Global Positioning Systems, Inertial Navigation, and Integration,
Mohinder S. Grewal, Lawrence R. Weill, Angus P. Andrews

Print ISBN 0-471-35032-X Electronic ISBN 0-471-20071-9

Global Positioning Systems, Inertial Navigation, and Integration

Global Positioning Systems, Inertial Navigation, and Integration

MOHINDER S. GREWAL
California State University at Fullerton

LAWRENCE R. WEILL
California State University at Fullerton

ANGUS P. ANDREWS
Rockwell Science Center

A John Wiley & Sons, Inc. Publication
NEW YORK / CHICHESTER / WEINHEIM / BRISBANE /SINGAPORE /TORONTO

ISBN 0-471-20071-9.

This title is also available in print as ISBN 0-471-35032-X.

For more information about Wiley products, visit our web site at www.Wiley.com.

Contents

Preface

This book is intended for people who will use Global Positioning Systems (GPS), Inertial Navigation Systems (INS), and Kalman filters. Our objective is to give our readers a working familiarity with both the *theoretical* and *practical* aspects of these subjects. For that purpose we have included "real-world" problems from practice as illustrative examples. We also cover the more practical aspects of implementation: how to represent problems in a mathematical model, analyze performance as a function of model parameters, implement the mechanization equations in numerically stable algorithms, assess its computational requirements, test the validity of results, and monitor performance in operation with sensor data from GPS and INS. These important attributes, often overlooked in theoretical treatments, are essential for effective application of theory to real-world problems.

The accompanying diskette contains MATLAB™ m-files to demonstrate the workings of the Kalman filter algorithms with GPS and INS data sets, so that the reader can better discover how the Kalman filter works by observing it in action with GPS and INS. The implementation of GPS, INS, and Kalman filtering on computers also illuminates some of the practical considerations of finite-wordlength arithmetic and the need for alternative algorithms to preserve the accuracy of the results. If the student wishes to apply what she or he learns, then it is essential that she or he experience its workings and failings—and learn to recognize the difference.

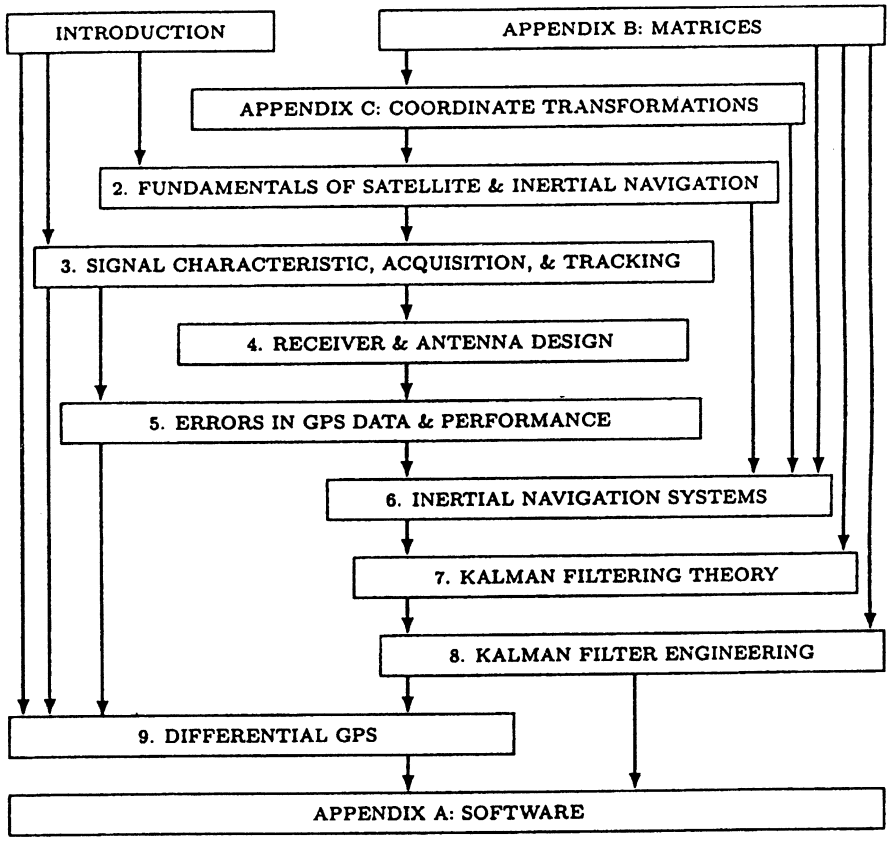
The book is organized for use as a text for an introductory course in GPS technology at the senior level or as a first-year graduate level course in GPS, INS, and Kalman filtering theory and application. It could also be used for self-instruction or review by practicing engineers and scientists in these fields.

