate three Euler angles for the attitude of the vehicle and all error parameters of interest for three gyros. The inaccuracy in gyro measurements can be modeled very well if all the sources of error are considered, including scale factor, misalignment, and bias drift, among others (See Appendix C). All these parameters can be estimated in the filter with nonlinear dynamic equations, but such a technique complicates the problem needlessly when some of the parameters are nearly constant for a given gyro. Inertial sensor error-parameter estimation can be simplified by determining some of these constant (and repeatable) parameters offline at a typical operating temperature. This type of calibration was found to be useful for the scale-factor error and misalignment errors, whose values were estimated experimentally and corrected in software without the need for real-time estimation. This simplification leaves the gyro biases as the only error parameters that are to be included as states. For automotive-grade inertial sensors, biases (or offset errors) appear to remain constant in the short-term, but always contain exponentially correlated noise that must be corrected in the long term to maintain the accuracy of the INS subsystem. Thereby, the error models for the gyros in this filter only consist of additive biases to the raw gyro measurements. The raw measurements from the gyros are components of the input vector to the filter (denoted with symbol u), and the body rates are expressed as follows in terms of the inputs and biases:

$$\text{Pitch body rate: } q = u_1 + \delta q \quad (4.1a)$$

$$\text{Roll body rate: } p = u_2 + \delta p \quad (4.1b)$$

$$\text{Yaw body rate: } r = u_3 + \delta r \quad (4.1c)$$

With these definitions, the state vector can now be written as:

$$\underline{x} = \begin{bmatrix} \theta & \phi & \psi & \delta q & \delta p & \delta r \end{bmatrix}^T \quad (4.2)$$

The first three elements are the vehicle pitch, roll and yaw, respectively, and the last three are the pitch-gyro bias, roll-gyro bias and yaw-gyro bias, respectively.

The next step in the development of this filter is modeling of the dynamic relationships among states and inputs. These equations will serve as the algorithms for state propagation in between GPS updates, and constitute the software component of the INS. In general, the relationship between body rates and Euler-angle rates is nonlinear with nonzero vehicle tilt, as roll and pitch cause coupling between the gyro outputs. For a yaw-pitch-roll rotation sequence from ENU to

body basis, the relationships between body rates and Euler-angle rates are:

$$\begin{bmatrix} q \\ p \\ r \end{bmatrix} = \begin{bmatrix} \cos \phi & 0 & -\sin \phi \cos \theta \\ 0 & 1 & \sin \theta \\ \sin \phi & 0 & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} \quad (4.3)$$

The inverse of the matrix in Eq. 4.3 would contain the nonlinear relationships needed to integrate the body rates into Euler angles. The nonlinear relationships between state elements and inputs would require the use of an extended Kalman filter [39], in which linearized equations are used to compute the state covariance and Kalman gain matrices, but the full nonlinear relationships are used to propagate the state. Extended Kalman filters are used widely in estimation applications of nonlinear problems, and are not difficult to implement. However, in this application the nonlinear problem can be simplified, by realizing that the roll and pitch of a car are small (typically $\leq 5^\circ$). With this simplification, the matrix in Eq. 4.3 is approximately equal to the identity matrix and the angle-integration equations become uncoupled, thus allowing the use of a simple linear filter. The validity of this simplification was tested in actual road tests, and the results of using both types of filters were indistinguishable. These results confirmed that a linear filter can be used for this application without appreciable loss of accuracy. The small tilt-angle approximations described herein can be summarized as the following relationships between Euler-angle rates and body rates:

$$\dot{\theta} \approx q = u_1 + \delta q \quad (4.4a)$$

$$\dot{\phi} \approx p = u_2 + \delta p \quad (4.4b)$$

$$\dot{\psi} \approx r = u_3 + \delta r \quad (4.4c)$$

The behavior of an automotive-grade gyro bias can be modeled fairly accurately as a constant plus a first order Markov process, with correlation time on the order of hundreds of seconds. This time period is relatively long when compared to correlation times of the GPS measurements used to update the bias, so the dynamics of the gyro bias will likely behave like a constant in a time period of less than about 100 seconds. Using this approximation, the dynamic equations of the gyro biases are:

$$\delta \dot{q} \approx 0 \quad (4.5a)$$

$$\delta \dot{p} \approx 0 \quad (4.5b)$$

$$\delta \dot{r} \approx 0 \quad (4.5c)$$

The continuous-time state-propagation equations can be written in matrix form by combining

Eqs. 4.2, 4.4 and 4.5:

$$\dot{\underline{x}} = \underline{A} \underline{x} + \underline{B} \underline{u} \quad (4.6)$$

where

$$\underline{A} = \begin{bmatrix} 0_{3 \times 3} & I_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} \quad (4.7)$$

$$\underline{B} = \begin{bmatrix} I_{3 \times 3} \\ 0_{3 \times 3} \end{bmatrix} \quad (4.8)$$

For using these equations in a discrete-time filter, a special notation needs to be adopted to account for the fact that the propagation equation will be augmented with GPS measurements, where the inputs (\underline{u}) and the GPS measurements have different update rates. The same notation as in Subsection 3.5 will be used, in which subscript i denotes the index of a GPS update, and subscript j denotes the index of an IMU sample in between GPS updates. With this notation, subscript j resets to zero whenever subscript i increments, as illustrated in Figure 3.8. Since the propagation step takes place in between GPS samples and involves IMU inputs only, subscript i remains fixed and subscript j increments with each new IMU sample. Therefore, the conversion of Eq. 4.6 to discrete time is done with the sampling period of the ADC ($\Delta t_{INS}=0.01$ sec):

$$\underline{x}(-)_{i,1} = \underline{\Phi} \underline{x}(+)_{i,0} + \underline{\Gamma} \underline{u}_{i,0} \quad (j=0) \quad (4.9a)$$

$$\underline{x}(-)_{i,j+1} = \underline{\Phi} \underline{x}(-)_{i,j} + \underline{\Gamma} \underline{u}_{i,j} \quad (j > 0) \quad (4.9b)$$

where

$$\underline{\Phi} = e^{\underline{A} \cdot \Delta t_{INS}} \quad (4.10)$$

$$\underline{\Gamma} = \int_0^{\Delta t_{INS}} e^{\underline{A} \eta} d\eta \underline{B} \quad (4.11)$$

In this application, the dynamics an automobile can be assumed to be much slower than the 100Hz sampling frequency of the ADC, so Eqs. 4.10 and 4.11 are simplified with the use of a first order approximation of the matrix exponential [30]:

$$\underline{\Phi} \approx \begin{bmatrix} I_{3 \times 3} & \Delta t_{INS} I_{3 \times 3} \\ 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix} \quad (4.12)$$

$$\underline{\Gamma} \approx \begin{bmatrix} \Delta t_{INS} I_{3 \times 3} \\ 0_{3 \times 3} \end{bmatrix} \quad (4.13)$$

The ‘ $-$ ’ signs in Eq. 4.9 are traditionally used to label a state estimate before an update step

(the pre-update state-estimate), while the ‘+’ sign indicates an estimate after the update step (the post-update state-estimate). This differentiation is necessary, as the two estimates are valid at the same time-epoch and will have the same time subscripts. Most literature on Kalman filtering will present Eq. 4.9 only as a relationship between the post-update state-estimate at epoch i and the pre-update state-estimate at epoch $i+1$, assuming that inputs and measurements are updated at the same rate. The notation used here emphasizes the fact that inputs are sampled faster than the measurement update rate, in which case the propagation step represented in Eq. 4.9a is used with the first input after the update ($j=0$) and Eq. 4.9b is used with all subsequent inputs until the next GPS update.

When a GPS measurement is available, the filter performs an update step that consists of combining the propagated state-estimate with the measurement. To maintain consistency with the established notation, the occurrence of a GPS update should increment the i subscript of the IMU time tags, and reset the j subscript. To simplify the notation of the update step, the propagated state estimate up to an instant before the current update should have $(i,0)$ subscripts. If n IMU samples occur between the last GPS update at time $t_{i-1,0}$ and the current GPS update at time $t_{i,0}$, then the subscript modification that precedes the update step would be as follows:

$$\underline{x}(-)_{i,0} = \underline{x}(-)_{i-1,n-1} \quad (4.14)$$

For this filter, the measurement vector is defined as the three Euler angle measurements from the GPS subsystem:

$$\underline{y}_i = \begin{bmatrix} \theta_{GPS}(t_i) & \phi_{GPS}(t_i) & \psi_{GPS}(t_i) \end{bmatrix}^T \quad (4.15)$$

The state-update process consists of blending this GPS measurement vector with the propagated state, and is described by the following equation:

$$\underline{x}(+)_{i,0} = \underline{x}(-)_{i,0} + \underline{K} \left(\underline{y}_i - \underline{H} \underline{x}(-)_{i,0} \right) \quad (4.16)$$

where

$$\begin{aligned} \underline{H} &= \begin{bmatrix} \underline{I}_{3 \times 3} & \underline{0}_{3 \times 3} \end{bmatrix} \\ \underline{K} &= \text{Gain Matrix} \end{aligned} \quad (4.17)$$

Note that when GPS measurements are not available, this step is bypassed and the propagation step is prolonged. This scenario is represented in Figure 4.1 with the open connection between the “Update” and “Propagate” blocks in the primary GPS/INS filter.

The gain matrix is usually selected based on the relative confidence levels between the propagated state and the measurement. If the errors in these quantities can be assumed to be uncorrelated and with Gaussian distributions, then a Kalman filter can be used which converges to the optimal gains based on the known variances of process noise (which affects the propagation step) and measurement noise. The Kalman gains tend to be high (near unity) when confidence in the measurements outweighs that of the propagated estimate, and low (near zero) otherwise. To avoid extra computation cost, Kalman gains can also be computed offline based on known variances [35, 39, 59], in which case the gain matrix would be constant. The sacrifice for this simplification is that the gains may not necessarily be optimal initially (depends on covariance of initial state estimate), but the estimated-state covariance would eventually converge such that the gains are optimal.

Kalman filters are usually implemented for systems with stationary random processes. A stationary random process is one whose statistics such as variance and time-correlation remain constant. As mentioned previously, however, the variances of the GPS measurements in this system are highly dependent on the signal environment and number of satellites in view, and are difficult to quantify in real time. The time-correlation of GPS attitude errors is also not always stationary as a result of imperfect phase-delay calibration, which results in distinct time-correlation for non-rotating antennas, but much less time-correlation for rotating antennas. These complexities make realization of a true optimal filter very challenging, and a sub-optimal filter was opted for ease of implementation. The constant gains were chosen near the Kalman values obtained under nominal conditions, but are more favorable towards the propagated state to adjust for frequent high phase-noise conditions. Gain scheduling was also implemented to allow for cases where the confidence in the propagated state is reduced as a result of long GPS outages or rapid yaw maneuvers. In such cases, the accuracy of the dead-reckoned attitude estimate is expected to degrade after a prolonged period without bias-calibration updates, and the gains are adjusted accordingly. In situations where the INS could have drifted by several degrees (about 30 seconds if there is a yaw maneuver) the INS is reset with the first available GPS solution.

Table 4.1 shows the values of the filter gains used in various situations. The columns of the table represent states of the system in which different sets of gains are used. The different cases are as follows:

Case A: Initialization (before convergence of gyro bias estimates) or after GPS outages during which significant yaw maneuvering occurred.

Case B: After Initialization; normal operation with small yaw rate.

Case C: Normal operation, but with significant yaw rate ($> 10^\circ/sec$). Gains are adjusted to mitigate errors from scale factors or gyro misalignment, as these quantities are not included as states and are not calibrated perfectly.

Case D: First epoch after a long GPS outage ($> 30\text{sec}$). Attitude estimates are reset to first GPS solution.

Table 4.1: Primary GPS/INS Filter Gains

Gain	A	B	C	D
$k_{1,1}$	0.15	0.1	0.3	1
$k_{2,2}$	0.15	0.1	0.3	1
$k_{3,3}$	0.15	0.05	0.15	1
$k_{4,1}$	0.04	0.001	0	no change
$k_{5,2}$	0.04	0.001	0	no change
$k_{6,3}$	0.04	0.0005	0	no change

Figure 4.3 contains plots of the GPS yaw and filtered yaw for a test case in which two 60-second GPS outages were induced by negating the *use_gps_att* flag artificially. During the first outage the antenna array was left static, and during the second outage the antenna array was rotated about 33° counterclockwise. The purpose of this test was to verify that the drift rate of the INS-only attitude solution was small for GPS outages on the order of a minute. The amount of INS error-drift accumulated during the outage can be gauged by the difference between the INS and GPS solutions when GPS is recovered after the outage; as shown, this difference is less than one degree in both cases, suggesting that the yaw rate-gyro was well calibrated before the outage and the bias did not change measurably during that time. Figure 4.4 shows the yaw rate-gyro bias estimate for this data set, and confirms this fact. The notable change in the variance of the bias estimate at $t = 35\text{sec}$ and $t=310\text{sec}$ is due to gain adjustments from the gain-scheduling algorithms, which occur after the bias has converged to a relatively constant value after initialization, and after long periods under dead reckoning. The running average of the gyro bias is also shown in this plot, and is simply a low-passed version of the filter's output. This cleaner estimate of the bias is used during GPS outages in case the outage occurs while the scheduled gains are high, which would result in high variance in the filter outputs. Although only the yaw and yaw-gyro bias estimates are shown in Figures 4.3 and 4.4, this performance is representative of the roll and pitch estimates as well. Of course, this comparison is subject to the caveat that GPS yaw measurements are more accurate than GPS roll or pitch measurements for this system (see Figure 2.11), and the filter gains reflect this difference.

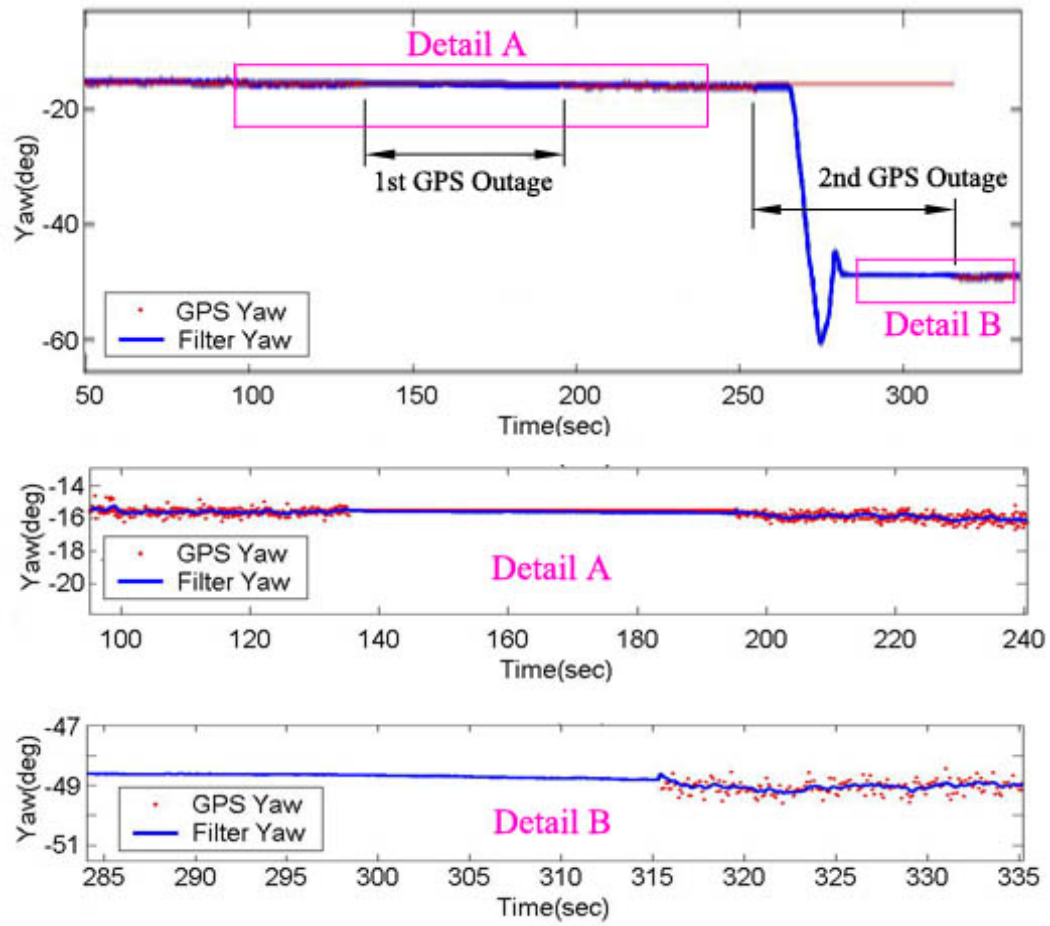


Figure 4.3: Primary GPS/INS Filter Yaw Estimates

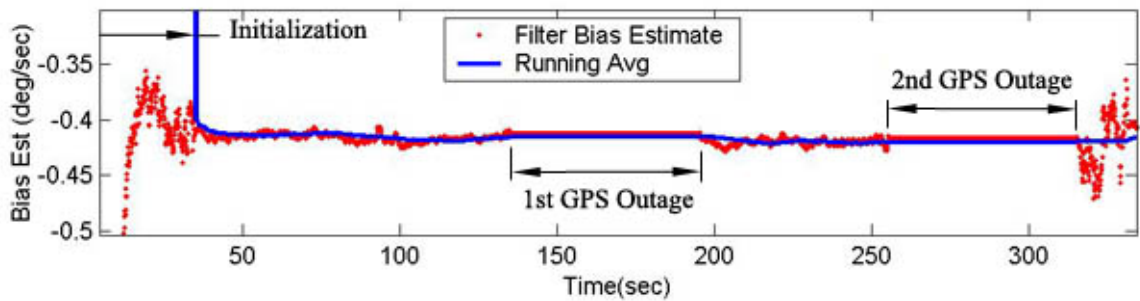


Figure 4.4: Primary GPS/INS Filter Yaw-Gyro Bias Estimates

dICP Filters

Under conditions where one or more channels are flagged as noisy, the raw dICP measurements for the appropriate channels can be replaced by filtered dICP estimates in the GPS attitude algorithms. These estimates are obtained through a traditional filtering process where the GPS dICP measurements are used in the update step, and the calibrated gyro rates are used in the propagation step. The state vector of the filter contains only one state, the dICP estimate for a given baseline and satellite.

The dynamic equation that constitutes the propagation step of this filter can be obtained by taking the time derivative of Eq. 2.9:

$$\frac{d}{dt} \left(\Delta\varphi_{AB}^k(t) \right) \approx \frac{d}{dt} \left(-\underline{1}_e^{kT}(t) {}^A\underline{B}_e^B(t) + n_{AB}^k + l_{AB}(t) \right) \quad (4.18)$$

Since the LOS vector and line bias change slowly with respect to the baseline in an Earth-fixed basis, this equation can be simplified by assuming that all terms are constant except the baseline vector:

$$\frac{d}{dt} \left(\Delta\varphi_{AB}^k(t) \right) \approx -\underline{1}_e^{kT} \frac{d}{dt} \left({}^A\underline{B}_e^B(t) \right) \quad (4.19)$$

The baseline vector in the ENU basis can be written as the product of the rotation matrix from body to ENU basis and the constant baseline vector in body basis:

$${}^A\underline{B}_e^B(t) = {}_{e \Leftarrow b} \underline{C}(t) {}^A\underline{B}_b^B \quad (4.20)$$

Substituting this equation into Eq. 4.19 results in an expression that shows the time rate-of-change of the dICP as a function of the time rate-of-change of the rotation matrix:

$$\Delta\dot{\varphi}_{AB}^k(t) \approx -\underline{1}_e^{kT} {}_{e \Leftarrow b} \dot{\underline{C}}(t) {}^A\underline{B}_b^B \quad (4.21)$$

The time rate-of change of the rotation matrix is a function of the vehicle's attitude and the body rates, which are available from the primary filter's output. This matrix quantity is most easily derived by expanding the rotation matrix as the product of three single degree-of-freedom rotation matrices where each is a function of one Euler angle:

$${}_{e \Leftarrow b} \underline{C}(\psi, \theta, \phi) = {}_{e \Leftarrow t_2} \underline{C}(\psi) {}_{t_2 \Leftarrow t_1} \underline{C}(\theta) {}_{t_1 \Leftarrow b} \underline{C}(\phi) \quad (4.22)$$

The t_1 and t_2 bases correspond to intermediate reference frames used between successive rotations. The time derivative of Eq. 4.22 is:

$$\dot{{}_{e \Leftarrow b} \underline{C}} = \dot{{}_{e \Leftarrow t_2} \underline{C}} {}_{t_2 \Leftarrow t_1} \underline{C} + {}_{e \Leftarrow t_2} \underline{C} \dot{{}_{t_2 \Leftarrow t_1} \underline{C}} + {}_{e \Leftarrow t_2} \underline{C} {}_{t_2 \Leftarrow t_1} \dot{\underline{C}} \quad (4.23)$$

where

$$\dot{\underline{C}}_{e \Leftarrow t_2} = \frac{\partial}{\partial \psi} \left(\underline{C}_{e \Leftarrow t_2} \right) \dot{\psi} \quad (4.24a)$$

$$\dot{\underline{C}}_{t_2 \Leftarrow t_1} = \frac{\partial}{\partial \theta} \left(\underline{C}_{t_2 \Leftarrow t_1} \right) \dot{\theta} \quad (4.24b)$$

$$\dot{\underline{C}}_{t_1 \Leftarrow b} = \frac{\partial}{\partial \phi} \left(\underline{C}_{t_1 \Leftarrow b} \right) \dot{\phi} \quad (4.24c)$$

The Euler-angle rates in these equations are approximated as in Eq. 4.4 from the gyro-rate inputs and estimated gyro biases. In discrete time, the timing relationship between the propagation and update steps is analogous to that of the primary GPS/INS filter, so the same timing subscripts as in Eqs. 4.9, 4.14 and 4.16 are applicable. As such, the propagation step for the dICP filters is:

$$\Delta \hat{\varphi}_{AB}^k(-)_{i,1} = \Delta \hat{\varphi}_{AB}^k(+)_{i,0} - \underline{1}_e^{k^T} \dot{\underline{C}}_{e \Leftarrow b}(t_{i,0})^A \underline{B}_b^B \Delta t_{INS} \quad (j=0) \quad (4.25a)$$

$$\Delta \hat{\varphi}_{AB}^k(-)_{i,j+1} = \Delta \hat{\varphi}_{AB}^k(-)_{i,j} - \underline{1}_e^{k^T} \dot{\underline{C}}_{e \Leftarrow b}(t_{i,j})^A \underline{B}_b^B \Delta t_{INS} \quad (j>0) \quad (4.25b)$$

The update step is:

$$\Delta \hat{\varphi}_{AB}^k(+)_{i,0} = \Delta \hat{\varphi}_{AB}^k(-)_{i,0} + k \left(\Delta \varphi_{AB}^k(t_{i,0}) - \Delta \hat{\varphi}_{AB}^k(-)_{i,0} \right) \quad (4.26)$$

where the gain k would have a value of about 0.1-0.3 for automotive grade gyros.

To simplify the implementation of this filter, the states and measurements in Eqs. 4.25a and 4.26 actually represent the dICP quantities corrected for the last known integer ambiguity. If instead they represented the raw dICP quantity, a cycle slip would create a large discontinuity between successive time-steps and complicate monitoring and visualization of the filters operation. If a cycle slip occurs between time-steps, the propagated state's integer correction will not match that of the new measurement (which is not known yet), so the update step is bypassed in these situations until the new integer is determined. Bypassing the update step is also used to prevent poor data from corrupting the filtered dICP estimate. To identify noisy data, the magnitude the dICP measurement innovation (also called the measurement residual) can be compared to a previously established noise threshold. The dICP innovation is the quantity inside the large parenthesis in Eq. 4.26:

$$res \left(\Delta \varphi_{AB}^k(t_{i,0}) \right) = \Delta \varphi_{AB}^k(t_{i,0}) - \Delta \hat{\varphi}_{AB}^k(-)_{i,0} \quad (4.27)$$

In this implementation, the magnitude of the dICP innovation is allowed to be as large as 0.05 cycles, which is equivalent to about two standard deviations of the expected phase-noise of a typical PLL. If the innovation exceeds this value, the dICP measurement is ignored in the filtering

process by bypassing the update step.

Figure 4.5 contains plots of the dICP measurements and estimates for a set of data which includes both static and dynamic conditions. The second plot shows the value of a flag that indicates whether the dICP innovation's magnitude exceeds the noise thresholds of ± 0.05 cycles. Detail A shows a section of data when the baseline is static, and the dICP measurements exhibit some time-correlated error due to phase-delay and line-bias drift. These errors cannot be mitigated with filtering because of their slow rate of change, and since their magnitude can exceed that of uncorrelated noise, no accuracy advantage would be gained by using the filtered dICP instead of the measurement in the attitude solution. Detail B focuses on a section of data between $t=102$ and $t=106$ where the dICP measurements appear corrupted by multipath, and the dICP filter successfully identifies and ignores the outlying data. Note how this situation is clearly distinct to other rapid changes in dICP caused by movement of the baseline, where the gyro rates capture the movement and properly propagate the filter estimate such that the innovations remain small.

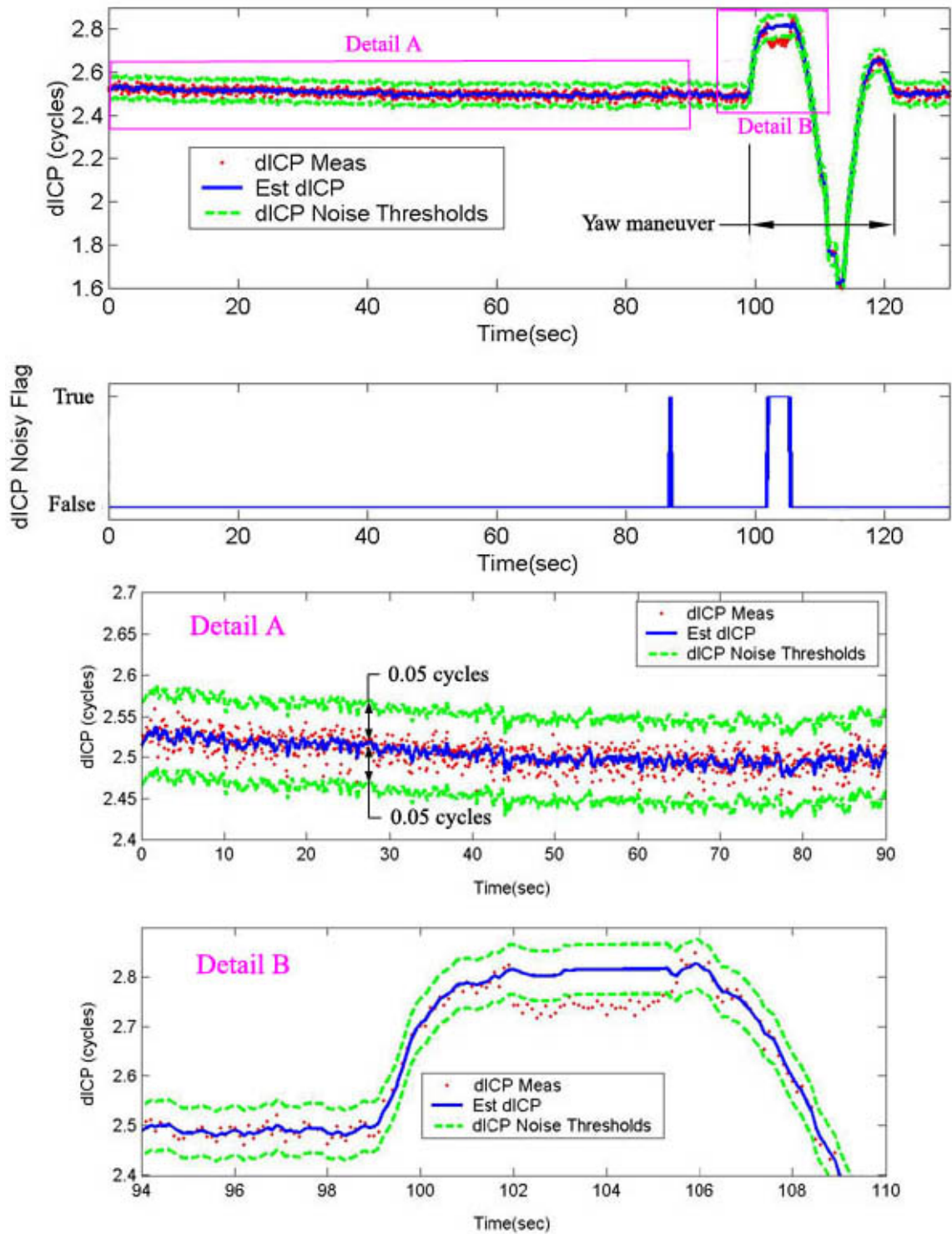


Figure 4.5: Performance of dICP Filters

4.2 Integrity Monitoring

Integrity monitoring refers to the inspection of measurements and program variables to ensure the continuous availability and robustness of the attitude system. This process can take place at various levels in the system, and may or may not capitalize on GPS/INS integration. At a low level, integrity monitoring is performed at the channel-level, and its purpose is detection of excessive phase-noise and cycle slips in individual channels. This information is critical for knowing when to exclude noisy measurements from the attitude solution and when to perform new integer-ambiguity searches. The next level of verification is at a single-baseline level, in which relationships that are common to all channels in a baseline are utilized, such as the line-bias and line-bias correction term. Geometry constraints are also applicable at the single-baseline level, consisting of such verifications as known baseline-length (for the linear attitude solution) and platform tilt-angle bounds. The highest level of integrity monitoring in the GPS subsystem involves multiple-baseline redundancy, in which the vector relationships between the baselines are used to verify various relationships between single-baseline parameters, including line-bias and integer ambiguities. The final level of integrity monitoring takes place within the primary GPS/INS filter, and prevents poor GPS attitude solutions from distorting the filtered attitude estimates.

4.2.1 dICP Fault Detection and Correction Measures

The discussion on dICP filters in Subsection 4.1.2 explained much of the methodology for dICP fault-detection and correction. The value of the dICP innovations was used as a measure of the noise content in the raw measurement, and values that exceeded a certain noise threshold were not used to update the filter state. Thereby, inspection of the innovations represents the first level of integrity monitoring of GPS measurements.

In instances where the residual exceeds the noise threshold in a given channel, the dICP state estimate is likely more accurate than the unprocessed GPS measurement, and is thus used subsequently in the attitude determination algorithms. Even though the state is not updated with the current GPS measurement, the state estimate still contains geometric information that is independent from the other channels, and is usable as a separate measurement. This corrective measure is useful as long as the noisy conditions are brief, such that the state estimates in the dICP filter retain correlation with past measurements from the channel. During a prolonged period where no updates occur, the estimated dICP for the channel may drift away from the true value simply as a result of integrating the noise on the gyro rates, and if used in the attitude-determination equations, it would deteriorate the accuracy of the solution. For this reason, monitoring the length of time without dICP updates is necessary for knowing when to reset the dICP estimate with the

first available measurement. In the current implementation, the filter is reset after a 30 second period without usable GPS measurements. This value is relatively conservative, as it assumes that no other channels are being used to calibrate the inertial sensors.

The value of the dICP filter innovation can also be used as a cycle-slip detector, since the change in integer ambiguity after a cycle slip will usually result in a innovation magnitude larger than multiple cycles, and this condition is clearly distinguishable from even the most severe multipath phase-noise in the channel. In the event of a possible cycle slip, a flag can be set to activate an integer determination process for the channel, and the decision to use the channel is deferred until the new integer is determined.

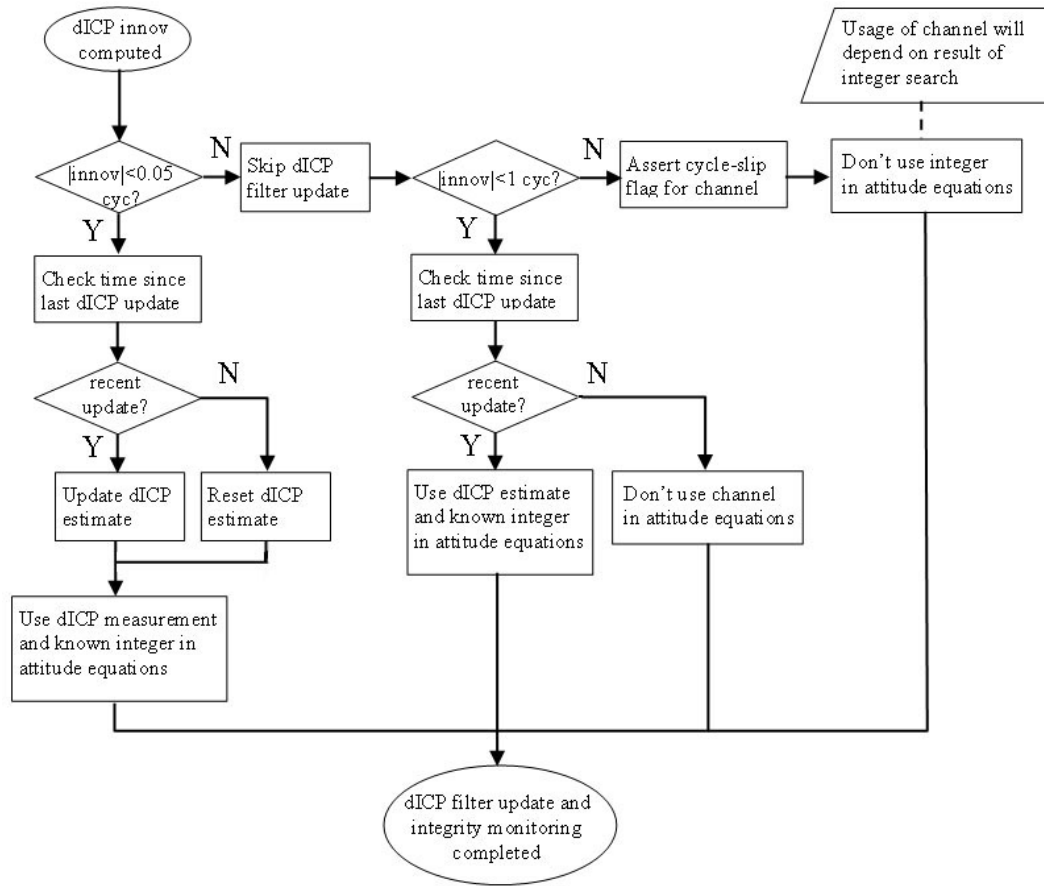


Figure 4.6: dICP Filter Updates and Integrity Monitoring

The flow chart in Figure 4.6 summarizes this fault-detection and correction procedure, including the simple decision processes utilized to flag noisy measurements and cycle slips and the actions taken in regards to the dICP filter updates and channel utilization.

4.2.2 Baseline-Level Integrity Monitoring

The next level of integrity monitoring takes place at the baseline level, and uses the relationships between the channels in a given baseline as well as geometric constraints to confirm the validity of a GPS solution. The techniques that fall under this category for the multiple-satellite solution include verification of the line-bias correction value, baseline length, and magnitude of inclination angle. Note, these methods are not dependent on the inertial subsystem. The one-satellite solutions do not allow for line-bias correction monitoring or baseline-length verification, and are thus more dependent on the inertial subsystem for fault detection and recovery. Thereby, integrity monitoring at the baseline level is different for the multiple-satellite and single-satellite solutions (as depicted in Figure 4.1) and are covered separately in the following subsections.

Baseline-Level Integrity Monitoring for Multiple-Satellite Solution

The first step in verifying the integrity of a multiple-satellite solution is inspection of the line-bias correction term. In Subsection 2.2.2, the line-bias correction term was introduced as a method of tracking changes in the line bias. Besides reflecting the common mode biases in all the channels in a baseline, this value also depends on the residuals in the individual channels, and thus on the noise content of the phase measurements. Since the line-bias correction term is supposed to be small and change slowly, any large, short-term deviations in its value can be used to identify instances of disagreement between channels for the multiple-satellite solution. In such cases, a simple inspection of the individual row-equation residuals yields the channels that cause the problem, and the noisy channels can be removed from the solution.

Figure 4.7 demonstrates the result of implementing this technique. The first plot shows the line-bias correction term (as computed with Eq. 2.14) and its running average (as computed with Eq. 2.28b) during a short test drive. The line-bias correction term clearly exhibits erratic behavior between 100sec and 120sec, when one or more channels became noisy as the car passed near some trees. The second plot shows the value of a flag (*dl_stable*) that is negated when the line-bias correction term exceeds 5cm difference with its running average. The running average is not updated during these relatively noisy conditions, but no further action is performed to amend the GPS measurement. The 5cm noise threshold was established empirically for this system, such that noise conditions that yield differences below this limit did not affect the attitude solution significantly. The third and fourth plots show the same data for a similar test drive, executed a few minutes later through the same path near the trees. In this case, corrective action is taken whenever the line-bias correction term exceeds 5cm difference with its running average. More specifically, up to two channels with the largest residuals are removed from the least-squares solution. As shown, the line-bias correction term is much better behaved through the

noisy section, though the scatter of points within the 10cm corridor around the running average suggests that there is a larger noise component in other channels that remain in the solution.

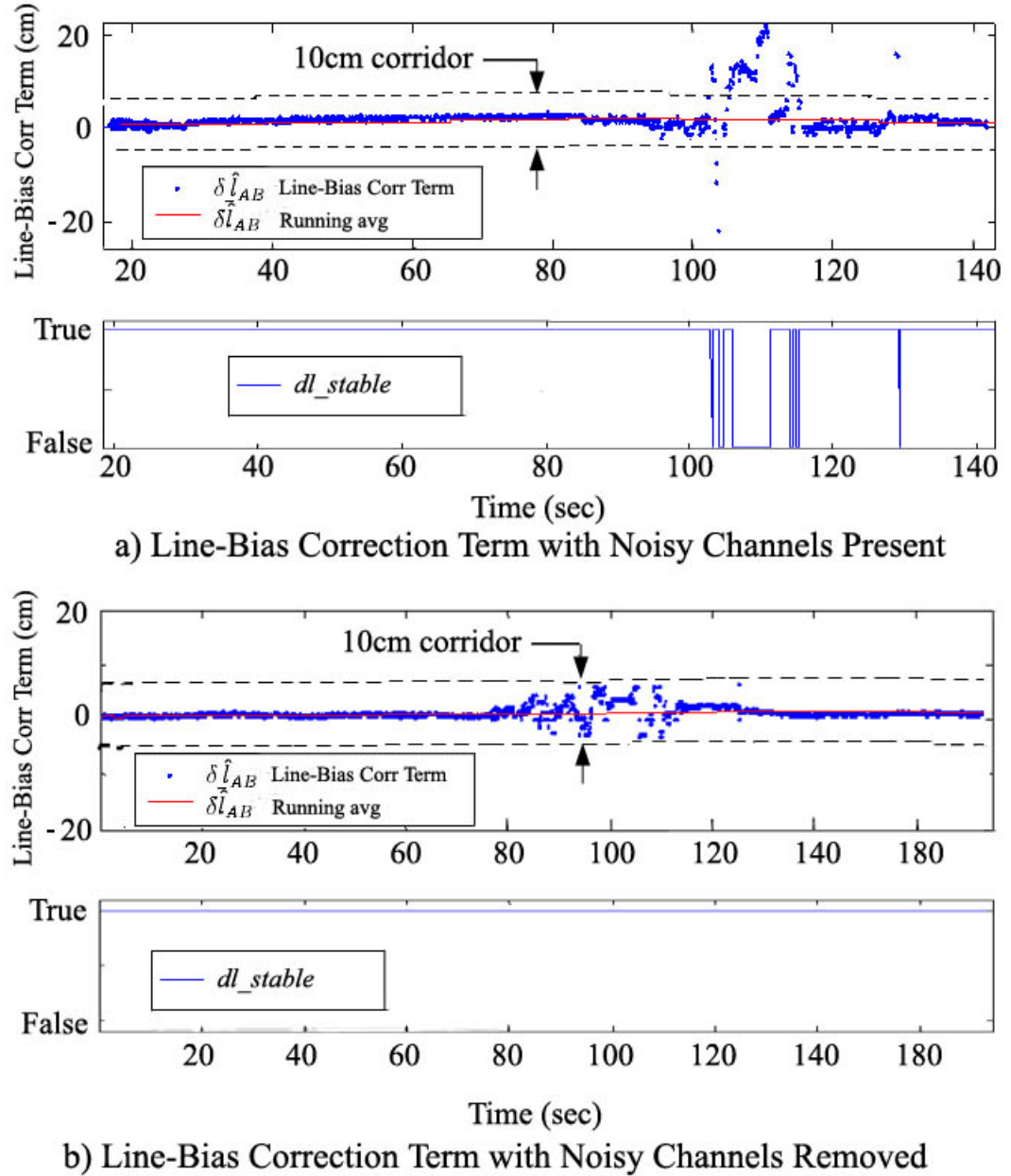


Figure 4.7: Noisy Channel Removal by Residual Inspection

Two other simple but useful verifications on the baseline vector are the baseline-vector length and the magnitude of the baseline inclination angle. The baseline-length test is identical to that used in the integer-verification process (see Eq. 2.33), and rejects solutions for which the norm of the baseline vector differs from the known baseline length by more than a specified tolerance. The inclination-angle test is similar in concept, in that the vertical component of the baseline vector is not allowed to exceed the value that would correspond to a maximum expected

inclination angle.

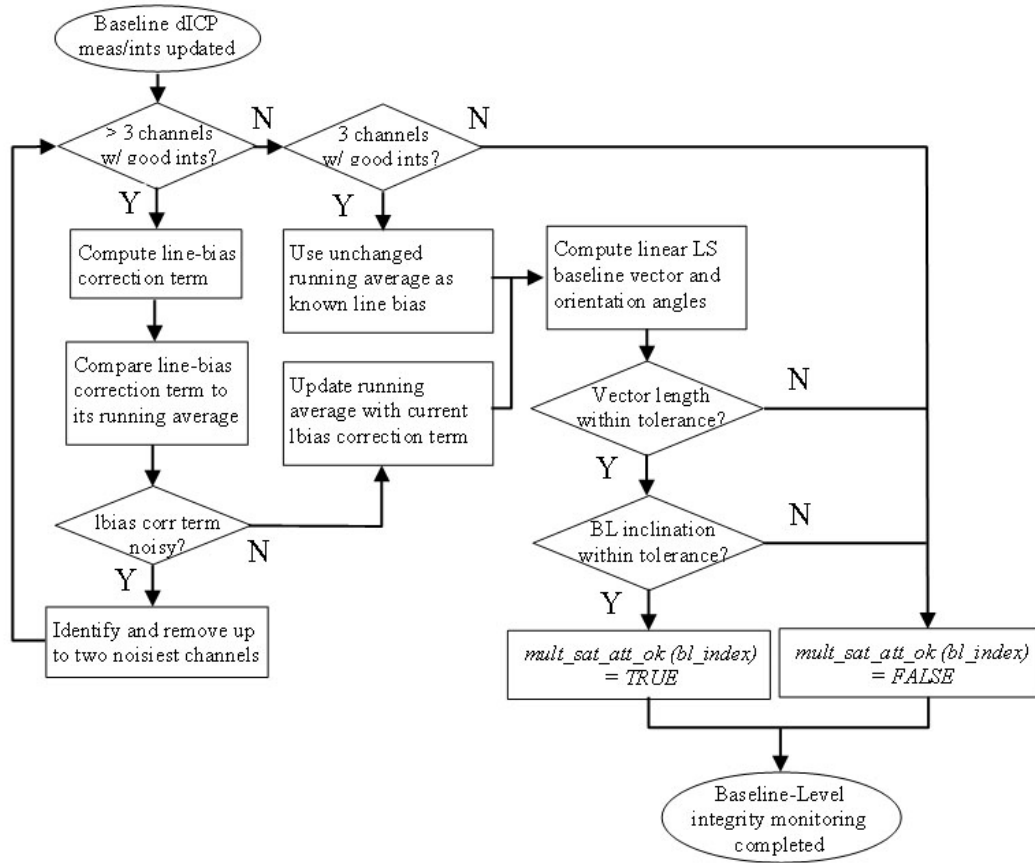


Figure 4.8: Baseline-Level Integrity Monitoring for Multiple-Satellite Solution

The integrity monitoring process at the baseline level takes place for each baseline individually, and includes the line-bias correction term test, the baseline length test and the baseline inclination test. All tests must pass for the baseline vector to be considered usable. The flow chart in Figure 4.8 illustrates this process as it is implemented in software, and begins after all required integer searches for the multiple-satellite solution have been performed. Near the end of the flow chart, the flag $mult_sat_att_ok(bl_index)$ represents the combined result of these tests. Note, this flag represents only one of the baselines, and is not the same as the $mult_sat_att_ok$ flag discussed in Section 4.1.1, which represents all of the baselines and is asserted only after the multiple-baseline integrity tests.