Chapter 1 informally introduces the general subject matter through its history of development and application. Chapters 2–5 and 9 cover the basic theory of GPS and

present material for a senior-level class in geomatics, electrical engineering, systems engineering, and computer science. Chapters 6–8 cover the application of GPS and INS integration with Kalman filtering. These chapters could be covered in a graduate level course in Electrical, computer, and systems engineering.

Chapter 6 gives the basics of INS. Chapter 7 covers linear optimal filters, predictors, and nonlinear estimation by "extended" Kalman filters. Applications of these techniques to the identification of unknown parameters of systems are given as examples. Chapter 8 deals with Kalman filter engineering, with algorithms provided for computer implementation. Chapter 9 covers current developments in the Wide Area Augmentation System (WAAS) and Local-Area Augmentation System (LAAS), including Local Area Differential GPS (LADGPS) and Wide-Area Differential GPS (WADGPS).

The following chapter-level dependency graph shows the book's organization and how the subject of each chapter depends upon material in other chapters. The arrows in the figure indicate the recommended order of study. Boxes above another box and



connected by arrows indicate that the material represented by the upper boxes is background material for the subject in the lower box.

MOHINDER S. GREWAL, Ph.D., P.E.
California State University at Fullerton

LAWRENCE R. WEILL, Ph.D.
California State University at Fullerton

ANGUS P. ANDREWS, Ph.D.
Rockwell Science Center
Thousand Oaks, California

Acknowledgments

M.S.G dedicates this work to his wife, Sonja Grewal, in recognition of her active support in the preparation of the manuscript and figures.

L.R.W. wishes to thank his mother, Christine R. Weill, who recently passed away, for her love and encouragement in pursuing his chosen profession. He also is indebted to the people of Magellan Systems Corporation, who so willingly shared their knowledge of the Global Positioning System during the development of the World's first hand-held receiver for the consumer market.

A.P.A. dedicates his work to his wife, Geraldine Andrews, without whose support and forbearance this could not have happened.

M.S.G also acknowledges the assistance of Mrs. Laura Cheung, graduate student at California State University at Fullerton, for her expert assistance with the Matlab programs, and Dr. Jya-Syin Wu and N. Pandya of the Raytheon Systems Company for their assistance in reviewing the manuscript.

Acronyms

A/D	Analog-to-digital (conversion)
ADC	Analog-to-digital converter
ADS	Automatic dependent surveillance
AGC	Automatic gain control
AIC	Akaike information-theoretic criterion
ALF	Atmospheric loss factor
AOR-E	Atlantic Ocean Region East (WAAS)
AOR-W	Atlantic Ocean Region West (WAAS)
ARINC	Aeronautical Radio, Inc.
ARMA	Autoregressive moving-average
AS	Antispoofing
ATC	Air traffic control
BIH	Bureau International de l'Heure
BPSK	Binary phase-shift keying
C/A	Coarse/acquisition (channel or code)
C&V	Correction and Verification (WAAS)
CDM	Code division multiplexing
CDMA	Code division multiple access
CEP	Circle of equal probability
CERCO	Comité Européen des Responsables de la Cartographie Officielle
CFAR	Constant false alarm rate

CONUS	Conterminous United States, also continental United States
DFT	Discrete Fourier transform
DGPS	Differential GPS
DME	Distance measurement equipment
DoD	Department of Defense
DOP	Dilution of precision
ECEF	Earth centered, earth fixed (coordinates)
ECI	Earth-centered inertial (coordinates)
EDM	Electronic distance measurement
EGM	Earth Gravity Model
EGNOS	European Geostationary Navigation Overlay Service
EIRP	Effective isotropic radiated power
EMA	Electromagnetic accelerometer
EMRBE	Estimated maximum range and bias error
ENU	East-north-up (coordinates)
ESA	European Space Agency
FAA	Federal Aviation Administration
FEC	Forward error correction
FLL	Frequency-lock loop
FM	Frequency modulation
FOG	Fiber-optic gyroscope
FPE	Final prediction error
FSLF	Free-space loss factor
FVS	Functional verification system
GBI	Ground-based interceptor
GDOP	Geometric dilution of precision
GEO	Geostationary earth orbit
GES	COMSAT GPS earth station
GIPSY	GPS-Infrared Positioning System
GIS	Geographical Information Systems
GIVE	Grid ionosphere vertical error
GLONASS	Global Orbiting Navigation Satellite System
GNSS	Global Navigation Satellite System
GOA	GIPSY/OASIS analysis
GPS	Global Positioning System
GUS	GEO uplink subsystem
HAL	Horizontal alert system
HDOP	Horizontal dilution of precision
HOT	Higher order terms

HOW	Hand-over word
HPL	Horizontal protection limit
IAG	International Association of Geodesy
IERS	International Earth Rotation Service
IF	Intermediate frequency
IGP	Ionospheric grid point (for WAAS)
ILS	Instrument Landing System
Inmarsat	International Mobile (originally "Maritime") Satellite Organization
INS	Inertial navigation system
IODC	Issue of data, clock
IODE	Issue of data, ephemeris
IOR	Indian Ocean Region (WAAS)
IRM	IERS reference meridian
IRP	IERS reference pole
IRU	Inertial reference unit
ISO	International Standardization Organization
ITRF	International Terrestrial Reference Frame
ITRS	International Terrestrial Reference System
ITS	Intelligent Transport Systems
ITU	International Telecommunications Union
JCAB	Japanese Commercial Aviation Board
JTIDS	Joint Tactical Information Distribution System
LAAS	Local Area Augmentation System
LADGPS	Local-area differential GPS
LEO	Low earth orbit
LHS	Left-hand side (of an equation)
LORAN	Long-range navigation
LPF	Low-pass filter
LSB	Least significant bit
LTP	Local tangent plane
MEDLL	Multipath-estimating delay-lock loop
MEMS	Micro-electromechanical systems
ML	Maximum likelihood
MLE	Maximum-likelihood estimate
MMSE	Minimum mean-squared error (estimator)
MMT	Multipath mitigation technology
MSAS	MTSAT Based Augmentation System
MSB	Most significant bit
MSL	Mean sea level

MTSAT	Multifunctional Transport Satellite
MVUE	Minimum-variance unbiased estimator
NAS	National Airspace System
NAVSTAR	Navigation System with Time and Ranging
NCO	Numerically controlled oscillator
NDB	Nondirectional beacon
NED	North–east–down (coordinates)
NGS	National Geodetic Survey
NIMA	National Imaging and Mapping Agency
NNSS	Navy Navigation Satellite System
NPA	Non-precision approach
NSTB	National Satellite Test Bed
OASIS	Orbit Analysis Simulation Software
PA	Precision approach
P-code	Precision code
PDF	Probability density function
PDOP	Position dilution of precision
PI	Proportional and integral (controller)
PIGA	Pulse-integrating gyroscopic accelerometer
PLGR	Personal low-cost GPS receiver
PLL	Phase-lock loop
PLRS	Position Location and Reporting System
PN	Pseudonoise
POR	Pacific Ocean Region (WAAS)
PPS	Precise Positioning Service
PRN	Pseudorandom noise or pseudorandom number
PRNAV	Precision Area Navigation
PSD	Power spectral density
RAAN	Right ascension of ascending node
RAG	Relative antenna gain
RF	Radio frequency
RINEX	Receiver Independent Exchange Format (for GPS data)
RLG	Ring laser gyroscope
RMS	Root mean squared, also Reference Monitoring Station
RNAV	Area navigation
ROC	Receiver operating characteristic
RPY	Roll pitch yaw (coordinates)
RTCM	Radio Technical Commission for Maritime Service
SA	Selective Availability (also abbreviated “S/A”)