Baseline-Level Integrity Monitoring for Average One-Satellite Solution

For the average one-satellite attitude solution, integrity monitoring at the baseline level can be done with or without interdependence among channels. Using other channels as a reference with which to assess the accuracy of a one-satellite attitude estimate amounts to a similar test as the line-bias correction term test discussed in the previous subsection, as it would consist of

comparing residual errors of one channel with respect to the other channels. As an alternative, the primary GPS/INS filter attitude output can be used instead of the other channels as a reference with which to gauge the noise content in a single dICP measurement. This option helps to maintain the decoupling of the channels, but strengthens the dependence of the one-satellite attitude techniques on the inertial subsystem. In a sense, using the primary filter output as a reference at this level is similar to its use in the dICP filters, in that a single GPS measurement is compared to a reference value partially based on the inertial subsystem. The difference between the two tests is that the reference value used in the baseline-level test is not a propagated estimate based on the channel being tested, and is thus less correlated to the tested measurement.

In this attitude system, baseline-level integrity monitoring for the one-satellite solutions is implemented with and without the use of the primary GPS/INS filter. Due to its simpler implementation and better reliability, using the GPS/INS filter as a reference is the preferred method when the gyros are well calibrated. The other channels are used as a reference for integrity monitoring only during the initialization phase of the inertial sensors or when the system operates without an INS. Regardless of the source of the reference attitude, the residual test on the channel is the same. The residual quantity to be tested with respect to the reference attitude is identical to that used in the one-satellite integer determination process, and is given by Eq. 2.88. This residual measurement was introduced as a measure of confidence in a new integer solution, but in this case it is also used to ascertain the noise content of the measurement when the integer is known.

In addition to the residual tests on each channel, integrity monitoring of the average one-satellite orientation angles also involves removal of channels with poor ADOP to prevent erratic measurements from disrupting the average attitude estimates. This process consists simply of rounding the weights of the one-satellite measurements (see Eqs. 2.91 and 2.92) to zero when they are considered low.

Figure 4.9 shows a flow chart that represents the integrity monitoring process for the average one-satellite solutions of a single baseline. As before, the algorithm depicted begins after any required integer searches have been completed, and the output of the algorithm is a flag that indicates whether the baseline vector based on one-satellite attitude measurements is usable. Note that since the yaw and roll angles are computed separately in the one-satellite attitude algorithms, the baseline vector is used even if the baseline roll is not determined due to poor ADOP. Since small vehicle pitch and roll are expected the majority of the time, the baseline-roll angle is assumed to be zero, and a usable one-satellite yaw measurement is sufficient to allow use of the baseline vector.

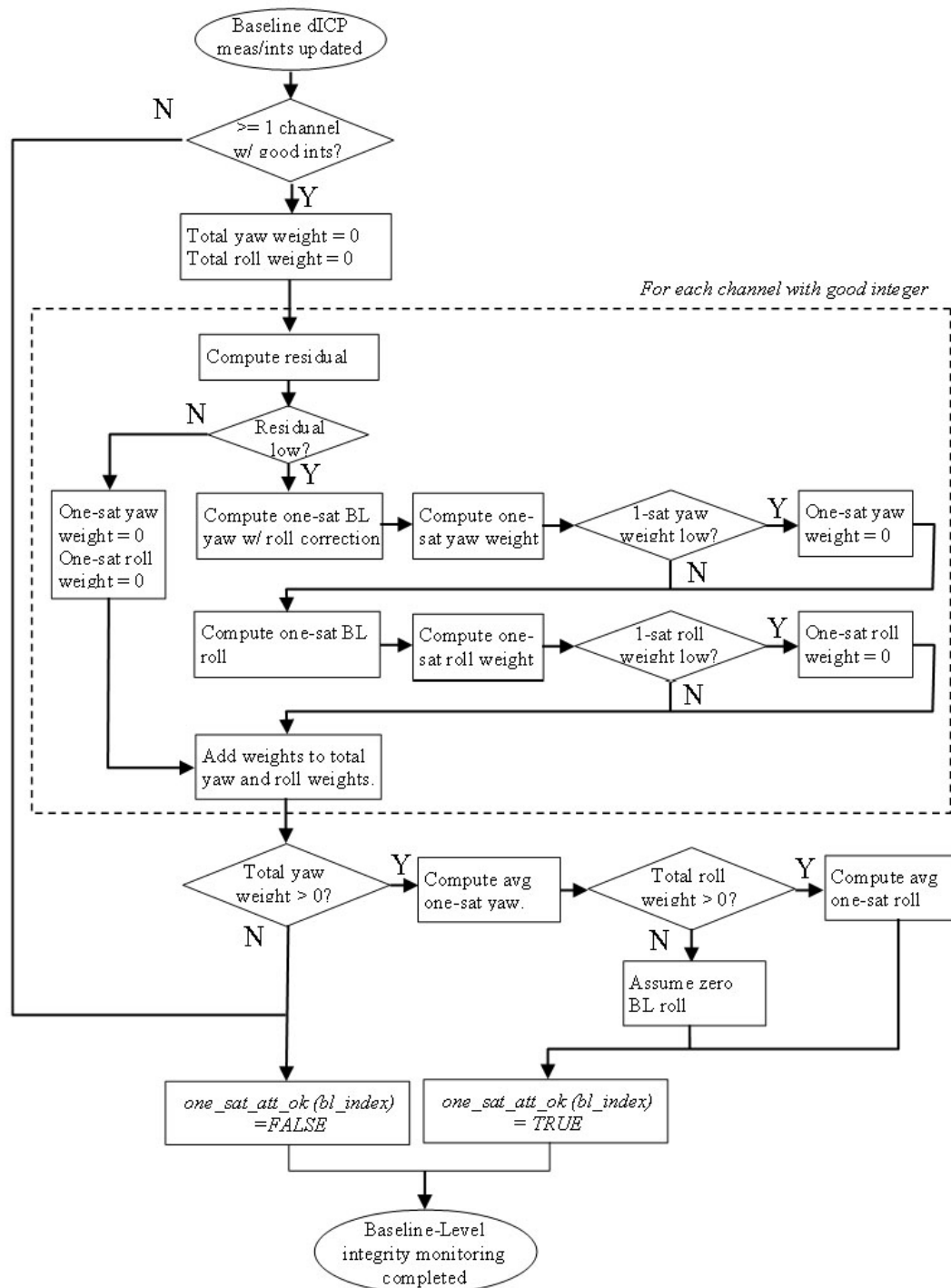


Figure 4.9: Baseline-Level Integrity Monitoring for One-Satellite Solution

4.2.3 Multiple-Baseline Integrity Monitoring and Recovery

Baseline redundancy is the main concept used to implement integrity monitoring with multiple baselines. The concept of redundancy is based on the fact that any one baseline vector can be obtained from the sum or difference of the other baseline vectors. As shown in Figure 2.7, the triangular baseline configuration used in this system is subject to the relationship

$$\vec{B}^{AC} = \vec{B}^{AB} + \vec{B}^{BC} \quad (4.28)$$

Since the three baselines are computed independently, the most obvious monitoring technique that can be used when all three baselines are available is making sure that this vector relationship is satisfied to within a specified tolerance. If one of the baselines fails lower-level integrity monitors, this higher-level integrity check cannot be used, but the full three-axis attitude estimate is still available.

The line biases of the baselines also obey the same relationship, since a line-bias simply represents a difference in propagation-delay times of the GPS signal along two paths. Thereby, the line bias of baseline AC should be the same whether it is measured between antennas A and C directly, or as the sum of the line biases of baselines AB and BC. Of course, this relationship must be adjusted for whole-number rollover, but for simplicity it will be expressed without emphasizing the integer correction:

$$l_{AC}(t) = l_{AB}(t) + l_{BC}(t) \quad (4.29)$$

The relationship also applies to the phase delays, which are affected by relative differences between perceived antenna phase-centers. Since all antennas are assumed to have the same LOS vector for a given satellite, the phase delays between the three baselines are related as follows:

$$\eta_{\varphi AC} = \eta_{\varphi AB} + \eta_{\varphi BC} \quad (4.30)$$

With this equation, the line biases and line-bias correction terms can be cross-checked when they are computed for all three baselines.

Equations 4.28-4.30 have another important use when they are combined such that the relationship between integer ambiguities for different baselines is revealed. To derive this useful property of redundant baselines, Eq. 4.28 is first multiplied by a LOS vector:

$$\vec{1}^{kT} \vec{B}^{AC} = \vec{1}^{kT} \vec{B}^{AB} + \vec{1}^{kT} \vec{B}^{BC} \quad (4.31)$$

Each of the terms in this equation now looks like the delta-range term from a row of the linear

attitude equation (see Eq. 2.9). Since Eqs. 4.29 and 4.30 relate the line bias and phase delay between baselines, the relationship between the remaining terms of the attitude equation rows is:

$$\Delta\varphi_{AC}^k - n_{AC}^k = (\Delta\varphi_{AB}^k - n_{AB}^k) + (\Delta\varphi_{BC}^k - n_{BC}^k) \quad (4.32)$$

This equation now provides a relationship between the integer ambiguities of the three baselines, and offers an additional means for verification that the integer values are correct.

The baseline-angle verification is one additional test that can be performed on two baselines. This test consists of computing the angle between two estimated baseline vectors, and comparing this angle to its known value. The cosine of the angle between two vectors can be computed with a well-known equation. Using baselines AB and AC as an example, the cosine of the angle between the baselines is:

$$\cos \angle_{CAB} = \frac{\vec{A-B} \cdot \vec{A-C}}{R_{AB}R_{AC}} \quad (4.33)$$

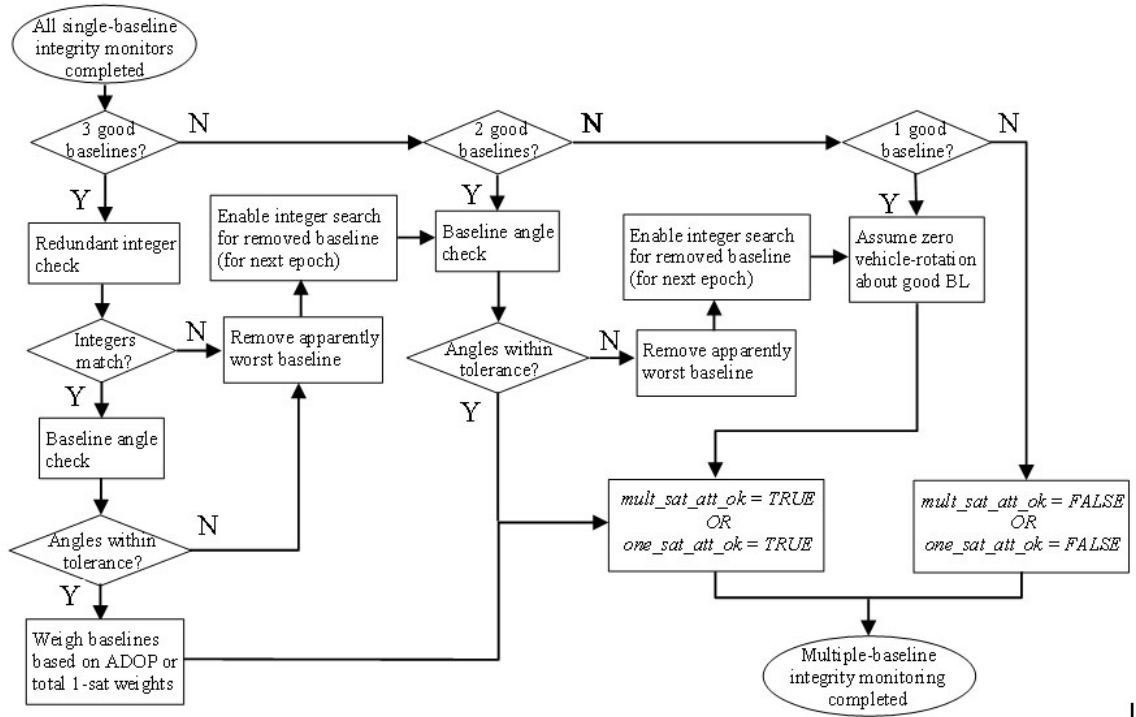


Figure 4.10: Integrity Monitoring with Multiple Baselines

Figure 4.10 shows a flow-chart that summarizes the integrity-monitoring and correction operations at the multiple-baseline level. The algorithm begins when the single-baseline integrity monitors are completed for all three baselines, and their output flags (*mult_sat_att_ok(bl_index)* and *one_sat_att_ok(bl_index)*) indicate whether individual baselines are usable. Note, this algorithm is essentially the same whether the baselines are computed with multiple-satellite solutions or from average one-satellite solutions. The flow chart includes branches that apply for cases

where there are three good baselines, two good baselines, or only one good baseline. In the latter case, only two rotational degrees of freedom can be measured; the yaw angle is observable with any single baseline, but roll and pitch depend on the position of the baseline on the vehicle. The outputs of these final integrity monitors in the GPS subsystem are simply the Boolean flags that were introduced in Subsection 4.1.1, *mult_sat_att_ok* and *one_sat_att_ok*.

4.2.4 Integrity Monitoring in the Primary GPS/INS Filter

The final level of integrity monitoring is in the primary GPS/INS filter, and uses the same innovation-monitoring technique that was used for the dICP filters. Since the small inclination-angle assumption decouples the three rotational degrees of freedom, each Euler angle can be monitored separately for large differences between the GPS measurement and corresponding propagated state. This type of error-checking capitalizes on the short-term stability of the INS subsystem to identify momentary errors in the GPS measurements that may have gone undetected in the GPS subsystem. After all the other integrity checks, this final level of fault-detection mostly targets single-baseline solutions (where the two other baselines are not considered usable), as this situation is not subject to the multiple-baseline integrity monitors (see Figure 4.10).

The innovation vector for the primary GPS/INS filter is shown in Eq. 4.16 as the vector quantity inside the larger parentheses. The individual components are:

$$\text{Pitch Innovation} = \theta_{GPS}(t_i) - \hat{\theta}(-)_{i,0} \quad (4.34a)$$

$$\text{Roll Innovation} = \phi_{GPS}(t_i) - \hat{\phi}(-)_{i,0} \quad (4.34b)$$

$$\text{Yaw Innovation} = \psi_{GPS}(t_i) - \hat{\psi}(-)_{i,0} \quad (4.34c)$$

These scalar values are considered independently to determine whether a GPS measurement should be used to update the corresponding state. In this implementation, up to a 5° innovation magnitude is used to update the state, and higher values are ignored in favor of dead reckoning during that epoch. In addition to checking the value of the innovations, the time since the last GPS update must also be considered, as done with the dICP filters. After a sufficiently long period without GPS measurements, the filter should be reset (instead of updated) with the GPS attitude measurement. However, to exceed 5° of error, this type of outage would have to be on the order of minutes and is very rare in most outdoor driving environments. The final stage of verification assures that the gyro-bias state-estimates remain relatively constant. This test consists of comparing a bias state-estimate with its running average, which is relatively free of wideband noise, and ignoring estimates that exceed a pre established threshold.

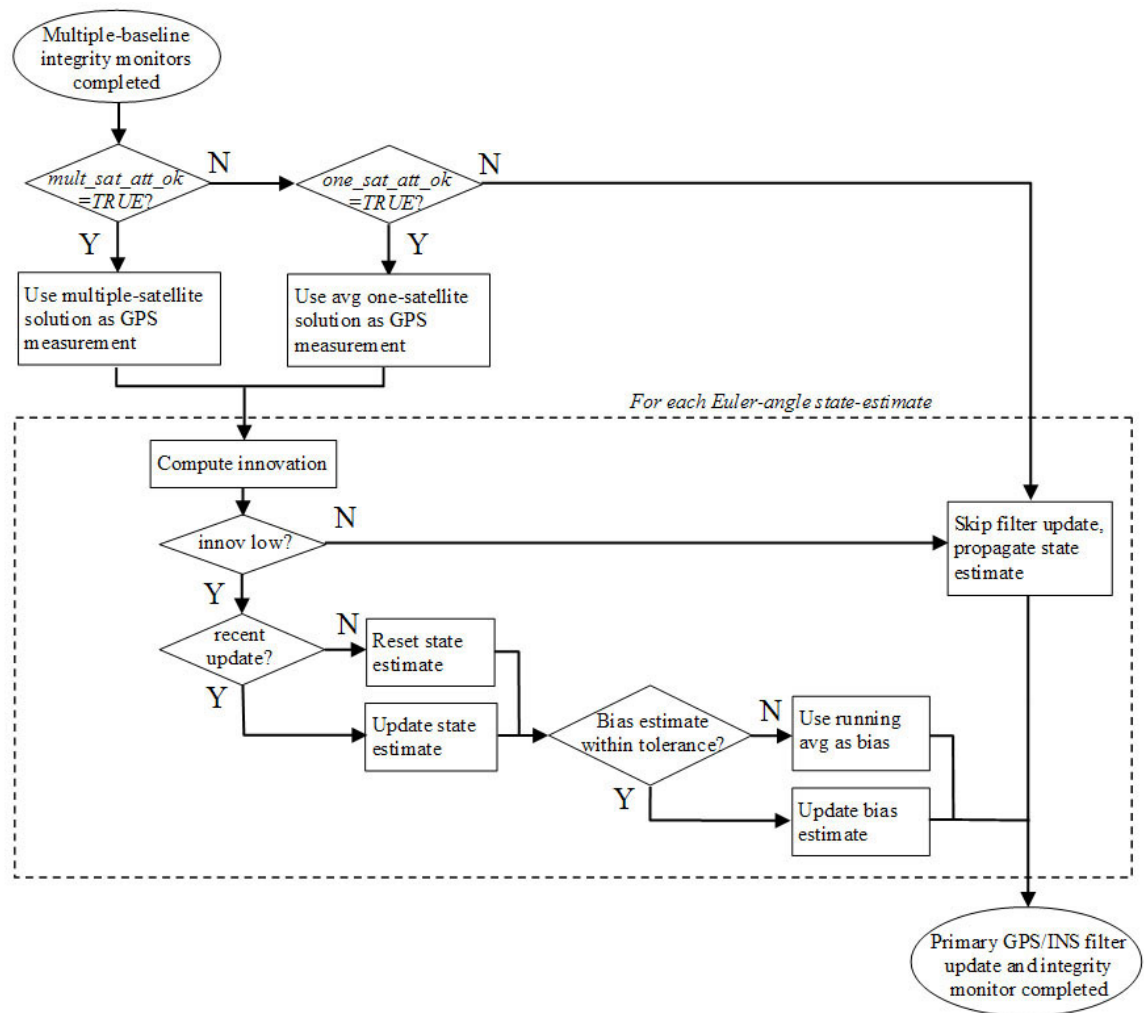


Figure 4.11: Primary GPS/INS Filter Updates and Integrity Monitoring

Figure 4.11 shows a flow chart that outlines the algorithm for integrity monitoring in the primary GPS/INS filter. Note, the residual inspection and the state-update and reset operations are independent for each degree of freedom. With this flexibility, the yaw angle estimate can be updated when having only one of the three baselines with valid GPS attitude, in which case vehicle roll and/or pitch may not be observable.

Chapter 5

GPS/INS Attitude System Performance

This chapter gives an overview of the performance of the GPS/INS attitude system prototype discussed in Chapters 3 and 4. The required performance of this attitude system encompasses both accuracy and robustness, and individual sections are dedicated to each of these criteria. Despite having no “truth system” with which to compare system outputs during road tests, the accuracy of the system in such tests is estimated by modeling the discrete filtering process as a continuous-time linear system and using some of the measurable noise characteristics of the filter inputs and innovations to derive a likely value for the attitude-error variance. The robustness of the system is gauged as the ability to maintain the integrity of GPS attitude solutions throughout numerous road tests through suburban roads, and it will be shown that this system performs very well when compared to a commercial attitude system with loose GPS/INS coupling.

5.1 Accuracy

This section presents some analysis and results that are used to estimate the attitude accuracy of this system during road tests. The method used for this purpose relies on a study of measured filter innovations. In the following development, the observed characteristics of the innovations (including variance and time correlation) are used to model their power spectral density (PSD). Using the approximate linear relationships between the attitude measurements, estimates, and innovations, the PSD of the system attitude error is estimated and used to calculate the error variance.

It is recognized that this method of estimating attitude accuracy does not capture low frequency errors well, such as the phase delay discussed in Section 2.5. With this type of error, the GPS/INS filter converges to the GPS solution within a few seconds, and the filter innovations would show no further time-correlated error. Thereby, the technique discussed in this section is best used for dynamic test cases where the baseline experiences angular motion most of the time (such as in Figure 5.4).

The primary GPS/INS filter behaves much like a low-pass filter of GPS measurements. During time-spans on the order of tens to hundreds of seconds, the error in the system output is

dominated by time-correlated errors in the GPS solutions. This behavior is evident in details A and B of Figure 4.3, where the filter follows the slowly-changing trends in the GPS solution but attenuates high-bandwidth noise. It is also evident from these figures that the inertial system errors are much smaller than GPS errors in the short term. These characteristics can be used to study the filtering process in a simplified fashion, where the propagated state estimates are assumed to have errors dominated by their initial condition (the last updated state estimate), and errors accrued by integrating gyro rates in between updates are very small in comparison. The validity of this assumption is fairly easy to visualize with static antennas, as the filter output is practically the output of a low-pass filter with the GPS measurements as inputs. In a dynamic situation, rapid transients may occur in the GPS inputs, but the filter output would track the faster dynamics in the propagation step, and with the bandwidth provided by the inertial sensors. Conclusively, if the inertial sensors are assumed to have small error in the short term when compared to the GPS inputs, the filter error equations can be studied as though the antennas are static. This simplification is implemented by assuming that the pre-update state estimate at the current epoch is equal to the post-update estimate at the last epoch:

$$\hat{x}(-)_{i+1} = \hat{x}(+)_i \quad (5.1)$$

Figure 3.22a shows a schematic of the filtering process after this assumption has been applied. The measurement and state can correspond to any of the three GPS Euler angles, since each angle is both a measurement and a state. The Z-transform relationship between the state and the input is exactly that of a single-pole low-pass filter with unity gain at low frequencies:

$$\frac{\hat{X}(z)}{Y(z)} = \frac{kz}{z - (1 - k)} \quad (5.2)$$

The gain k is an element of the $\underline{\underline{K}}$ matrix in Eq. 4.16 that relates an Euler-angle measurement to its estimated state (k_{11}, k_{22} or k_{33}). The filter innovation is denoted δy_i and the Z-transform function that relates it to the input is a high-pass filter:

$$\frac{\delta Y(z)}{Y(z)} = \frac{z - 1}{z - (1 - k)} \quad (5.3)$$

The block diagram of the system in the continuous-time domain is illustrated in Figure 3.22b and the transfer functions are as follows:

$$\frac{\hat{X}(s)}{Y(s)} = \frac{a}{s + a} \quad (5.4)$$

$$\frac{\delta Y(s)}{Y(s)} = \frac{s}{s + a} \quad (5.5)$$

where

$$a = -\ln(1 - k)/\Delta t \quad (5.6)$$

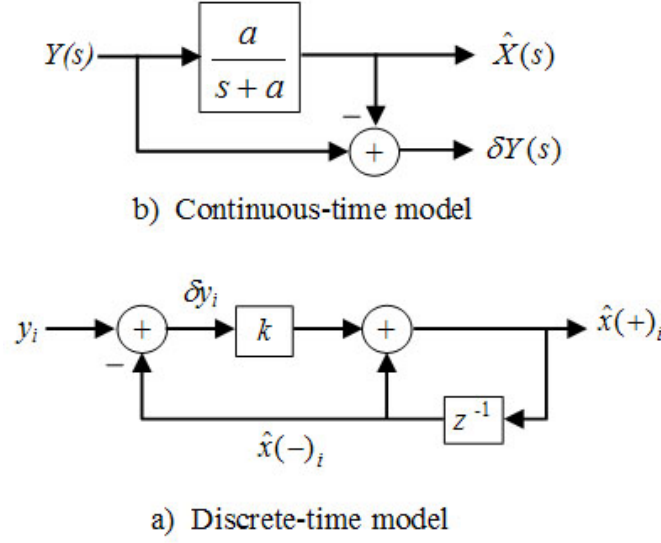


Figure 5.1: Filter Models for Error Analysis

Based on the observed error characteristics of GPS attitude measurements (see Figure 2.11) the inputs to this system can be modeled as the sum of the true value of the state (x), an exponentially-correlated noise component, and a wide-band noise component. The exponentially-correlated noise is well modeled as a first-order Gauss-Markov process (abbreviated m_y) and the wide-band noise is assumed to be Gaussian (abbreviated n_y). Using this model, the GPS measurements can be expressed in the time domain as follows:

$$y(t_i) = x(t_i) + m_y(t_i) + n_y(t_i) \quad (5.7)$$

The error in a state estimate is defined as the difference between the state estimate and true value:

$$\delta \hat{x}(t) = x(t) - \hat{x}(t) \quad (5.8)$$

Using Eqs. 5.5 and 5.7, this error can be expressed in the frequency domain, in terms of the GPS measurement components and filter gain:

$$\begin{aligned} \delta \hat{X}(s) &= X(s) - \hat{X}(s) \\ &= X(s) - \frac{a}{s+a} (X(s) + M_y(s) + N_y(s)) \end{aligned} \quad (5.9)$$

Likewise, the innovation can be expressed as a function of these terms:

$$\delta Y(s) = \frac{s}{s+a} (X(s) + M_y(s) + N_y(s)) \quad (5.10)$$

As explained previously, removing the dynamics of the system by assuming small INS errors allows the state in these equations to be represented by a nearly constant value. Thereby, the true value of the state and its low-pass filtered value are approximately equal, and its high-pass filtered value is approximately zero:

$$\delta \hat{X}(s) \approx \frac{a}{s+a} (M_y(s) + N_y(s)) \quad (\text{for bandwidth of } X(s) \ll a) \quad (5.11)$$

$$\frac{s}{s+a} X(s) \approx 0 \quad (\text{for bandwidth of } X(s) \ll a) \quad (5.12)$$

With these simplifications, Eqs. 5.9 and 5.10 become

$$\delta \hat{X}(s) \approx \frac{a}{s+a} (M_y(s) + N_y(s)) \quad (5.13)$$

$$\delta Y(s) \approx \frac{s}{s+a} (M_y(s) + N_y(s)) \quad (5.14)$$

As shown, there is a close relationship between the errors in the estimated state and the innovations of the filter. While the errors cannot be measured directly, the innovations can, and the relationship between the inputs and the high-pass filter in Eq. 5.14 can be used to approximate the characteristics of the inputs.

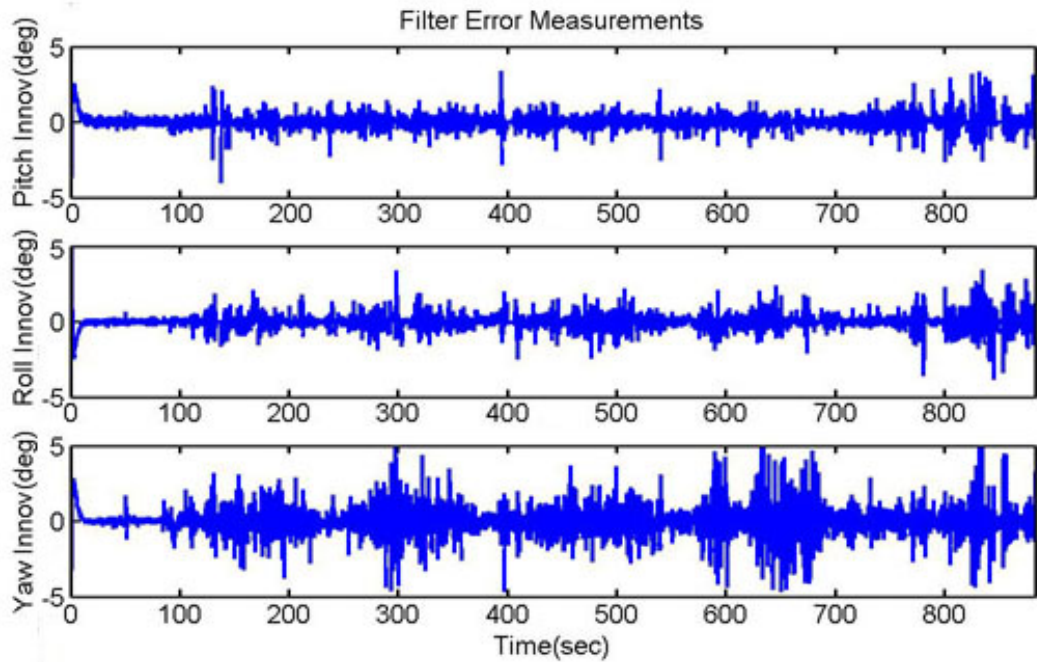


Figure 5.2: Innovation Measurements in Primary GPS/INS Filter

Figure 5.2 shows time histories of these innovations during a road test. The antennas are static at initialization, and a transient is visible as the filter converges. The clear fluctuations in variance as a function of time are a typical characteristic of the phase errors that are induced while driving through urban environments. Another important characteristic of these innovations is that they contain some time correlation; this observation agrees with the signal model in Eq. 5.14, as the high-pass filter allows some low frequencies, though attenuated, into its output. Thereby, the time correlation in the innovations can be interpreted as a diminished effect from the m_y input, and the uncorrelated noise component can be attributed to the n_y input, which is passed nearly unmodified by the high-pass filter.

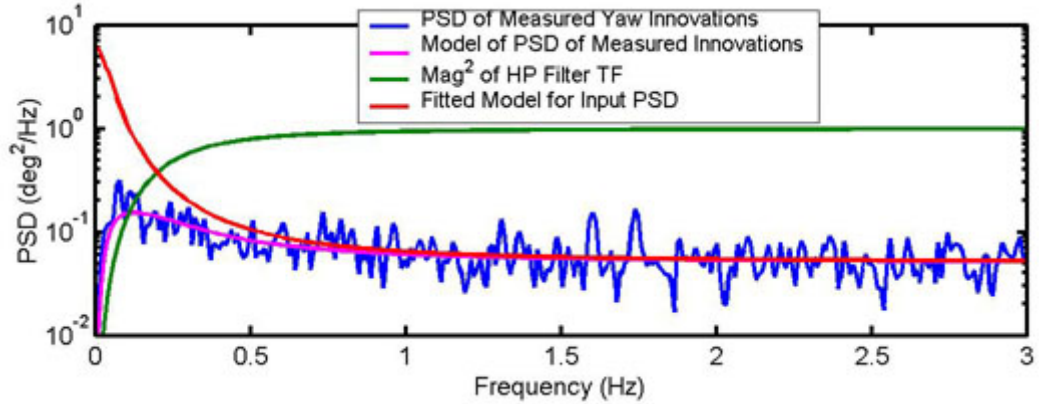


Figure 5.3: PSD Measurements and Models for Yaw Innovations

The PSD of the yaw innovations is shown in Figure 5.3, and confirms the presence of both correlated and uncorrelated components. The correlated noise is characterized by the higher spectral content at low frequencies, and the uncorrelated Gaussian noise is evident as a constant at all frequencies. As expected, the high-pass filter attenuates frequencies near DC, but sufficient correlation is still present to extract an estimate of the input PSDs. Using the PSD model of the high-pass filter (with $a=1.62$), a model of the input functions PSD ($m_y + n_y$) can be formulated such that a model for the PSD of the inputs fits the measured data. This fitted model is shown in Figure 5.3 as the red graph, and consists of the PSD of an exponentially correlated process plus that of a wide-band noise process:

$$PSD(m_y(t) + n_y(t)) = \frac{2\beta_{m_y}\sigma_{m_y}^2}{\omega^2 + \beta_{m_y}^2} + \sigma_{n_y}^2 \quad (5.15)$$

where

$$\beta_{m_y} = 0.3 \text{ sec}^{-1}, \sigma_{m_y}^2 = 0.9 \text{ deg}^2, \sigma_{n_y}^2 = 0.05 \text{ deg}^2 \quad (\text{for yaw}) \quad (5.16)$$

With this estimate of the input functions' noise content, the PSD of the state-estimate errors can

also be computed from Eq. 5.13:

$$PSD(\delta\hat{x}) = \frac{a^2}{\omega^2 + a^2} \left(\frac{2\beta_{m_y}\sigma_{m_y}^2}{\omega^2 + \beta_{m_y}^2} + \sigma_{n_y}^2 \right) \quad (5.17)$$

The variance of the estimate errors can be obtained readily from the PSD:

$$\text{var}(\delta\hat{x}(t)) = \frac{1}{2\pi} \int_{-\infty}^{\infty} PSD(\delta\hat{x}(t)) d\omega \quad (5.18)$$

The solution for this equation has a closed form:

$$\text{var}(\delta\hat{x}(t)) = \frac{a^3\sigma_{n_y}^2 + 2a^2\sigma_{m_y}^2 - a\beta_{m_y}(\sigma_{n_y}^2 + 2\sigma_{m_y}^2)}{2(a^2 - \beta_{m_y}^2)} \quad (5.19)$$

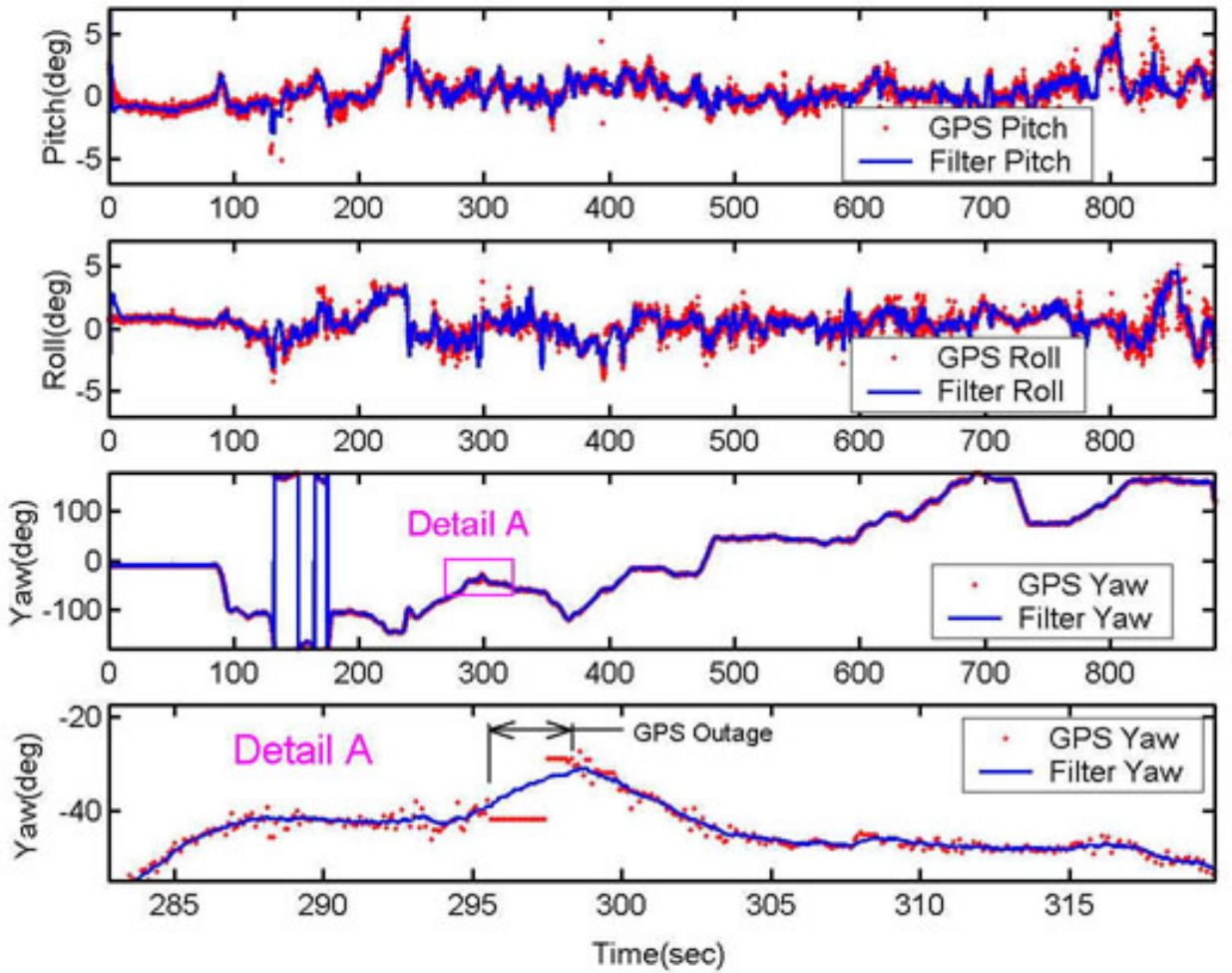


Figure 5.4: Attitude Measurements During a Road Test

For this particular data set, Eq. 5.19 yields a yaw-error variance of about 0.80 deg^2 , which satisfies the original goal of maintaining the yaw-error variance below 1.0 deg^2 . Note, this error estimate is averaged over the entire data set, so there are instances where the variance may be higher or lower than this value. Using the same technique, the estimated dynamic variances of the roll pitch error are each about 0.35 deg^2 . These accuracy estimates are representative of other dynamic data-sets obtained with the same filter gains. The difference between these accuracy estimates and those observed in static tests (such as in Figure 2.11) can be attributed to the fact that phase delay, which affects mostly pitch and roll accuracy in static tests, is more uncorrelated in dynamic conditions and is thus partially filtered out.

The actual attitude measurements that correspond to this data are shown in Figure 5.4. The graphs show the GPS measurements and the filter outputs for comparison, and a detail of the yaw graph is included to show the system behavior during a short GPS outage, during which GPS measurements are shown to remain constant.

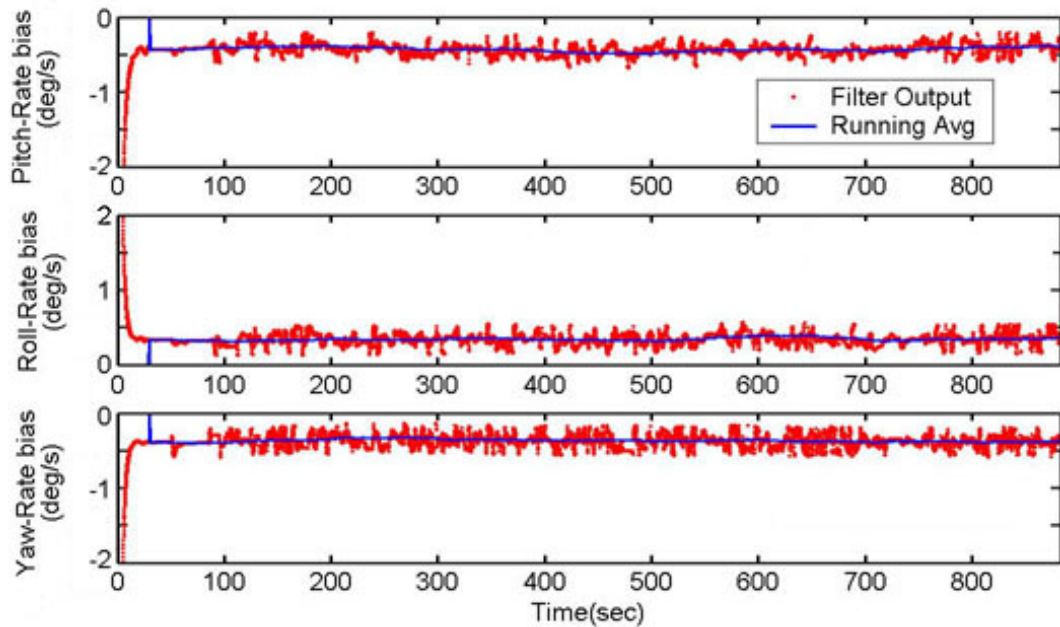


Figure 5.5: Gyro-Bias Estimates During a Road Test

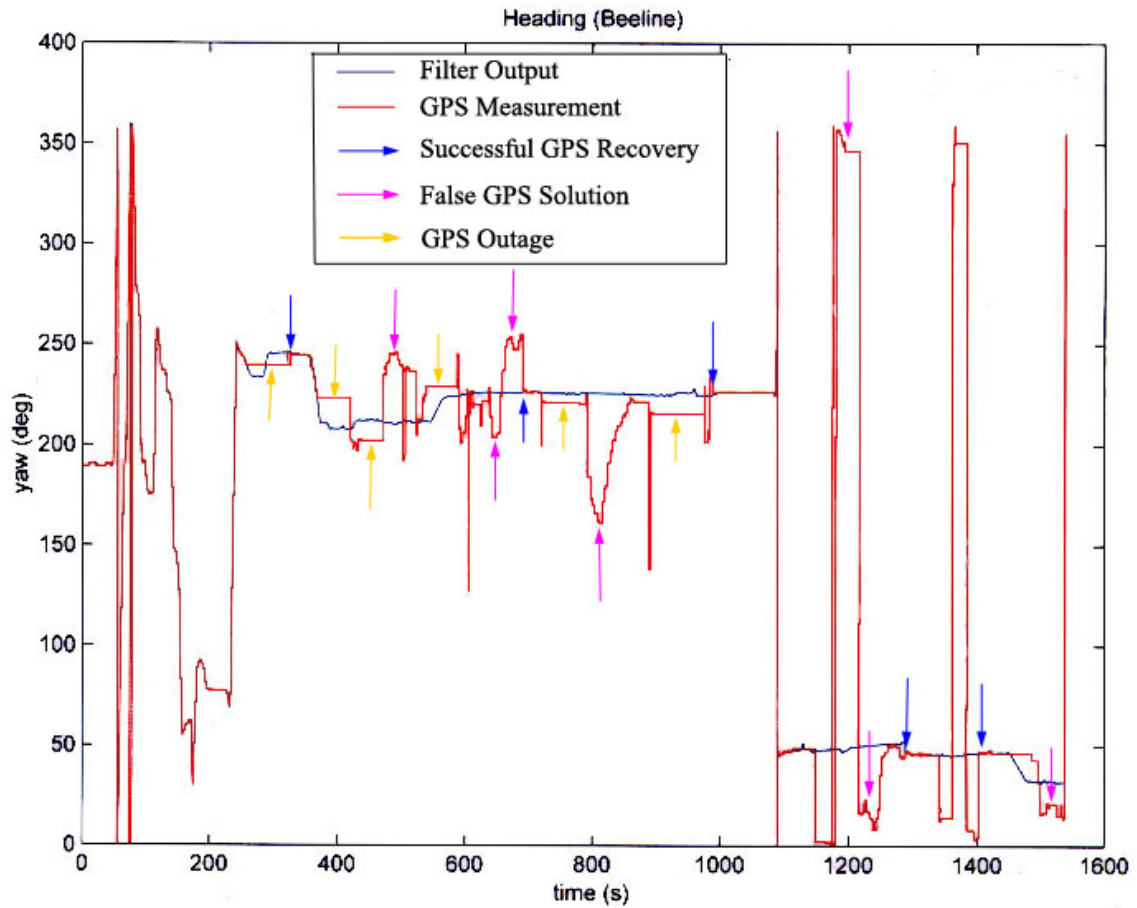
Estimation of the gyro biases in driving conditions is also an important criterion to evaluate, as errors in these estimates are the primary source of error-growth during periods of dead reckoning. Figure 5.5 shows the three gyro-bias state estimates for the same test shown in Figure 5.4. The plots show the same information as Figure 4.4, but for all three gyros and for real driving conditions. As shown, the noise in the state estimates can vary significantly, depending on the gains used, and the filter-level integrity monitors allow the bias estimates to be within 0.2 deg/sec of the running average, as explained in Subsection 4.2.4. No truth reference is available with

which to compare the accuracy of the gyro-bias estimates, but the behavior of the running average is roughly indicative of the bias-stability of automotive-grade gyros. Furthermore, the dead-reckoning performance throughout all observed GPS outages was consistent with having sufficiently accurate bias estimates.

5.2 Robustness

In the context of this attitude system, robustness refers to the prevention of undetected false GPS solutions (mostly a result of using wrong integer ambiguities) and to rapid recovery after GPS-signal outages. In systems with short baselines, assessment of the robustness of the system can be achieved with a relatively qualitative comparison of the GPS measurements and the filter outputs. Failures of integrity (i.e. robustness) in a short baseline usually cause large disagreement between the channels, which results in high residuals and/or large attitude errors. In a loosely-coupled system, integrity monitoring at the filter level (as covered in Subsection 4.2.4) can be used to reject large attitude errors. However, the absence of deeper-level integrity monitoring would prevent aiding of the GPS subsystem to recover successfully from noisy conditions and outages, and to screen for more subtle integrity failures that are not discernible as large attitude errors. Figure 5.6 shows the output of a loosely-coupled attitude system which was implemented with a Novatel Beeline attitude system. The GPS measurements and the output of the GPS/INS filter are shown in this plot. As shown, the system works very well at the beginning of the test, but afterwards there are various instances where the GPS subsystem gives false solutions that are not used by the filter, and several minutes may elapse until a successful GPS recovery occurs. During this time, the dead-reckoning attitude solution from the filter can accrue several degrees of error.

In contrast, Figure 5.7 shows the yaw measurements taken with the tightly-coupled GPS/INS attitude system discussed in Chapters 3 and 4, under similar driving conditions as those applicable to Figure 5.6. To emphasize the conditions in which this test was taken in terms of GPS signal quality, the number of satellites used to generate an attitude solution for each baseline is shown in the first plot in Figure 5.7. The frequent rapid fluctuations in the number of satellites used is caused by the various integrity monitors disabling and enabling channels with intermittent noisy measurements. The second plot shows whether the multiple-satellite solution, average one-satellite solution, or no GPS solution is used by the primary GPS/INS filter. As shown, even with only two or three usable satellites, the one-satellite yaw solution is often available. When total GPS outages do occur, they are sufficiently short to not accrue observable error through dead-reckoning.



Courtesy of Stanford Dynamic Design Laboratory

Figure 5.6: Performance of a Loosely-Coupled GPS/INS Attitude System using Novatel Beeline

It is important to emphasize the type of driving environment in which these tests were conducted. The types of urban environments tested include those in which there is continuous view of at least one useful GPS satellite most of the time. Rapid recovery of the GPS subsystem has been tested with total outages on the order of 100 seconds, but outages on the order of several minutes (two or more) sometimes accrue sufficient error through dead reckoning such that the INS-aided integer recovery is invalid, and a full integer-space search is required. Such scenarios include prolonged dwelling in an indoor garage, in very narrow urban canyons (such as downtown areas), or under very dense canopies of trees. To guarantee a successful recovery after such conditions, the current form of this prototype requires that the GPS signals are relatively free of noise to reinitialize the integer ambiguities. Obviously this condition is not always possible, and implementation of a safety-critical system based on similar technology would require further development in this area.

Another interesting empirical result obtained through various tests with this attitude system is the relative amount of time in which the two GPS attitude solutions are used. As indicated in

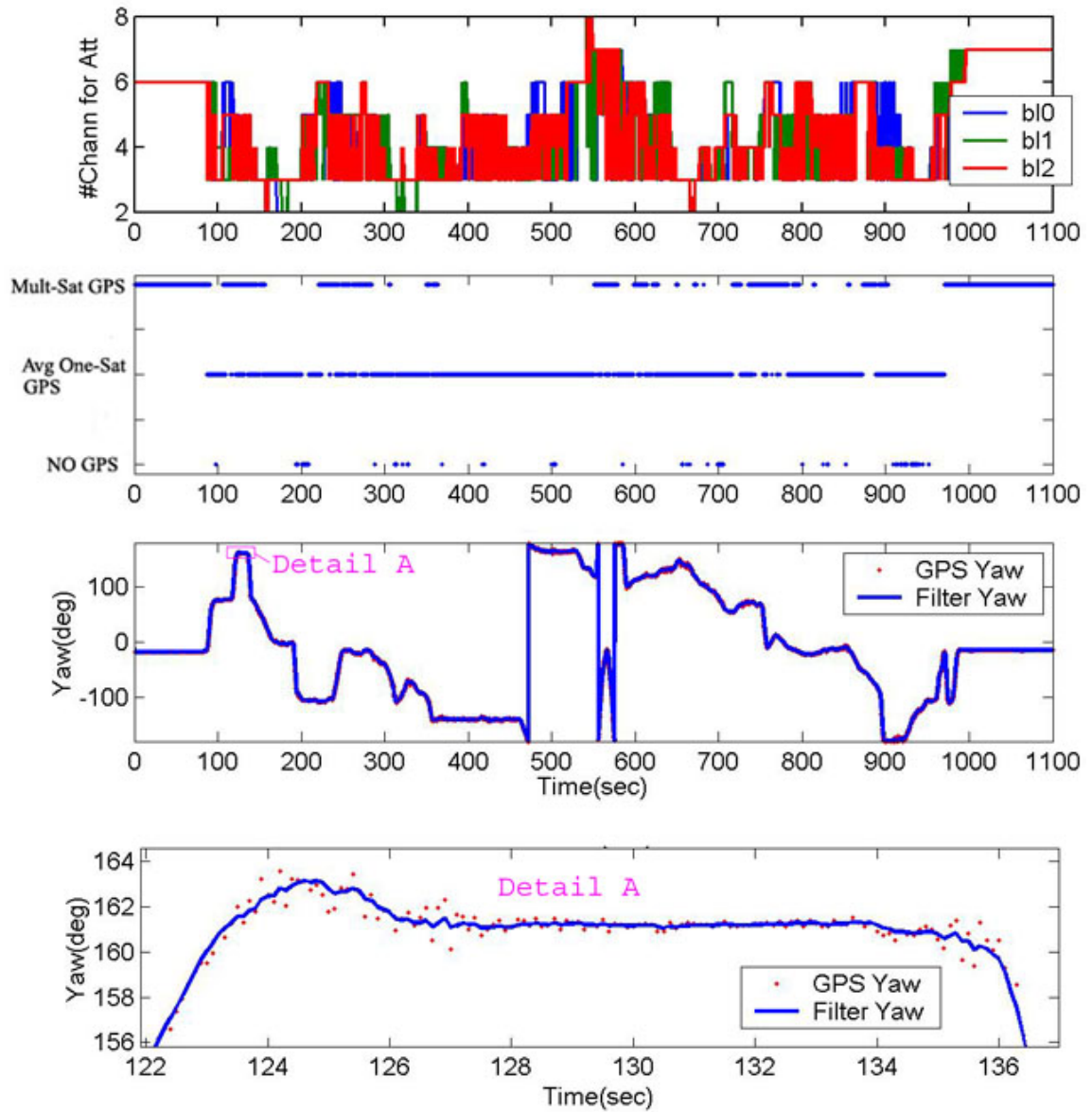


Figure 5.7: Robustness Performance During a Road Test

Figure 5.7, the average one-satellite solutions are used the majority of the time in urban environments. Note, this particular test begins and ends at the same location (and orientation) in an open parking lot to further verify the validity of the attitude solution; this location has a relatively good view of the sky, so the multiple-satellite solution is used during the stationary times near the beginning and end of the test, which do not represent the conditions of a typical dynamic driving environment. To assess the value of using the average one-satellite solution, statistics were collected from numerous similar tests (totaling about 1 hour of driving) to compile an estimate of the relative amount of time that each method of attitude determination is used and how often no GPS attitude is available. Table 5.1 summarizes the results of this study, and shows that the one-satellite solution can improve GPS yaw availability to over 90%, and the lengths of the

outages are easily within the range of dead-reckoning capability of automotive-grade gyros.

Table 5.1: Statistics of GPS Attitude Availability and Outages in Road Tests

	Mult-Sat GPS (% time)	One-Sat GPS (% time)	No GPS (% time)	Avg Outage (sec)	Longest Outage (sec)
Pitch	50%	36%	14%	0.5	11
Roll	50%	38%	12%	0.4	11
Yaw	50%	45%	5%	0.1	8

5.3 Limitations of Tight Coupling

Chapter 4 presented a detailed description of the measures that must be taken to provide robustness to the system output of an integrated GPS/INS attitude system. The principal idea behind all the techniques for filtering and integrity monitoring under tight coupling is the necessity to favor the stability of the inertial sensors in the short-term instead the fragile carrier-phase outputs of the GPS receivers. However, the GPS outputs must also be used as much as possible to provide long-term stability. Assuming that fault-detection at the channel level and beyond is sufficient to discriminate good GPS solutions from poor ones, the remaining fundamental failure point of this design involves situations where GPS carrier-phase outputs remain unusable for extended periods. This situation does not necessarily imply that no GPS signals are visible, but rather that the carrier-phase outputs are too noisy to track properly and/or with unknown integer ambiguities following one or numerous cycle slips. Thereby, further improvements to system integrity that target this limitation must be made to the phase-tracking ability of the GPS receivers. As mentioned in Section 1.2, this type of enhancement can be realized through ultra-tight integration, which can improve the accuracy and robustness of a GPS phase-lock loop in terms of phase noise and cycle slips. The details of this powerful technique are explored in the next chapter through a method designated as Doppler-aiding, which can be considered one of the more straightforward variants of ultra-tight coupling.

Chapter 6

Inertial Aiding of GPS Tracking Loops

This chapter explores GPS/INS coupling at its deepest level, where the GPS tracking loops utilize inertial aiding to track GPS signals received by a moving platform. This level of integration requires considerably more effort than loose or tight coupling, as the architecture of the GPS tracking-loops (particularly the PLL) is different from that in a traditional receiver. For this reason, a typical end user does not currently have the flexibility to implement ultra-tight coupling with a standard commercial GPS unit, as existing receivers do not provide such flexible features. Realization of a real-time ultra-tightly coupled system requires access to the control software of the receiver, and thus entails development of very customized hardware. The required capital for this type of effort is typically beyond the relatively modest means of academic researchers, so efforts to produce real-time, ultra-tightly coupled systems are currently being pioneered by private corporations and military researchers.

Some organizations have published limited information that outlines their current efforts, including Interstate Electronics Corporation [6], The Aerospace Corporation, and Draper Laboratories [40]. The approach taken in their implementations involves the use of a single large filter, or smaller federated filters (one per channel) that use in-phase and quadrature samples from the GPS receiver channels as measurements for updating the filter states. Control of the replicated carrier and code generators also comes from navigation filter outputs, which are propagated with IMU measurements to achieve high bandwidth carrier-phase tracking and anti-jam capability. This more complex architecture variant of ultra-tight integration is also termed “deep integration”, and is characterized by implementing the closed-loop signal tracking for all channels through the navigation filter itself, thus precluding the need to maintain separate code and carrier tracking loops [56]. The details of these implementations are not discussed in this chapter, though Appendix D contains a discussion on the development of a PLL estimator which could comprise a channel-level subset of a global filter.

The integration structure presented in this thesis was first introduced in [3] as a simpler approach to ultra-tight integration that utilizes classical control theory. To differentiate this technique from the other various forms of ultra-tight coupling, some researchers refer to this method as “INS-aided GPS tracking loops” [34] or simply “Doppler aiding”. The concepts that will be

presented can be used to study the theory and performance improvements of a GPS/INS navigation system with ultra-tight integration, without the added complexity of global estimation. The straightforward architecture of the system is not limited to be a simple learning tool, but can also potentially be used to realize a system with all the important benefits of ultra-tight coupling. Such a system was implemented in software as a testbed for the concepts discussed in this chapter, and is strictly a post-processing tool for tracking recorded real or simulated GPS signals with or without inertial aiding.

The remaining sections of this chapter will focus on the design of a GPS/INS system that incorporates inertial aiding of GPS tracking loops. The GPS receivers of the system feature a modified phase-lock loop that utilizes external Doppler estimates, and a rate-aided delay-lock loop for maintaining code tracking during brief signal outages. Although much of the discussion involves details of the GPS tracking loops, some attention is also given to external Doppler-frequency and clock-error frequency estimation. Other important design issues such as GPS receiver clock and inertial sensor quality are also covered, and evaluated versus expected system performance.

6.1 GPS Tracking Loops

This section reviews some of the basic concepts of traditional GPS phase and code tracking loops, including modeling and implementation. The ideas that will follow in the development of tracking loops that incorporate inertial aiding will build upon these traditional loop structures to realize the structure of a GPS receiver with Doppler aiding. More detailed explanations and analysis of GPS tracking loops can be found in [19] and [62].

Figure 6.1 shows a schematic of the basic components of a typical GPS receiver, and identifies the location of the PLL and DLL. These loops are responsible for tracking the carrier and code components of a GPS signal after the acquisition phase. As labeled in the figure, these control loops would be implemented in a correlator chip and a microprocessor. The functions of correlator chip include mixing the GPS signal with local replicas of the channel code and carrier components (code and carrier wipeoff), and providing measurements of the phase and code offsets (between signal and replica) for each of these signal components; the microprocessor uses these measurements and executes the control-logic that defines the dynamic characteristics of the signal-tracking process. In addition, the microprocessor also contains the software that constructs PR and ICP values from tracking-loop measurements and ultimately generates position, velocity and time (PVT) solutions.

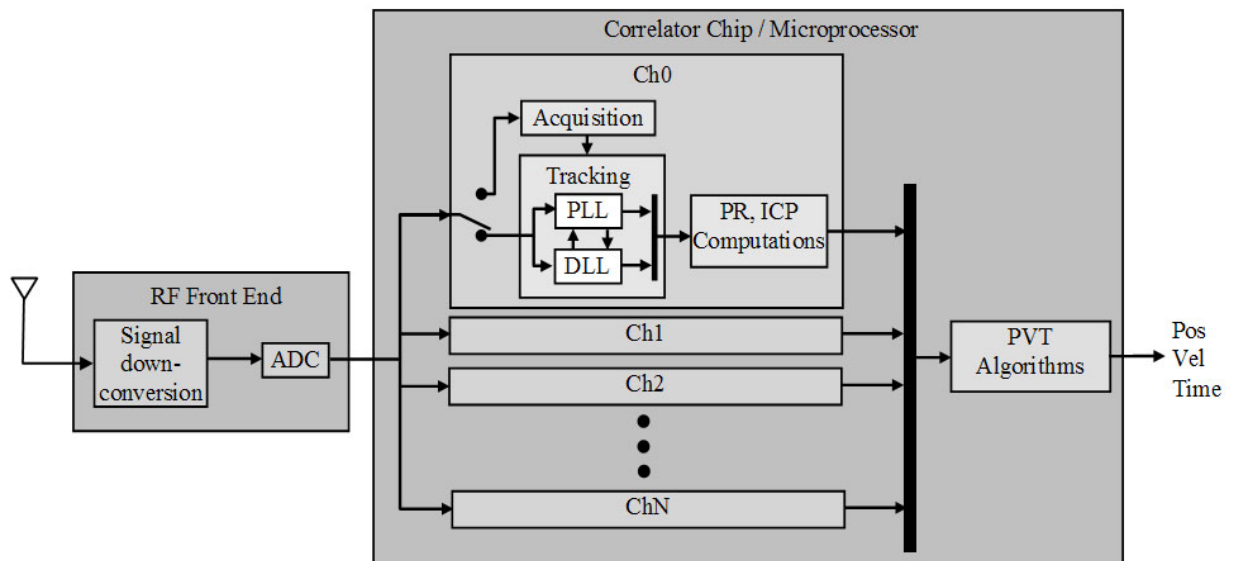


Figure 6.1: Schematic of a GPS Receiver

6.1.1 The Phase Lock Loop

The Doppler frequency and carrier-phase of a GPS signal are tracked by the phase-lock loop. The Doppler frequency is one of the components needed to properly mix the downconverted GPS signal to baseband, such that the bits of the GPS navigation message can be extracted. The phase of the carrier is tracked in order to identify the transition between navigation bits, as the changes are encoded as phase shifts in the carrier (see Figure 1.1). The precise carrier replica and phase-offset measurements that are generated as part of this process also provide a means for constructing the ICP measurement, which makes GPS attitude and carrier-phase positioning possible.

A PLL is designed to track the dynamic variations in Doppler frequency that occur as a result of relative motion between the satellite and receiver. Satellite velocity results in large but slowly-changing Doppler, while the antenna motion of a moving vehicle causes smaller but more rapidly-changing Doppler. The bandwidth of the PLL determines the dynamic range of receiver motion that can be tolerated without losing track of the carrier-phase. However, this bandwidth cannot be set to an arbitrarily large value, since wide-band phase-noise is amplified with increasing loop bandwidth. This tradeoff between dynamic tracking and noise suppression is the crux of the design of the PLL loop filter.

The PLL also tracks changes in the carrier frequency that originate from local oscillator instability, as these errors are introduced through the downconversion and sampling of the analog GPS signal. While such frequency variations are undesirable, they become part of the downconverted GPS signal within the receiver and they must be tracked by the PLL. In a given channel,

these local oscillator dynamics are indistinguishable from satellite-clock dynamics or Doppler frequency changes. The one characteristic that differentiates the local oscillator contribution is that it is common to all the channels, and can thus be resolved as a common-mode term when utilizing the Doppler frequencies from multiple channels to solve for user velocity. The impact of the local oscillator on PLL design would be more pronounced in a receiver that is known to be stationary and has PLLs fittingly designed for low-bandwidth motion and high noise-suppression. In such a case, the PLL bandwidth would have a lower limit set by the stability of the local oscillator. The same concept applies to a loop that receives Doppler aiding from an inertial subsystem, as most of the motion dynamics of the receiver will be captured by the inertial sensors and the PLL would only need to track frequency variations caused by the local oscillator and errors in the external Doppler estimate. This idea will be explored further in this chapter.

Most modern GPS receivers have second-order PLLs [19] with loop bandwidths of 15-30Hz, which accommodates the dynamic necessities of most terrestrial and aerial vehicles. Higher-order loops are only necessary in certain specialized high-bandwidth applications, such as carrier-tracking in fighter aircraft [34]. Since automobile navigation does not entail very rapidly-changing dynamics, the treatment of PLLs in this thesis is limited to second-order loops. Although the implementation of the loop is done in the discrete domain, the sampling frequencies (50- 1000Hz, depending on the averaging time of the correlators) are usually much greater than the desired bandwidth of the loop, and allow design and analysis in the continuous domain. However, design in the discrete domain may be necessary if long averaging times that result in low sampling frequencies are used in conjunction with high-bandwidth loops. The analysis and results that are presented in this thesis assume the minimum averaging time of 1ms, which is equivalent to using a 1000Hz sampling frequency and allows for loop design in the continuous domain.

Figure 6.2 shows a block diagram of a second-order PLL model. Such a model is a traditional representation of a PLL in the continuous frequency domain. The main components of the loop are the phase discriminator, the loop filter or controller, and the NCO which represents the plant of the system. The main input to the loop is the phase of the reference signal ($\varphi_r(s)$), and the second input ($w_\varphi(s)$) represents external phase-noise sources that do not need to be tracked and degrade the accuracy of the replicated carrier. Such noise sources may include multipath, RF interference, or wideband phase-noise in the reference carrier. The output is the phase of the replicated signal ($\varphi_{PLL}(s)$). The NCO is modeled as an integrator, as it relates changes in frequency to changes in phase. The phase detector performs a difference operation between the phases of the reference and output signals, and this error signal ($\delta\varphi$) is the input to the loop filter. The loop filter generates a control signal ($f_{PLL}(s)$), an input to the plant, that represents a change in oscillator frequency needed to correct phase offsets. The goal of the loop filter is to maintain

the phase offset at zero, which also implies maintaining the frequency of the replicated signal locked to that of the reference signal. This loop does not include the IF component of the carrier, so the dominant components of $f_{PLL}(s)$ in steady-state tracking are the Doppler and local-clock frequency error, as represented in the following equation:

$$f_{PLL} = f_{dopp} + f_{clk} + f_{noise} \quad (6.1)$$

There are many choices available for the configuration of the loop filter. One of the most widely used is a proportional-integral (PI) compensator, which makes the closed-loop system a second order PLL. This same type of controller will be utilized in the PLLs discussed in this chapter. The gain (k_1) and loop-filter zero ($1/\tau_1$) are the design parameters of the loop, and determine the dynamic characteristics of the phase-tracking process.

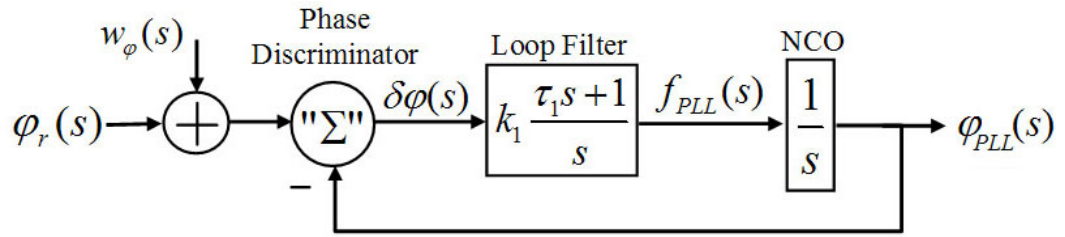


Figure 6.2: Model of a Second Order PLL

The output phase is related to the sum of the two inputs by a second order transfer function:

$$\varphi_{PLL}(s) = H_1(s) (\varphi_r(s) + w_\varphi(s)) \quad (6.2)$$

where

$$H_1(s) = \frac{k_1(\tau_1 s + 1)}{s^2 + k_1 \tau_1 s + k_1} \quad (6.3)$$

The design process of the loop filter may specify a desired damping ratio and closed-loop bandwidth. To simplify the design process, a damping ratio of can be chosen to establish a relationship between the two design parameters. For a damping ratio of 0.707, the relationship is:

$$\tau_1 = \sqrt{2/k_1} \quad (6.4)$$

Using this or a similar relationship, the more important bandwidth parameter is only a function of the gain k_1 . Bandwidth is commonly quantified as either the -3dB bandwidth (B_{3dB}) or the noise-equivalent bandwidth (B_n). The -3dB bandwidth is the frequency at which the closed-loop transfer function magnitude is 3dB below the magnitude at zero frequency. The one-sided, noise-equivalent bandwidth is more commonly used in literature pertaining to PLLs, and is defined as

follows for a closed-loop transfer function $H(s)$:

$$B_n = \frac{1}{2\pi} \int_0^\infty |H(j\omega)|^2 d\omega \quad (\text{Hz}) \quad (6.5)$$

For the transfer function in Eq. 6.3 and using the relationship in Eq. 6.4, the bandwidth expressions for the second order PLL are:

$$B_{3dB_{1\varphi}} = \sqrt{(2 + \sqrt{5})k_1} / 2\pi \approx 0.32\sqrt{k_1} \quad (\text{Hz}) \quad (6.6)$$

$$B_{n\varphi_1} = \frac{3\sqrt{k_1}}{4\sqrt{2}} \approx 0.53\sqrt{k_1} \quad (\text{Hz}) \quad (6.7)$$

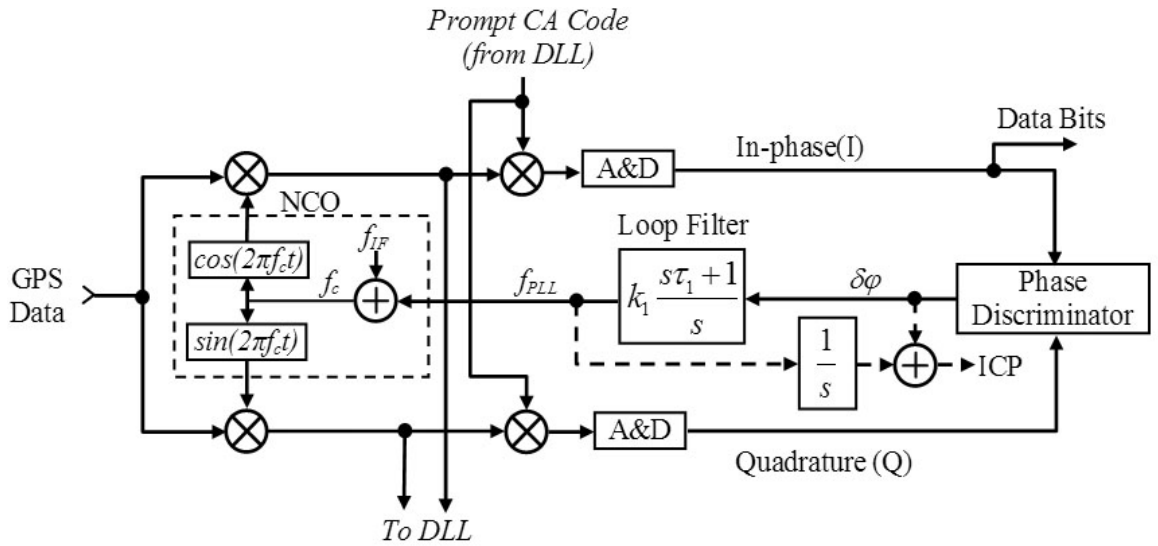


Figure 6.3: Implementation of a PLL

A PLL is constructed from hardware and software components, with a structure similar to that shown in Figure 6.3. The NCO generates in-phase and quadrature estimates of the down-converted carrier signal, at a frequency equal to the sum f_{PLL} and f_{IF} . These estimates are mixed with the incoming GPS samples and with a prompt replica of the channel's PRN code. The PRN code replica comes from the coupled DLL, which must achieve code lock before the PLL can track. These operations result in measurements of the cosine and sine of the phase-offset measurement (in the in-phase and quadrature branches). The cosine and sine of the phase offset are commonly abbreviated I and Q , respectively. The blocks labeled "A&D" are "Accumulate and Dump" operations, which perform averaging of the correlation results for at least one millisecond (one code period). In steady-state tracking, the in-phase branch contains resolved estimates of the navigation bits, and the quadrature branch approximates the value of small phase offsets. The phase discriminator uses the I and Q measurements to compose the phase-offset measurement.

Different types of discriminators can be used for this purpose; two of the more commonly used are arctangent and Costas discriminators, given by the following equations:

$$\delta\varphi = \arctan\left(\frac{Q}{I}\right) \quad (\text{Arctangent discriminator}) \quad (6.8)$$

$$\delta\varphi = QI \quad (\text{Costas discriminator}) \quad (6.9)$$

The ICP measurements used in Chapters 2-5 also originate in the PLL. An ICP measurement from time t_A to time t_B can be constructed by using the phase of the carrier replica at t_A as an initial condition, adding the total whole-number of carrier-cycles elapsed until t_B , and finally adding the phase of the carrier replica at time t_B . If using the unmodified replicated carrier, this integrated value will include integrated cycles of the IF. This form of the measurement is certainly usable, but would also have the unnecessary component of the integrated IF. For this reason, the ICP measurement reported to the user does not normally include this component, but only the equivalent of an integrated carrier with frequency f_{PLL} . To maintain accurate ICP measurements during dynamic transients that cause measurable misalignment of the replicated carrier, the phase-offset measurement can be part of the ICP. Figure 6.3 includes this equivalent representation of an ICP measurement, but it should be noted that it may not necessarily be constructed in this manner in a GPS receiver.

6.1.2 The Delay-Lock Loop

The function of the DLL is to track the CA code component of the GPS signal. In addition to generating the prompt code needed for tracking the carrier in the PLL, the code phase from the code generator is also the ranging signal used to determine the pseudoranges for position determination. Although the structure of the DLL will not change with the implementation of Doppler-aiding, some explanation is necessary as to why a rate-aided DLL should be used to complement the robustness enhancements in the PLL achieved through this level of integration.

The specific type of DLL that will be emphasized in this section is a noncoherent DLL. The “noncoherent” characteristic implies that the incoming navigation data-bits are not known, so the I and Q branches must be squared and added to ascertain a power measurement from the early and late correlators. This configuration keeps the operation of the DLL independent of the short-term tracking of the PLL, as only a rough estimate of f_{PLL} (within a few hundred Hz), not the carrier-phase, is necessary to discern a distinct correlation peak in the early and late correlators. The penalty for squaring I and Q is an loss in SNR, as the noise in the signal is also squared.

Figure 6.4 shows the model of a noncoherent DLL. The code generator (CG) produces a replica of the CA code at a frequency imposed by the loop compensator, which is usually a

simple gain. The error signal in this model ($\delta\tau(s)$) represents the difference between the phase of the reference CA code ($\tau_r(s)$) and that of the replicated CA code ($\tau_{DLL}(s)$). The bandwidth of the loop is set by the gain of the compensator, and can be much lower than that of the PLL. A lower bandwidth ($\approx 1\text{Hz}$) and lower-order loop can be used in the DLL because the dynamics of the code-phase are less influenced by user motion than the carrier phase. This distinction is a result of the difference in the lengths of a carrier cycle (19cm) and a CA code chip (300m), which span the range of measurement of the carrier and code discriminators, respectively.

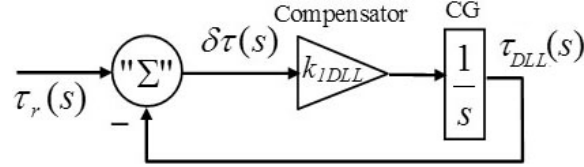


Figure 6.4: Model of a First Order DLL

For code-ranging precision, it is advantageous to have the bandwidth of the DLL as low as possible. Code-rate aiding from the PLL is a well-known method that allows further reduction of the DLL bandwidth. This technique consists of using the f_{PLL} frequency to obtain an estimate of the rate of the code-phase. Using this rate aiding, the DLL only has to track corrections to the estimate and can use a lower loop-gain to do so, as long as the error dynamics of the code-rate estimate are slower than those of the code-rate itself. In this case, the rate of code-carrier divergence would be a driving factor in determining the bandwidth of the DLL [19].

Figure 6.5 shows the model of a rate-aided PLL. The gain k_{CDLL} transforms f_{PLL} to the offset code-rate from the nominal of 1.023MHz, and is a function of various constants:

$$k_{CDLL} = \frac{\lambda_{L1} 1.023 \times 10^6}{c} \approx 6.50 \times 10^{-4} \quad (6.10)$$

Note, this relation is valid as long as the early minus late power measurement is normalized by the prompt power of the signal. Since this type of DLL uses f_{PLL} for code tracking, performance of the loop can also benefit from providing external Doppler estimates to the PLL. If estimates of f_{PLL} are less noisy and more robust as a result of external Doppler aiding, then the code-rate estimates will reflect the same benefits. In addition to improved noise performance, a rate-aided DLL can provide a synchronized code despite brief outages in the GPS signal, as long as an external frequency aiding is uninterrupted and the outages are brief. This advantage will be exploited when external Doppler aiding is integrated with the tracking loops.

The code-rate derived from f_{PLL} results in a very good approximation of the required rate of the local code generators, such that the replicated code remains aligned with the reference code. However, the closed-loop configuration is still needed to track the slow and unpredictable drifts

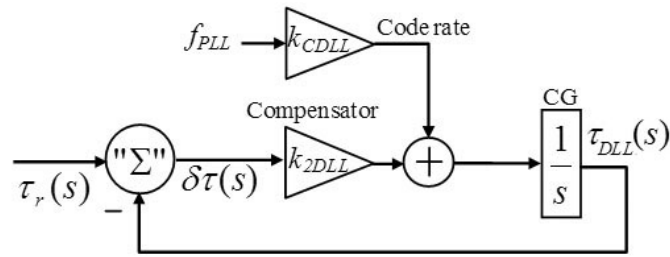


Figure 6.5: Model of a Rate-Aided DLL

caused by code-carrier divergence [19], which is caused by slight differences in delay of different frequencies through the ionosphere.

The implementation of a rate-aided DLL is illustrated in Figure 6.6. The noncoherent characteristic of this loop is evident from the squaring operations that follow the A&D blocks.

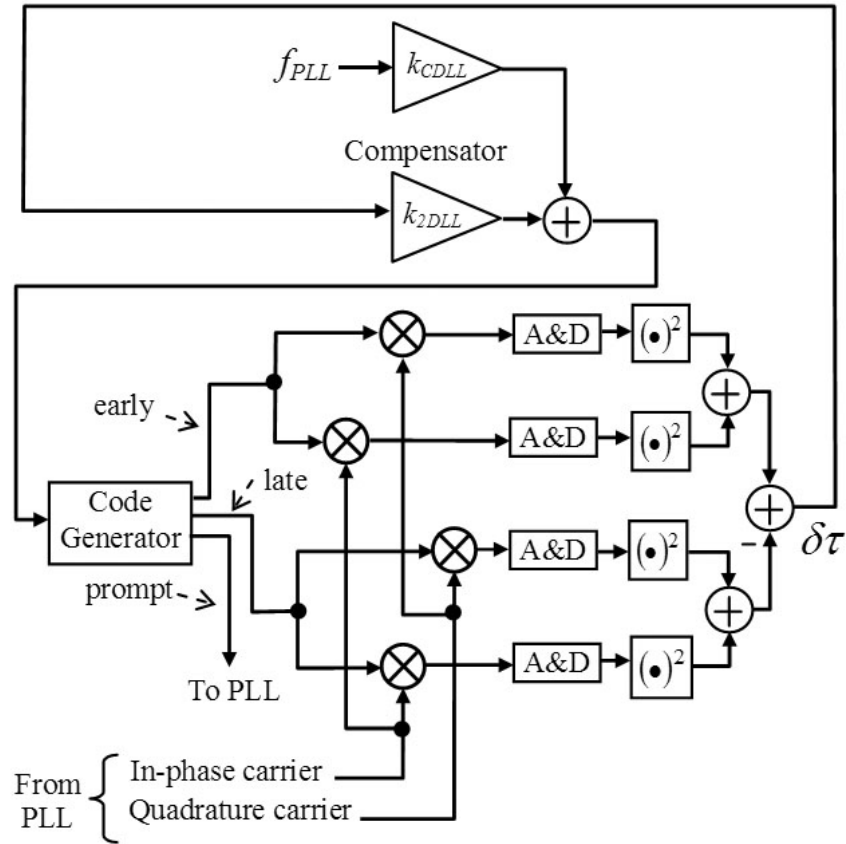


Figure 6.6: Implementation of a Rate-Aided DLL

6.2 Doppler and Clock Error-Frequency Estimation

As explained in Section 6.1, the main components of the frequency tracked by a PLL are the Doppler and clock-error frequencies. If these components can be estimated externally, they can be used to aid the PLL to track with more efficiency and robustness. The Doppler frequency of the carrier can be estimated from the outputs of an INS and satellite ephemeris data. With the use of the accelerometers in an IMU, a GPS/INS attitude system such as that discussed in Chapters 3-5 would contain all the necessary components of a navigation system that can be used to estimate Doppler frequencies and implement Doppler-frequency aiding to the GPS tracking loops. In a typical GPS receiver with a TCXO, the clock error-frequency is primarily composed of errors from the local oscillator. However, much smaller frequency-errors in the satellite clock are also present, and with a single channel, are indistinguishable from effects of the local oscillator.

This section discusses methods for estimating the Doppler and local clock error-frequencies. The methods that will be discussed are used later in this chapter to develop some of the components in a GPS receiver with Doppler aiding.

6.2.1 Doppler-Frequency Estimation

The Doppler frequency of the carrier signal can be expressed simply as the velocity of the receiver antenna relative to the satellite, projected onto the LOS vector and scaled by the carrier wavelength. The relationship is expressed as follows:

$$f_{dopp}^k = \frac{1}{\lambda_{L1}} \left(\vec{V}^{RX} - \vec{V}^k \right) \cdot \vec{I}^k \quad (6.11)$$

where

$$\begin{aligned} \vec{V}^{RX} &= \text{Velocity of receiver antenna in an Earth-fixed frame} \\ \vec{V}^k &= \text{Velocity of SV\#k in an Earth-fixed frame} \end{aligned}$$

The left superscript on the velocity symbols denotes the reference frame in which the velocity is measured. The lower-case e denotes an ENU frame as before, but any other Earth-fixed frame can be used, such as the ECEF frame (denoted by upper-case E).

The computation of the ECEF satellite velocity from ephemeris data is a well-known procedure, described in [46]. Another very simple and more accurate method for computing satellite velocity is to estimate the time derivative of its position. The advantage of this technique is being able to capitalize on the second-order corrections to SV position calculation that are included in the ephemeris message. The computation of the ECEF satellite position from ephemeris data

is described in [5]. Using this method to compute the satellite position as a function of GPS time-of-week, the velocity of the satellite can be approximated as the discrete time-derivative of its position, with a small time-step ($\Delta t \approx 1ms$):

$${}^e\mathbf{V}_E^k(t) \approx \frac{{}^o\mathbf{R}_E^k(t + \Delta t) - {}^o\mathbf{R}_E^k(t)}{\Delta t} \quad (6.12)$$

where

$${}^o\mathbf{R}_E^k = \text{Vector from ECEF origin to SV\#k expressed in ECEF basis}$$

The ENU or ECEF velocity of the receiver antenna is one of the common measurements from GPS receivers, and comes from Doppler measurements of the channel PLLs. However, an integrated GPS/INS system can provide velocity measurements with sufficient bandwidth and time stability to aid the PLL through an intermediate Doppler calculation. Since the inertial sensors provide accelerations in a body-fixed frame, an attitude reference is needed to transform them to an Earth-fixed frame. The attitude system can be implemented in a number of ways, including an unaided inertial navigator (provided it has relatively stable rate-gyro and accelerometers), or a loosely or tightly-coupled GPS/INS system like that discussed in Chapters 3-5. The latter option can be significantly less expensive, but more difficult to construct.

The receiver-antenna velocity is the crucial term that links inertial measurements to the GPS tracking loops. As mentioned earlier, the high-frequency dynamics of the Doppler are primarily caused by receiver motion, and if these dynamics can be estimated by means of an INS, the effort to track them with a PLL can be reduced. This concept is practical only if the error dynamics of the Doppler estimates are slower than those of the Doppler itself, such that the bandwidth of the PLL can be reduced to improve noise performance. Of course, the error dynamics of the Doppler estimates are dependent on the type of inertial sensors used, level of GPS availability, and on the length of time that dead-reckoning (for both velocity and attitude) is employed. A study of these error dynamics under different circumstances of GPS availability is included later in this chapter, as part of the discussion on selecting a Doppler-aided PLL loop-bandwidth and quantifying its performance.

6.2.2 Clock Error-Frequency Estimation

The clock error-frequency component of f_{PLL} is part of the GPS velocity computation. Since it is a common error term on all the channels, four or more satellites are necessary to solve for three components of the user velocity (V_x, V_y, V_z) and the clock error-frequency, as shown in the

following equation:

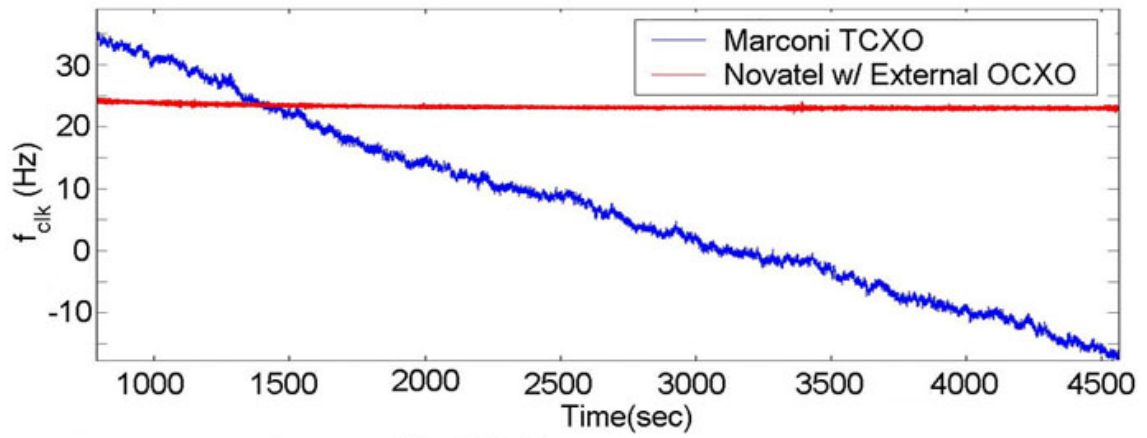
$$\begin{bmatrix} -f_{PLL}^1 \lambda_{L1} + \vec{V}^1 \cdot \vec{1}^1 \\ -f_{PLL}^2 \lambda_{L1} + \vec{V}^2 \cdot \vec{1}^2 \\ \vdots \\ -f_{PLL}^N \lambda_{L1} + \vec{V}^N \cdot \vec{1}^N \end{bmatrix} = \begin{bmatrix} \vec{1}^{1^T} & 1 \\ \vec{1}^{2^T} & 1 \\ \vdots & \vdots \\ \vec{1}^{N^T} & 1 \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ V_z \\ V_{clk} \end{bmatrix} \quad (6.13)$$

where

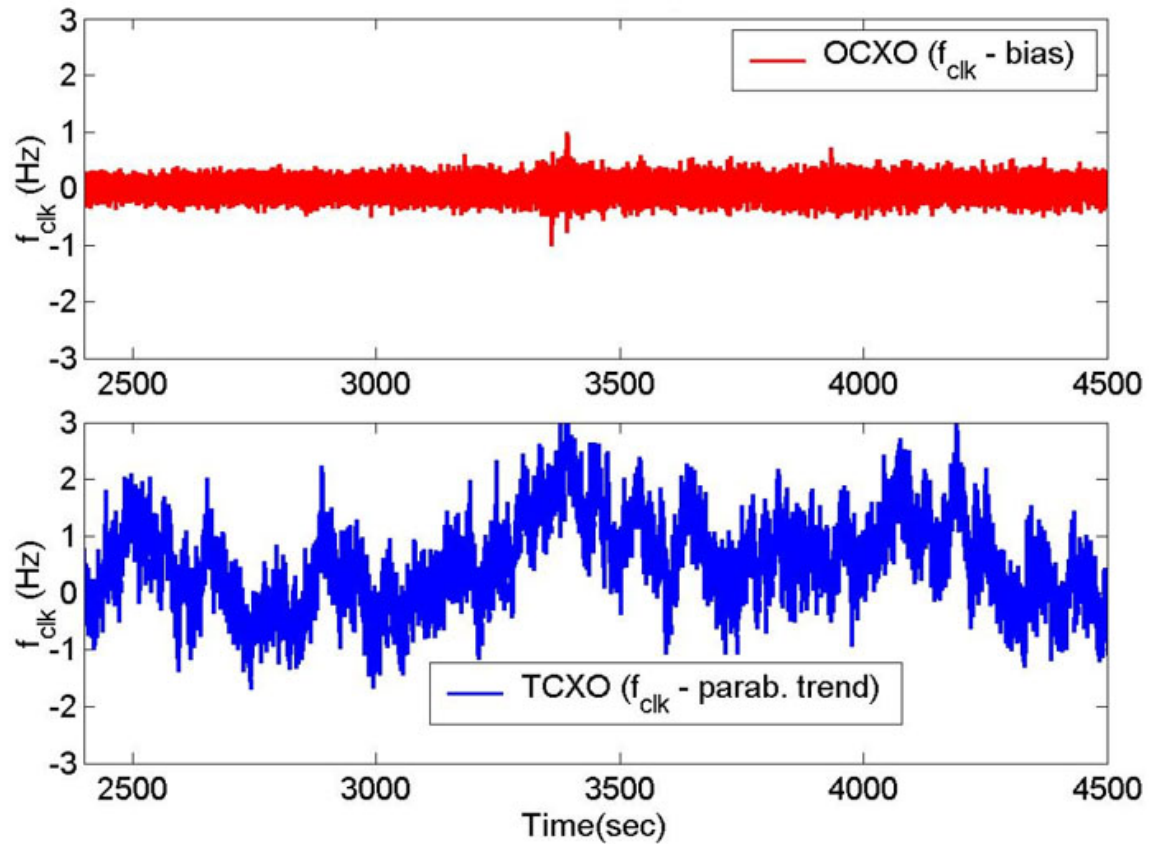
$$V_{clk} = \lambda_{L1} f_{clk} \quad (6.14)$$

The measurements of user velocity and clock error-frequency are obtained with a least-squares estimate of the vector of unknowns. Note, no basis is specified for the vectors of this matrix equation, since any Earth-fixed basis can be utilized as long as it is used consistently.

The behavior of the clock error-frequency varies significantly with the type of reference oscillator in the GPS receiver. Figure 6.7a shows measurements of f_{clk} for a common TCXO used in most consumer GPS receivers, and for a higher quality OCXO. As shown, the OCXO has very little change in frequency over time after an initial transient, and f_{clk} is composed primarily of a constant bias and broadband noise. In contrast, the TCXO has a distinct frequency drift over time that may span hundreds of Hertz over a few hours, and a more prevalent component of exponentially-correlated noise. The difference in exponentially-correlated frequency noise between the two clocks is depicted more clearly in Figure 6.7b, which shows the same comparison with the bias and parabolic trend removed from the OCXO and TCXO data, respectively. Broadband noise in the measurements is also clearly visible in both plots, and suggests that low-pass filtering these measurements before using them to aid the PLL would be beneficial.



a) Unmodified Clock Frequency-Error Measurements



b) Clock Frequency-Error Measurements Corrected for Bias or Parabolic Trend

Figure 6.7: Measurements of Clock Error-Frequency with a TCXO and an OCXO

6.3 Development and Design of Doppler-Aided Phase-Tracking Loops

This section introduces an approach to ultra-tight coupling based on the concept of feed-forward to reconfigure the PLL model. Feed-forward is used in classical control theory when a component or form of the reference signal can be measured, and processed in some way to aid the controller. A textbook example of this idea is when the DC bias of the reference signal is known or can be measured, thus precluding the need to track DC with the controller. The known bias is multiplied by the inverse of the plant model and added to the output of the controller. This scheme allows the use of a controller that requires less power, since its output would no longer contain a DC bias.

In the context of a rate-aided PLL, feed-forward is useful for constructing a model of the loop, analogous to Figure 6.2, that incorporates external frequency aiding. This architecture will change the fundamental function and of the PLL, and the design of the loop filter will be intrinsically coupled to the characteristics of the INS subsystem that provides Doppler estimates. Therefore, this level of GPS/INS integration can be considered a form of ultra-tight coupling. To relate this form to other variants of ultra-tight coupling which utilize optimal estimation, a discussion is included in Appendix D that presents a method for implementing a frequency-aided PLL in state-space form, with an estimator whose state-vector includes in-phase and quadrature phase estimates.