SAE	Society of Automotive Engineers
SAVVAN	Système Automatique de Vérification en Vol des Aides a la Navigation
SAW	Surface acoustic wave
SBAS	Space-based augmentation system
SBIRLEO	Space-based infrared low earth orbit
SIS	Signal in space
SNR	Signal-to-noise ratio
SPS	Standard Positioning Service
SV	Space vehicle (time)
SVN	Space vehicle number (= PRN for GPS)
TCS	Terrestrial communications subsystem (for WAAS)
TCXO	Temperature compensated Xtal (crystal) oscillator
TDOP	Time dilution of precision
TEC	Total electron count
TLM	Telemetry word
TOA	Time of arrival
TOW	Time of week
TTFF	Time to first fix
UDDF	Universal Data Delivery Format
UDRE	User differential range error
UERE	User-equivalent range error
UPS	Universal Polar Stereographic
URE	User range error
UTC	Universal Time Coordinated (or Coordinated Universal Time)
UTM	Universal Transverse Mercator
VAL	Vertical alert limit
VDOP	Vertical dilution of precision
VHF	Very high frequency (30–300 MHz)
VOR	VHF OmniRange (radio navigation aid)
VPL	Vertical protection limit
WAAS	Wide Area Augmentation System
WADGPS	Wide-area differential GPS
WGS	World Geodetic System
WMS	Wide-area master station
WN	Week number
WNT	WAAS network time
WRE	Wide-area reference equipment
WRS	Wide-area reference station

Global Positioning Systems, Inertial Navigation, and Integration

Global Positioning Systems, Inertial Navigation, and Integration,
Mohinder S. Grewal, Lawrence R. Weill, Angus P. Andrews

Print ISBN 0-471-35032-X Electronic ISBN 0-471-20071-9

1

Introduction

The five basic forms of navigation are as follows:

1. Pilotage, which essentially relies on recognizing landmarks to know where you are. It is older than human kind.
2. Dead reckoning, which relies on knowing where you started from, plus some form of heading information and some estimate of speed.
3. Celestial navigation, using time and the angles between local vertical and known celestial objects (e.g., sun, moon, or stars) [115].
4. Radio navigation, which relies on radio-frequency sources with known locations (including Global Positioning System satellites).
5. Inertial navigation, which relies on knowing your initial position, velocity, and attitude and thereafter measuring your attitude rates and accelerations. It is the only form of navigation that does not rely on external references.

These forms of navigation can be used in combination as well [16, 135]. The subject of this book is a combination of the fourth and fifth forms of navigation using Kalman filtering.

Kalman filtering exploits a powerful synergism between the *Global Positioning System* (GPS) and an *inertial navigation system* (INS). This synergism is possible, in part, because the INS and GPS have very complementary error characteristics. Short-term position errors from the INS are relatively small, but they degrade without bound over time. GPS position errors, on the other hand, are not as good over the short term, but they do not degrade with time. The Kalman filter is able to take advantage of these characteristics to provide a common, integrated navigation

implementation with performance superior to that of either subsystem (GPS or INS). By using statistical information about the errors in both systems, it is able to combine a system with tens of meters position uncertainty (GPS) with another system whose position uncertainty degrades at kilometers per hour (INS) and achieve bounded position uncertainties in the order of centimeters [with differential GPS (DGPS)] to meters.