6.3.1 Doppler-Aided PLL Model

The feed-forward representation of a PLL model with frequency aiding utilizes the fact that the sum of external Doppler and clock-error frequencies is a measurement of the *rate of change* of the reference carrier-phase. As indicated in Eq. 6.1, these two frequencies are the two major components of loop-filter output, which is used to control the NCO and maintain phase lock. Figure 6.8 shows the model of a rate-aided PLL, with the feed-forward branch originating from the signal to be tracked. This model is analogous to that in Figure 6.2 for a standard PLL, and its purpose is only to obtain the transfer functions that *model* the behavior of a Doppler-aided PLL, not to depict a physical realization. To clarify this point, the gray background in the figure encompasses components and signals that do not reflect a physical architecture, as a designer would not have access to these signals. In a physical implementation, a designer would only have access to the components and signals outside the gray block, such as the signal labeled “External frequency aid”, which would be the sum of the Doppler frequency and clock frequency-error estimates.

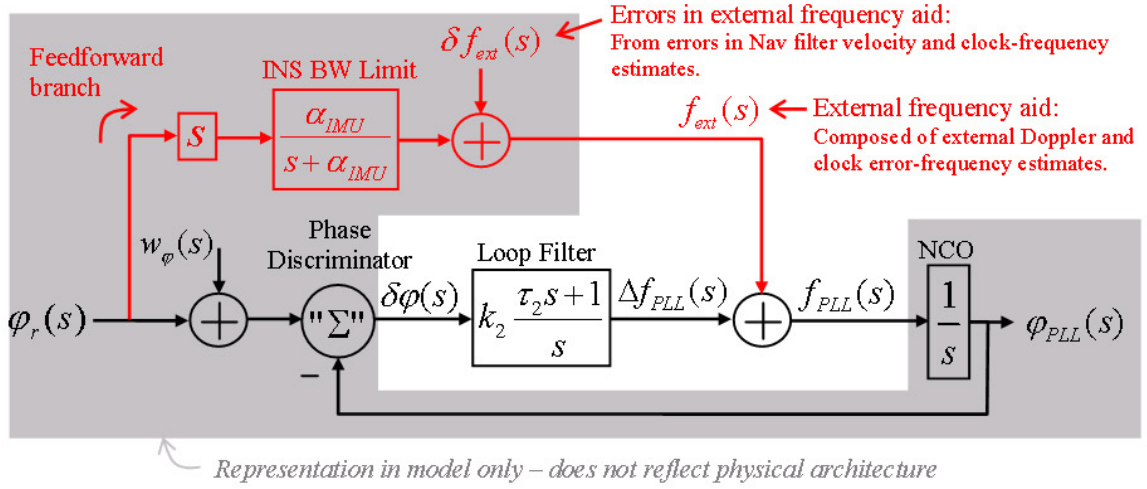


Figure 6.8: Model of a Frequency-Aided PLL

The feed-forward branch (emphasized in red) models the behavior of external Doppler estimates as they relate to the reference phase input. The branch contains a derivative operation (s) to relate carrier phase to carrier frequency (inverse of NCO plant), and a single-pole low-pass filter to represent the limited bandwidth of the INS that provides the Doppler component. An error input (δf_{ext}) is also added to the branch to account for imperfect frequency estimates.

The PLL configuration in Figure 6.8 offers advantages over the traditional structure (Figure 6.2) in terms of phase-noise suppression and tracking bandwidth. These benefits are possible because the reference input and phase noise are no longer treated by the same transfer function, as they are in Eq. 6.2; in this case, the external frequency estimates from the feed-forward branch are related to the reference signal, but have virtually no relation to the phase-noise input. The phase-noise input is still related to the output by a similar transfer function as Eq. 6.3 (with different poles and zeros), but the relationship between the reference input and output is now largely dependent on the inertial sensors. The following equations show these relationships, assuming zero input from $\delta f_{ext}(s)$:

$$\varphi_{PLL}(s) = H_3(s)\varphi_r(s) + H_2(s)w_\varphi(s) \quad (6.15)$$

where

$$H_3(s) = \frac{\frac{\alpha_{IMU}}{s + \alpha_{IMU}} + \frac{k_2(\tau_2 s + 1)}{s^2}}{1 + \frac{k_2(\tau_2 s + 1)}{s^2}} \quad (6.16)$$

$$H_2(s) = \frac{k_2(\tau_2 s + 1)}{s^2 + k_2\tau_2 s + k_2} \quad (6.17)$$

Equation 6.16 is expressed in such a way to illustrate that as the INS bandwidth gets large

($\alpha_{IMU} \rightarrow \infty$), $H_3(s)$ approaches unity. This result suggests that the phase dynamics originating from user motion can be tracked with the bandwidth provided by the inertial sensors. Therefore, the design of the loop filter no longer needs to account for high-bandwidth user dynamics, but it must still be able to track the dynamics of $\delta f_{ext}(s)$. The term Δf_{PLL} in Figure 6.8 represents the signal that tracks errors in the external frequency estimates, based on the transfer function specified in Eq. 6.17:

$$\Delta f_{PLL} = -H_2(s)\delta f_{ext}(s) \quad (6.18)$$

The presence of $H_2(s)$ in Eqs. 6.15 and 6.18 indicates that the tracking-bandwidth versus noise-rejection tradeoff still takes place, but with the important difference that the required bandwidth is lower, so the loop bandwidth can be reduced to improve phase-noise suppression. The lower limit on the bandwidth of the loop filter is now determined by the dynamics of $\delta f_{ext}(s)$.

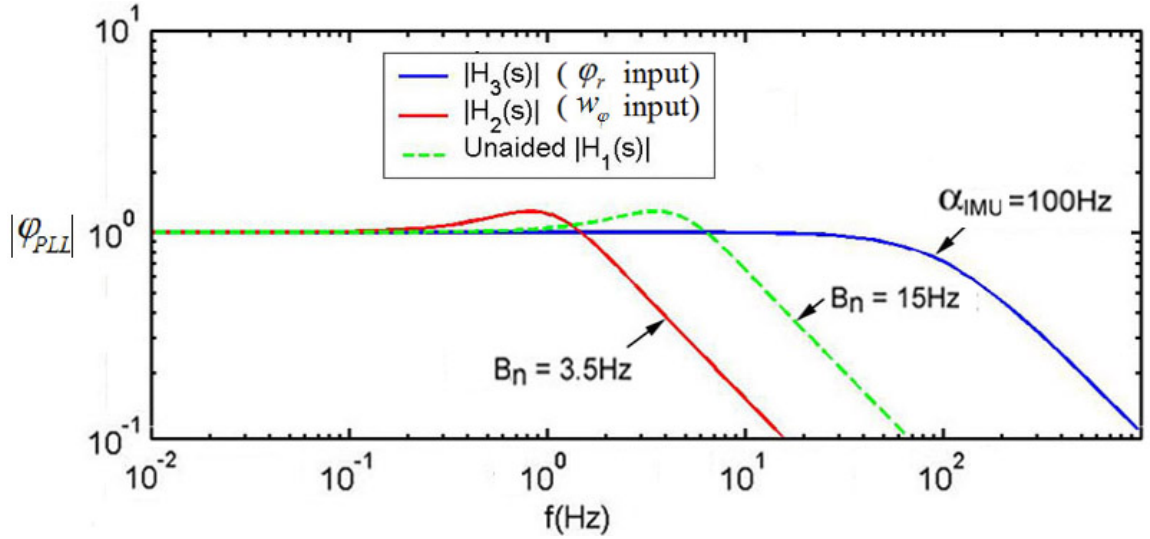


Figure 6.9: PLL Transfer Functions

To illustrate the benefits of Doppler aiding in terms of phase-noise rejection and phase-tracking bandwidth, Figure 6.9 shows the magnitude of the closed-loop transfer function $H_1(s)$ used in a traditional PLL with a noise bandwidth of 15Hz, versus $H_3(s)$ and $H_2(s)$ used in a Doppler-aided system with 100Hz bandwidth inertial sensors ($\alpha_{IMU} = 2\pi \times 100\text{Hz}$) and 3.5Hz noise bandwidth for error tracking. Selection of the closed-loop bandwidth with external frequency aiding will be discussed in Subsection 6.3.7; for the purposes of this illustration, it is assumed that 3.5Hz is sufficient bandwidth to track the error dynamics of the external frequency estimates, as will be shown later in chapter.

6.3.2 Implementation Details

Parallel Loop-Filter Structure

In many low-cost applications of ultra-tight integration, it may be necessary to have a GPS system in operation before the external Doppler frequency estimates are available. The most common example of this limitation is using inexpensive inertial sensors with unknown biases, and using normal GPS attitude and velocity outputs to calibrate the biases. After the inertial sensors are properly calibrated, the attitude and velocity of the vehicle can be measured with the accuracy and bandwidth needed for effective Doppler aiding.

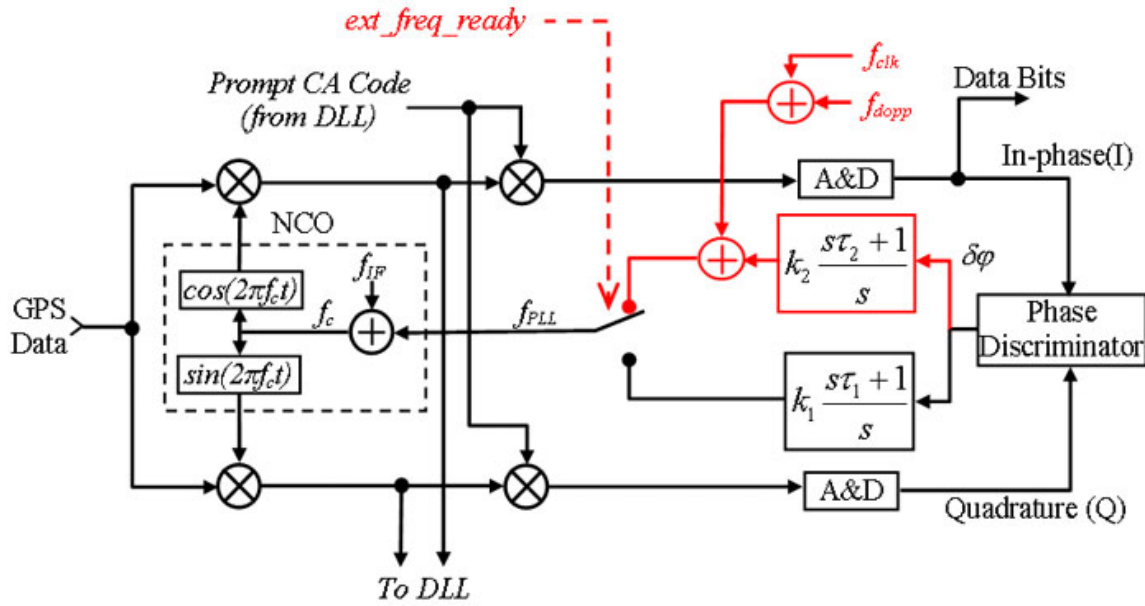


Figure 6.10: Implementation of a Frequency-Aided PLL

The need to initialize an INS with GPS before Doppler aiding can be used suggests that a GPS system equipped with ultra-tight integration should also function as a stand-alone navigation system during an initial calibration phase. The simple architecture introduced in the previous section differs from a traditional PLL only in the loop filter and use of external frequency estimates, so the transition between the two options can take place by selecting outputs from one of two parallel branches; one branch contains a conventional loop filter, and the other branch contains a lower-bandwidth loop filter with external frequency aiding. A schematic of this structure is shown in Figure 6.10. The transition from the traditional loop-filter branch to the frequency-aided branch is controlled by a flag labeled *ext_freq_ready*, which is asserted when Doppler estimates and clock error-frequency estimates are available. The modifications that differentiate this loop from a traditional structure are emphasized in red.

GPS Signal Integrity Monitoring

One of the most important benefits of using ultra-tight coupling is more robust tracking of GPS signals. In the event that a signal is impossible to track, such as during a momentary LOS blockage, it is also desirable to maintain a forecast of the signal to allow tracking to resume without a cycle slip when the signal is recovered. This type of robustness is especially beneficial to an automobile GPS attitude system, since brief dropouts of signals in urban environments are a primary source of cycle slips. With an accurate replica of the reference carrier and code frequencies, it is possible to duplicate the GPS signal (without the navigation data bits) through open-loop tracking with sufficient accuracy to prevent a cycle slip upon recovery of the signal. During the outage, replicas of the carrier and code frequencies originate from the external frequency estimates, and there may or may not be other good GPS channels integrated with the INS. This subsection focuses on a design that can be used to fine-tune the external frequency estimates such that they can be used temporarily to estimate the GPS signal without the use of closed-loop tracking. A scheme for integrity monitoring of the GPS signal is also needed to detect situations where further tracking of a corrupt or very weak signal is detrimental, and should be suspended in favor of open-loop tracking.

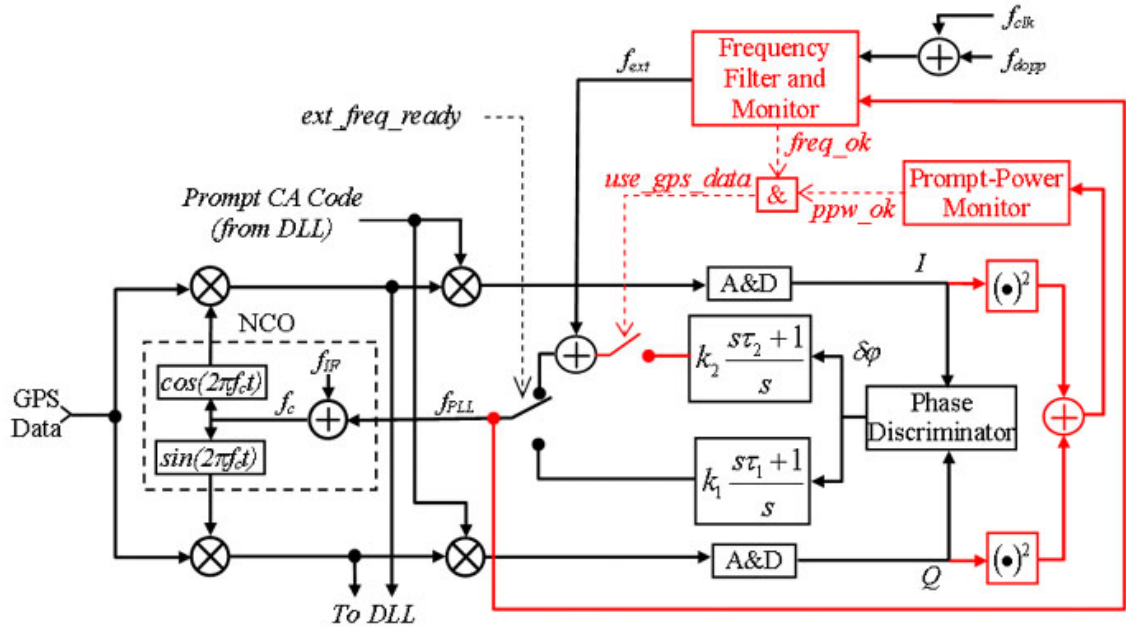
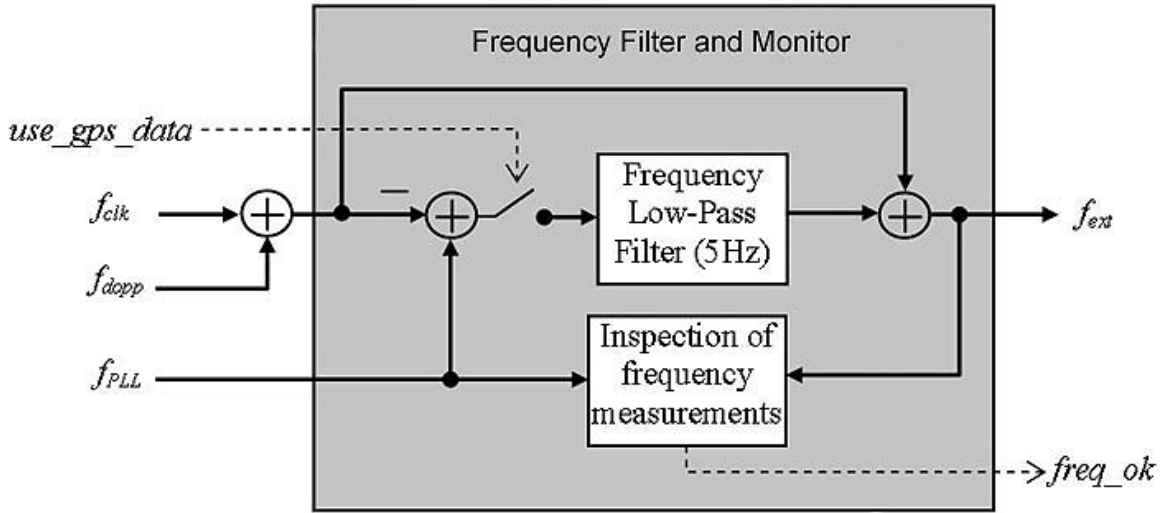


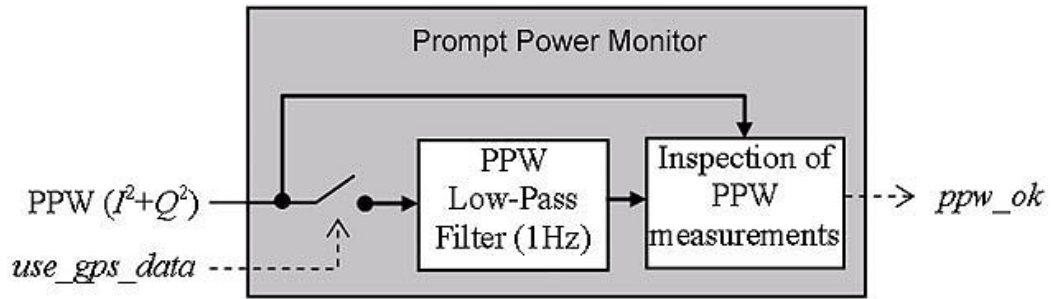
Figure 6.11: GPS Signal Integrity Monitoring in Frequency-Aided PLL

The first variation to the design illustrated in Figure 6.10 is to include the option to disable closed-loop tracking, but continue to generate I and Q measurements through open-loop tracking. This modification can be achieved by breaking the closed-loop immediately after the loop filter, such that the external frequency aid is still fed into the PLL. Figure 6.11 illustrates this

feature, and a control flag labeled *use_gps_data* that enables closed-loop tracking when asserted. The value of this flag is controlled by the outputs of two fault-detection mechanisms, whose purpose is to determine when the integrity of the GPS signal is compromised. The prompt-power monitor contains a low-pass filter that establishes an estimate of the signal power under normal tracking conditions, and it negates the *ppw_ok* flag when the signal power falls below a certain threshold relative to the filtered prompt-power value. The second block in Figure 6.11 is labeled “Frequency Filter and Monitor”. This block is designed to remove any biases from the external frequency estimates by comparing these values to f_{PLL} when in closed-loop tracking. This correction fine-tunes the external frequency estimates, such that they are as accurate as possible if open-loop tracking is needed. The monitoring component of this block ensures that the output of the loop filter remains continuous and near zero. A failure of these conditions may indicate an anomalous phase-noise situation, as the proportional element of the loop filter directly reflects the behavior of the phase-offset measurement. Such a condition is indicated by negation of the *freq_ok* flag, which also negates the *use_gps_data* flag.



a) Frequency Filter and Monitor



b) Prompt Power Filter and Monitor

Figure 6.12: GPS Signal Integrity Monitoring in Frequency-Aided PLL

The implementations of the frequency filter and monitor and the prompt-power monitor are shown in Figures 6.12a and b, respectively. When the PLL is operating in closed-loop tracking (*use_gps_data* asserted), the frequency low-pass filter produces an estimate of any low-frequency biases in the external frequency estimates. The use of this estimate to correct the frequency value used by the PLL does not affect closed-loop tracking significantly, but it is necessary in the event that open-loop tracking is needed. The frequency monitor uses the corrected frequency estimates as a reference to identify any anomalous behavior in f_{PLL} , such as sudden discontinuities and large phase-noise, and negates the *freq_ok* flag in such cases to invoke open-loop tracking. The prompt-power monitor has a similar architecture, in that the raw measurement is compared with its low-passed value to identify instances of low signal power, in which case the *ppw_ok* flag is negated.

The phase and frequency tracking performance of the PLL architecture discussed in this section is demonstrated in Figures 7.6 and 7.7. A detailed discussion of these results is deferred until Section 7.2, but in relevance to this section, the simulation case includes a GPS outage on one of the receivers, during which open-loop phase-tracking is successful in averting a cycle slip.

6.3.3 Frequency Error Models and Loop-Bandwidth Design

In a typical dynamic receiver, the bandwidth specification of the PLL is set to track the dynamics of the receiver's motion, which are usually much faster than those of the frequency instability of the local oscillator. In the frequency-aided PLL introduced in Subsection 6.3.1, the bandwidth needed to track the dynamics of the receiver is provided by the inertial sensors, so the objective of the closed-loop filter design was to track the errors in the external frequency estimates. The next few sections present a quantitative analysis of the sources of these errors, as a tool for designing the loop bandwidth in accordance with the type of GPS clock and inertial sensors used.

The dominant errors in the external frequency estimates can be subdivided into errors from the clock error-frequency estimates and errors from the Doppler frequency estimates, and each of these components will be considered separately. Contribution from the satellite clock phase-jitter is of concern if published specifications (5.7° RMS for 10Hz loop BW [66]) are taken literally [44, 34], but observations by some researchers have shown that these specifications are overly conservative [34]. To corroborate this conclusion, Subsection 6.3.6 discusses a simple procedure for measuring satellite phase noise and shows the results from this process. The acceleration spectrum of the vehicle (which includes vibration) also has an effect on the frequency output of an oscillator that experiences the same accelerations. The sensitivity of a TCXO to physical accelerations is about 0.5-2.0 parts-per-billion/g [60], which corresponds to a change

of about 0.02 Hz for a 10MHz oscillator under a 1g acceleration. While this effect is small, the error is multiplied within the RF front-end by a factor equal to the ratio of the L1 carrier to the oscillator frequency (about 157 for a 10MHz oscillator) [25]. In consequence, the effect of vibration can be significant depending on the dynamic environment of the receiver, and it can be quantified based on the vibration spectral density of a vehicle.

To study the phase-tracking error as a function of bandwidth in a frequency-aided PLL, only the closed-loop portion of the model in Figure 6.8 needs to be considered. Figure 6.13 shows a modified model that can be used to study the frequency-error tracking performance of the loop. Note, the relationship between $\delta f_{ext}(s)$ and $\Delta f_{PLL}(s)$ (see Eq. 6.18) is the same as that between $\delta \varphi_{ext}(s)$ and $\delta \varphi_{PLL}(s)$, so either set of symbols can be used interchangeably. The symbols δf_{dopp} and δf_{clk} represent errors in the Doppler and clock error-frequency estimates, respectively, and are inputs to the loop. The symbol $\delta \varphi_{vib}$ accounts for phase errors due to vibration, which is treated separately from the static receiver's frequency instability.

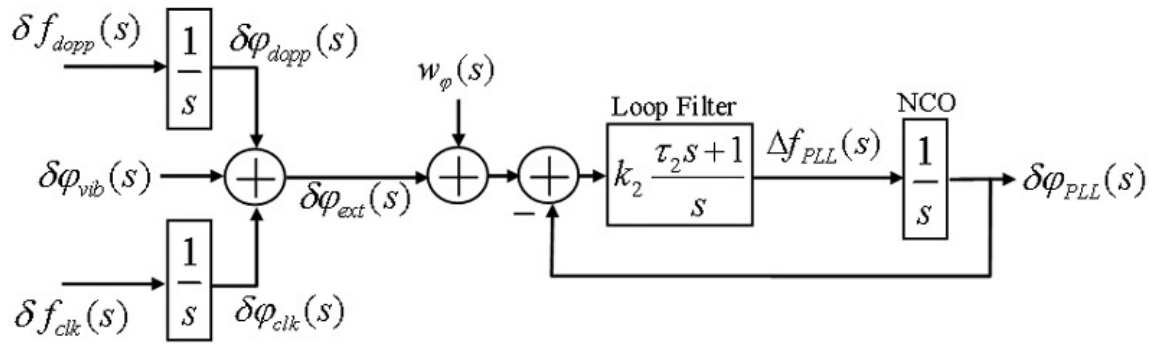


Figure 6.13: Model Used for Error-Tracking Analysis of Frequency-Aided PLL

The tracking error of the loop is defined as

$$\begin{aligned} \varepsilon_{\varphi}(s) &= \delta \varphi_{ext}(s) - \delta \varphi_{PLL}(s) \\ &= (1 - H_2(s)) \delta \varphi_{ext}(s) + H_2(s) w_{\varphi}(s) \end{aligned} \quad (6.19)$$

Note, this tracking error is the sum of two terms; the first decreases with increasing bandwidth and the second increases with increasing bandwidth. If the design of the PLL is not concerned with user motion, then the goal of the loop-bandwidth selection process may be to find an optimal bandwidth that minimizes the sum of these terms under typical operating conditions. Since each of the terms in Equation 6.19 will be treated separately, the equation can be expressed for convenience as the sum of two independent tracking errors:

$$\varepsilon_{\varphi}(s) = \varepsilon_{\varphi_{ext}}(s) + \varepsilon_{\varphi_w}(s) \quad (6.20)$$

where

$$\varepsilon_{\varphi_{ext}}(s) = (1 - H_2(s)) \delta\varphi_{ext}(s) \quad (6.21)$$

$$\varepsilon_{\varphi_w}(s) = H_2(s) w_\varphi(s) \quad (6.22)$$

$\varepsilon_{\varphi_{ext}}$ corresponds to tracking error induced by external frequency-estimate errors and oscillator vibration, and ε_{φ_w} is the tracking error caused by thermal noise and RF interference.

The tracking performance of the PLL can be quantified in terms of the tracking-error variance, which is the sum of the variances of $\varepsilon_{\varphi_{ext}}(t)$ and $\varepsilon_{\varphi_w}(t)$. The former term can be determined as a function of the loop transfer function and the PSD of the reference input:

$$\text{var}(\varepsilon_{\varphi_{ext}}(t)) = \int_0^\infty |1 - H_2(j\omega)|^2 S_{\delta\varphi_{ext}}(f) df \quad (6.23)$$

where the expression involving H_2 is evaluated with the second-order loop filter in Eq. 6.17:

$$|1 - H_2(j\omega)|^2 = \frac{(\omega/\omega_n)^4}{1 + (\omega/\omega_n)^4} \quad (6.24)$$

$$\omega_n = \sqrt{k_2} \quad \left(\text{for } \tau_2 = \sqrt{2/k_2} \right) \quad (6.25)$$

$$S_{\delta\varphi_{ext}}(f) = \text{PSD}(\delta\varphi_{ext}(t))$$

The symbol $S(f)$ is used hereafter to denote a PSD of the time-domain variable indicated in the subscript. The tracking error caused by thermal noise and RF interference is modeled as follows for a Costas discriminator:

$$\text{var}(\varepsilon_{\varphi_w}(t)) = \frac{B_{n\varphi}}{(S/N_0)} \left[1 + \frac{1}{2T_{av}(S/N_0)} \right] \quad (6.26)$$

Note that Eq. 6.26 is identical to 2.96, except for the exclusion of the superscript k to denote SV# k , and the various symbols in this expression were defined previously. Indeed, these two equations refer to the same quantity, as the PLL tracking error is introduced into the ICP measurements through the phase-offset measurement (see Figure 6.3).

Given these equations for the tracking-error variance, the remaining task is to specify the PSD of $\delta\varphi_{ext}(t)$. From Figure 6.13, $\delta\varphi_{ext}(t)$ is the sum of $\delta\varphi_{dopp}$, $\delta\varphi_{clk}$ and $\delta\varphi_{vib}$ so the following relationship follows for the PSDs of the phase-noise terms:

$$S_{\delta\varphi_{ext}}(f) = S_{\delta\varphi_{dopp}}(f) + S_{\delta\varphi_{clk}}(f) + S_{\delta\varphi_{vib}}(f) \quad (6.27)$$

Furthermore, the $1/s$ relationship between frequency and phase allows the use frequency quantities that can be more easily measured to express the PSD of the reference input components:

$$S_{\delta\varphi_{dopp}}(f) = \frac{1}{f^2} S_{\delta f_{dopp}}(f) \quad (6.28a)$$

$$S_{\delta\varphi_{clk}}(f) = \frac{1}{f^2} S_{\delta f_{clk}}(f) \quad (6.28b)$$

One possible design approach for selecting the bandwidth of the PLL is to pick the loop filter parameters that minimize the sum of Eqs. 6.23 and 6.26 for a given SNR. However, it should be noted that the variance due to changing SNR, multipath and RF interference can vary significantly depending on the signal environment, and these variable conditions will change the optimal bandwidth. The following subsections will focus on modeling techniques for the clock and Doppler-frequency errors, such that a closed form for the PLL tracking-error variance can be specified as a function of the loop bandwidth.

6.3.4 Oscillator Phase-Noise PSD Models

The spectral density model of a static oscillator's phase noise is well known. The model consists of the sum of the different types of frequency and phase noise that are observed in practice, where each term is a function of frequency [62]:

$$S_{\delta\varphi_{clk}}(f) = \frac{k_a}{f^4} + \frac{k_b}{f^3} + \frac{k_c}{f^2} + \frac{k_d}{f} + k_e \quad (\text{rad}^2/\text{Hz}) \quad (6.29)$$

where

$$\begin{aligned} k_a/f^4 &= \text{Frequency random walk} \\ k_b/f^3 &= \text{Frequency flicker noise} \\ k_c/f^2 &= \text{Frequency white noise/ Phase random walk} \\ k_d/f &= \text{Phase flicker noise} \\ k_e &= \text{Phase white noise} \end{aligned}$$

The coefficients that define the oscillator's phase-noise PSD are not normally available as specifications, but they can be determined by fitting the model to measurements or specifications that are given. The phase-noise stability of an oscillator is typically given in specification document as the single side-band (SSB) phase noise for several frequency points in units of dBc/Hz [60, 69], which describe the decay of the oscillator's output spectrum as a function of frequency

offset from the nominal [17]. For large frequency offsets, the SSB phase-noise can be easily converted to a phase-noise PSD (in rad^2/Hz) [37], which are the units needed for evaluating Eq. 6.23:

$$S_{\delta\varphi_{clk}}(f) \approx 2L_{\delta\varphi_{clk}}(f) \quad \text{if} \quad \int_f^\infty S_{\delta\varphi_{clk}}(f')df' \ll 1\text{rad}^2 \quad (6.30)$$

where

$$L_{\delta\varphi_{clk}}(f) = \text{SSB phase noise in dBc/Hz}$$

The low-frequency components of the phase-noise PSD can be obtained through measurements, such as those shown for the frequency noise of a TCXO and OCXO in Figure 6.7. By the relationship in Eq. 6.28, frequency and phase-noise PSDs are easily interchangeable, and measurements of frequency noise can be used to extract a model of the phase-noise PSD. This technique was used to obtain the graphs shown in Figure 6.14, which show the measured PSDs for the TCXO and OCXO (for the same data illustrated in Figure 6.7), the PSD points obtained from the oscillators' specifications sheets, and the model that was fit to both measurements and specifications. As shown, the frequency range of the measured part of the PSD is limited by the sampling period of the ICP measurements (5Hz for the TCXO, 10Hz for the OCXO), but the higher-frequency portion of the PSDs can be filled in with the points given in the specification sheets. In these examples, the coefficients of the models were adjusted manually to fit the measured PSDs at low frequencies and the oscillators' specifications at high frequencies. The coefficients determined for each clock are listed in Table 6.1.

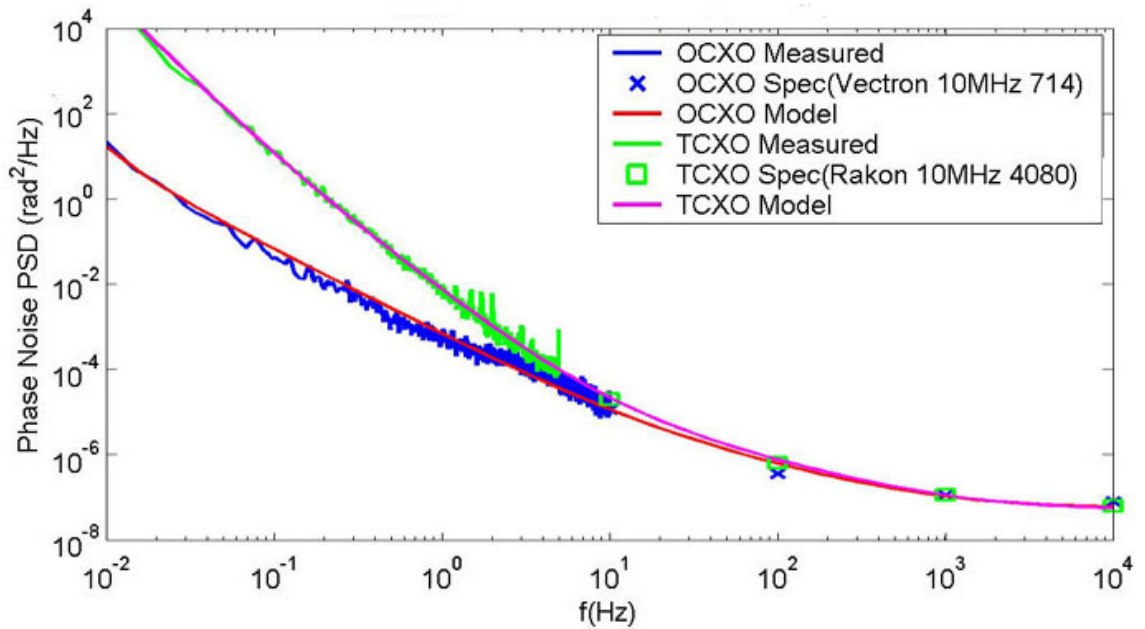


Figure 6.14: Phase-Noise PSDs of a TCXO and an OCXO

Table 6.1: Clock Phase-Noise PSD Model Coefficients

	k_a	k_b	k_c	k_d	k_e
OCXO	1.0×10^{-7}	9.0×10^{-7}	6.5×10^{-4}	5.0×10^{-5}	5.5×10^{-8}
TCXO	6.0×10^{-4}	6.0×10^{-3}	9.6×10^{-4}	6.2×10^{-5}	5.5×10^{-8}

As demonstrated by the frequency noise measurements shown in Figure 6.7, the frequency instability of an oscillator can be observed as part of the GPS velocity measurements. Since the frequency error typically contains a relatively constant bias or predictable long-term parabolic trend, current measurements can be used to estimate future values. However, the exponentially correlated noise, especially in the TCXO, is essentially a stochastic process and its higher frequency components cannot be predicted accurately. Furthermore, measurements of the clock error-frequency can only take place with four or more measurements, which may not always be available. For these reasons, the spectrum of $\delta\varphi_{clk}$ is assumed to be the entire spectrum of the oscillators phase noise, and the PLL should be designed to track this component without external aiding. When it can be estimated by other channels, however, an accurate external estimate of f_{clk} can be used to achieve open-loop tracking, as will be demonstrated in Chapter 7.

If the receiver is mounted on a vibrating platform, as it is on practically all moving vehicles, the effects of vibration on the reference oscillator must also be considered. The spectral density of the oscillator's phase noise due to vibration is given in [44] as follows:

$$S_{\delta\varphi_{vib}}(f) = (k_g N_f f_o)^2 \frac{S_{vib}(f)}{f^2} \quad (6.31)$$

where

$$\begin{aligned} k_g &= \text{Oscillators sensitivity to acceleration}(\text{parts}/g) \\ f_o &= \text{Nominal frequency of oscillator}(Hz) \\ N_f &= f_{L1}/f_o \\ S_{vib} &= \text{PSD of vehicle vibration}(g^2/Hz) \end{aligned}$$

Typical values for the constants in Eq. 6.31 are $k_g \approx 1 \times 10^{-9}$, $f_o = 10\text{MHz}$, and $N_f \approx 157$. The spectral density of vehicle vibration is easily estimated experimentally from accelerometer outputs in the IMU, and a model can be formulated that overbounds the observed vibration PSD. Figure 6.15 illustrates this approach; the figure shows the PSD of accelerometer measurements taken during a road test, and a corresponding model. The measured quantity is the resultant of the three accelerometer outputs from the IMU. The measured and modeled PSDs reflect the presence

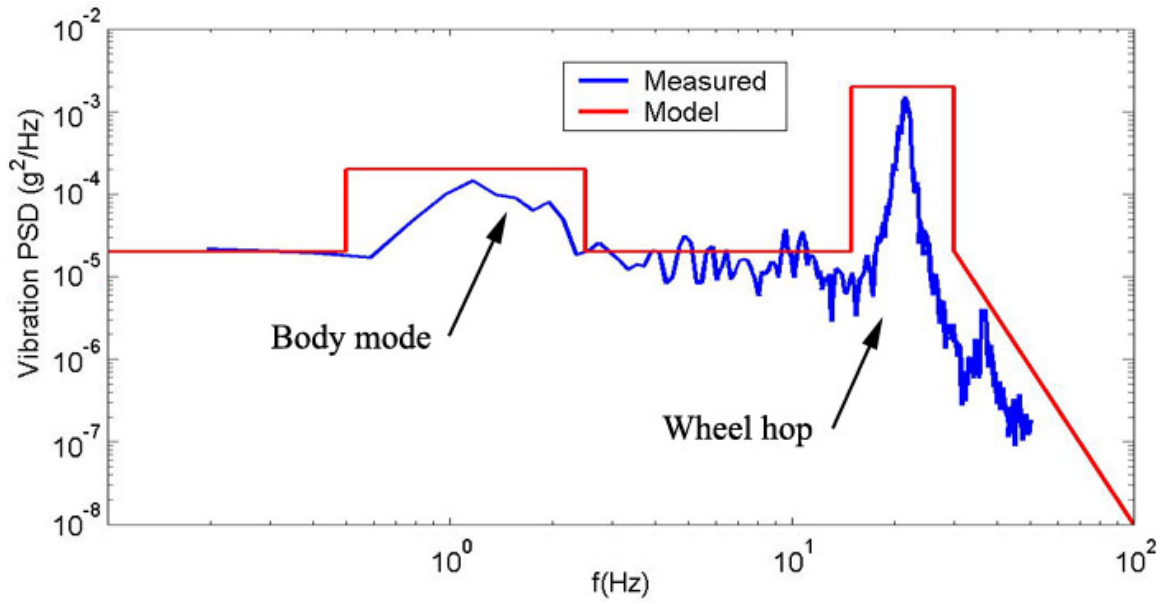


Figure 6.15: Measured and Modeled Vibration PSD of a Test Automobile in Motion

of a body mode and a wheel-hop mode around 1Hz and 20Hz, respectively. The equation that describes the model is as follows:

$$S_{vib}(f) = \begin{cases} 2 \times 10^{-5} & (0 < f \leq 0.5Hz) \\ 2 \times 10^{-4} & (0.5 < f \leq 2.5Hz) \\ 2 \times 10^{-5} & (2.5 < f \leq 15Hz) \\ 2 \times 10^{-3} & (15 < f \leq 30Hz) \\ \frac{30^6}{f^6} 2 \times 10^{-5} & (30Hz < f) \end{cases} \quad (g^2/Hz) \quad (6.32)$$

6.3.5 Doppler-Estimate Error Models

An equation for the error in the external Doppler-frequency estimates can be obtained by applying a perturbation to the receiver velocity in Eq. 6.11. Assuming that the SV velocity and LOS vectors are deterministic or have small errors when compared to estimates of receiver velocity, the following equation reveals a relationship between errors in receiver-velocity estimates and errors in the Doppler estimates:

$$f_{dopp}^k + \delta f_{dopp}^k = \frac{1}{\lambda_{L1}} \left(\left({}^e\vec{V}^{RX} + \delta {}^e\vec{V}^{RX} \right) - {}^e\vec{V}^k \right) \cdot \vec{1}^k \quad (6.33)$$

The perturbation to the Doppler frequency estimate is evident:

$$\delta f_{dopp}^k = \frac{1}{\lambda_{L1}} \delta {}^e\vec{V}^{RX} \cdot \vec{1}^k \quad (6.34)$$

Given an error in the estimated receiver velocity vector, this equation shows that the Doppler-estimate error will be maximized if the velocity-error vector is parallel to the LOS vector. Since this condition cannot be anticipated, a reasonable conservative simplification is to assume the worst-case scenario of the two vectors being parallel, such that maximum external Doppler-estimate error is proportional to the magnitude of the velocity-error vector:

$$\left| \delta f_{dopp}^k \right|_{\max} = \frac{1}{\lambda_{L1}} \left| \delta^e \vec{V}^{RX} \right| \quad (6.35)$$

These Doppler-estimate errors can have very different characteristics, depending on whether the navigation filter has GPS solutions available. That is, the GPS subsystem must be tracking four or more satellites to provide position and velocity measurements to bound the error growth of dead-reckoning navigation with inertial sensors. In automobile applications, it is expected that four or more channels are not always available, so designing for the case when GPS position and velocity are not available is a reasonable conservative choice. Without GPS and using inexpensive inertial sensors, it is expected that the Doppler estimate errors will resemble a random walk at best, and the rate at which the errors grow will be largely dependent on the initial condition of the stochastic process. This initial condition is the last Doppler estimate that was computed *with* GPS aiding, so the model for the case without GPS aiding must ultimately be related to that with GPS aiding.

Doppler-Estimate Error Model with GPS Aiding

Modeling of the error in the receiver velocity estimates begins at the source of the estimates themselves, the navigation filter. A three-dimensional GPS/INS navigation filter will likely have three states that represent the velocity of the IMU or that of a GPS antenna on some other fixed location on the vehicle. When the GPS antenna and the IMU are not collocated, the velocities of the two points are related by the vector between them and the angular velocity of the vehicle (pitch, roll, and yaw rates). If multiple GPS antennas are fixed on the vehicle, as is the case with a GPS attitude system, the velocity measurements from the three antennas inherently contain information about angular velocity and can be combined such that they yield velocity measurements on other fixed points on the vehicle. With plans to devise a 3D navigation system based on the GPS/INS attitude system discussed in Chapters 3-5, the GPS-antenna frame was designed to allow mounting of the IMU very close to the horizontal centroid of the triangle defined by the three antennas (see Figure 3.5). This design allows for a convenient simplification for converting GPS velocity measurements to velocity measurements at the IMU. That is, the average of the GPS velocity measurements from the three receivers is a very good approximation of the velocity of the IMU, even when the vehicle is turning. Of course, this simple relationship will not

apply for any mounting configuration of an IMU and multiple GPS antennas, but the principle can be generalized by stating that the velocity of any point on a rigid body can be determined from the velocities of three other distinct points on the body.