A key function performed by the Kalman filter is the statistical combination of GPS and INS information to track drifting parameters of the sensors in the INS. As a result, the INS can provide enhanced inertial navigation accuracy during periods when GPS signals may be lost, and the improved position and velocity estimates from the INS can then be used to make GPS signal reacquisition happen much faster when the GPS signal becomes available again.

This level of integration necessarily penetrates deeply into each of these subsystems, in that it makes use of partial results that are not ordinarily accessible to users. To take full advantage of the offered integration potential, we must delve into technical details of the designs of both types of systems.

1.1 GPS AND GLONASS OVERVIEW

1.1.1 GPS

The GPS is part of a satellite-based navigation system developed by the U.S. Department of Defense under its NAVSTAR satellite program [54, 56, 58–63, 96–98].

1.1.1.1 GPS Orbits The fully operational GPS includes 24 or more (28 in March 2000) active satellites approximately uniformly dispersed around six circular orbits with four or more satellites each. The orbits are inclined at an angle of 55° relative to the equator and are separated from each other by multiples of 60° right ascension. The orbits are nongeostationary and approximately circular, with radii of 26,560 km and orbital periods of one-half sidereal day (≈11.967 h). Theoretically, three or more GPS satellites will always be visible from most points on the earth's surface, and four or more GPS satellites can be used to determine an observer's position anywhere on the earth's surface 24 h per day.

1.1.1.2 GPS Signals Each GPS satellite carries a cesium and/or rubidium atomic clock to provide timing information for the signals transmitted by the satellites. Internal clock correction is provided for each satellite clock. Each GPS satellite transmits two spread spectrum, L-band carrier signals—an L_1 signal with carrier frequency $f_l = 1575.42$ MHz and an L_2 signal with carrier frequency $f_2 = 1227.6$ MHz. These two frequencies are integral multiples $f_1 = 1540 f_0$ and $f_2 = 1200 f_0$ of a base frequency $f_0 = 1.023$ MHz. The L_1 signal from each satellite uses *binary phase-shift keying* (BPSK), modulated by two *pseudorandom noise* (PRN) codes in phase quadrature, designated as the C/A-code and P-code. The L_2

signal from each satellite is BPSK modulated by only the P-code. A brief description of the nature of these PRN codes follows, with greater detail given in Chapter 3.

Compensating for Propagation Delays This is one motivation for use of two different carrier signals L_1 and L_2. Because delay varies approximately as the inverse square of signal frequency f (delay $\propto f^{-2}$), the measurable differential delay between the two carrier frequencies can be used to compensate for the delay in each carrier. (See [86] for details.)

Code Division Multiplexing Knowledge of the PRN codes allows users independent access to multiple GPS satellite signals on the same carrier frequency. The signal transmitted by a particular GPS signal can be selected by generating and matching, or correlating, the PRN code for that particular satellite. All PRN codes are known and are generated or stored in GPS satellite signal receivers carried by ground observers. A first PRN code for each GPS satellite, sometimes referred to as a precision code or P-code, is a relatively long, fine-grained code having an associated clock or chip rate of $10f_0 = 10.23$ MHz. A second PRN code for each GPS satellite, sometimes referred to as a clear or coarse acquisition code or C/A-code, is intended to facilitate rapid satellite signal acquisition and hand-over to the P-code. It is a relatively short, coarser grained code having an associated clock or chip rate $f_0 = 1.023$ MHz. The C/A-code for any GPS satellite has a length of 1023 chips or time increments before it repeats. The full P-code has a length of 259 days, during which each satellite transmits a unique portion of the full P-code. The portion of P-code used for a given GPS satellite has a length of precisely one week (7.000 days) before this code portion repeats. Accepted methods for generating the C/A-code and P-code were established by the satellite developer[1] in 1991 [42, 66].