One special and convenient feature of a navigation filter that incorporates GPS, is that the primary navigation states, position and velocity in an Earth-fixed frame, are directly observable with the GPS subsystem. That is, position and velocity are part of the state vector and part of the measurement vector used in the update step of the filtering process. The consequence of this arrangement is that the post-update state-estimate for each of the navigation states is essentially a weighted average between the pre-update state-estimate and a direct measurement of the state itself, where a gain determines the relative weight of each component. Using a general notation to represent an arbitrary state (x) and a measurement (y), this form for the update step of the discrete filter is as follows:

$$\hat{x}_i(+) = (1 - k)\hat{x}_i(-) + ky_i \quad (0 < k < 1) \quad (6.36)$$

The state measurement can be written as the sum of its true value and measurement-noise:

$$y_i = x_i + v_i \quad (6.37)$$

The value of the gain k is based on the relative accuracy of the propagated state and the measurement. Assuming that the accuracy of the GPS measurements is fixed, the steady-state value of the gain depends only on the type of IMU used; the better the IMU, the lower the gain. The propagation step consists of propagating the time derivative of the state, and for position and velocity, the time derivatives are also part of the state vector:

$$\hat{x}_{i+1}(-) = \hat{x}_i(+) + \hat{\dot{x}}_i(+) \Delta t \quad (6.38)$$

The time derivative of the state can be written as the sum of its true value and process noise:

$$\hat{\dot{x}}_i(+) = \dot{x}_i + w_i \quad (6.39)$$

The true value of the time derivative of the state is defined in this model in terms of the state itself and for a given sampling period:

$$\dot{x}_i \triangleq \frac{x_{i+1} - x_i}{\Delta t} \quad (6.40)$$

Finally, the state-estimate error is defined as the difference between the true value of the state

and its post-update state estimate:

$$\delta x_i \triangleq x_i - \hat{x}_i(+)$$
(6.41)

Equations 6.36-6.41 can be combined and manipulated to yield an expression for the propagation of the state-estimate error as a function of an initial error and values of measurement and process noise:

$$\delta x_{i+N} = (1-k)^N \delta x_i - \Delta t \sum_{n=0}^{N-1} (1-k)^{N-n} w_{i+n} - k \sum_{n=0}^{N-1} (1-k)^n v_{i+N-n}$$
(6.42)

The discrete autocorrelation of the state-estimate error can be computed from this function, as the variable N indicates the number of offset samples ($N \geq 0$):

$$\begin{aligned} \phi_{\delta x \delta x}(N) &= E(\delta x_i \delta x_{i+N}) \\ &= (1-k)^N E(\delta x_i^2) - \Delta t \sum_{n=0}^{N-1} (1-k)^{N-n} E(\delta x_i w_{i+n}) \\ &\quad - k \sum_{n=0}^{N-1} (1-k)^n E(\delta x_i v_{i+N-n}) \end{aligned}$$
(6.43)

This expression can be complicated if the measurement and process noises are correlated in time, as they would also be correlated with the state-estimate error. To simplify the model, it is assumed that these terms have Gaussian white noise characteristics, which makes the summation terms drop out:

$$\phi_{\delta x \delta x}(N) = (1-k)^N E(\delta x_i^2) \quad \text{if } E(w_i w_j) = 0 \text{ and } E(v_i v_j) = 0 \quad \forall i \neq j$$
(6.44)

Utilizing the symmetry property of the autocorrelation, this equation can be rewritten as follows:

$$\phi_{\delta x \delta x}(N) = \sigma_{\delta x}^2 e^{-\beta |N| \Delta t}$$
(6.45)

where

$$\begin{aligned} \beta &= \frac{-\ln(1-k)}{\Delta t} \\ \sigma_{\delta x}^2 &= \text{var}(\delta x_i) = E(\delta x_i^2) \end{aligned}$$
(6.46)

In this form, the autocorrelation function is recognizable as that of a first-order Gauss-Markov process. The convenient form of this result facilitates modeling of the PSD of the state-estimate error, as Markov processes have well defined PSDs (see Appendix B):

$$S_{\delta x}(\omega) = \frac{2\beta \sigma_{\delta x}^2}{\omega^2 + \beta^2}$$
(6.47)

To finalize this model, the variance of the state-estimate error needs to be ascertained. The accuracy of the state estimate is obviously dependent on the accuracy of the GPS measurements and on the stability of the inertial sensors in the IMU. As stated previously, these two factors influence the selection of the gain k . In the short term, the measurement noise in the GPS measurements is typically greater than accumulated error due of the inertial system, and typical values of k (less than 0.3) reflect this contrast. In a Kalman filter, the gain would be optimal in the sense that the noise contribution to the state-estimate from the pre-update state estimate and from the measurement itself would be equal. In the GPS/INS filters discussed in this thesis, most of the filter gains are adjusted to be somewhat higher than the Kalman gains for the given dynamic models. The reason for this adjustment is that the dynamic models for the inertial sensor errors are simplified, and do not account for the nonlinear error sources, such as misalignment and scale-factor variations. These sources may result in larger inertial-sensor errors than expected, especially during highly dynamic maneuvers. The consequence of this condition would be an undesired transient in the state estimate after a dynamic maneuver that is possibly worse than the GPS measurement errors, and can be detrimental to the system's integrity. To prevent these situations, the gains of the filter can be increased in favor of the GPS measurements, at the expense of sacrificing optimality under more nominal conditions. As a result, GPS errors have a predominant influence on the errors in state estimate, as was also discussed and demonstrated in Subsection 5.1 with regard to the accuracy of the GPS/INS attitude system.

If GPS measurement noise is the prevalent contributor to the error in the state estimate, it follows that the variance of the state estimate should be expressible in terms of the GPS measurement variance. To this end, Eqs. 6.36 and 6.37 can be substituted into Eq. 6.39 to yield the following:

$$\delta x_i = (1 - k)(x_i - \hat{x}_i(-)) + kv_i \quad (6.48)$$

Assuming zero mean for the state error and uncorrelated GPS measurement noise, the variance of this equation is:

$$E(\delta x_i^2) = (1 - k)^2 E((x_i - \hat{x}_i(-))^2) + k^2 E(v_i^2) \quad (6.49)$$

The assumption that INS errors are small in the short term when compared to GPS errors implies that the errors in the pre-update state estimate will be dominated by error in their initial condition. This initial condition is the post-update state-estimate from the previous time-step, whose error is mostly due to GPS measurement noise. Thus, if the variance of the GPS measurements is assumed to be constant (not always true), then

$$\begin{aligned} E((x_i - \hat{x}_i(-))^2) &\approx E((x_i - \hat{x}_{i-1}(+))^2) \\ &= E((x_i - \hat{x}_i(+))^2) \end{aligned}$$

$$= E(\delta x_i^2) \quad (6.50)$$

Substitution of this equation into Eq. 6.49 gives an expression for the state-estimate error in terms of the variance of the measurement noise:

$$E(\delta x_i^2) \approx \frac{k}{2-k} E(v_i^2) \quad (6.51)$$

Finally, the continuous-time PSD model for the state-estimate error can be written in terms of the gain k and the variance of the measurement noise:

$$S_{\delta x}(\omega) = \frac{-2 \left(\frac{\ln(1-k)}{\Delta t} \right) \frac{k}{2-k}}{\omega^2 + \left(\frac{\ln(1-k)}{\Delta t} \right)^2} E(v^2) \quad (6.52)$$

This result can now be applied to the specific case where velocity is the estimated state in a navigation filter. The velocity-vector error magnitude is the term of interest in Eq. 6.35, and it is the resultant of three velocity-error vector components. The variance of the error-magnitude is simply the sum of the variances of the velocity-vector components. For simplicity, it can be assumed that each velocity-vector component is measured with equal accuracy with GPS, and the ‘up’ component should be used since it is slightly less accurate than the horizontal components. In this case, the continuous-time PSD model of the Doppler-estimate error, properly scaled for the GPS velocity update rate, can be written as follows:

$$S_{\delta fdopp}(\omega) = \frac{1}{\lambda_{L1}^2} \left(\frac{-2 \left(\frac{\ln(1-k)}{\Delta t} \right) \frac{k}{2-k}}{\omega^2 + \left(\frac{\ln(1-k)}{\Delta t} \right)^2} \right) 3 \Delta t_{GPS} \text{var}(V_{GPS}) \quad (6.53)$$

The validity of this model can be verified by an analysis of the Doppler-estimate errors obtained from an actual navigation filter. Figure 6.16 shows a comparison between the noise characteristics predicted by this model, and those observed for a navigation filter implemented with the GPS/INS attitude system discussed in Chapters 3-5. The gain of the filter was $k = 0.25$, and the sample variance of the 10Hz GPS velocity measurements was 1.6 (cm/sec)^2 , which corresponds to the variance of the vertical velocity component. The errors in velocity estimates were determined by comparing the filter-state velocity estimates with a post-processed approximation of the true velocity; the true-velocity approximation was estimated with the use of a non-causal window filter that uses past and future data to determine more accurate velocity estimates than those of the causal navigation filter. The time and frequency-domain comparisons of the model in Eq. 6.53 agree relatively well, especially in the time domain. However, two distinct differences appear in the frequency domain. The first difference is that the model does not capture

the effect of the 10Hz GPS measurement update rate and its harmonics; the second difference is that the model predicts higher spectral content at low frequencies than is observed in the data. The latter disagreement is likely due to the method employed for estimating the Doppler frequency errors, since the low-pass filtering technique used to estimate the “truth” value does not capture low-frequency errors well. Therefore, the behavior of the model at low frequencies is expected to be realistic, given that some time-correlated errors are expected from both GPS and INS measurements.

In conclusion, the model for Doppler-estimate errors that has been developed in this subsection demonstrates similar time-domain and frequency-domain characteristics as observed errors obtained from a real navigation filter with GPS aiding. In the next subsection, this model will be used as a basis from which to obtain a similar model for the case without GPS aiding of the navigation filter, as that will be the model used in Subsection 6.3.7 with regard to setting the bandwidth of a Doppler-aided PLL for automobile applications.

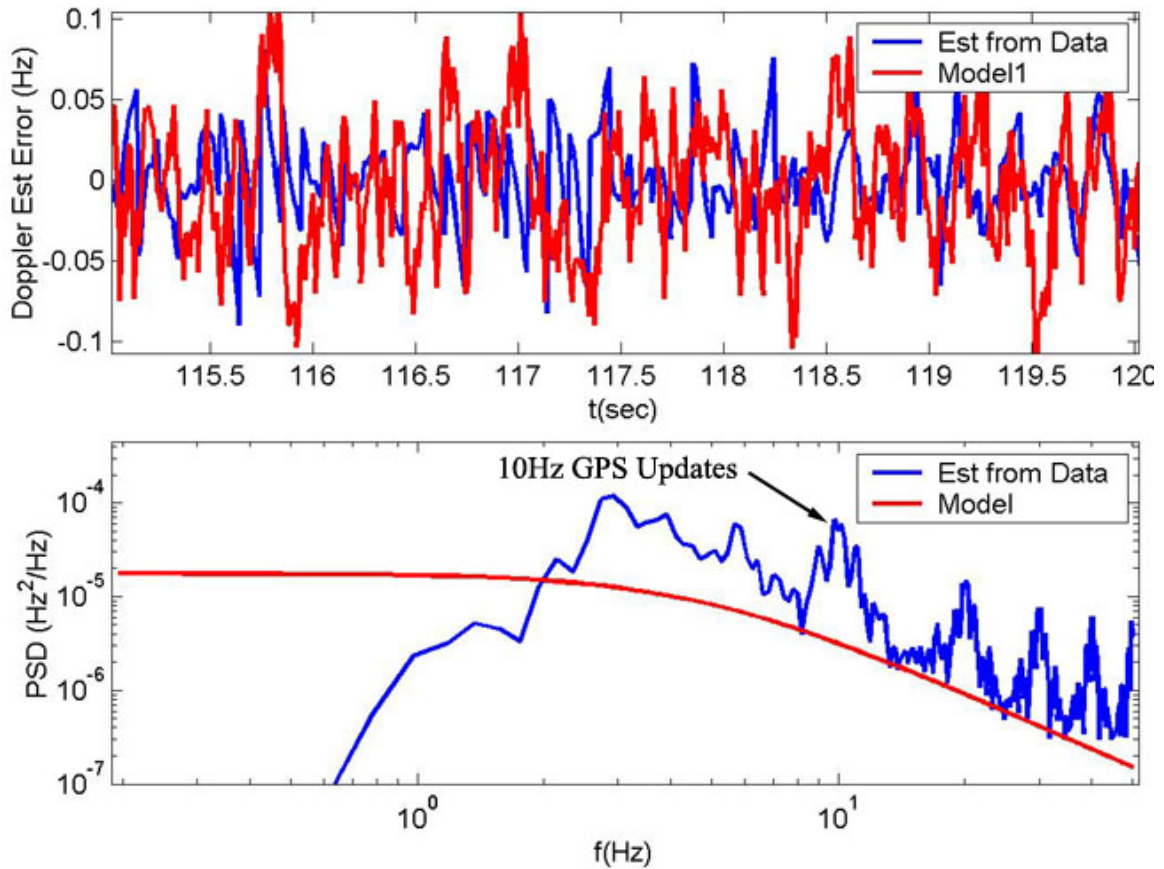


Figure 6.16: Time and Frequency-Domain Comparisons of Doppler-Estimate Error Model with Approximate Experimental Errors from Navigation Filter

Doppler-Estimate Error Model without GPS Aiding

In the event that the GPS subsystem cannot track four or more satellites, the assumption of having GPS measurements to limit the magnitude of Doppler-estimate errors is lost. In this situation, the vehicle velocities are estimated by dead-reckoning after the last GPS measurement is obtained, and the Doppler estimate errors would grow with time after that point. The dead reckoning process used to estimate velocities involves integrating gyro measurements to determine vehicle attitude and integrating accelerations in an Earth-fixed frame into velocities. Therefore, estimation of velocity by dead reckoning accrues errors from integrating noisy measurements from two sources, the accelerometers and the gyros. Furthermore, the error growth may be dependent on the trajectory, as fast or highly banked attitude maneuvers would amplify the effects of sensor scale-factor errors and misalignments. For these reasons, the varied error characteristics of Doppler estimates under dead reckoning are difficult to encompass with one model. However, a few observations can be made that will lead to an educated approximation of the behavior of the Doppler estimate errors without GPS aiding.

The first qualitative observation that can be made is that the variance of Doppler-estimate errors under dead reckoning is not constant, but grows with time after the last GPS update in the navigation filter. Secondly, it can be stated that the variance of these Doppler-estimate errors will be equal to or greater than when GPS is available, and it will be roughly equal for a short time after GPS is lost. With these observations, several characteristics of the spectral density of Doppler-estimate errors can be inferred. The unbounded error growth with time is representative of the behavior of a random walk, which has a well defined PSD shape, but not magnitude (see Appendix B). To scale the magnitude of the random walk PSD, it can be matched to that of the model in Fig 6.16 at high frequencies, as the two models will have the same slope above 10Hz. This choice of scaling factor guarantees that the variance of dead-reckoning Doppler estimates are always higher than those with GPS aiding. Furthermore, this scaling enforces that the spectral content for the two models is similar at high frequencies since the two models apply to the same inertial sensors and error-growth with time is a *low-frequency* effect. The PSD model of Doppler-estimate errors that corresponds to this description is as follows:

$$S_{\delta f_{dopp}}(\omega) = \frac{1}{\lambda_{L1}^2} \left(\frac{-2 \left(\frac{\ln(1-k)}{\Delta t} \right) \frac{k}{2-k}}{\omega^2} \right) 3 \Delta t_{GPS} \text{var}(V_{GPS}) \quad (6.54)$$

Note that GPS terms appear in this model despite the fact that it represents errors of a dead-reckoning process. These terms are not only needed to scale the model as described previously, but they also add the dependence on GPS accuracy that is expected for the initial condition of the random walk model.

Figure 6.17 shows plots of the models described by Eqs. 6.53 and 6.54, and comparisons with PSDs of estimated Doppler errors from the actual navigation filter, with and without GPS aiding. As expected, the data without GPS aiding has no peaks at multiples of 10Hz due to the absence of GPS updates, and the spectral content is correspondingly lower than that for the data with GPS aiding above 10Hz. At low frequencies (below 3Hz), the effect of low-frequency, unbounded error-growth is evident, and appears higher than anticipated by the model. As stated previously, this error growth is the result of integrating two or more noisy sensor outputs and unpredictable trajectory-dependent error sources like inertial-sensor scale factor and cross-axis sensitivity. The combined effect from these sources can indeed be worse in the long term than a random walk, as indicated by the data below 1Hz, and the model may not always be valid at low frequencies. However, the use of this model for short time spans without GPS aiding is valuable, as Doppler aiding by dead-reckoning with automotive-grade sensors is expected to be beneficial only on the order of tens of seconds without GPS. During longer periods without GPS navigation, a traditional PLL loop filter without Doppler aiding would be preferable if external Doppler estimates are severely degraded.

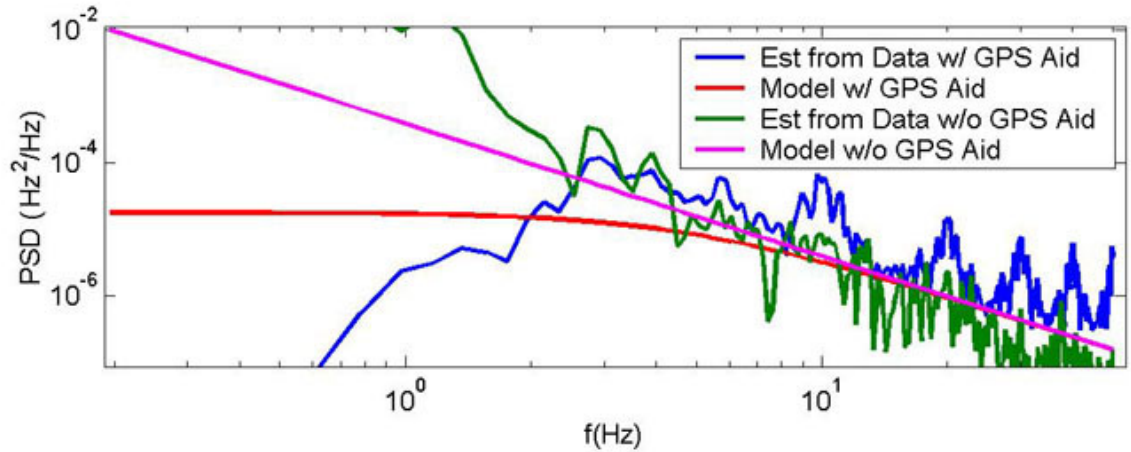


Figure 6.17: PSD Models of Doppler-Estimate Errors with and without GPS Aiding of Navigation Filter with Automotive-Grade Sensors

6.3.6 Satellite Clock Phase-Noise

The GPS satellites contain very stable atomic clocks when compared to oscillators used in consumer receivers. With frequency standards composed of both cesium and rubidium technologies, the GPS satellites atomic clocks can measure time with only a few nanoseconds of drift per day. Naturally, such oscillators are expected to introduce very little phase noise into the transmitted GPS signal, and thereby should have a small effect on the phase error of the receiver PLL. Nevertheless, the specification for the contribution of satellite phase-noise alone to the tracking

error of a 10Hz-bandwidth PLL is 5.7° RMS (0.1 rad) [66]. To put this number in perspective, the phase-tracking variance of a 15Hz PLL (noisier than a 10Hz PLL) with an OCXO and tracking a signal with about 45dB-Hz SNR, is about 0.08 rad. Therefore, observations from actual tracking loops suggest that the satellite phase-noise is significantly lower than current specifications. To verify this hypothesis, the following discussion presents a simple method for estimating a less conservative value for the effect of satellite phase noise based on clock frequency-error residuals.

For a stationary receiver, the frequency tracked by a PLL which tracks SV# k can be represented as follows:

$$f_{PLL}^k = \frac{-1}{\lambda_{L1}} \vec{1}^k \cdot^e \vec{V}^k + f_{clk} + \delta f_{PLL}^k \quad (6.55)$$

where

$$\delta f_{PLL}^k = \text{Frequency residual for SV\#}k$$

Not including the frequency residual, this equation is a rearranged form of one row of Eq. 6.13, but assumes that the receiver is stationary. The frequency residual term accounts for all components that make up f_{PLL} that are not Doppler or the common local-oscillator frequency error. Components in this category include the satellite clock error-frequency, cross-channel correlation noise, multipath, and other channel-dependent noise.

To proceed with this analysis, it will be assumed that multipath effects are minimized by placing the antenna away from obstructions, such that the frequency residual can be assumed to be uncorrelated with other channels. If this is the case, then the best estimate for f_{clk} is computed from average of all the channels. For N channels, this estimate is determined as follows:

$$\hat{f}_{clk} = \frac{1}{N} \sum_{k=1}^N \left(f_{PLL}^k + \frac{1}{\lambda_{L1}} \vec{1}^k \cdot^e \vec{V}^k \right) \quad (6.56)$$

Substituting Eq. 6.55 into Eq. 6.56 gives an expression for this estimate in terms of the truth value and the frequency residuals:

$$\hat{f}_{clk} = f_{clk} + \frac{1}{N} \sum_{k=1}^N \delta f_{PLL}^k \quad (6.57)$$

The error variance of this estimate can be computed by assuming that the variance of the frequency residuals is approximately equal in all channels:

$$\text{var}(\hat{f}_{clk} - f_{clk}) \approx \frac{1}{N} \text{var}(\delta f_{PLL}^k) \quad (6.58)$$

With the result of Eq. 6.56, an estimate of the frequency residual in each channel can also be computed:

$$\begin{bmatrix} \delta \hat{f}_{PLL}^1 \\ \delta \hat{f}_{PLL}^2 \\ \vdots \\ \delta \hat{f}_{PLL}^N \end{bmatrix} = \begin{bmatrix} f_{PLL}^1 + \frac{1}{\lambda_{L1}} \vec{1}^1 \cdot e \vec{V}^1 - \hat{f}_{clk} \\ f_{PLL}^2 + \frac{1}{\lambda_{L1}} \vec{1}^2 \cdot e \vec{V}^2 - \hat{f}_{clk} \\ \vdots \\ f_{PLL}^N + \frac{1}{\lambda_{L1}} \vec{1}^N \cdot e \vec{V}^N - \hat{f}_{clk} \end{bmatrix} \quad (6.59)$$

An analysis of these frequency residual estimates reveals information about the accuracy of \hat{f}_{clk} . That is, a vector of very small residuals relative to \hat{f}_{clk} would indicate that all the PLL frequencies contain a very similar common term that can be attributed to the local oscillator, and the clock error-frequency estimate would be expected to be very accurate. In contrast, a vector of large residual estimates relative to \hat{f}_{clk} would indicate that a common term among the channels is not necessarily dominant after the Doppler is removed, and the clock error-frequency estimate may not be accurate. Of course, the accuracy of \hat{f}_{clk} impacts the accuracy of the frequency residuals, which is the focus of this analysis to investigate the contribution from the satellite clock.

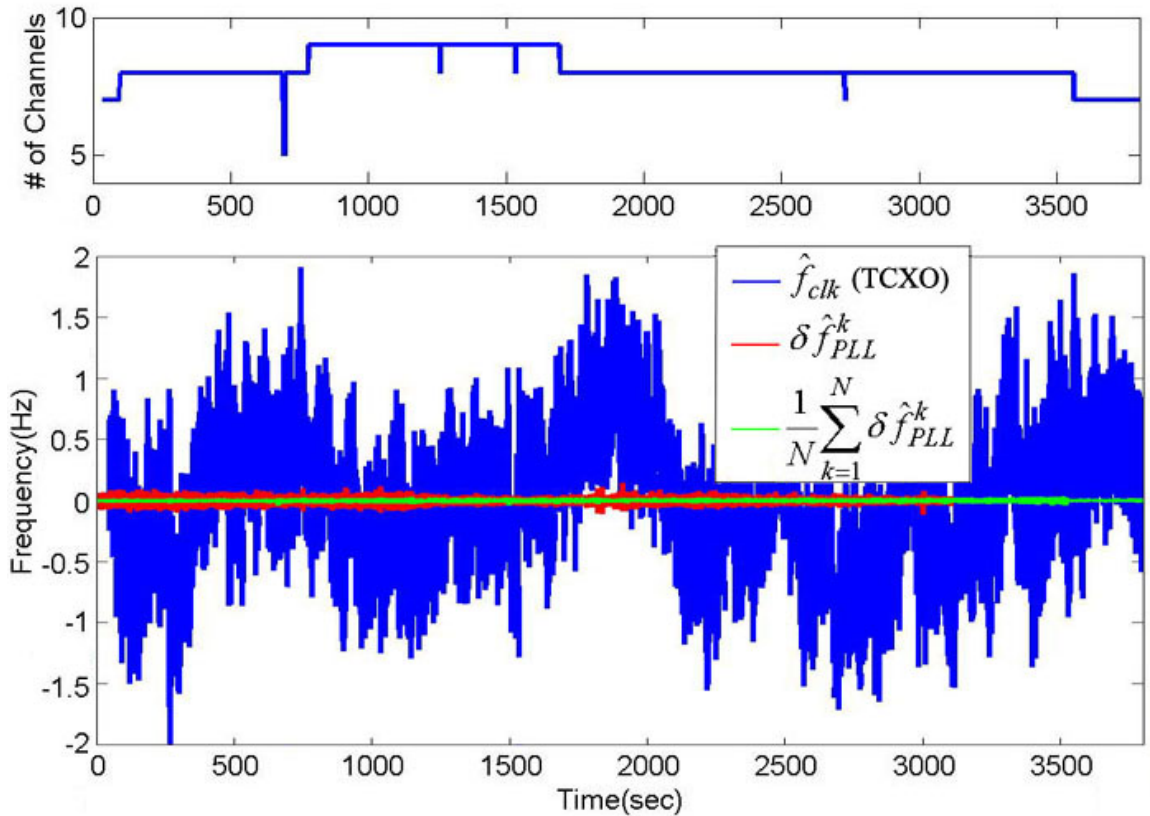


Figure 6.18: Clock Error-Frequency and Frequency Residual Measurements

Figure 6.18 shows measurements of \hat{f}_{clk} for a TCXO, a single-channel frequency residual, and the average of all the frequency residuals for the number of channels indicated in the

first graph. The parabolic trend has been removed from \hat{f}_{clk} , such that only the exponentially-correlated noise and wideband noise are visible. As shown, the frequency residual from a single channel (shown in red) is unbiased, and at least an order of magnitude smaller than the exponentially-correlated frequency noise of the TCXO; this result supports the conclusion that the clock error-frequency measurements obtained by Eq. 6.56 are more accurate than any single-channel residual. In fact, the error in \hat{f}_{clk} is approximately the average of the frequency residual estimates, which is also shown in the figure as the green trace.

The update rate of the ICP measurements used to generate the data in Figure 6.18 was 10Hz, which implies that frequency content only up to 5Hz was measured. Since the PLL of a dynamic receiver typically has a bandwidth of at least 15Hz, it is assumed that the transfer function of the PLL is approximately unity for these measurements. This assumption allows decoupling of the PLL transfer function from the behavior of the measurements, and the tracked frequencies are assumed to represent the input frequencies very well.

The next step in quantifying the effect of satellite-clock phase noise is actually verifying its presence in the frequency residuals. To perform this experiment, frequency residuals were measured as shown in Eq. 6.59 with the three Allstar receivers of the automobile attitude system prototype, which share a common reference oscillator. As expected, the use of a common clock resulted in virtually identical measurements of the clock error-frequency in all three receivers. To verify the presence satellite-specific noise, a correlation analysis of the frequency residuals was performed for channels tracking the same SV in different receivers. Since the receivers use the same reference oscillator, the test for a correlation among same SV frequency-residuals would need to differentiate between correlation due to satellite clock or to local clock (present because of imperfect values of \hat{f}_{clk}). To make this differentiation, two correlation quantities were used. The first is the average cross correlation between channels tracking the same SV in different receivers, and the second is the average cross correlation between different channels in the same receiver:

$$\text{Same SV Cross Corr} = \frac{1}{3N} \sum_{l=1}^2 \sum_{m=l+1}^3 \sum_{k=1}^N \text{Cross Corr} \left(\delta \hat{f}_{PLL}^{k,l}, \delta \hat{f}_{PLL}^{k,m} \right) \quad \forall l \neq m \quad (6.60)$$

$$\text{Same RX Cross Corr} = \frac{1}{3 \sum_{i=1}^{N-1} i} \sum_{m=1}^3 \sum_{k=1}^N \sum_{j=k+1}^N \text{CrossCorr} \left(\delta \hat{f}_{PLL}^{k,m}, \delta \hat{f}_{PLL}^{j,m} \right) \quad \forall j \neq k \quad (6.61)$$

where

m, l = Receiver indices

j, k = Channel indices (for channels tracking common SV in all receivers)

If the local clock error-frequency is estimated as accurately as expected, then the same RX cross-correlation would be low; if the same SV cross-correlation is higher, then the difference in correlation between the two quantities can be attributed to a satellite-specific source, not related to the local oscillator. Satellite-specific sources that introduce the same noise in different receivers would include some multipath if the antennas are close together, and satellite clock error-frequency. The antennas in the attitude system are only 50cm apart, but they were placed on the roof of a four-story building to place them above any possible sources of multipath other than from below. The results of the two cross-correlation operations are shown in Figure 6.19, for the same data set in Figure 6.18. As shown, the same SV cross-correlation peak was considerably more distinctive than that for the same RX, suggesting that satellite-specific noise was present in the data. These results were found to be repeatable only with relatively long data sets (over one hour), as other noise sources in the frequency residuals blur out the correlation peaks in short time spans.

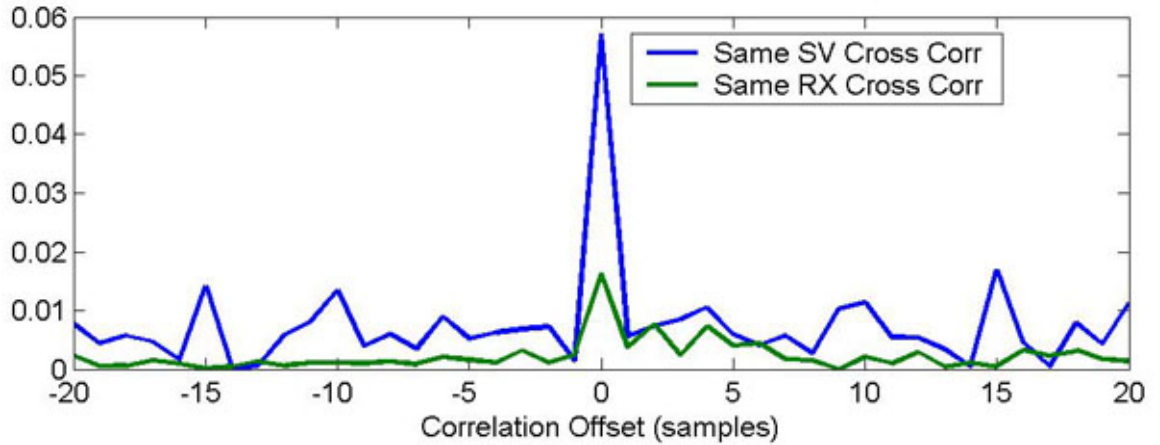


Figure 6.19: Cross-Correlation Results of PLL Frequency Residuals

The results of this frequency-residual correlation analysis support the fact that satellite phase-noise has some discernable effect on the GPS signal, and the observed magnitude of the residuals verifies that these effects are very small when compared to the frequency instability of a local oscillator. The PSD of the frequency residuals measured in this experiment can be transformed into a phase-noise PSD, which represents a less conservative estimate of the satellite phase-noise effect on a PLL than that specified in [66]. Figure 6.20 shows the phase-noise PSDs obtained from frequency-residual measurements from a single channel, and from all the common channels averaged over the three receivers. If the emerging downward trend of the SV phase-noise PSDs is extrapolated to higher frequencies, the SV phase-noise would be about 10-100 times smaller than that of the TCXO at 10Hz. This result agrees with the experimental PLL tracking-error results shown in [34], in which it is subsequently assumed that the SV phase-noise PSD is similar in shape to a TCXO, but one hundred times smaller than the specification.

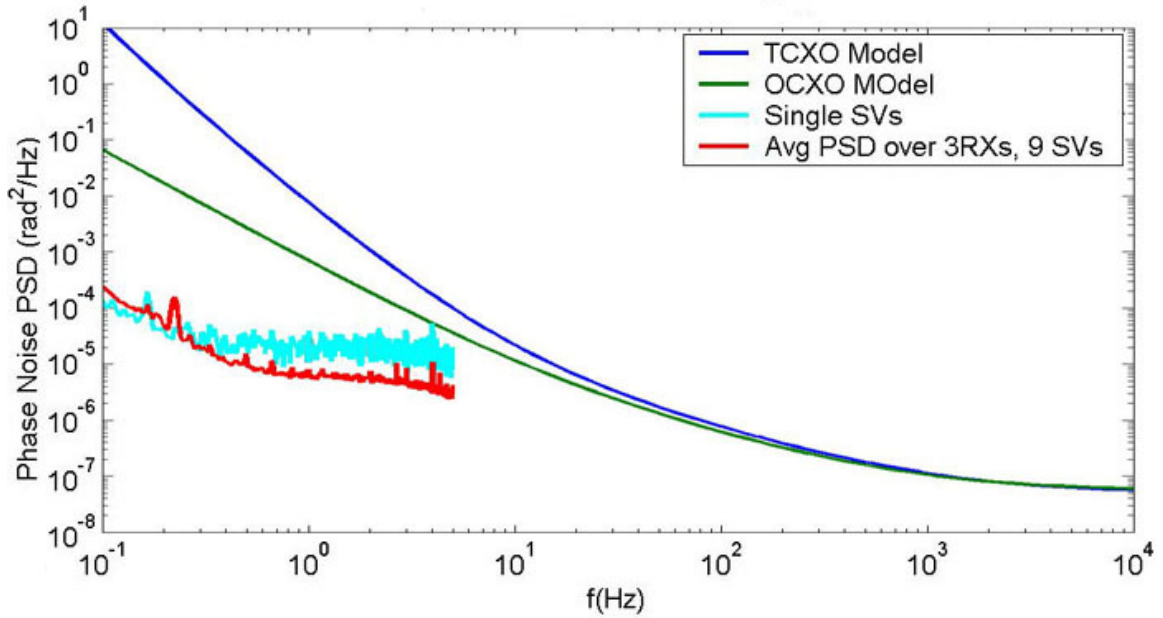


Figure 6.20: Phase-Noise PSDs of Satellite-Specific Noise

In conclusion, observed measurements of frequency/phase-noise induced by satellite-specific sources, including the satellite clock, are much smaller than those from a TCXO or even an OCXO. This result is important to the design of frequency-aided tracking loops, as the overly conservative specification given for the effects of the satellite clock does not allow much reduction of the PLL bandwidth to improve noise performance with Doppler aiding. Having shown experimentally that the satellite clock phase-noise effect is minor when compared to other sources, it will be assumed that a PLL designed to track the more dominant Doppler-error and local clock error-frequencies combined will incur a negligible phase noise penalty from the satellite clock. The alternative to this simplification is to establish a PSD model for the satellite-clock phase noise, and include it as another contributor (along with thermal/interference noise, Doppler-estimate errors, local clock error-frequency, and vibration) to the tracking error of the PLL.

6.3.7 Closed-Loop Bandwidth Selection

Equations 6.26, 6.29, 6.31 and 6.53/ 6.54 comprise the set of models needed to specify the phase tracking-error (also called phase jitter) of the Doppler-aided PLL. Figure 6.21 shows the contribution from each of these contributors to the phase-jitter of the PLL output as a function of noise-equivalent bandwidth. The wide-band noise component in this graph, given by Eq. 6.26, also represents interference from multipath or other sources, and unlike all the other tracking-error contributors, it increases with increasing loop bandwidth. The figure also includes a representative model of dynamic stress for a second-order PLL with no Doppler aiding, which will be

discussed in the next section with respect to cycle slips. In a traditional PLL *without* inertial aiding, this dynamic stress model would be used instead of the Doppler-estimate error models. This image shows clearly that the loop bandwidth must be reduced to suppress interference noise, but not reduced so far as to incur larger tracking error from all other sources. Thereby, the object of the design procedure is to select the lowest bandwidth possible that meets a tracking-error specification. The specification that will be used in this analysis was obtained from [62], which states that the RMS phase error should not exceed 0.1 rad to produce high correlation in coherent demodulation.

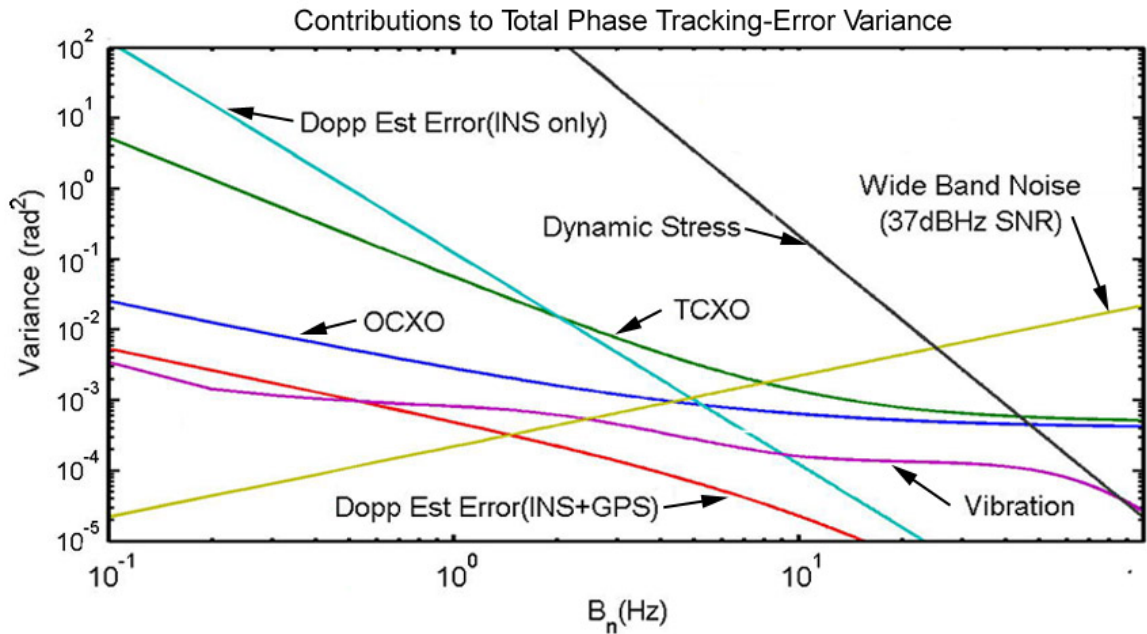


Figure 6.21: Contributions to PLL Output Phase-Jitter

Figure 6.22 shows the total phase jitter from all contributors (except dynamic stress), as a function of loop bandwidth, and for two different grades of INS and GPS reference oscillators. The graphs also include variation on whether the navigation filter has GPS updates (labeled “w/ GPS aiding”), or is in dead-reckoning mode (labeled “w/o GPS aiding”). The appropriate loop bandwidth to use for a Doppler-aided PLL can be obtained from these graphs, as the minimum bandwidth that meets the 0.1 rad specification. Table 6.2 summarizes the result of this procedure for the eight cases shown. The general conclusion of this analysis from a loop-bandwidth reduction standpoint is that it does not make sense to combine inexpensive with expensive components, such as a TCXO with a tactical-grade IMU. With such combinations, the loop bandwidth would have to be set to accommodate the lower-quality component and no benefit would be gained from having another higher cost component. However, PLL loop-bandwidth reduction is not the only design goal of a GPS/INS system, and other performance specifications (such as dead-reckoning

accuracy through GPS outages) may justify the use of a mixture of high-quality and low-quality components.

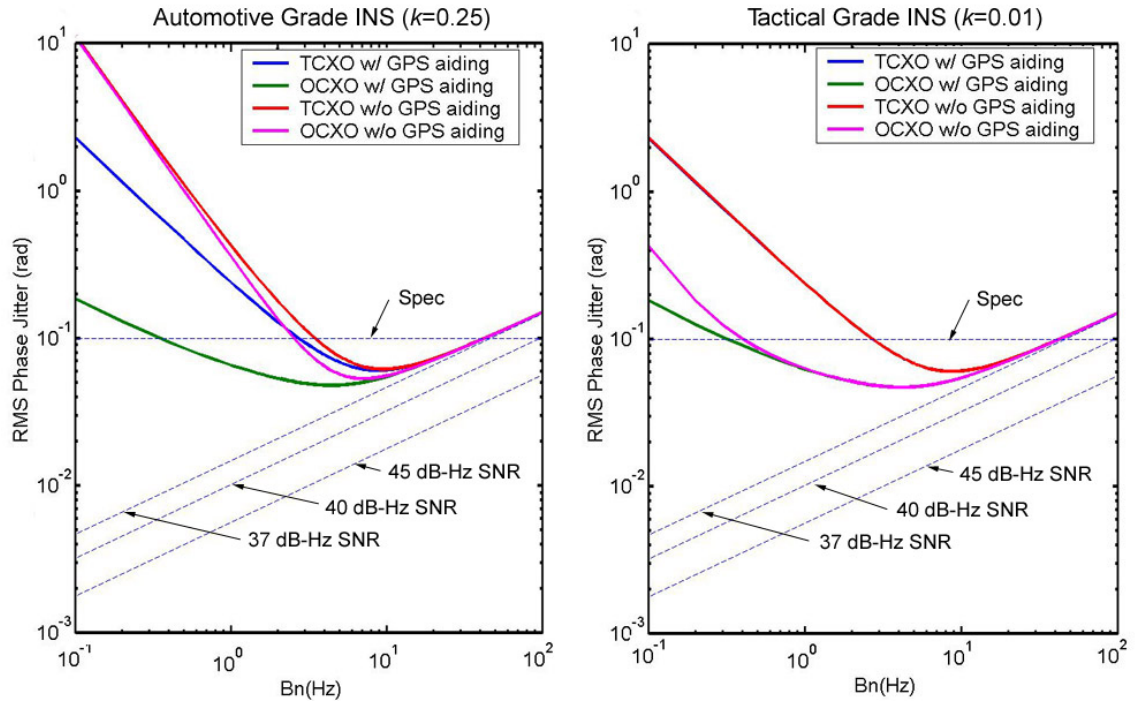


Figure 6.22: Phase Jitter vs PLL Noise-Bandwidth

Within the design space presented in Table 6.2, the appropriate choice for an automobile system would be that corresponding to the lowest-cost components: a TCXO and an automotive-grade IMU. Furthermore, limited satellite visibility in urban areas is expected to cause frequent GPS outages, in which case the design without GPS aiding is appropriate. Based on this analysis, the loop bandwidth of a Doppler-aided PLL in such a system should be 3-4Hz.

Table 6.2: Minimum Loop Bandwidths (Hz) for Doppler-Aided PLL with 0.1rad Phase Jitter

IMU Type	With GPS-Aided INS (4+ Channels Tracking)		Without GPS-Aided INS (<4 Channels Tracking)	
	TCXO	OCXO	TCXO	OCXO
Automotive ($k=0.25$)	2.8	0.34	3.4	2.4
Tactical ($k=0.01$)	2.7	0.30	2.7	0.4

Chapter 7

Performance of Doppler-Aided Phase-Lock Loops

Chapter 6 presented the development of Doppler-aided tracking loops, and showed how this technique allowed reduction of PLL loop bandwidth for improving the robustness of carrier-phase tracking. In terms of benefits to an attitude-system, the performance improvement gained through Doppler aiding is measured as reduced probability of cycle-slips and better phase-noise mitigation without sacrificing phase-tracking bandwidth. In this chapter, these ideas are explored further, and verified with actual performance results of PLLs in a custom software receiver. In the first section, the cycle slip analysis that will be presented is analogous to equivalent analysis of standard PLLs in relevant literature. The benefits of a Doppler-aided PLL in terms of cycle-slip robustness are presented in analytical terms, which are largely functions of the PLL noise bandwidth and dynamic stress due to vehicle dynamics. The second section of this chapter includes results which highlight the tracking performance of a Doppler-aided PLL. Results are presented for both simulated and real data, and the capability for open-loop carrier-phase tracking is demonstrated.

7.1 Cycle-Slip Analysis

For automotive GPS attitude applications, one of the most significant improvements to be gained from integration with inertial sensors is robustness against cycle slips. In Chapter 4, tight coupling was used to detect cycle slips and improve reliability of subsequent integer recovery in the presence of phase noise. Despite these measures, the occurrence of cycle slips and phase noise at the same time could not be avoided, and posed a challenge to the continuous availability and integrity of the GPS attitude subsystem. Chapter 6 demonstrated the concept of how Doppler aiding of the PLL can be used to address part of this problem by reducing phase noise as much as possible. In this section, it is shown that Doppler aiding can also reduce the rate of cycle slips in a PLL under dynamic stress, and can even be used to execute open-loop phase-tracking for short periods of time in the event of a brief line-of-sight blockage.

The cycle-slip performance of Doppler-aided PLLs will be discussed for four scenarios:

- The channel is tracking with enough other channels to generate a GPS navigation solution

for the GPS/INS filter.

- The channel is tracking, but with not enough other channels to generate a GPS navigation solution for the GPS/INS filter.
- The channel is not tracking, but enough other channels are available to generate a GPS navigation solution.
- The channel is not tracking, nor are there enough other channels to generate a GPS navigation solution.

The first two cases can be grouped as closed-loop tracking, while the last two are categorized as open-loop tracking. As such, the analysis for these cases will be presented in two subsections, one for closed-loop tracking and one for open-loop tracking, and the impact of availability of other channels is considered in each subsection.

7.1.1 Closed-Loop Cycle-Slip Performance

The average time to cycle slips for an unstressed, first order Costas PLL is given in [44] as follows:

$$\bar{T}_{slip1} = \frac{\pi^2}{8\sigma_\varphi^2 B_{n\varphi}} I_0^2 \left(\frac{1}{4\sigma_\varphi^2} \right) \text{ (sec)} \quad (7.1)$$

where

$$\sigma_\varphi^2 = \text{var}(\varepsilon_\varphi) \text{ } (\varepsilon_\varphi \text{ is in rad, and is defined in Eq. 6.20})$$

$$I_0 = \text{Modified Bessel function of zeroth order}$$

For a first-order Costas loop with dynamic stress, the equivalent expression is also given in [44]:

$$\bar{T}_{slip2} = \frac{\pi}{4B_{n\varphi}\gamma} \tanh \left(\frac{\pi\gamma}{2\sigma_\varphi^2} \right) \left[I_0^2 \left(\frac{1}{4\sigma_\varphi^2} \right) + 2 \sum_{n=1}^{\infty} (-1)^n \frac{I_n^2 \left(\frac{1}{4\sigma_\varphi^2} \right)}{1 + \left(\frac{2n\sigma_\varphi^2}{\gamma^2} \right)^2} \right] \text{ (sec)} \quad (7.2)$$

where

$$\gamma = \text{Constant steady-state error of PLL (rad) due to dynamic stress}$$

$$I_n = \text{Modified Bessel function of } n^{th} \text{ order}$$

Using the result from either of these equations, the probability of a cycle slip can be computed as a function of time:

$$P_{slip}(t) = 1 - e^{-t/\bar{T}_{slip}} \quad (7.3)$$

While no closed-form expressions have been found for higher order loops, it has been shown by simulation that a second order loop has similar characteristics if σ_ϕ^2 is about 1dB higher than that for a first-order loop [47]. That is, the average time to slip of a second-order loop would be only a few percent shorter than that for a first-order loop with the same SNR. Thereby, the cycle-slip analysis for a first-order loop is assumed to be representative of a second-order loop.

The condition of having a dynamically “unstressed” loop simply requires that the loop have no steady-state error. The second-order loops discussed in Chapter 6 meet this requirement while the antenna is stationary or has constant velocity, as the Type II system can track reference-phase constants and ramps with zero steady-state error [29]. However, the loop incurs dynamic stress or steady-state error if the reference-phase input is parabolic. A parabolic reference-phase input corresponds to a ramp reference-frequency input, which occurs under constant acceleration, and is a common condition of any moving vehicle. For the second-order loops discussed in this chapter, the steady-state error for a constant Doppler rate is:

$$\gamma \approx \frac{\dot{f}_{dopp}}{3.56B_{n\phi}^2} \text{ (cycles)} \quad (7.4)$$

From the time-derivative of Eq. 6.11, the Doppler rate turns out to be a function of the velocity and acceleration of the vehicle, the LOS vector, and the rate of change of the LOS vector. A reasonable simplification assumes that the rate of the change of the LOS vector is much smaller than that of the acceleration of the vehicle; furthermore, to drop the dependence on the LOS vector, a worst-case scenario can be assumed where the vehicle acceleration is parallel to the LOS vector. Implementing these assumptions on Eq. 7.4 results in the following equation:

$$\gamma_{dyn} \approx \frac{a_{max}}{3.56\lambda_{L1}B_{n\phi}^2} \text{ (cycles)} \quad (7.5)$$

where a_{max} is the magnitude of the maximum expected acceleration of the vehicle. In the following analysis for cases with dynamic stress, a_{max} is assumed to be 5m/sec², which corresponds to a change in velocity of 60mph in about 5 seconds.

Equations 7.1 and 7.2 can both be used to analyze the cycle slip performance of a PLL, with or without Doppler aiding. For a standard PLL without Doppler aiding, dynamic stress would apply and Eq. 7.2 would be used. For a Doppler-aided PLL with GPS navigation available, the bandwidth of the inertial sensors would be expected to remove most of the effect of dynamic stress from vehicle motion, and the error of the Doppler estimates would be bounded; in this case, Eq. 7.1 can be used under the assumption that Doppler-estimate errors have the unbiased Gauss-Markov properties discussed in Subsection 6.3.5, and thus incur little dynamic stress.

If GPS is not available and the navigation filter is in dead-reckoning mode, the growing

error of the external Doppler-estimate will have a similar effect as dynamic stress. If the model presented in Eq. 6.54 is used to model the error growth in the Doppler estimates, the standard deviation of the errors would grow proportionally to the square-root of time (not as severe as a ramp). For this type of dynamic stress, a maximum value of Doppler-error rate cannot be specified exactly, because the random walk model is a stochastic process. However, the statistics of the slope of a line-segment connecting any two points in a random walk sequence can be expressed as a function of the driving-noise variance and the time elapsed between the two points. Using this technique, a conservative value of three standard deviations of the slope (for a 1-second elapsed time between points) will be used to quantify the dynamic stress due to growing Doppler-estimate error:

$$\gamma_{\delta f_{dopp}} = \frac{3 \sigma_{\delta \dot{f}}|_{t=1 \text{ sec}}}{3.56 B_{n\varphi}^2} \quad (\text{cycles}) \quad (7.6)$$

where $\sigma_{\delta \dot{f}}|_{t=1 \text{ sec}} \approx 0.33 \text{ Hz/sec}$ if the model in Eq. 6.54 is used.

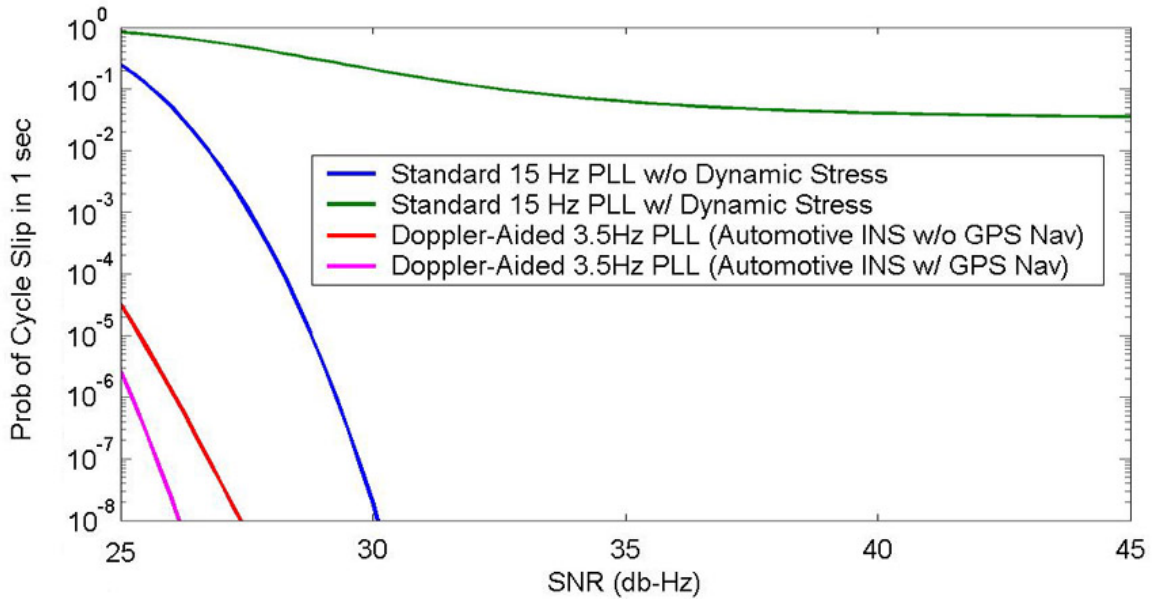


Figure 7.1: Closed-Loop Cycle-Slip Performance of Traditional and Doppler-Aided PLLs

Figure 7.1 shows the probabilities of encountering a cycle slip in a one-second period for all scenarios of closed-loop tracking. For the standard 15Hz PLL with no Doppler-aiding, the dynamic stress in Eq. 7.5 has a significant impact on cycle slip performance. In fact, a GPS receiver meant to operate with this dynamic range would likely be designed with a higher bandwidth than 15Hz (or have third-order PLLs) to reduce the likelihood of cycle slips during vehicle acceleration. With SNR below 35dB-Hz, which can occur with RF interference or signal blockage, the adverse impact of larger phase noise on the 15Hz PLL is clearly evident through the high probabilities of cycle-slips. In contrast, the 3.5Hz Doppler-aided PLL shows a vast reduction in

cycle-slip rates. This improvement is due to better suppression of phase-noise with lower PLL bandwidth, and to the removal of most dynamic stress by the high-bandwidth of the inertial sensors. It should be noted that the plot that applies to Doppler-aiding with the INS only (w/o GPS navigation) refers to a one-second period of dead-reckoning since the last GPS update.

7.1.2 Open-Loop Cycle-Slip Performance

As discussed in Subsection 6.3.2, accurate estimates of the Doppler and clock-error frequencies can be used for short periods of time to execute open-loop tracking, such that no cycle slip occurs when closed-loop tracking is recovered. This section explores the capability of this feature by quantifying the probability of a cycle slip as a function of time with open-loop phase tracking.

The model that will be used to study phase-error growth in open-loop tracking is shown in Figure 7.2. The lower branch represents a scenario in which the channel is not tracking, but there are enough other channels tracking to generate a GPS navigation solution for the navigation filter. As shown before in Subsection 6.3.5, the Doppler-estimate errors, originating from the navigation filter's velocity-estimate errors, resemble the exponentially correlated noise of a first-order Markov process if GPS is available. The single-pole low-pass filter shown in Figure 7.2 comprises this model, and is consistent with the PSD model in Eq. 6.53, with β given by Eq. 6.46. The upper branch represents the case where no GPS navigation is available, and the Doppler-estimate errors resemble a random walk. The variance of the uncorrelated, Gaussian driving noise (σ_v^2) for both models is equivalent to the numerator of Eqs. 6.53 and 6.54. Note that the upper branch also has an initial condition specified for the integration process of the NCO, and this initial condition will originate from the last Doppler estimate obtained with GPS navigation. This initial condition is included, as it can have a large impact on the phase-error growth in open-loop tracking with no GPS navigation available.

The probability of a cycle slip upon recovery of signal tracking can be expressed in terms of the accumulated phase error during open-loop tracking:

$$P_{slip}(t) = P(|\delta\varphi| > 1/4 \text{ cycle}) \quad (7.7)$$

If the phase-error growth is greater than a quarter cycle, then the PLL may lock onto a different carrier cycle than that before tracking was lost, essentially resulting in a cycle slip. The probability of this event is a function of time, and is determined analytically by treating the accumulated phase-error as a Gaussian stochastic process with zero mean. The discrete equations that are

used to compute the accumulated phase-error are as follows:

$$\delta f_1(t_{i+1}) = e^{-\beta \Delta t_{PLL}} \delta f_1(t_i) + \sqrt{\frac{1}{2\beta} (1 - e^{-2\beta \Delta t_{PLL}})} v(t_i) \quad (7.8)$$

$$\delta f_2(t_{i+1}) = \delta f_2(t_i) + \sqrt{\Delta t_{PLL}} v(t_i) \quad (7.9)$$

$$\delta \varphi_1(t_{i+1}) = \delta \varphi_1(t_i) + \Delta t_{PLL} \delta f_1(t_i) \quad (7.10)$$

$$\delta \varphi_2(t_{i+1}) = \delta \varphi_2(t_i) + \Delta t_{PLL} \delta f_2(t_i) \quad (7.11)$$

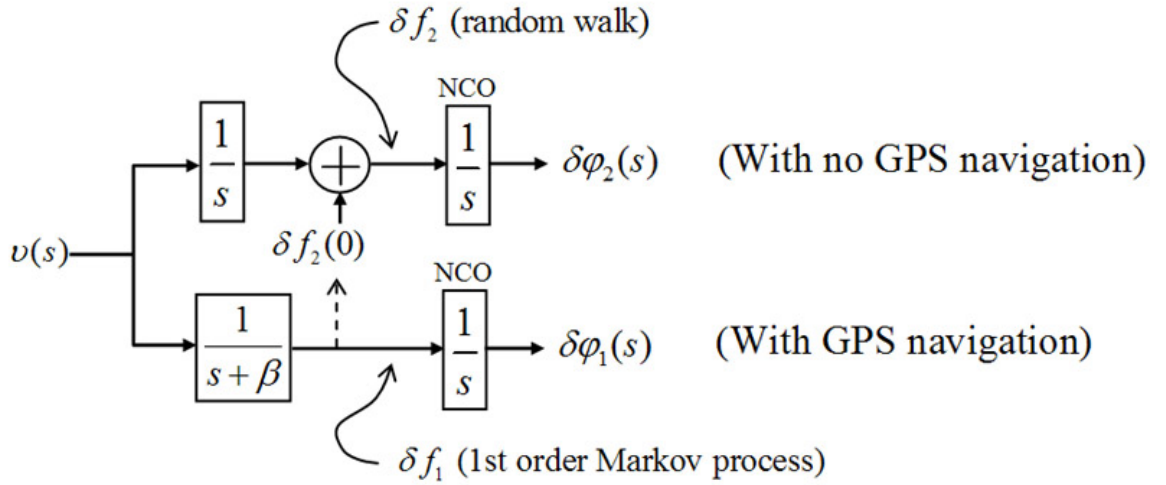


Figure 7.2: Model for Open-Loop Phase-Tracking Performance

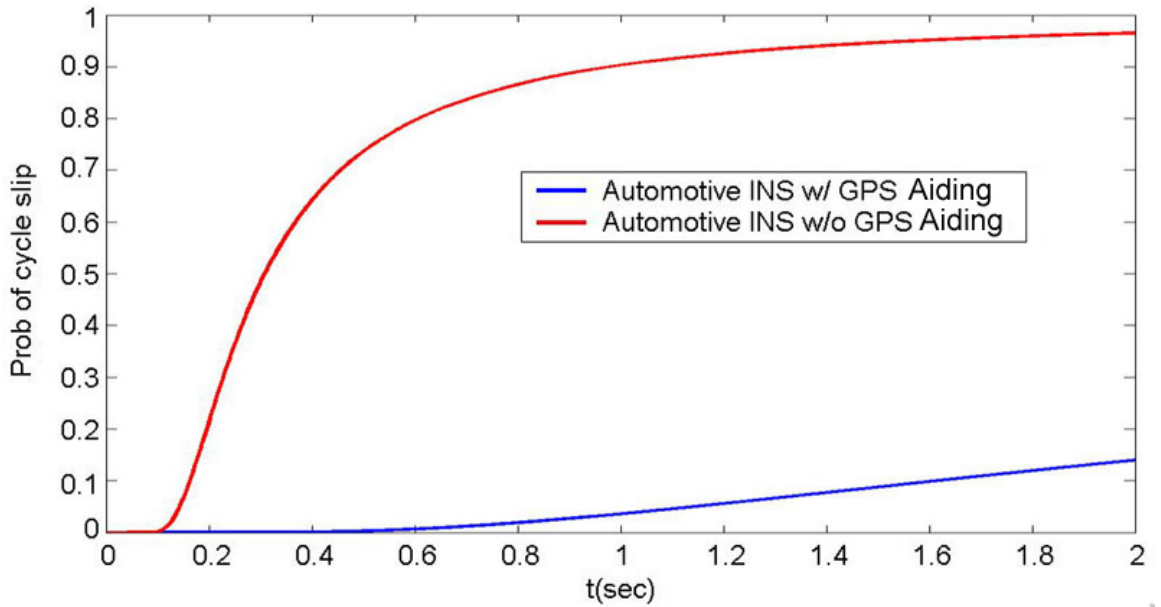


Figure 7.3: Open-Loop Cycle-Slip Performance of a Doppler-Aided PLL

The effective sampling time, Δt_{PLL} , is the length of the averaging time of the GPS receiver correlators, and the minimum value of 1ms is used in this analysis. The variances of the accumulated phase errors are time-dependent variables, and N samples after loss of closed-loop tracking, are given by the following equations:

$$\sigma_{\delta\varphi_1}^2(N) = \Delta t_{PLL}^2 \sigma_{\delta f_1}^2 \left(N + \sum_{n=1}^{N-1} 2n e^{-\beta(N+n)\Delta t_{PLL}} \right) \quad (7.12)$$

$$\sigma_{\delta\varphi_2}^2(N) = \Delta t_{PLL}^2 \left(N^2 \sigma_{\delta f_1}^2 + \Delta t_{PLL} \sigma_v^2 \sum_{n=1}^{N-1} n^2 \right) \quad (7.13)$$

$\sigma_{\delta\varphi_1}^2$ and $\sigma_{\delta\varphi_2}^2$ are the variances of $\delta\varphi_1$ and $\delta\varphi_2$, respectively, and $\sigma_{\delta f_1}^2$ is the variance of the Markov-process that represents Doppler-estimate errors with GPS navigation. Note, these equations assume zero initial conditions, as they are negligible when compared to the rapid growth of phase-error during open-loop tracking. Based on the variance values of Eqs. 7.12 and 7.13, Figure 7.3 shows how the probability of a cycle slip grows a function of time with open-loop tracking. The two graphs illustrate how having GPS navigation is a major advantage for open-loop tracking for more than about a tenth of a second with an automotive-grade INS. However, even without GPS navigation, the results based on these models suggest that open-loop tracking could have a significant reduction in the number of cycle slips during normal driving conditions, as the average lengths of time of GPS outages presented in Table 5.1 are on the order of fractions of a second.

7.2 Performance Results

This section presents results that illustrate most of the benefits of Doppler aiding that have been discussed, including reduced sensitivity to phase-noise without sacrificing phase-tracking bandwidth, and open-loop tracking to avert cycle slips. These results will be discussed in two subsections; the first will show the PLL tracking results from a detailed simulation of GPS signals and IMU data through a maneuver with relatively fast dynamics, and includes a two-antenna GPS attitude system; the second subsection will include post-processed results with real GPS samples from one of the antennas in the automobile attitude system, and real inertial-sensor data from the accompanying automotive-grade INS. The two subsections are meant to be complementary, as the real-data test was unable to recreate the fast dynamics or dual-antenna system that was achievable through simulation. However, the validation of simulated results with real GPS and IMU data taken from a moving car, even if it is for a single receiver, is an important result that supports the feasibility of implementing Doppler aiding with inexpensive hardware in an

automobile GPS attitude system.

The results that will be shown in this section were obtained with the use of a software receiver, written specifically for this project. This tool was needed to implement GPS tracking-loops that could incorporate external frequency-estimates into the tracking loops. In addition to having PLLs and DLLs for multiple channels, the software receiver computes PVT at a user-specified update rate, and if data from two antennas is available, attitude measurements are also given. The algorithms used to generate these measurements are essentially the same to those used in a real GPS receiver. The primary input file to the software receiver is a binary data file that contains sequential samples of the GPS signal after downconversion by an RF front-end. The GPS data samples may be real or simulated, as long as simulated data is true to the structure of the GPS signal including Doppler shifts, range delays, and navigation data bits. The intermediate frequency, sampling frequency, and bits-per-sample values are specified by the user for a specific input file.

A few different versions of this software were developed as part of this research, including variants that read GPS samples from one input file (for single-antenna) or from two input files (two antennas for attitude determination). In either case, the software offers the option to enable Doppler aiding in all tracking loops if input files containing IMU or INS data are provided. These inertial-data input files can be also be real or simulated. Other useful features of the software include the capability to corrupt specific channels by reducing SNR or removing them entirely to simulate brief outages, and to change parameters of the tracking loops very easily to alter bandwidth or loop-filter structure.

7.2.1 Results from Simulation

In addition to the software receiver described above, GPS signal-simulation software was also developed as a tool for advancing the study of inertially-aided GPS tracking loops before having the means to collect real GPS samples. This software generates a GPS signal as it would appear at the output of the RF front end of a GPS receiver. The Doppler-shifts, signal delays (due to true range), and navigation data-bits are recreated from real satellite-ephemeris data from an input file. Inputs from the user include the sampling frequency, intermediate frequency, initial GPS time, and trajectory information (including attitude for dual-antenna simulation). The simulated signal also includes frequency/phase noise from a local reference oscillator, and the desired noise statistics of the clock can also be specified by the user. In its current version, this software does not include the capability to specify variable SNR on the different channels or atmospheric delays on the signals; these aspects are not critical to the work in this thesis, though they could be included easily to improve the fidelity of the GPS signal simulations.

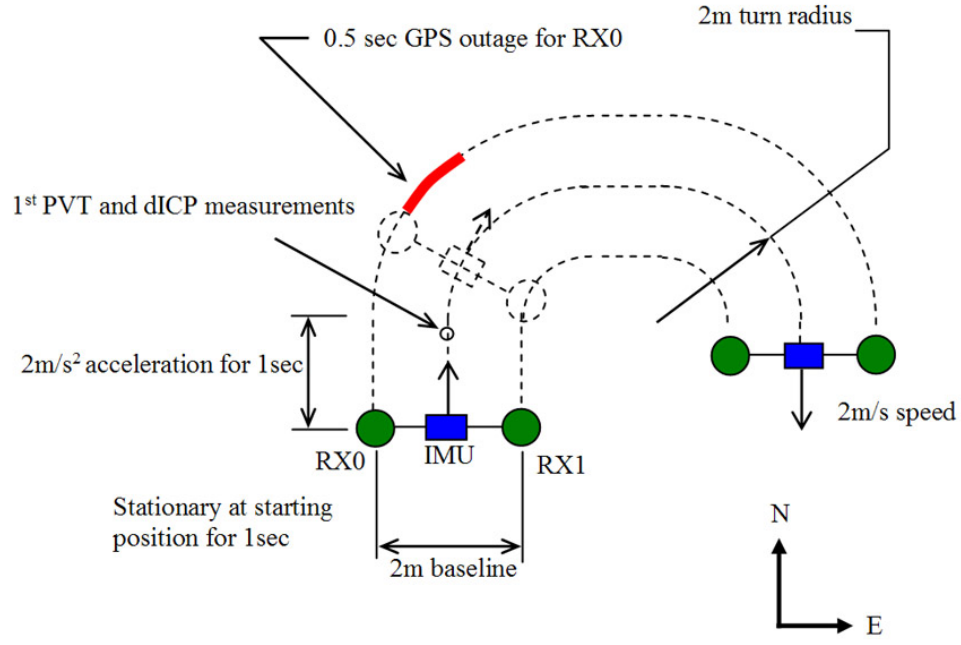


Figure 7.4: Simulated Trajectory with Dual Antennas

The simulated trajectory that applies to the results in this section is shown in Figure 7.4. The simulation consisted of generating synchronized GPS samples (containing signals from six SVs) for both antennas and for the inertial sensors, and included noise characteristics of automotive-grade inertial sensors and of a TCXO. The final speed of the baseline center point (also the INS) was 2m/s after the initial 1-second acceleration. The two 90° turns incorporate fast dynamics into the simulation, as the endpoints of the baseline experience very fast acceleration as the turns begin and end. A half-second outage for one of the antennas was also included during the first turn to demonstrate open-loop tracking.

The files generated from this simulation were then used as the input files to the software receiver, which generated GPS PVT solutions for both antennas, and attitude measurements for the one baseline. Some of these high-level outputs from the software receiver are shown in Figure 7.5, which includes the GPS position, velocity, and yaw angle measurements throughout the simulated trajectory. Note that the velocity plots have an anomalous GPS velocity measurement at $t \approx 2.9\text{sec}$; this point corresponds to the first PVT measurement after the GPS outage on RX0, as f_{PLL} experiences a transient when the PLL realigns the phase of the replicated carrier.

Figure 7.6 includes graphs of various frequencies relevant to one of the channels in the software receiver, including f_{PLL} , the estimated Doppler frequency (\hat{f}_{dopp}), the estimated clock error-frequency (\hat{f}_{clk}), and the external frequency estimate (\hat{f}_{ext}), which is the sum of \hat{f}_{dopp} and \hat{f}_{clk} with any biases removed. These quantities are shown for each of the receivers, and are distinctly different for each receiver during the turns as the velocities of the antennas are different when

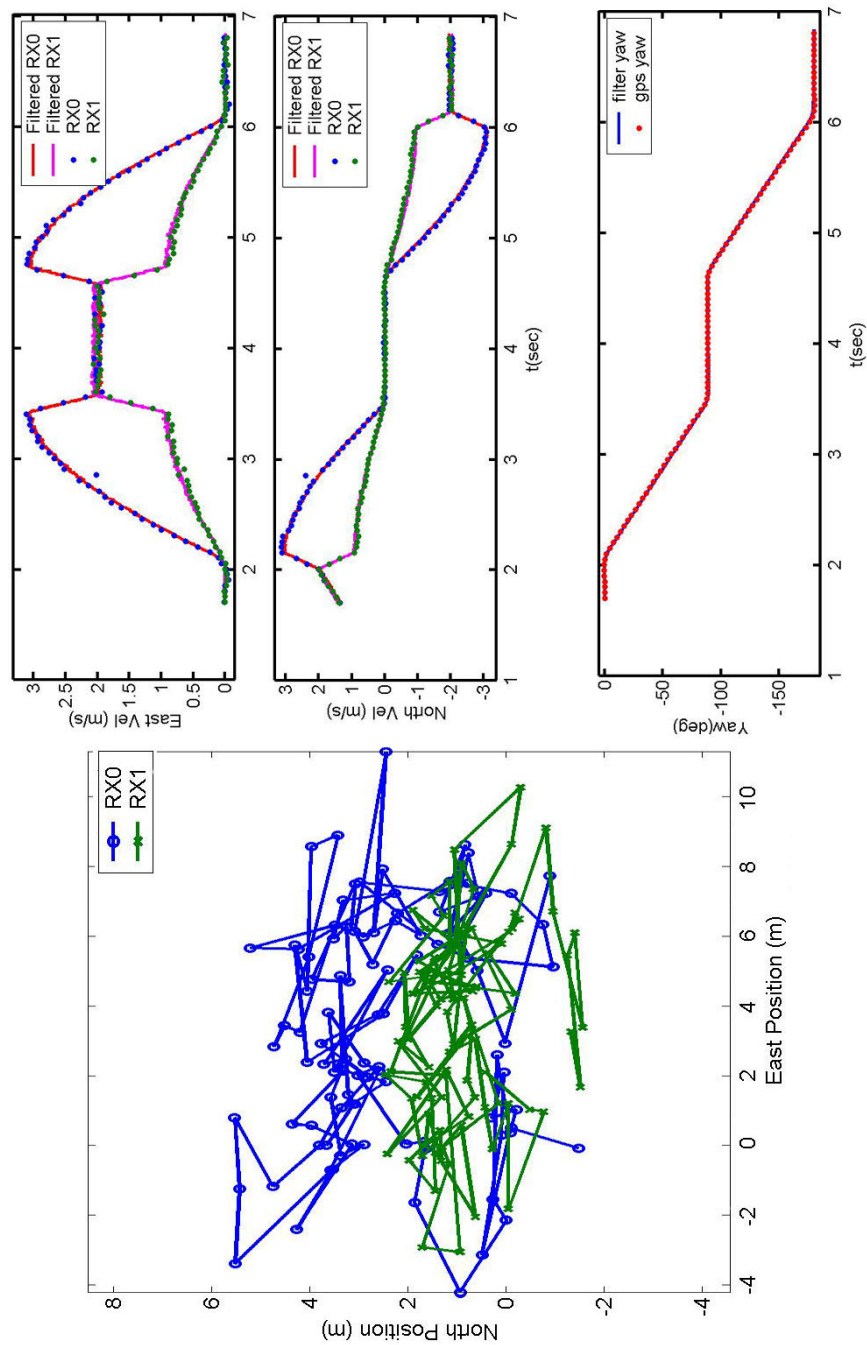


Figure 7.5: Software-Receiver Outputs from Simulation Test Case

the baseline rotates. As shown, Doppler aiding is enabled at $t \approx 1.8\text{sec}$, shortly after the first PVT solution and initialization of the INS. Prior to this time, a traditional 15Hz PLL is used. A reduction in frequency noise is clearly evident after the transition to the 3.5Hz PLL with external frequency aiding. As expected, the attenuation of high-bandwidth frequency noise improves by a factor approximately equal to the ratio of PLL loop bandwidths, in this case by a factor of 4.3. It is also important to note that despite the use of a lower bandwidth PLL, f_{PLL} tracks the fast changes in Doppler frequency during turns with no apparent transients, as the external Doppler estimates capture these fast dynamics very well.

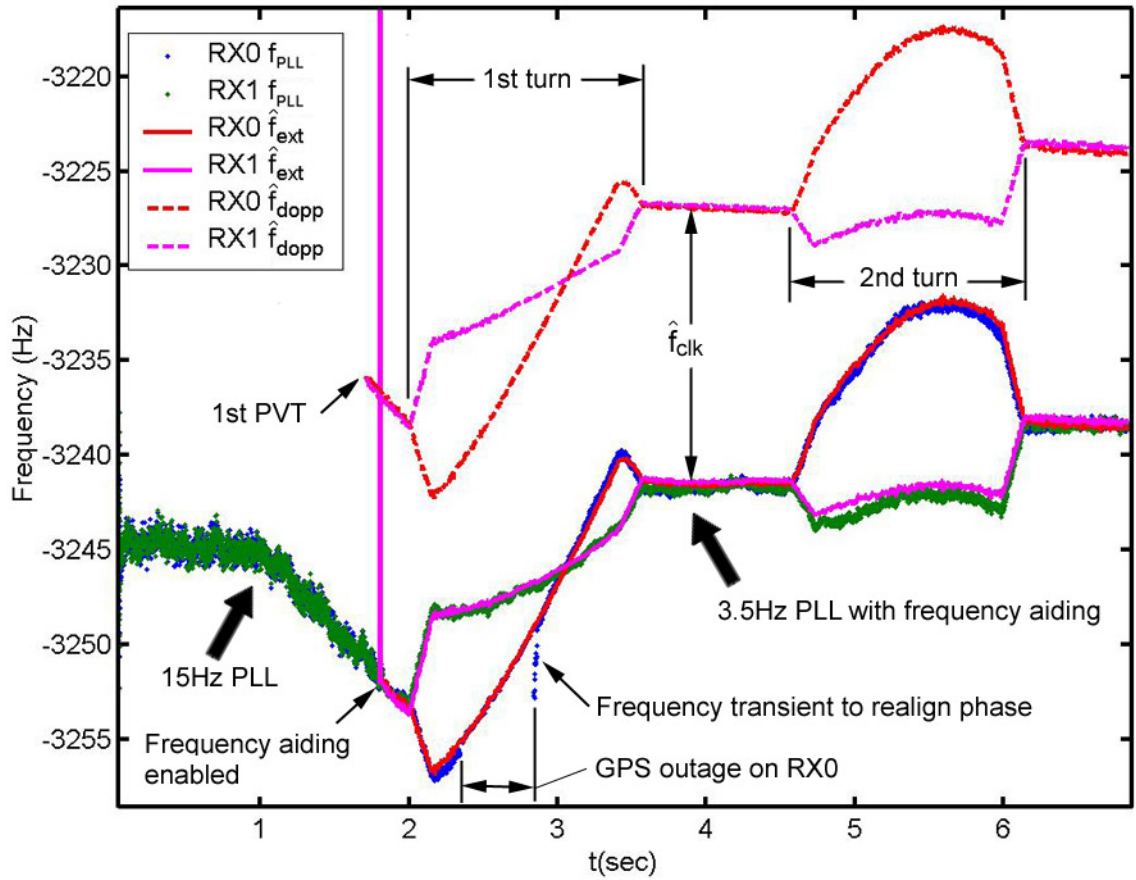


Figure 7.6: Channel 1 Frequencies for Simulation Test-Case

The open-loop tracking period during the GPS outage is labeled in Figure 7.6, between $t \approx 2.3\text{sec}$ and $t \approx 2.8\text{sec}$. Note how the Doppler-frequency estimates are similar in shape but different in magnitude from f_{PLL} . However, if \hat{f}_{clk} is also used, then the combined external frequency estimate is a very good approximation of f_{PLL} , and makes open-loop tracking possible. The frequency transient at the end of the outage is imposed by the loop filter as it strives to line up the phase offset that was accumulated during the outage.

The measured phase-jitter for RX0 is shown in Figure 7.7, where the transition to a Doppler-aided PLL is also labeled. Since the simulated channel has good SNR, the loop-bandwidth design

graphs in Figure 6.22 suggest that a 15Hz PLL may have lower phase-jitter than a Doppler-aided 3.5Hz PLL, though the difference will depend on the amount of dynamic stress. Nevertheless, the phase jitter should remain close to the desired limit of 0.1 rad for the Doppler-aided case. As shown, the specification is slightly exceeded, probably because of exaggerated simulated TCXO noise.

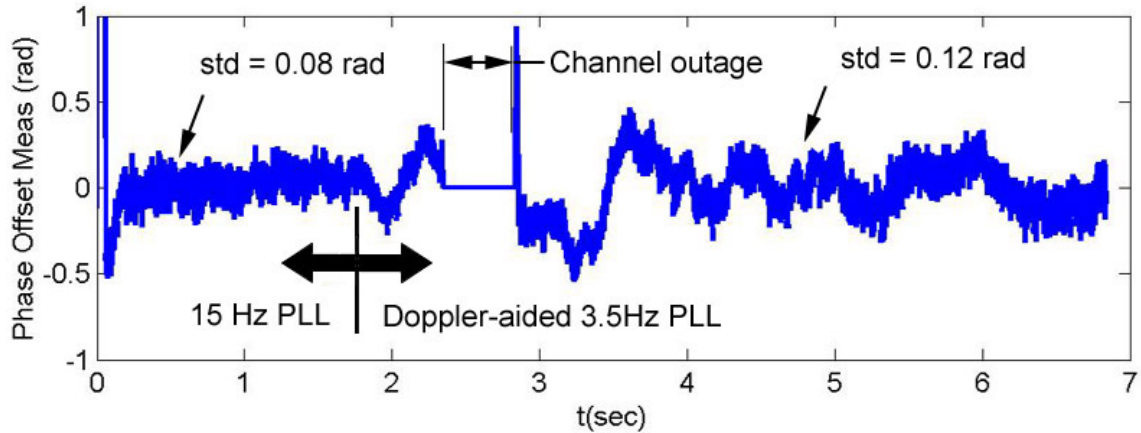


Figure 7.7: Channel 1 Phase-Jitter for Simulation Test-Case

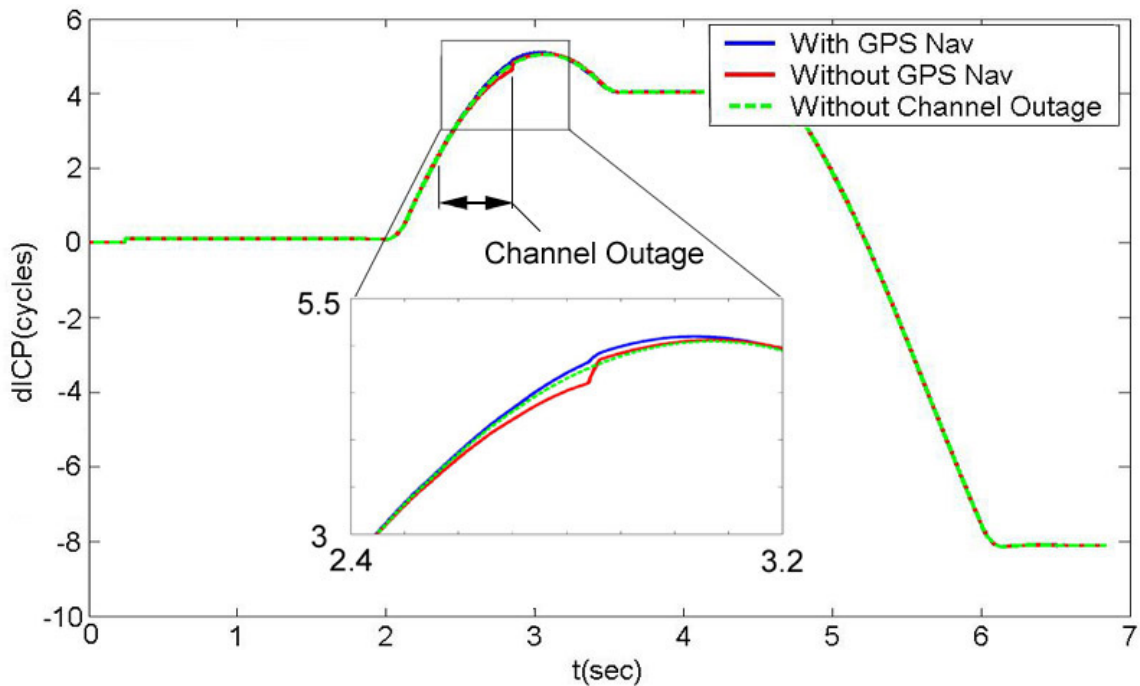


Figure 7.8: dICP Measurements and Cycle-Slip Aversion for Simulation Test Case, with and without GPS Navigation during Channel Outage

The dICP value for this channel is shown in Figure 7.8. The cases shown include simulations without the GPS outage (for comparison), and with the GPS outage with and without GPS navigation during open-loop tracking; in both cases a cycle slip is averted after the outage. The subtle

distinction between the graphs is emphasized in the detail window, which focuses on the section of the plot where closed-loop tracking is reacquired. As expected, open-loop tracking performs better (accrues less phase error) if GPS navigation is available during the channel outage; this difference is evident as the smaller dICP transient for the case with GPS navigation. This result supports the cycle-slip analysis results shown in Figure 7.3, which indicate that the probability of a cycle slip after half a second is significantly higher if GPS navigation is not available.

Figure 7.9 shows the performance of the DLL throughout the simulation. As shown, there is no noticeable effect in the code-phase offset upon recovery of the signal after the outage, as the 300m code chip is much more tolerant to accumulated INS error than the 19cm carrier wavelength in open-loop tracking. This contrast is especially true if the DLL is rate-aided, as it is in this case. Note, this plot is normalized such that the mean prompt power is unity.

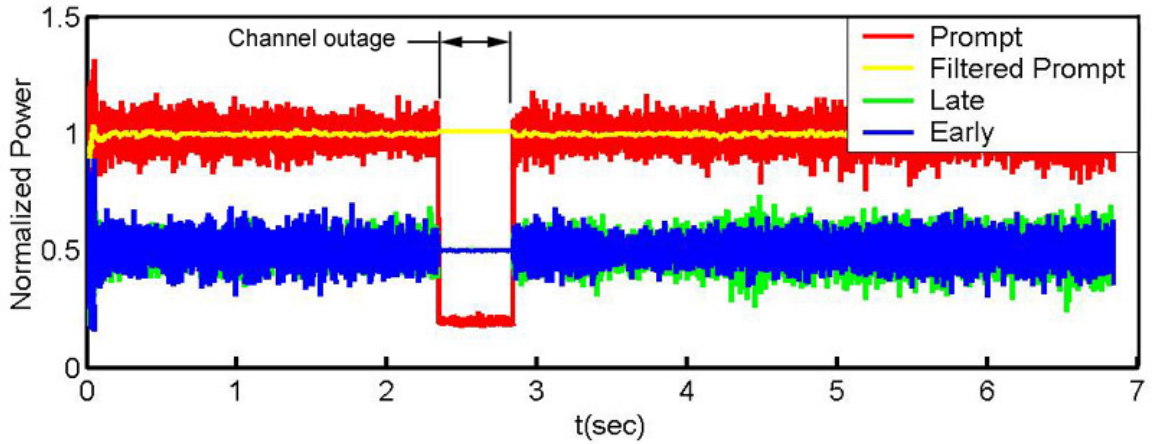
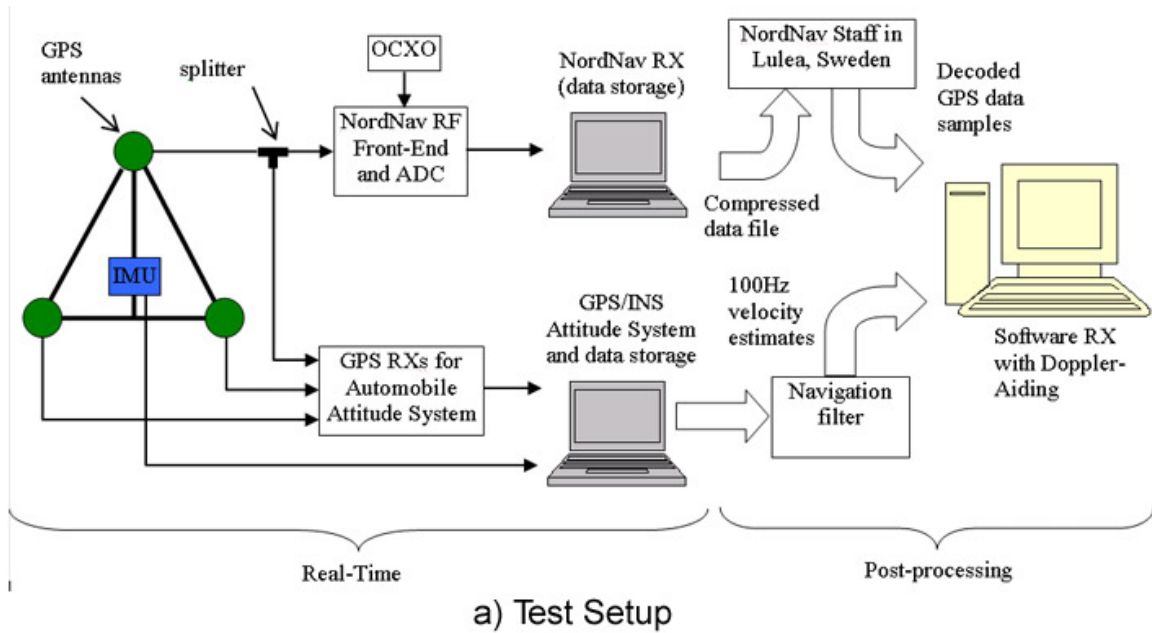


Figure 7.9: Early, Late, and Prompt Power from DLL Outputs for Simulation Test Case

7.2.2 Results with Real Data

The results presented in the previous subsection are very encouraging with regards to improving phase-tracking performance with Doppler aiding. The next logical step in the pursuit of a GPS/INS system that has such capabilities is to verify these results with real inertial-sensor data and real GPS data. With a version of the GPS signal-simulation tool that uses arbitrary trajectory waypoints, studies with real inertial-sensor data and simulated GPS samples were also performed, and yielded results similar to those in the previous section. The more challenging part of validation with real data is obtaining real GPS-signal samples to track with the software receiver. At this time, this difficulty is due to the lack of inexpensive equipment that can be used to sample the GPS signal from an RF front-end at sampling frequencies over 4MHz and store the samples without discontinuities to computer memory for an extended period of time (10 seconds or more). Many different hardware and software setups were attempted to collect this type of

data, and most were partially successful, but for some shortcoming that made the data unusable. Such problems included gaps in the data and phase discontinuities in the replicated carrier.



b) Test Location



c) Test Vehicle

Figure 7.10: Real-Data Test Setup, Location, and Vehicle

With cooperation of the JPALS group (part of the GPS lab) at Stanford University, a set of equipment was finally compiled that was capable of collecting the necessary data, at least for several seconds. The setup consisted of the automobile attitude system discussed in Chapters 3-5 to collect attitude and accelerometer data, and a NordNav software receiver (in data collection mode) tapped to the front antenna to collect GPS samples. To accommodate the needs of concurrent tests for the JPALS group, the baselines were 1 meter long, the antennas were of survey-grade quality, and an OCXO reference oscillator was used. The compressed data files generated by the NordNav software required decoding to make them readable by the software

receiver, and this task was done by NordNav staff for 10 seconds of the data set which included the first turn after the vehicle started moving. A schematic of the test setup and photographs of the test location and test vehicle are shown in Figure 7.10.

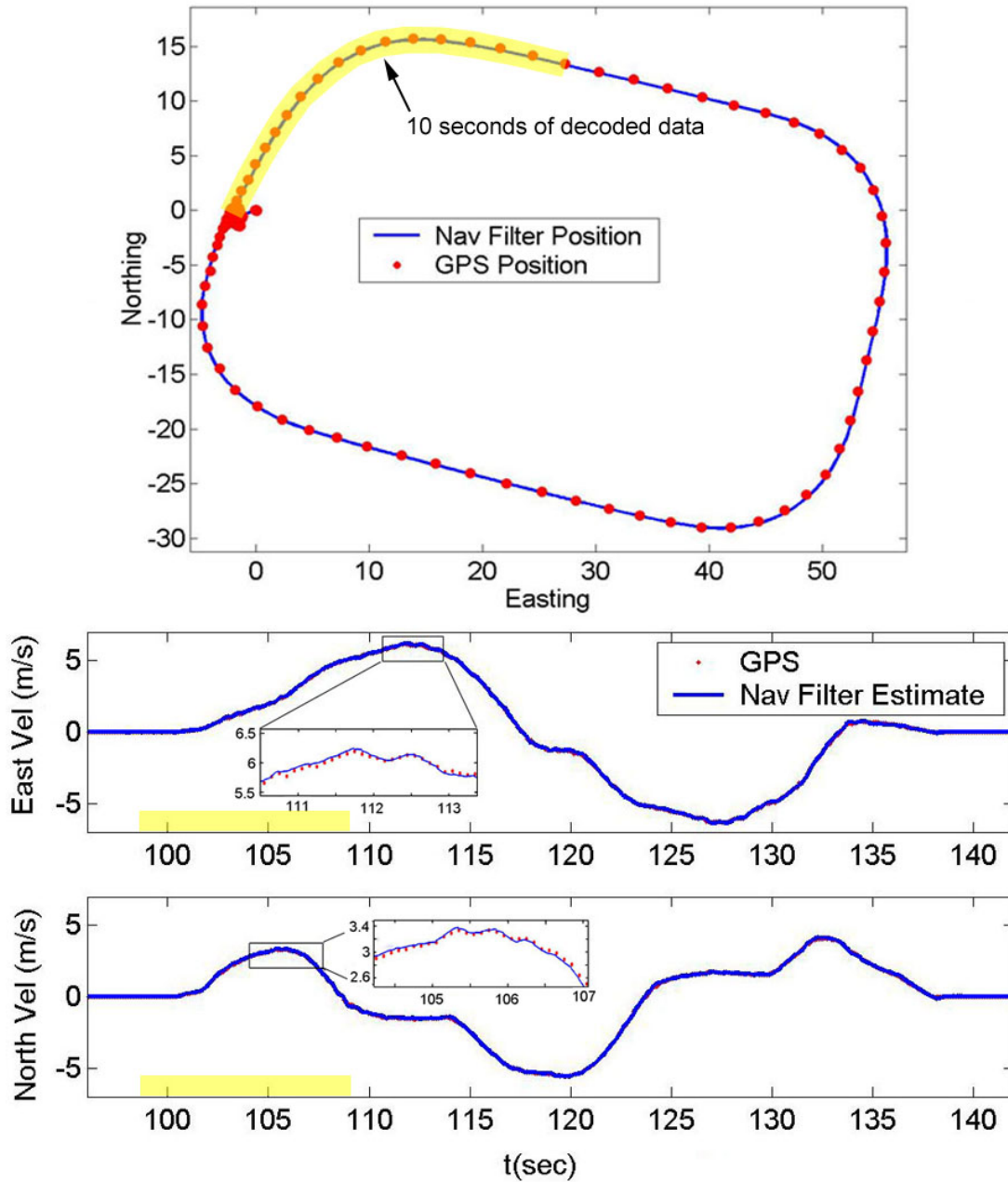


Figure 7.11: Position and Velocity Measurements from GPS and Navigation Filter for Real-Data Test Case

After the data collection, the stored attitude and accelerometer data were post-processed with a navigation filter to get vehicle velocity estimates at the sampling rate of the IMU (100Hz) to implement Doppler aiding. The position and velocity outputs of the navigation filter are shown in Figure 7.11, and the 10-second portion of decoded data from NordNav is highlighted. The

velocity plots include detail windows to illustrate some of the effects of vehicle dynamics on estimated velocities, which will also affect the the Doppler estimates.

The software receiver was able to acquire and track four satellites from the decoded data: SV#s 5,17,24 and 30, which all had high elevations at the time and location of the test (Stanford University, at GPS time 259951 on Dec 16, 2003). The PLL frequencies for channel 4 (corresponding to SV#30), are depicted in Figure 7.12. In comparison to Figure 7.6, which shows similar information for the simulation test-case (but for two antennas), the slower dynamics of the real vehicle produce no sharp changes in Doppler-rate. However, they do contain dynamic elements that are not included in the simulation, such as the suspension dynamics pointed out in the figure. Note, the INS captures the nuances of these dynamics in the Doppler-estimates, and they are aptly passed on to \hat{f}_{ext} and show up in f_{PLL} . In regard to suppressing frequency noise, performance is remarkably similar to the simulation case.

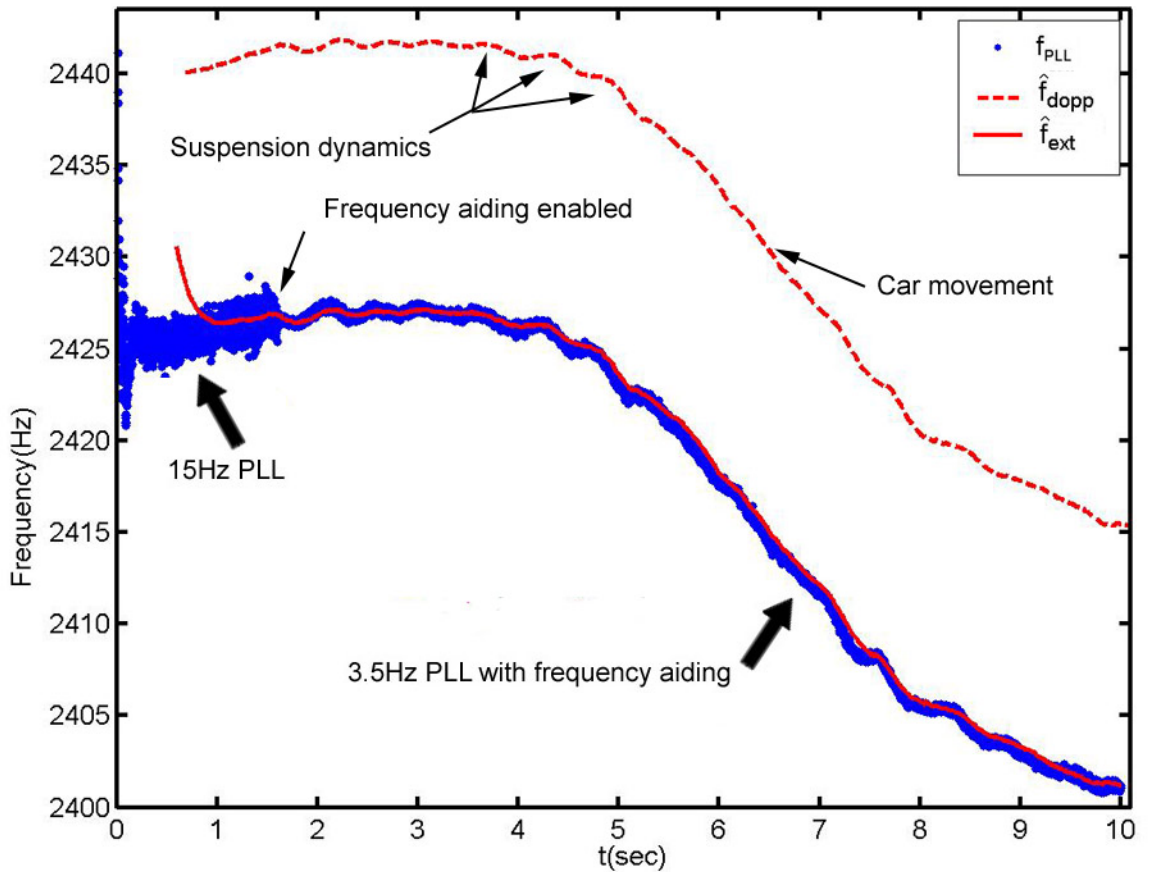


Figure 7.12: Channel 4 Frequencies for Real-Data Test Case

The presence of only four channels tracking high-elevation satellites, resulted in relatively poor HDOP for determining GPS position and velocity, and correspondingly, estimates of the clock error-frequency were also not as accurate as they can be. For this reason, the initial value of \hat{f}_{ext} at $t \approx 0.5sec$ is about 5Hz off from the apparent value of f_{PLL} . The transient that occurs

as \hat{f}_{ext} approximates f_{PLL} is due to the low-pass filter shown in Figure 6.12a, which is part of the mechanism to remove any biases from the external frequency estimates.

The phase-jitter of the PLL is shown in Figure 7.13, and is also very similar to what was observed in simulation (Figure 7.7). Once again, the value of the phase-jitter after Doppler-aiding is invoked is about 0.02 rad higher than the specification. In contrast to the simulated case, however, the larger phase-jitter errors seem to coincide with the suspension dynamics rather than with random clock noise. This behavior suggests that performance of the navigation filter could be improved to better account for the non collocation of the GPS antenna and the IMU.

Figure 7.14 shows the performance of the DLL throughout the section of decoded data. The only notable difference in this graph when compared to the simulated case (Figure 7.14) is that the SNR in the real case is not as favorable as in the simulated case. The relative SNR in the two cases can be ascertained from these normalized plots by comparing the magnitude of the noise in the any of the correlator power measurements (prompt, early or late) between the figures.

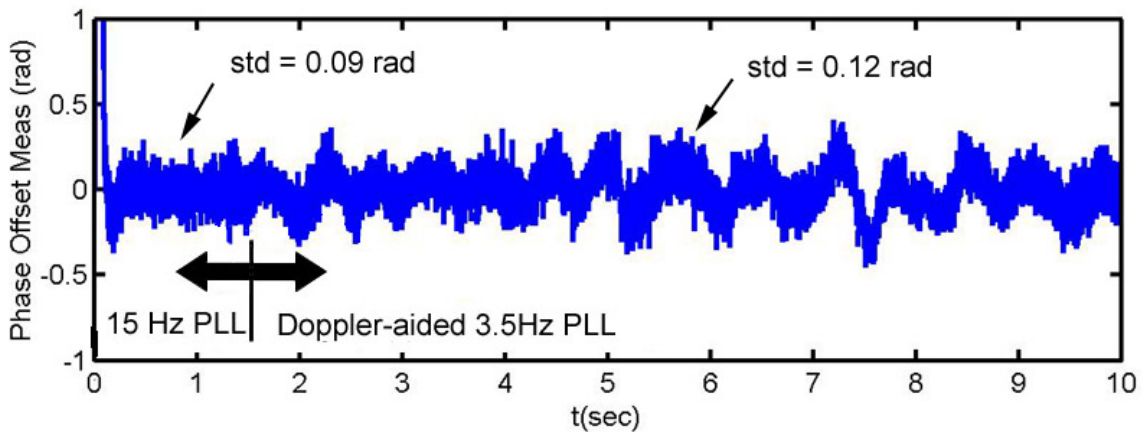


Figure 7.13: Channel 4 Phase-Jitter for Rea-Data Test Case

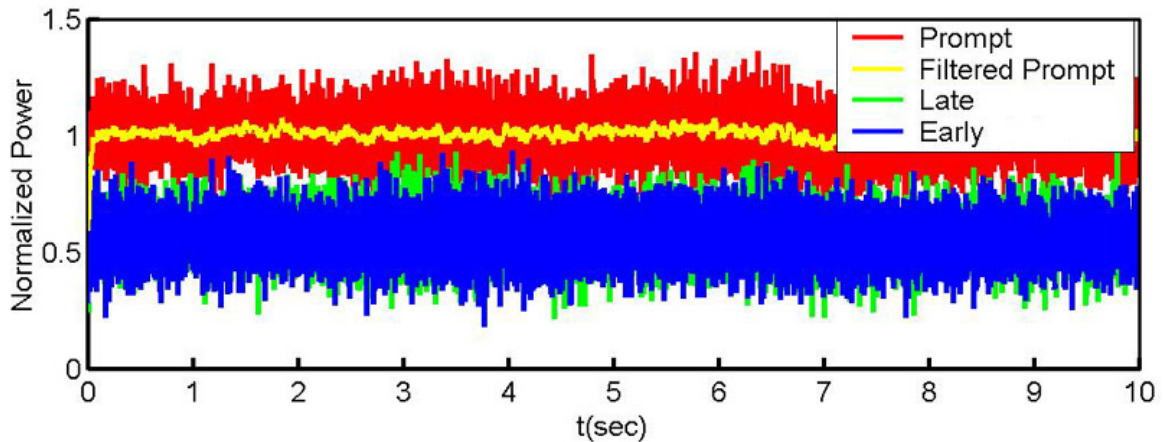


Figure 7.14: Early, Late, and Prompt Power from DLL Outputs for Real-Data Test Case

Chapter 8

Conclusions

8.1 Summary

Many of the emerging sophisticated control and navigation applications in modern automobiles require measurements of the vehicle's attitude. Some of these applications involve detection of hazardous situations (such as skidding and drifting off the road) and corrective action. As such, these systems and the sensors on which they depend need a high level of availability and integrity. The emergence of these important safety features is slowed by the absence of attitude sensors that meet the low-cost, accuracy, and reliability standards needed for practical implementations in consumer vehicles. To address this problem, the research covered in this thesis has presented a possible solution for the mechanization of a robust and inexpensive attitude system for automobiles, based on GPS and automotive-grade inertial sensors.

GPS differential carrier-phase is currently one of the most accurate means of measuring a vehicle's attitude, and has been used widely in applications where signal integrity and satellite visibility are rarely compromised, such as in aircraft and farm vehicles. Unfortunately, the operating conditions of a car in a city compromise the integrity and availability of GPS carrier-phase significantly. The topics presented in this thesis focused largely on methods for improving availability and integrity of an attitude system based on GPS carrier-phase, subject to the challenging conditions of urban driving. The main ideas of this work are summarized as follows:

- The number of GPS carrier-phase measurements in typical driving conditions can often drop below three as a result of signal blockages or inability to recover integer ambiguities in the presence of phase-noise. With less than four good carrier-phase measurements, traditional attitude-determination methods with GPS are often unusable, which may result in unavailability of attitude-measurements for long periods of time. Coupling with an automotive-grade INS can provide availability throughout short GPS-attitude outages, but would be limited by large error accumulation during periods where only one or two satellites are visible for prolonged periods. The one-satellite attitude algorithms developed in Chapter 2 were shown to have a very positive impact on availability, as they capitalize on

every GPS satellite that is visible to measure either of the degrees of freedom of a baseline. These algorithms were incorporated with the more accurate but less robust methods of attitude determination with multiple satellites in a real-time prototype. One-satellite attitude determination was found to increase GPS yaw availability from about 50% of the time to over 90% of the time during several suburban driving tests, and still maintain accuracy of approximately one degree.

- Integration with inertial sensors was found to be a necessity, not only for providing availability throughout short GPS outages, but also to provide a much needed robustness improvement to the processing of carrier-phase measurements. To achieve this level of improvement, the prototype system discussed in Chapter 4 featured tight integration at the carrier-phase level, in which inertial measurements were used to filter and monitor raw GPS carrier-phase measurements. This technique allowed cycle-slip monitoring at the channel-level, and provided a means for identifying and rejecting channels with excessive phase noise. Another powerful benefit from tight coupling was INS-aided integer-ambiguity resolution, in which the global integer-search process was replaced with a much more reliable local search for integers in the neighborhood of the INS attitude estimate. In addition to channel-level fault detection, baseline-level and multiple-baseline level integrity monitoring algorithms were also invoked to enforce the relationships between multiple channels for the same baseline and between multiple baselines fixed on a vehicle. The prototype system developed with all these improvements performed very favorably in terms of robustness, when compared to much more expensive commercial attitude systems with loose GPS/INS coupling.
- Despite the many enhancements achieved through tight coupling, no improvements could be made to the raw GPS carrier measurements in terms of phase-noise and frequent rate of cycle-slips. Situations where cycle-slips and phase-noise occur together are a major integrity threat even to a tightly-coupled system, as phase-noise can prevent successful recovery of integer ambiguities during multiple successive epochs; depending on the duration of such difficulty, the INS may accrue sufficient error during dead-reckoning to cause a system integrity failure. To reduce the likelihood of such scenarios, Chapters 6 and 7 explored the concept of inertial aiding of phase-lock loops, in which Doppler-frequency estimates derived from a GPS/INS navigation filter were employed to improve the noise-performance, carrier-tracking bandwidth, and reduce cycle-slip rate in harsh conditions. The benefits of this powerful technique were demonstrated through analysis, simulation

and post-processing of real data. Among these benefits were reduction of cycle slip probability in dynamic conditions, and the possibility of using open-loop tracking to avert cycle-slips through the short GPS outages (less than a second) that were shown to be common in the driving-test results shown in Chapter 5. Chapter 6 also presented a design technique for Doppler-aided PLLs, based on classical control techniques and modeling of contributing sources of phase-jitter. This procedure culminated in a concise result of how the use of different grades of GPS reference oscillators and IMUs affect system design and performance. From a standpoint of the automotive industry, the important conclusion from this study is the fact that Doppler aiding did exhibit significant performance improvements with inexpensive clocks (TCXO) and automotive-grade inertial sensors.

8.2 Future Research

To bring these areas of research to fruition in the form of a commercial product, a number of issues remain to be investigated. A recommended sequence of steps will be presented in this section.

The GPS/INS attitude system discussed in Chapters 3-5 could be modified and embellished in a number of ways to improve performance. It is expected that the use of shorter baselines would result in even more robust integer ambiguity recovery, and careful phase-delay calibration of such a system would minimize the penalty in accuracy. Furthermore, code-phase or carrier-phase GPS velocity measurements along with the IMU's accelerometers could be used to provide further observability and accuracy of vehicle pitch and roll. The use of velocity heading can also be utilized to provide yaw measurements for a car, as long as the vehicle is not turning or sideslip angle is known. Since this technique can be implemented with a single antenna, it would be a less complex and more inexpensive solution than the multiple-antenna system discussed in this thesis. The performance of a single-antenna system when compared to a multiple antenna system for attitude determination in a car has not been evaluated, and would make a strong complement to the research presented here.

As discussed in Subsection 7.2.2, comprehensive testing of Doppler-aided GPS tracking loops with real data was limited by the unavailability of hardware to collect usable GPS samples for a long periods of time (on the order of minutes). By decoding ten seconds of a compressed data file, the NordNav staff provided a temporary solution that sufficed for the scope of this thesis. However, the capability to obtain much longer sets of data for post-processing would be a necessary step for validating performance benefits in the presence of degraded SNR by multipath, and intermittent signal blockages. This step could be accomplished in a number of ways ranging from simple, though expensive solutions to more complicated but inexpensive

solutions. A simple and expensive alternative would involve acquisition of high-end data collection ADC boards that have large amounts of on-board memory (gigabytes), such that the data does not have to be streamed through a PCI bus to computer memory. A less expensive, though very much more labor-intensive solution, would be to construct a data collection system with an inexpensive combination of microprocessors and/or FPGAs to buffer samples from an RF front end (such as the two-bit samples in the GP2010 front end), and stream them to a hard drive without the use of a computer. Theoretically, the write speed of current hard drives is sufficient to accommodate the sampling resolution/rate combination of a typical GPS front end. This option was attempted as part of this research and was partially successful, but was abandoned in favor of a less time-consuming solution when it became available through cooperation with the Stanford GPS group.

Once such a setup is successfully assembled and tested, expansion to a dual-antenna version would be a logical step towards completing a testing platform for a true attitude system. By collecting and tracking data from at least two antennas, the performance of an attitude system with Doppler-aided carrier-tracking loops would not have to be extrapolated from the ICP measurements of a single antenna (as it was in Chapter 7), and the dual-antenna validation that was done with simulation in this thesis could be done with real data.

The development of hardware (or software) for real-time, Doppler-aiding of GPS tracking-loops would involve access to a GPS development system. With the increasing speed of modern computers, such a task could be accomplished entirely with a software receiver (similar to NordNav's) that would run on a computer with concurrent access to IMU outputs and running a real-time GPS/INS navigation filter. Since multiple GPS channels must be tracked concurrently, such a setup is likely to tax even a modern computer with 2-3GHz processor speed. If implemented in hardware, as an eventual commercial product would be, the RF front-end and correlator chips of standard chipsets could be used, and perhaps even the same dedicated microprocessors to interface with the correlator chips. The difference would be in the software that closes the PLL control loops, as it would use an additional input from external frequency aiding, and perhaps the option to switch between a standard PLL or a Doppler-aided PLL, as well as the other modifications shown in Figure 6.10.

The realization of this real-time system would represent a ground-breaking improvement in GPS receiver technology, especially if it could be accomplished with low-cost components. Based on the results of this thesis, such a system would have superior carrier-phase measurement noise performance and cycle-slip robustness when compared to existing GPS receivers. Assuming that prior post-processing tests concluded that this technology should be part of an attitude sensor for consumer vehicles, remaining work would involve incorporating it as a sensor for higher-level applications of vehicle control.

GPS modernization, due around the year 2010, is expected to have a momentous impact on accuracy and robustness of GPS navigation. Like most other GPS applications, attitude determination will benefit from the higher power-level of the civil signal and the availability of a second civil frequency. The availability of two carrier signals will introduce interesting possibilities in the realm of integer-ambiguity determination, as it does for dual-frequency wide-lane integer resolution in carrier-phase positioning [19].

With the current performance levels of automotive-grade sensors, no level of GPS/INS integration could provide accurate attitude or navigation outputs after only a few minutes without GPS. Such situations are expected to occur in very deep urban canyons, covered parking lots, and long stretches of road under tall trees and heavy foliage. To provide attitude availability in these cases, better inertial sensors or a tertiary set of sensors like magnetometers will have to be included. With the increasing performance and decreasing cost of MEMS technology, it is expected that MEMS-based inertial sensors for automotive applications will get better in the future, and only time will reveal the best sensor combination to use. Alternatively, current research involving very sophisticated vehicle-parameter modeling has the potential to act as a dead-reckoning system in the absence of GPS [10, 11]. This approach would involve the use of GPS and inertial sensors to calibrate the vehicle model, including moments of inertia, tire stiffness, suspension parameters, and road friction. If GPS is lost, these calibrated parameters along with odometer, wheel-speed and steering-angle inputs, constitute sufficient information to allow estimation of vehicle position, velocity, and attitude. This scheme is likely to have many of the same pitfalls as a stand-alone INS, such as growing error with time due to imperfections in the model and changing road conditions. Nevertheless, the vast possibilities of sensor-fusion and vehicle-modeling techniques are a potential area of research that should be explored to achieve the full performance capabilities of a practical attitude sensor for automobiles.

Appendix A

Coordinate Frames and Vector Notation

This thesis makes use of various coordinate frames that will be summarized in this Appendix. In addition, the notation that is used to express and manipulate vectors in these different frames is reviewed. Illustrations of the different frames are included in Figure A.1.

A.1 Reference Coordinate Frames

Earth-Centered, Earth-Fixed (ECEF):

The cartesian ECEF coordinate-frame is of particular importance to GPS, as the satellite and user-position vectors are computed in this basis before they are converted to more comprehensible navigation coordinates like latitude, longitude and altitude. The origin the ECEF frame is at the center of mass of the Earth, and its x-axis passes through the intersection of the Equator and the Mean Greenwich meridian. Its y-axis also intersects the Equator, and its z-axis passes through the conventional, terrestrial North pole [19]. Due to Earth's rotation and orbit around the Sun, the ECEF is not technically an inertial reference frame. However, for dynamic applications that move relatively slowly and near the Earth's surface, such an Earth-fixed frame can be used to approximate an inertial reference frame, as the gravitational field of the Earth dominates other accelerations induced by the rotation and orbit of the Earth. This approximation is valid for automobile navigation, and is used implicitly in the INS navigation equations in this thesis which ignore rotation rates and accelerations due to the Earth's motion.

Vectors specified in this basis are denoted with right subscript E .

Ellipsoidal Coordinates (lat,lon,alt):

The conventional angular measures of latitude and longitude are two of the coordinates in an ellipsoidal reference frame. These two angles define the position of a user on the Earth's surface, and constitute the rotations needed to transform vectors between the ECEF basis and the ENU basis. The third measurement in Ellipsoidal coordinates is altitude, and for GPS applications, it is referenced relative to the surface of a reference ellipsoid (WGS84) that approximates the shape of the Earth's surface. Since altitude is not utilized much in this thesis, interested readers can

find detailed information on the WGS84 ellipsoid and strict definitions of altitude measurements in [19].

East-North-UP (ENU):

The ENU frame is defined with two axes (East and North) defining a tangent plane to the reference ellipsoid. This frame is one of the more commonly used bases in this thesis, as it is used as the reference to define the attitude of a vehicle. It is also a useful reference frame for expressing user velocity, and in some cases, position relative to an initial point. Thereby, the origin of an ENU frame can be defined as a fixed, nearby point on the Earth's surface, or at the current location of the vehicle, in which case it is a locally-level frame. The latter case is the more traditional way of using an ENU frame to reference attitude measurements, though the former can also be used if a flat-Earth model is assumed in short-distance navigation applications.

Vectors specified in this basis are denoted with right subscript e .

Body frame:

The body frame is simply a coordinate system fixed to a vehicle. Traditionally, the body frame lines up with the roll and pitch axes of the vehicle, such that the Euler angles that define attitude can be expressed as successive rotations about the body axes. In this thesis, the body frame is defined such that it lines up with an ENU frame if the vehicle is level and pointed directly North. It is also convenient to line up strapdown IMUs with the body frame, as it simplifies transformations of gyro measurements to body rates.

Vectors specified in this basis are denoted with right subscript b .

Baseline frames:

Baseline-fixed frames are another type of body frame, but with the x-axis lined up with a baseline vector. These bases are used when computing the two baseline-orientation angles, which simplifies the process if the baseline does not happen to line up with the body axes of the vehicle. In this thesis, all baselines are orthogonal to the yaw-axis of the vehicle, such that a single yaw rotation transforms a vector from a baseline frame to the vehicle body frame.

Vectors specified in this basis are denoted with right subscript b followed by the double-letter baseline identification.

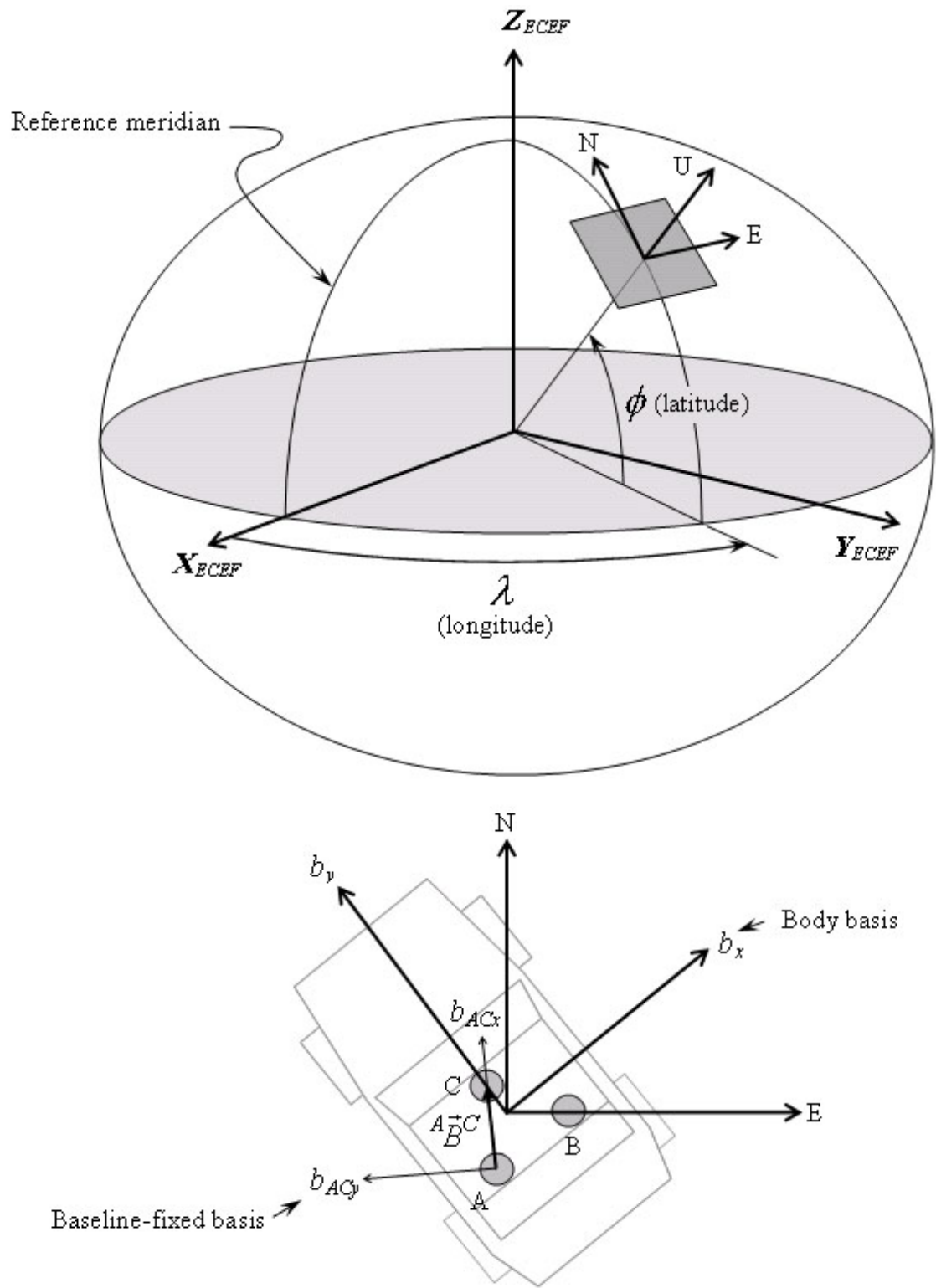


Figure A.1: Illustration of Various Coordinate Frames

A.2 Vector Notation

Physical quantities like distance, velocity, acceleration (among many others) represent real phenomena, and exist in the physical world whether or not they are represented as vectors in a mathematical world. As such, there need not be a coordinate frame (basis) specified to refer to these quantities and operations involving them, but they *can* be expressed numerically in any basis of choice. However, if numerical values are sought that involve these quantities, then a basis must be specified to give the numbers meaning. Transformations from one basis to another are done with rotation matrices, which are symbolized with the new basis as a left subscript and the old basis a right subscript. For example, the rotation matrix ${}^e\mathbf{C}_b$ transforms a vector in expressed in the body basis to the ENU basis.

If vector quantities are discussed without reference to a specific basis, they are denoted with an overhead arrow symbol. If the vector is specified in a particular basis, then an underscore is used instead of the overhead arrow, and a right subscript is used to indicate the basis. For example, ${}^A\vec{B}B$ is the vector representing baseline AB , and ${}^A\underline{B}_b^B$ is the baseline vector expressed in body basis.

The above example also serves to illustrate the notation for vectors with units of length. These vectors always connect two points in space, and the notation for the vector includes the starting point of the vector as a left superscript and the end-point of the vector as a right superscript.

The time derivative of a vector with units of length is a velocity vector. This derivative must be taken relative to a specific reference frame, as the velocity of an object depends on the reference frame from which it is observed. The same concept applies to acceleration vectors. A left superscript on a velocity or acceleration vector indicates the applicable basis, but the vector could still be expressed numerically in any basis. For example, ${}^e\vec{V}^A$ is the velocity of antenna A as observed in the ENU basis, and ${}^e\underline{V}_b^A$ would be this velocity expressed in the vehicle body basis.

Vector dot-products and cross-products also have notation that depends on whether quantities are expressed with a basis. Without a basis, the more traditional vector-operation notations are used. For example, the dot product of a satellite-velocity vector with the LOS vector is written as ${}^e\vec{V}^k \cdot \vec{I}^k$, but if these vectors are expressed in an ECEF basis the operation is ${}^e\underline{V}_E^{kT} \underline{I}_E^k$, where the T superscript represents the transpose operation. The cross-product between two baselines would be written as ${}^A\vec{B}B \times {}^A\vec{B}C$; this operation in the ENU basis would be ${}^A\underline{B}_e^{CX} \underline{B}_e^C$. The X superscript on the left term stands for the skew-symmetric cross-product operator, which is a 3×3 matrix of

the following form:

$$\underline{\underline{v}}^X = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \quad \text{where} \quad \underline{v} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Appendix B

Gauss-Markov Processes and Random Walks

B.1 Gauss-Markov Processes

A special kind of random process known as Gauss-Markov (GM) process can be generated by passing Gaussian white noise through a linear low-pass filter. The output of the filter also has Gaussian characteristics, but will be exponentially correlated in time. The order of the filter determines the order of the GM process; that is, a first-order filter (one with a single pole) results in a first-order GM process. GM processes can be used to model many stochastic physical phenomena with correlated noise. Some that are that are relevant to this thesis include the bias behavior of inertial sensors, frequency instability of oscillators, GPS attitude errors due to uncalibrated phase-delay, and GPS/INS navigation-filter velocity errors. Since a first-order GM process is used exclusively in this thesis, it will be the focus of the following discussion. Further information on GM processes, including those of higher order, can be found in [35] and [59].

A first-order GM process (x) is described by a first-order differential equation:

$$\dot{x} + \beta x = w \quad (\text{B.1})$$

where $w(t)$ is the input white noise. In the frequency domain, the input-output relationship is given by the following transfer function:

$$\frac{X(s)}{W(s)} = \frac{1}{s + \beta} \quad (\text{B.2})$$

As shown, the transfer function is that of a first-order low-pass filter. To illustrate the effect of passing noise through this filter, Figure B.1 shows an image the driving noise (with unity variance) and resulting GM sequence (with $\beta=10$). One of the most distinct characteristics of the GM process is its time-correlation. The autocorrelation of the process is:

$$\phi_{xx}(\tau) = \sigma_x^2 e^{-\beta\tau} \quad (\tau = \text{time offset}) \quad (\text{B.3})$$

where

$$\sigma_x^2 = \frac{1}{2\beta} \sigma_w^2 \quad (\text{B.4})$$

Another of the more useful quantities of a GM process is its PSD, which is the Fourier transform of the autocorrelation:

$$S_x(\omega) = \frac{2\beta\sigma_x^2}{\omega^2 + \beta^2} \quad (\text{B.5})$$

In many practical applications, including those of this thesis, it is useful to have a discrete-domain equivalent of the GM process model. The difference equation derived from Eq. B.1 is:

$$x_{i+1} = e^{-\beta \Delta t} x_i + w_i \quad (\text{B.6})$$

A relationship that relates the variances of the continuous and discrete driving noises can be obtained by enforcing that the variance of the continuous GM process is equal to that of the discrete GM process. The result of this exercise is:

$$\sigma_{w_i}^2 = \frac{\sigma_w^2}{2\beta} \left(1 - e^{-2\beta \Delta t} \right) \quad (\text{B.7})$$

B.2 Random Walks

A random walk occurs as a result of integrating white noise. In relation to GM processes, a random walk can be interpreted as the limiting case as $\beta \rightarrow 0$. The differential equation and the difference equation of a random walk are consistent with this analogy:

$$\dot{x} = w \quad (\text{B.8})$$

$$x_{i+1} = x_i + w_i \quad (\text{B.9})$$

Likewise, the relationship between the continuous and discrete driving noise can be obtained from Eq. B.7 by taking the limit as $\beta \rightarrow 0$:

$$\sigma_{w_i}^2 = \Delta t \sigma_w^2 \quad (\text{B.10})$$

Like a GM process, the expected value of a random walk depends on its initial condition. However, unlike a GM process, a random walk is not a stationary process, as its variance grows linearly with time. That is, the longer the integration time of the process, the larger the value of its variance, as illustrated in Figure B.1. The random walk depicted in the bottom chart uses the same driving noise as the GM process; the variance of the GM process reaches steady-state, but that of the random walk clearly grows with time.

The autocorrelation of a random walk is not defined only as a function of τ (time offset), but some authors ([35]) define it as a function of time and time-offset. The PSD of a random walk is typically defined as being proportional to $1/\omega^2$ but with unspecified magnitude (as it is unbounded at DC). However, in cases where the random walk resembles a GM process at high frequencies, the magnitude of its PSD can be matched to that of the GM PSD at high frequencies. If the driving noise is scaled as in Eqs. B.7 and B.10 for the same continuous driving noise, the PSDs at high frequencies would match exactly. This similarity is evident (in the time domain) upon inspection of the two plots in the bottom graph of Figure B.1, which exhibit nearly identical high-frequency content.

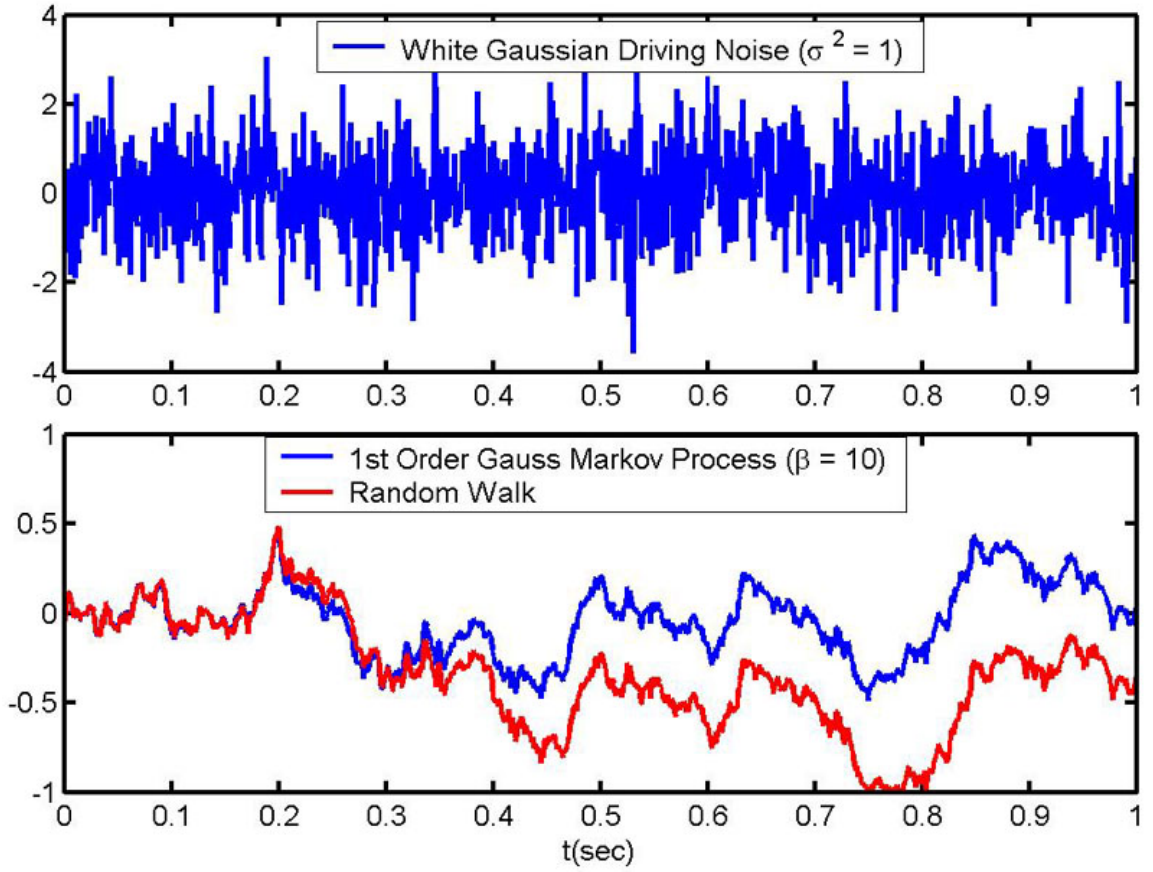


Figure B.1: Images of White Gaussian Driving Noise and Resulting Gauss-Markov Process and Random Walk

Appendix C

Inertial-Sensor Error Modeling

Proper calibration of the inertial sensors in an INS is critical to navigation performance. In Section 4.1.2 of this thesis, the gyro errors in the primary GPS/INS filter were modeled as constant biases, but an offline calibration process was necessary to approximate alignment errors and the yaw-gyro scale factor. This appendix gives a more general treatment of inertial-sensor errors, and covers the steps that are necessary to simplify the general error-model of a gyro to a single bias term. The reasoning behind this simplification and its implications are also examined. The error models of a gyro and of an accelerometer are nearly identical, and only that of a gyro will be presented here for conciseness. The only notable difference (according to [22]) between the models of the two types of inertial sensors is that accelerometers may contain a nonlinearity term instead of the gyro's acceleration sensitivity. Note, this similarity is limited to the raw sensor measurements alone, as the propagation of uncompensated sensor errors into the inertial navigation solutions is very different.

The first error source to be considered is sensor misalignment. This type of error refers to the physical misalignment of the sensor's axis of sensitivity from where it is intended to be. To simplify the discussion, it will be assumed that the gyros are supposed to be aligned with the vehicle's body axes, as they are in the attitude system prototype discussed in Chapters 3-5. Using the subscript g to denote a basis aligned with they gyros' axis of sensitivity, the rotation of the angular velocity vector from the gyro basis to the body basis is expressed as:

$${}^e\omega_b^b = {}_{b=g}C_g {}^e\omega_g^b \quad (C.1)$$

Note, the notation for the angular velocity vector represents the angular velocity of the body frame relative to the ENU frame. To allow for each gyro misalignment to be independent of the other two, the matrix ${}_{b=g}C_g$ is allowed to be a nonorthogonal transformation, and a small rotation-angle approximation is used:

$${}_{b=g}C_g \approx I_{3 \times 3} + \delta {}_{b=g}C_g \quad (C.2)$$

where

$$\delta_{b \equiv g} \underline{\underline{C}} = \begin{bmatrix} 0 & g_{pr} & -g_{pq} \\ -g_{qr} & 0 & g_{qp} \\ g_{rq} & -g_{rp} & 0 \end{bmatrix} \quad (\text{C.3})$$

The terms in the $\delta_{b \equiv g} \underline{\underline{C}}$ matrix represent small rotation angles (in radians) due to misalignment, and the notation is consistent with that used in [22]. For example, the term g_{pr} represents a positive rotation angle about the body r (yaw) axis from the body $p-r$ plane to the gyro p axis. This particular misalignment causes pitch rotation to be measured as a disturbance in the roll gyro. Figure C.1 shows an illustration of these rotations.

In addition physical misalignment errors, the gyro sensors themselves have imperfect measurements that require calibration. Thereby, the notation will be ammended to reflect the fact that the angular velocity is an imperfect estimate if it is based on gyro measurements:

$${}^e\hat{\underline{\omega}}_b^b = \left(\underline{\underline{I}}_{3 \times 3} + \delta_{b \equiv g} \hat{\underline{\underline{C}}} \right) {}^e\hat{\underline{\omega}}_g^b \quad (\text{C.4})$$

The sources of error in the gyro measurements include using inexact scale factors and gyro biases, sensitivity to linear acceleration, nonlinearity, quantization error, and wideband noise. The gyro scale-factor is a quantity specified by the manufacturer. This constant specifies the conversion factor from an analog electrical signal to an angular rate for a particular gyro (such as the 18mV/ $^\circ$ /s spec in Table 3.2). Deviations from the nominal value exist from sensor to sensor, and may change slightly with temperature. The sensor bias refers to an offset in the gyro output from the true angular rate, and is usually the most important error to correct. This bias is not necessarily a constant, as it also varies with temperature. Furthermore, gyros of consumer or automotive quality exhibit distinct exponentially-correlated noise in the bias, which can be modeled as a first-order GM process [32]. Sensitivities to acceleration, nonlinearity and quantization effects are more difficult to model than other quantities, but can be included as additive terms that contribute to the variance of the wideband noise. Expressing these quantities symbolically, the gyro measurements can be written as:

$${}^e\hat{\underline{\omega}}_g^b = \left(\underline{\underline{I}}_{3 \times 3} + \delta \underline{\underline{K}}_{sf} \right) (\underline{u}_g + \delta \underline{b}_g + \underline{v}_g) \quad (\text{C.5})$$

where

$$\delta \underline{\underline{K}}_{sf} = \begin{bmatrix} \delta k_{sfp} & 0 & 0 \\ 0 & \delta k_{sfq} & 0 \\ 0 & 0 & \delta k_{sfr} \end{bmatrix} \quad \text{Scale factor errors}$$

$$\underline{u}_g = \text{Vector of gyro measurements}$$

$$\begin{aligned}
\delta \underline{b}_g &= \text{Vector of gyro biases} \\
\underline{v}_g &= \text{Vector of uncorrelated, wideband gyro noise}
\end{aligned} \tag{C.6}$$

Note, \underline{v}_g is an aggregate of all other terms that are not typically modeled, including including acceleration sensitivity, nonlinearity, and quantization effects. If estimated scale factors and biases are used in this equation and it is substituted in to Eq C.4, an estimate of the vehicle angular velocity is computed from the gyro measurements as follows:

$${}^e\hat{\underline{\omega}}_b^b = \left(\underline{I}_{3 \times 3} + \delta \underline{\hat{C}}_{b=g} \right) \left(\underline{I}_{3 \times 3} + \delta \underline{\hat{K}}_{sf} \right) (\underline{u}_g + \delta \underline{\hat{b}}_g) \tag{C.7}$$

Theoretically, all the gyro-error model terms could be included in an estimator to maximize the accuracy of the dynamic model. However, the observability of misalignment and scale-factor errors is dependent on the dynamics of the vehicle. For example, to be able to estimate the yaw gyro scale-factor, there would have to be a distinct nonzero turning-rate such that the scale-factor error would result in a measurable residual (filter innovation). If the gyro does not experience relatively large rotation rates, then the gyro scale-factor is not well observed, nor does it incur large errors into the navigation solution.

As might be expected, the dynamics needed to make misalignment and scale factor errors observable are not always available. In such cases, it is advantageous to perform off-line calibration of these parameters and they can be hard-coded in the software without the need to include them as states in an estimator. This technique is more ideally suited for misalignment errors, since they are normally constant in a fixed IMU installation. As mentioned previously, the scale factor may not necessarily be a constant due to temperature variations, but the offline calibration can be done at a steady-state temperature consistent with the expected operating conditions of the vehicle.

In applications where not all the gyros experience large turning rates, some simplifications can be made. Such is the case in automobile applications, where the pitch and roll rates are small in comparison to yaw rate (during turns). If the pitch and roll gyro outputs are expected to be small, then the presumably small misalignment errors that incur cross-axis sensitivity due to pitch and roll motion will have a small effect. The same reasoning applies to the roll and pitch scale factors, which will have a small effect if the roll and pitch rates are small. Thereby, a reasonable simplification for an automobile attitude system would be to concentrate the calibration effort on the terms that incur error through the yaw rate; that is, g_{pq} , g_{qp} , and δk_{sfr} . Since other terms

have little effect in comparison to these, the matrices in Eq C.7 can be simplified:

$$\delta_{b \equiv g} \hat{\underline{C}} \approx \begin{bmatrix} 0 & 0 & -\hat{g}_{pq} \\ 0 & 0 & \hat{g}_{qp} \\ 0 & 0 & 0 \end{bmatrix} \quad (C.8)$$

$$\delta \hat{\underline{K}}_{sf} \approx \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \delta \hat{k}_{sfr} \end{bmatrix} \quad (C.9)$$

If these alignment and scale-factor errors can be estimated off-line, then the only error parameters left to estimate from Eq C.7 are the gyro biases. These terms are additive to the gyro inputs, and can thus be estimated with a linear filter as explained in Section 4.1.2. Note, by only including the sensor bias in the estimator, any errors in the off-line calibration will be absorbed into the estimation process of the bias, and appropriate process noise or gain adjustment should be used to compensate for this effect.

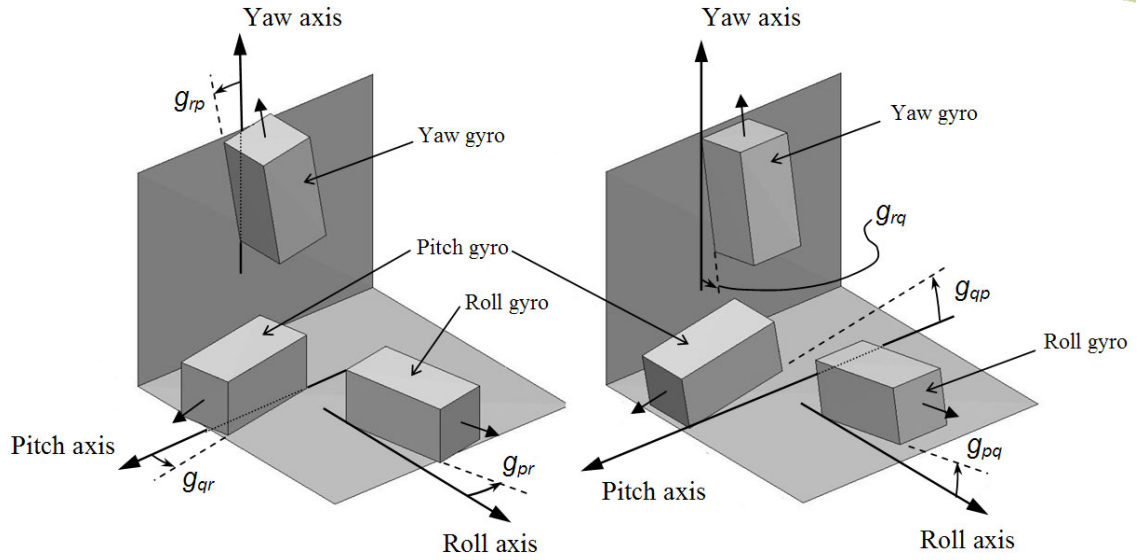


Figure C.1: Illustration of Gyro Misalignments Relative to Platform

Appendix D

Development of a PLL Estimator for Ultra-Tight Coupling

This appendix presents a development of the state-space implementation of a PLL estimator and controller, which comprises the low-level design of the more commonly accepted mechanizations of ultra-tight GPS/INS integration. These architectures utilize a large global estimator or a less optimal bank of smaller estimators (a federated approach with one estimator per channel) to blend inertial measurements with the GPS tracking loops. These techniques offer the additional advantage of having filtered estimates of the I and Q samples, which results in less noisy carrier-phase offset measurements (and ICP measurements). In fact, the use of estimated I and Q quantities, as opposed to the original measurements, to compute the carrier-phase offset is the primary difference between the controls approach described in Section 6.3 and the more complicated estimation approach. The scope of this discussion will cover the implementation of an estimator for a single channel, with the external frequency aiding applied as a measurement and the frequency rate as an input. Of course, these quantities are actually formed from the outputs of another estimator (the navigation filter), so the estimator described here can be interpreted as a component of a federated filtering architecture. Alternatively, the dynamic equations used in this single-channel estimator can also be used to assemble a global filter that encompasses all of the navigation, inertial sensor, and GPS signal-tracking states. In the following discussion, it may be helpful to the reader to refer to Figure D.2 as the characteristics of the dynamic equations are described.

As described in Subsection 6.1.1, the I and Q quantities, which are outputs from a correlator chip, are measurements of the cosine and sine of the phase-offset measurement:

$$I = \cos(\delta\varphi) \quad (\text{D.1a})$$

$$Q = \sin(\delta\varphi) \quad (\text{D.1b})$$

Taking the time derivative of these relationships yields a pair of dynamic equations that can be

used in the propagation step of an estimator:

$$\dot{I} = -Q \delta\dot{\phi} \quad (\text{D.2a})$$

$$\dot{Q} = I \delta\dot{\phi} \quad (\text{D.2b})$$

The phase-offset rate in these equations is equivalent to the error in frequency of the replicated carrier, and the error in frequency is the difference between the true frequency of the reference carrier and the frequency input into the NCO:

$$\delta\dot{\phi} = \omega_r - \omega_{PLL} \quad (\text{D.3})$$

where

$$\omega_r = \dot{\phi}_r \quad (\text{D.4})$$

$$\omega_{PLL} = 2\pi f_{PLL} \quad (\text{D.5})$$

The loop filter discussed in Subsection 6.3.1 can be implemented precisely in state-space form with the inclusion of a state that represents the integrated phase-offset measurement. For an arctangent phase discriminator, the dynamic equation for this state, designated z_{PLL} , is:

$$\dot{z}_{PLL} = \arctan\left(\frac{Q}{I}\right) \quad (\text{D.6})$$

The loop is closed by assigning ω_{PLL} as a function of the phase offset, z_{PLL} , and the reference frequency:

$$\omega_{PLL} = \omega_r - k_2 z_{PLL} - \sqrt{2k_2} \arctan(Q/I) \quad (\text{D.7})$$

As written, this equation represents the feedback of a Doppler-aided PLL implemented in state-space form, where the feedback signal is a function of direct measurements, not estimates. If an estimator is used, then ω_r , I , Q , and z_{PLL} would be state estimates rather than direct measurements. This variation is shown in Figure D.2, which shows the feedback from the controller to the plant as a function of estimated elements.

Substituting Eqs. D.7 and D.3 into Eq. D.2a yields the closed-loop dynamic equations for the I and Q states:

$$\dot{I} = -Q \left(k_2 z_{PLL} + \sqrt{2k_2} \arctan(Q/I) \right) \quad (\text{D.8})$$

$$\dot{Q} = I \left(k_2 z_{PLL} + \sqrt{2k_2} \arctan(Q/I) \right) \quad (\text{D.9})$$

The frequency of the reference carrier is also a state to be estimated, as it is not known precisely. The dynamic equation for this state is simply the sum of the input frequency rates:

$$\dot{\omega}_r \approx 2\pi (\dot{f}_{dopp} + \dot{f}_{clk}) \quad (\text{D.10})$$

The rate of change of the Doppler frequency can be obtained by taking the time derivative of Eq. 6.11. If it is assumed that the LOS vector changes very slowly relative to the satellite and user velocities, this operation yields the following approximation for SV# k :

$$\dot{f}_{dopp}^k \approx \frac{1}{\lambda_{L1}} \left(\frac{d}{dt} \vec{a}^{RX} - \vec{a}^k \right) \cdot \vec{1}^k \quad (\text{D.11})$$

In this equation, the acceleration of the receiver antenna relative to an earth-fixed frame (\vec{a}^{RX}) is a function of the INS outputs (accelerations, body rates, and vehicle attitude), and the acceleration of the SV (\vec{a}^k) can be obtained as the time derivative of its velocity.

The rate of change of the clock error-frequency in Eq. D.10 must be obtained from an estimated model of the behavior of f_{clk} . As shown in Figure 6.7 the behavior of the clock frequency-error is much more easily predicted from past data for an OCXO than for a TCXO, though it is possible to do so for a TCXO as long as measurements of f_{clk} can be taken with sufficient frequency.

If using a federated filtering architecture where the navigation filter is separate from the GPS-tracking filters, it is beneficial to include a model for exponentially correlated noise in the external frequency measurements. This type of error may be present in the estimates of both f_{dopp} and f_{clk} , especially if using low-cost components like automotive grade inertial sensors and a TCXO. The state that represents this error will be designated as n_{fext} , and its dynamic model is that of a first-order Markov process:

$$\dot{n}_{fext} = \frac{-1}{\tau_{fext}} n_{fext} + w_{fext} \quad (\text{D.12})$$

In this filter implementation, it is assumed that the parameters of the Markov process, τ_{fext} and w_{fext} , can be determined empirically by analyzing the velocity outputs of the navigation filter and f_{clk} measurements similar to those shown in Figure 6.7.

The measurements of the filter include the I and Q correlator outputs and the external frequency estimates. Note, the external frequency measurements to the filter are derived from the navigation filter velocity estimates, and will thus use information from the other GPS channels indirectly.

In summary, the state vector of the single-channel PLL estimator is:

$$\underline{x}_{PLL} = \begin{bmatrix} I & Q & z_{PLL} & \omega_r & n_{fext} \end{bmatrix}^T \quad (\text{D.13})$$

The nonlinear dynamic equations that describe the frequency-aided PLL are Eqs. D.6, D.8, D.9, D.10, and D.12. Additive process noise should also be included to account for inexact dynamic modeling and input errors. This term is especially important for Eq. D.10, because noisy inputs are part of this equation, and other factors that are not modeled (such as the SV clock noise) also contribute to the reference frequency. The input is the time rate of change of f_{dopp} and f_{clk} , and the measurement vector is:

$$\underline{y}_{PLL} = \underline{H}_{PLL} \underline{x}_{PLL} \quad (D.14)$$

where

$$\underline{H}_{PLL} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \quad (D.15)$$

Since the dynamic equations of this system are nonlinear, implementation of an optimal estimator is done with an extended Kalman filter. The propagation of the state-covariance and computation of the gain matrix is similar the equivalent operations in a linear filter, but uses linearized equations about the current state estimate. The details of this well-known procedure are beyond the scope of this discussion, but can be found in [35] and [59].

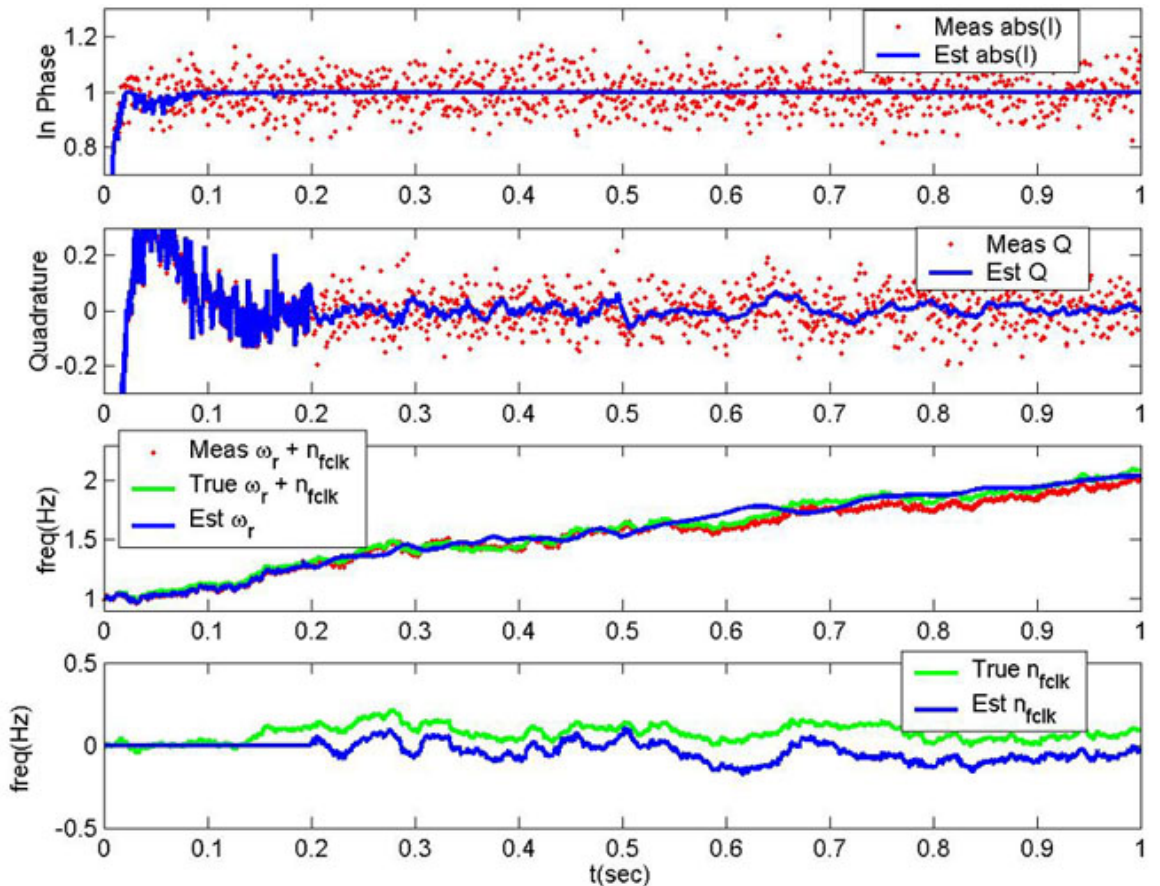


Figure D.1: Performance of a PLL Implemented with an Estimator

The performance of this PLL architecture is demonstrated in Figure D.1, which shows the various estimated state elements and measurements. As shown, the I and Q estimates are less noisy when estimated, and would result in less noisy ICP measurements. In this test case, the estimator is enabled at $t = 0.2$ sec, at which time the external frequency estimates are used; prior to this transition, the PLL is in an initial convergence phase and is configured as a traditional 15Hz PLL. At $t = 0.5$ sec, the bandwidth of the closed-loop branch is reduced from 15Hz to about 6Hz, but little change is noticeable in the state estimates. This example illustrates the impact of the use of an estimator along with a controller, as the poles of the estimator (as decided by the estimator gains) also have an impact on the behavior of the system. Thereby, modifying the controller gains alone will not alter the PLL dynamics in the same way as was described in Subsection 6.3.1, and the design guidelines would be more complicated.

In the context of the GPS/INS attitude system described in this thesis, the use of an estimator at the PLL level may provide more accurate ICP measurements, but it would be more complicated and offer virtually no advantage over the simpler frequency-aided design in terms of cycle-slip reduction and phase. Since this type of robustness improvement was the original motivation for pursuing ultra-tight coupling for an automobile attitude system, the simpler design described in the Section 6.3 was favored and is applicable to all analysis in Chapters 6 and 7.

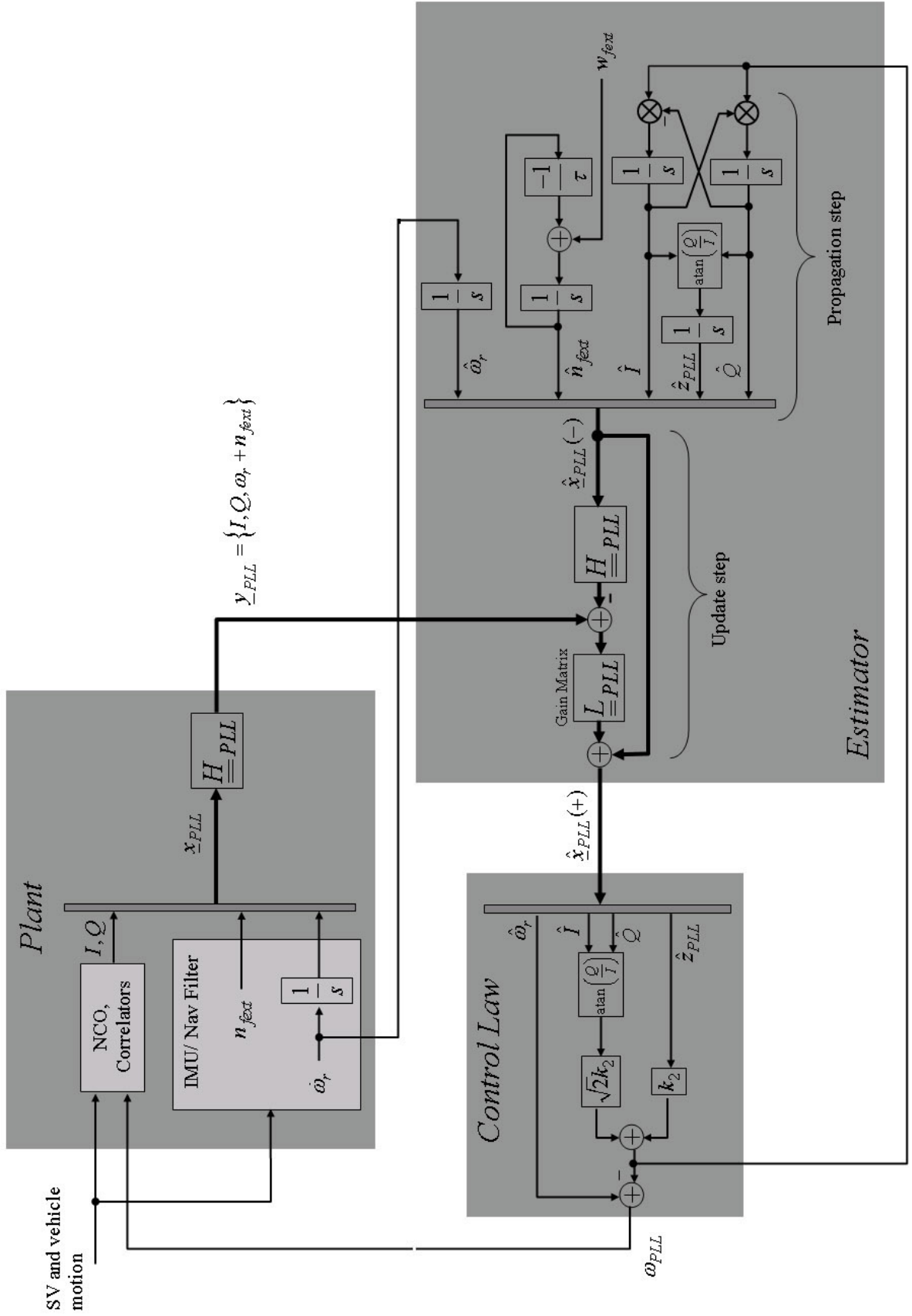


Figure D.2: Schematic of a Single-Channel PLL Estimator

Appendix E

Accronyms and Symbols Guide

E.1 List of Acronyms

ADC: Analog-to-Digital Converter

ADOP: Attitude Dilution of Precision

API: Application Programmers Interface

A&D: Accumulate and Dump

BPSK: Binary Phase-Shift Keying

BW: Bandwidth

bps: Bits per second

CA (or C/A): Coarse/Acquisition

CG: Code Generator

CP: Carrier Phase

DLL: Delay-Locked Loop

DC: Direct Current. Term used when referring to spectral content at zero frequency.

DGPS: Differential GPS

dICP: Single Difference of Integrated Carrier Phase (differential carrier phase)

DME: Distance Measuring Equipment

DOP: Dilution of precision

ENU: East-North-Up Coordinate Frame

ECEF: Earth-Centered, Earth-Fixed Coordinate Frame

FLL: Frequency-Locked Loop

GM: Gauss Markov

GPS: Global Positioning System

GUI: Graphical User Interface

ICP: Integrated Carrier Phase

IF: Intermediate Frequency

IMU: Inertial Measurement Unit

INS: Inertial Navigation System

ILS: Instrument Landing System
 LAAS: Local-Area Augmentation System
 LOS: Line of Sight
 NCO: Numerically Controlled Oscillator
 OCXO: Oven Compensated Crystal Oscillator
 OS: Operating System
 PLL: Phase-Lock Loop
 PPS: Pulse per second
 PR: Pseudorange
 PRN: Pseudo Random Noise
 PPW: Prompt power
 PSD: Power Spectral Density
 PVT: Position, Velocity, Time
 PVA: Position, Velocity, Attitude
 RF: Radio Frequency
 RX: Receiver
 SSB: Single-Side Band
 SV: Space Vehicle. Common abbreviation used when referring to a GPS satellite.
 SW: Software
 TCXO: Temperature Compensated Crystal Oscillator
 UTC: Universal Time Coordinated
 WAAS: Wide-Area Augmentation System

E.2 List of Symbols

Scalar quantities related to carrier-phase measurements: Unless otherwise noted, all the following symbols will have superscript k to denote measurements from SV# k and letter subscripts (A, B, C) to refer to a particular antenna. Double-letter subscripts (such as AB) identify differential quantities between antennas.

φ : Integrated carrier-phase (ICP)
 φ_f : Fractional component of ICP
 $\Delta\varphi$: Differential carrier-phase (dICP)
 m : Integer component of ICP
 r : Range from satellite to antenna

- T_φ : Troposphere delay of carrier
- I_φ : Ionosphere delay of carrier
- n : Integer ambiguity
- δt : Clock error (from SV or local clock, depending on superscript or subscript)
- l : Line bias (no k superscript)
- δl : Line-bias correction term (no k superscript)
- ε_φ : Phase noise (includes phase-delay and broadband noise)
- η_φ : Phase-delay component of ε_φ
- v_φ : Broadband noise component of ε_φ

Vector quantities related to carrier-phase measurements: Vectors compiled from some of the differential scalar quantities listed above are denoted with underscores and double-letter subscripts to identify a particular baseline. Each vector row corresponds to one channel (one SV).

- $\underline{\Delta\Phi}$: Vector of dICPs
- \underline{N} : Vector of integer ambiguities
- \underline{L} : Vector of line-bias (all rows equal)
- $\underline{\delta L}$: Vector of line-bias correction terms
- \underline{E}_φ : Vector of phase-noise terms (ε_φ)

Vectors of physical quantities: 3×1 vectors that represent physical quantities (like 3D coordinates, velocities, accelerations) are identified with an overhead arrow if they are not specified in any particular basis. If represented in a particular basis (like ENU, ECEF), these vectors are identified with an underscore and a right subscript to indicate the basis used.

- $\vec{1}^k$: LOS unit vector to SV# k
- $\underline{1}_e^k$: LOS unit vector specified in ENU basis
- $C_{x,y,z}$: Vector components of $\underline{1}_e^k$
- ${}^A\vec{B}^B$: Baseline vector between antennas A and B (left/right superscripts may vary depending on baseline)
- ${}^A\underline{B}_e^B$: Baseline vector between antennas A and B , specified in ENU basis (left/right superscripts may vary depending on baseline, subscript may vary depending on basis)
- ${}^0\vec{R}^k$: Vector from origin of ECEF frame to SV# k
- \vec{V}^k : Velocity vector of SV# k with respect to an Earth-fixed frame
- \vec{V}^{RX} : Velocity vector of a receiver with respect to an Earth-fixed frame
- \vec{a}^k : Acceleration vector of SV# k with respect to an Earth-fixed frame
- \vec{a}^{RX} : Acceleration vector of a receiver with respect to an Earth-fixed frame

Related Matrices:

$\underline{\underline{H}}_e$: Observation matrix of LOS unit vectors (1 vector per row). Subscript always indicates basis.

$\underline{\underline{H}}_{1e}$: Observation matrix of LOS vectors (1 vector per row) in ENU basis with fourth column of ones for resolving line-bias correction term. Subscript always indicates basis.

Vectors and Matrices for Filters or Estimators: Subscript *PLL* on any of these symbols pertains specifically to implementation of PLL with an estimator.

\underline{u} : Input vector

\underline{x} : State vector

\underline{y} : Measurement vector

$\underline{\underline{A}}$: Continuous domain dynamic matrix

$\underline{\underline{B}}$: Continuous domain input-to-state mapping matrix

$\underline{\underline{H}}$: State-to-measurement mapping matrix

$\underline{\underline{\Phi}}$: Discrete domain dynamic matrix

$\underline{\underline{\Gamma}}$: Discrete domain input-to-state mapping matrix

$\underline{\underline{K}}$: Gain matrix

Angular and angular rate quantities:

ϕ : Vehicle Roll

θ : Vehicle Pitch

ψ : Vehicle Yaw

p : roll body rate

q : pitch body rate

r : yaw body rate

${}^e\vec{\omega}^b$: Angular velocity of reference frame *b* relative to reference frame *e*

δp : roll gyro bias

δq : pitch gyro bias

δr : yaw gyro bias

ϕ_{AB} : Baseline Roll (subscript denotes baseline)

ψ_{AB} : Baseline Yaw (subscript denotes baseline)

ϕ_{AB}^k : One-satellite baseline roll (subscript denotes baseline)

ψ_{AB}^k : One-satellite baseline yaw (subscript denotes baseline)

β_{AB}^k : Angle between baseline AB and LOS vector to SV#*k*

El^k : Elevation of SV#*k*

Az^k : Azimuth of SV#*k*

Other symbols related to GPS attitude determination:

- R_{AB} : Baseline length (subscript denotes baseline)
 δB_{AB} : Baseline length error for linear attitude solution (subscript denotes baseline)
 e_{AB} : Residual magnitude error for linear attitude solution (subscript denotes baseline)
 J_N : Cost function for integer search with linear attitude solution
 J_{AB} : Cost function for nonlinear attitude solution (subscript denotes baseline)
 $r_{\phi_{AB}}$: Ratio of external roll-estimate error variance to dICP-error variance
 $r_{\psi_{AB}}$: Ratio of external yaw-estimate error variance to dICP-error variance
 $W_{\psi_{AB}}^k$: Weighing function for one-satellite yaw (for SV# k and baseline AB)
 $W_{\phi_{AB}}^k$: Weighing function for one-satellite roll (for SV# k and baseline AB)
 $\underline{\underline{F}}$: ADOP matrix
 $\underline{\underline{G}}$: Matrix quantity used in the definition of ADOP matrix
 $\underline{\delta A}$: 3×1 vector of attitude perturbations

Terms related to Phase-Lock Loops:

- $B_{n\varphi}$: Noise equivalent bandwidth of a PLL
 $H_1(s)$: Closed-loop transfer function of traditional PLL
 $H_2(s)$: Closed-loop transfer function that relates phase-noise input to phase output in Doppler-aided PLL
 $H_3(s)$: Closed-loop transfer function that relates reference-phase input to phase output in Doppler-aided PLL
 f_{PLL} : Frequency tracked by a PLL
 φ_{PLL} : Phase tracked by PLL
 φ_r : Reference phase input for PLL model
 w_φ : Phase noise input for PLL model (from thermal noise and interference)
 $\delta\varphi$: Phase offset measurement (output of phase discriminator)
 f_{dopp} : Doppler component of f_{PLL} (may have superscript to indicate specific channels)
 f_{clk} : Clock error-frequency of reference oscillator
 δf_{dopp} : Error in external Doppler-frequency estimate
 δf_{clk} : Error in external clock error-frequency estimate
 δf_{PLL}^k : Frequency residual for SV# k
 $\delta\varphi_{ext}$: Reference phase used for model of Doppler-aided PLL

- $\delta\varphi_{dopp}$: Component of $\delta\varphi_{ext}$ induced by external Doppler-estimate errors
- $\delta\varphi_{clk}$: Component of $\delta\varphi_{ext}$ induced by external clock error-frequency estimates
- $\delta\varphi_{vib}$: Component of $\delta\varphi_{ext}$ induced by vibration of reference oscillator
- $k_{1,2}$: Loop filter gain of PLL (subscripts denote different gains for different bandwidths)
- k_{a-d} : Constants used to model oscillator phase-noise
- $\tau_{1,2}$: Inverse of loop-filter zero (subscripts denote different zeros for different bandwidths)
- α_{IMU} : Value of pole used to model finite-bandwidth inertial measurements
- δf_{ext} : Error input to Doppler-aided PLL model, to account for imperfect external frequency estimates
- Δf_{PLL} : Closed-loop frequency corrections of a Doppler-aided PLL
- Q : Quadrature component of phase-offset measurement
- I : In-phase component of phase-offset measurement
- T_{av} : Averaging time in GPS signal correlators (equivalent to Δt_{PLL})
- ε_{φ} : Tracking error of PLL
- $\varepsilon_{\varphi_{ext}}$: Tracking error of PLL induced by $\delta\varphi_{ext}$
- ε_{φ_w} : Tracking error of PLL induced by w_{φ}
- \bar{T}_{slip} : Mean time to cycle slip for a PLL
- γ : Steady-state error of a PLL due to dynamic stress (subscripts *dyn* or *fdopp* denote source of dynamic stress)
- n_{fext} : Symbol for Markov-process model of external frequency estimate errors in PLL estimator
- z_{PLL} : Symbol used for integrated phase-offset in PLL estimator

Terms related to Delay-Lock Loops:

- τ_r : Reference code-phase in DLL model
- τ_{DLL} : Code-phase tracked by DLL
- $\delta\tau$: Code-phase offset measurement
- k_{CDLL} : Gain to convert f_{PLL} to code rate
- k_{1DLL} : Gain of DLL with no rate aiding
- k_{2DLL} : Gain of DLL with rate aiding

General symbols:

c	: Speed of light
λ_{L1}	: Wavelength of L1 carrier
f_{L1}	: Frequency of L1 carrier
Δt	: Sampling period (subscripts GPS, INS, or PLL)
σ	: Standard deviation
σ^2	: Variance (of variable specified in subscript)
β	: Time constant of a Gauss-Markov Process
$\phi(N)$: Discrete autocorrelation, N indicates offset samples
$S(f)$: Power spectral density (of variable specified in subscript)
$L(f)$: SSB Phase noise in dBc/Hz
S/N_0	: Signal to noise ratio
$I_n(x)$: Modified Bessel function of order n
q	: Quaternion element
\underline{q}	: Quaternion vector
$\underline{C}_{a \leftarrow b}$: Rotation matrix from basis b to basis a

Notation Variants: On occasion, some variables contain additional symbology to emphasize a particular variant of the quantity. Such variants include an estimate, a guess, a mean value, or a combination of these. An “estimate” is different from a “guess”, in that estimates are typically equivalent to a measurement; guesses normally refer to a iteration variable in recursive processes like integer-ambiguity searches or line-bias determination, and may be entirely random.

Symbol	Meaning	Example	Meaning of example
$\hat{}$	Estimate	$\hat{\underline{B}}_e^{AB}$	Estimate of baseline AB in ENU basis
\sim	Guess	$\tilde{\underline{B}}_e^{AB}$	Guess of baseline AB in ENU basis
$\bar{}$	Mean	$\bar{\underline{l}}_{AB}$	Average of line-bias guesses over all channels
\dagger	Pseudoinverse (Matrices only)	$\underline{\underline{H}}_e^\dagger$	$\left(\underline{\underline{H}}_e^T \underline{\underline{H}}_e\right)^{-1} \underline{\underline{H}}_e^T$

Bibliography

- [1] Abbot, Eric and J.D. Powell, "Land Vehicle Navigation Using GPS", *Proceedings of the IEEE*, Vol. 87, No. 1, 1999. pp. 145-162.
- [2] Alban, S., "An Inexpensive and Robust GPS/INS Attitude System for Automobiles", Proceedings of the ION GPS-02, Portland, OR, September 2002.
- [3] Alban, S. and D. Gebre-Egziabher, "Performance Analysis and Architectures for INS-Aided GPS Tracking Loops", Proceedings of the ION NTM-03, Anaheim, CA, January 2003.
- [4] Anonymous, (2001, April) *How Things Work: GPS Antennas*, Professional Surveyor Magazine. (Retrieved from <http://www.profsurv.com/howthng.htm>).
- [5] Anonymous, *NAVSTAR GPS Space Segment/Navigation User Interfaces*, Interface Control Document No. ICD-GPS-200C, Department of Defense, June 2000.
- [6] Besser, J. et al., "Trunav: A Low-Cost Guidance/Navigation Unit Integrating a SAASM-Based GPS and MEMS IMU in a Deeply Coupled Mechanization", Proceedings of the ION GPS-02, Portland, OR, September 2002.
- [7] Brenner, Mats, "Integrated GPS/Inertial Fault Detection Availability", *Journal of the Institute of Navigation*, Vol 43, No.2, 1996. pp. 339-358.
- [8] Bevely, David, "High Speed, Dead Reckoning, and Towed Implement Control for Automatically Steered Farm Tractors Using GPS", Ph.D. Thesis, Department of Aeronautics and Astronautics, Stanford University, June 2001.
- [9] Canadian Marconi Company. User's Manual for Allstar: PN 220-600944-0XX. Publication No. 1200-GEN- 0101A. November 1998.
- [10] Carlson, Christopher R. and J.C. Gerdes, "Identifying Tire Pressure Variation by Nonlinear Estimation of Longitudinal Stiffness and Effective Radius", Symposium on Advanced Vehicle Control (AVEC), Hiroshima, Japan, September 2002.
- [11] Carlson, Christopher R., J.C. Gerdes and J.D. Powell, "Practical Position and Yaw Rate Estimation with GPS and Differential Wheelspeeds", Symposium on Advanced Vehicle Control (AVEC), Hiroshima, Japan, September 2002.

- [12] Carlson, Neal A., "Federated Filter for Distributed Navigation and Tracking Applications", Proceedings of the ION 58th Annual Meeting, Albuquerque, NM, June 2002.
- [13] Cohen, C.E., "Attitude Determination Using GPS", Ph.D. Thesis, Department of Aeronautics and Astronautics, Stanford University, 1992.
- [14] Conway, Andrew, P. Montgomery, S. Rock, R. Cannon, and B.W. Parkinson, "A New Motion-Based Algorithm for GPS Attitude Integer Resolution", *Journal of the Institute of Navigation*, Vol. 43, No. 2, Summer 1996, pp. 179-190.
- [15] DeLorenzo, D. et al., "GPS Attitude Determination for a JPALS Testbed: Integer Initialization and Testing", Proceedings of the IEEE Position Location and Navigation Symposium, Monterey, CA, April 2004.
- [16] Diesel, J. W. and S. Luu, "GPS/IRS AIME: Calculation of Thresholds and Protection Radius Using Chi-Square Methods", Proceedings of ION NTM-95, Palm Springs, CA, September 1995.
- [17] Drucker, Eric, (2000, February). "*Model PLL Dynamics and Phase-Noise Performance*". Microwaves & RF Magazine.
- [18] Dumaine, Michael, "High Precision Attitude Using Low Cost GPS Receivers", Proceedings of the ION GPS-96, Kansas City, MO, September 1996.
- [19] Enge, Per, and Pratap Misra, *Global Positioning System: Signals, Measurements, and Performance*, Ganga-Jamuna Press, 2001.
- [20] Euler, H.J. and C.H. Hill, "An Optimal Ambiguity Resolution Technique for Attitude Determination", Proceedings of the 1996 IEEE Position Location and Navigation Symposium (PLANS), Atlanta, GA, April 1996.
- [21] Euler, H.J. and C.H. Hill, "Attitude Determination: Exploring all Information for Optimal Ambiguity Resolution", Proceedings of the ION GPS-95, Palm Springs, California, September 1995.
- [22] Farrell, Jay and Matthew Barth, *The Global Positioning System and Inertial Navigation*, McGraw-Hill, New York, NY, 1999.
- [23] Farrell, J. and M. Barth, "Integration of GPS-aided INS into Advanced Vehicle Control and Safety Systems", MOU-374 Final Report, University of California, Riverside, October 2000.

- [24] Fathy, Hosam, "Direct Determination of Aircraft Quaternion Parameters Using a Combined Inertial-GPS Instrumentation System", Proceedings of ION-NTM 99, San Diego, CA, January 1999.
- [25] Filler, R., "The Acceleration Sensitivity of Quartz Crystal Oscillators: A Review", IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control, Vol. 35, No. 3, May 1988. pp. 297-305.
- [26] Ford, Tom, "Magnetic Beeline: Satellite Derived Attitude for Marine Navigation", Proceedings of ION-GPS-98, Nashville, TN, September 1998.
- [27] Ford, Tom, "Beeline RT20 A Compact, Medium Precision Positioning System with an Attitude", Proceedings of ION GPS-97, Kansas City, MO, September 1997.
- [28] Furuno Electric Co. Ltd., Product Specifications for Satellite Compass, Model SC-120. 2002. <http://www.furuno.co.jp/english/sc/sc_120.html>
- [29] Franklin, G.F., J.D. Powell and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, Addison Wesley Longman Inc., 3rd Edition, Menlo Park, CA, 1994.
- [30] Franklin, G.F., J.D. Powell and M. Workman, *Digital Control of Dynamic Systems*, Addison Wesley Longman Inc., 3rd Edition, Menlo Park, CA, 1998.
- [31] Gautier, J.D., "GPS/INS Generalized Evaluation Tool (GIGET) for the Design and Testing of Integrated Navigation Systems", Department of Aeronautics and Astronautics, Ph.D. Thesis, Stanford University, 2003.
- [32] Gebre-Egziabher, Demoz, "Design and Performance Analysis of a Low-Cost Dead-Reckoning Navigator for General Aviation Applications", Ph.D. Thesis, Department of Aeronautics and Astronautics, Stanford University, 2001.
- [33] Gebre-Egziabher, Demoz, R. Hayward and J.D. Powell, "A Low-Cost GPS/Inertial AHRS for General Aviation Applications", Proceedings of the 1998 IEEE Position Location and Navigation Symposium, (PLANS), Palm Springs, CA, March 1998.
- [34] Gebre-Egziabher, Demoz, A. Razavi et al., "Doppler Aided Tracking Loops for SRGPS Integrity Monitoring", Proceedings of the ION GPS-03, Portland, OR, September 2003.
- [35] Gelb, Arthur, *Applied Optimal Estimation*, MIT Press, Cambridge, MA, 1976.
- [36] Gomez, S.F., "Attitude Determination and ADOP Results for International Space Station GPS Receiver", Proceedings of the ION GPS-00, Salt-Lake City, UT, September 2000.

- [37] Grebenkemper, C. John, (2001), “*Local Oscillator Phase Noise and its Effect on Receiver Performance*”, The Communications Edge, Watikns-Johnson Communications Inc. Tech Notes, Vol. 8 No. 6. <http://www.wj.com/pdf/technotes/LO_phase_noise.pdf>
- [38] Grewal, Mohinder S., L.R. Weill, and A.P. Andrews, *Global Positioning Systems, Inertial Navigation, and Integration*, John Wiley & Sons Inc., New York, New York, 2001.
- [39] Grewal, Mohinder S. and Angus P. Andrews, *Kalman Filtering: Theory and Practice Using Matlab*, John Wiley & Sons Inc., New York, New York, 2001.
- [40] Gustafson, Donald et al., “A High Anti-Jam GPS Based Navigator”, Proceedings of the ION-NTM 2000, Anaheim, CA, January 2000.
- [41] Hayward, R.C., Demoz Gebre-Egziabher et al., “Inertially Aided GPS Based Attitude Heading Reference System (AHRS) for General Aviation Aircraft”, Proceedings of ION GPS-97, Kansas City, MO, September 1997.
- [42] Hayward, R.C. and J.D. Powell, “Real Time Calibration of Antenna Phase Errors for Ultra-Short Baseline Attitude Systems”, Proceedings of the ION GPS-98, Nashville, TN, September 1998.
- [43] Hayward, R.C., A. Marchick and J.D. Powell, “Single Baseline GPS Based Attitude Heading Reference System (AHRS) for Aircraft Applications”, Proceedings of the American Control Conference, San Diego, CA, June 1999.
- [44] Hegarty, Christopher J., “Analytical Derivation of Maximum Tolerable In-Band Interference Levels for Aviation Applications of GNSS”, *Journal of the Institute of Navigation*, Vol. 44, No. 1, Spring 1997, pp. 25-34.
- [45] Heo, M.B, Boris S. Pervan, et. al. “Airborne Autonomous Fault Detection for Shipboard Landing Navigation Using Carrier Phase DGPS”, Proceedings of the ION-NTM-03, Anaheim, CA, January 2003.
- [46] Hoffman-Wellenhof, Bernhard, H. Lichtenegger and J. Collins, *GPS: Theory and Practice*, 5th Edition, Springer-Verlag Wien, New York, NY, 2001.
- [47] Holmes, J.K., *Coherent Spread Spectrum Systems*, John Wiley & Sons Inc., New York, New York, 1982.
- [48] Knight, Don, “A New Method of Instantaneous Ambiguity Resolution”, Proceedings of the ION GPS-94, Salt-Lake City, UT, September 1994.

- [49] Kraye, Christian et al., “Performance Analysis and Development of a Tightly Coupled GNSS/INS System”, Institute of Geodesy and Navigation, University of the Federal Armed Forces, Munich, 2002.
- [50] Montgomery, P.Y., “Carrier Differential GPS as a Sensor for Automatic Control”, Ph.D. Thesis, Stanford University, 1996.
- [51] Mueller, K.T., Darren Dow et al., “Field Test Results with a High Speed Rail Prototype GPS Locomotive Location System”, Proceedings of ION GPS-03, Portland, OR, September 2003.
- [52] Navtech Seminars and GPS Supply. Product information page for Allstar boards.2003.
<<http://www.navtechgps.com/supply/allstar.asp>>
- [53] O’Connor, Michael, Thomas Bell et al., “Automatic Steering of Farm Vehicles Using GPS”, Proceedings of the 3rd International Conference on Precision Agriculture, Minneapolis, Minnesota, June 1996.
- [54] Pervan, Boris S, Fang-Cheng Chan, et. al, “System Concepts for Cycle Ambiguity Resolution and Verification for Aircraft Carrier Landing”, Proceedings of ION GPS-01, Salt Lake City, UT, September 2001.
- [55] Petzold, Charles, *Programming Windows*, Microsoft Press, 5th edition, November 1998.
- [56] Phillips, Richard E. and George T. Schmidt, “GPS INS Integration”, AGARD Lecture Series MSP LS 207 on System Implications and Innovative Application of Satellite Navigation, NATO, Paris, July 1996.
- [57] Robert Bosch Corporation. DRS-MM1 Yaw-Rate Sensor Specification Sheet, 1999.
- [58] Ryu, J., E. Rossetter, J.C. Gerdes, “Vehicle Sideslip and Roll Parameter Estimation using GPS”, Proceedings of the International Symposium on Advanced Vehicle Control, Hiroshima, Japan, September 2002.
- [59] Stengel, Robert F., *Optimal Control and Estimation*, Dover Publications Inc., New York, New York, 1994.
- [60] Rakon Inc. Product Specifications for TXO4080. 2003.
<http://www.rakon.com/models/browse-model?model_id=89&model_type=O>
- [61] Rounds, Steve, (2004, January) *Jamming Protection of GPS Receivers: Part I: Receiver Enhancements*, GPS World Magazine. (Retrieved from <http://www.gpsworld.com/gpsworld/article/articleDetail.jsp?id=81907&pageID=1>).

- [62] Spilker, J., *Digital Communications by Satellite*, Prentice Hall Inc., New Jersey, 1977.
- [63] Sutton, E., "Calibration of Differential Phase Map Compensation Using Single Axis Rotation", Proceedings of the ION GPS-98, Nashville, TN, September 1998.
- [64] Sutton, E., "Performance Improvements Achieved Using Differential Phase Map Compensation in a Direction Finding System", Proceedings of the ION GPS-98, Nashville, TN, September 1998.
- [65] Tome, Phillip, "Integrating Multiple GPS Receivers With A Low Cost IMU For Aircraft Attitude Determination", Proceedings of ION NTM-99, San Diego, CA, January 1999.
- [66] US Department of Defence, *Global Positioning System Standard Positioning Service Signal Specification*, 2nd Edition, June, 1995.
- [67] van Graas, F., and A. Soloviev, "Precise Velocity Estimation Using a Stand-Alone GPS Receiver", Proceedings of the ION NTM-03, Anaheim, CA, January 2003.
- [68] F. Van Graas and M. Braasch, "GPS Interferometric Attitude and Heading Determination: Initial Flight Test Results", Journal of the Institute of Navigation, Vol. 38, No. 3, Winter 1991.
- [69] Vectron International. Product Specifications for CO-714 OCXO. 2003.
<<http://www.vectron.com/products/ocxo/co714718s.htm>>
- [70] Wagner, J.F. and G. Kasties, "Improving the GPS/INS Integrated System Performance by Increasing the Distance Between GPS Antennas and Inertial Sensors", Proceedings of the ION NTM-02, San Diego, CA, January 2002.
- [71] Wang, C. and G. Lachapelle, "GPS Attitude Determination Reliability Performance Improvement Using Low-Cost Receivers" Proceedings of the ION GPS-02, Portland, OR, September 2002.
- [72] Zarlink Semiconductor. Product Specifications for GP2021 RF Front End. 2003.
<http://products.zarlink.com/product_profiles/GP2010.htm>
- [73] Zarlink Semiconductor. Product Specifications for GP2021 12-Channel Correlator. 2003.
<http://products.zarlink.com/product_profiles/GP2021.htm>