Navigation Signal The GPS satellite bit stream includes navigational information on the ephemeris of the transmitting GPS satellite and an almanac for all GPS satellites, with parameters providing approximate corrections for ionospheric signal propagation delays suitable for single-frequency receivers and for an offset time between satellite clock time and true GPS time. The navigational information is transmitted at a rate of 50 baud. Further discussion of the GPS and techniques for obtaining position information from satellite signals can be found in Chapter 3 and in [84, pp. 1–90].

1.1.1.3 Selective Availability

Selective Availability (SA) is a combination of methods used by the U.S. Department of Defense for deliberately derating the accuracy of GPS for "nonauthorized" (i.e., non–U.S. military) users. The current satellite configurations use only pseudorandom dithering of the onboard time reference [134], but the full configuration can also include truncation of the

[1] Satellite Systems Division of Rockwell International Corporation, now part of the Boeing Company.

transmitted ephemerides. This results in three grades of service provided to GPS users. SA has been removed as of May 1, 2000.

Precise Positioning Service Precise Positioning Service (PPS) is the full-accuracy, single-receiver GPS positioning service provided to the United States and its allied military organizations and other selected agencies. This service includes access to the unencrypted P-code and the removal of any SA effects.

Standard Positioning Service without SA Standard Positioning Service (SPS) provides GPS single-receiver (stand-alone) positioning service to any user on a continuous, worldwide basis. SPS is intended to provide access only to the C/A-code and the L_1 carrier.

Standard Positioning Service with SA The horizontal-position accuracy, as degraded by SA, currently is advertised as 100 m, the vertical-position accuracy as 156 m, and time accuracy as 334 ns—all at the 95% probability level. SPS also guarantees the user-specified levels of coverage, availability, and reliability.

1.1.2 GLONASS

A second configuration for global positioning is the Global Orbiting Navigation Satellite System (GLONASS), placed in orbit by the former Soviet Union, and now maintained by the Russian Republic [75, 80].

1.1.2.1 GLONASS Orbits GLONASS also uses 24 satellites, but these are distributed approximately uniformly in three orbital plans (as opposed to four for GPS) of eight satellites each (six for GPS). Each orbital plane has a nominal inclination of 64.8° relative to the equator, and the three orbital planes are separated from each other by multiples of 120° right ascension. GLONASS orbits have smaller radii than GPS orbits, about 25,510 km, and a satellite period of revolution of approximately $\frac{8}{17}$ of a sidereal day. A GLONASS satellite and a GPS satellite will complete 17 and 16 revolutions, respectively, around the earth every 8 days.

1.1.2.2 GLONASS Signals The GLONASS system uses frequency division multiplexing of independent satellite signals. Its two carrier signals corresponding to L_1 and L_2 have frequencies $f_1 = (1.602 + 9k/16)\,\text{GHz}$ and $f_2 = (1.246 + 7k/16)\,\text{GHz}$, where $k = 0, 1, 2, \ldots, 23$ is the satellite number. These frequencies lie in two bands at 1.597–1.617 GHz (L_1) and 1240–1260 GHz (L_2). The L_1 code is modulated by a C/A-code (chip rate = 0.511 MHz) and by a P-code (chip rate = 5.11 MHz). The L_2 code is presently modulated only by the P-code. The GLONASS satellites also transmit navigational data at a rate of 50 baud. Because the satellite frequencies are distinguishable from each other, the P-code and the C/A-code are the same for each satellite. The methods for receiving and analyzing

GLONASS signals are similar to the methods used for GPS signals. Further details can be found in the patent by Janky [66].

GLONASS does not use any form of SA.

1.2 DIFFERENTIAL AND AUGMENTED GPS

1.2.1 Differential GPS

Differential GPS (DGPS) is a technique for reducing the error in GPS-derived positions by using additional data from a reference GPS receiver at a known position. The most common form of DGPS involves determining the combined effects of navigation message ephemeris and satellite clock errors (including propagation delays and the effects of SA) at a reference station and transmitting pseudorange corrections, in real time, to a user's receiver, which applies the corrections in the process of determining its position [63, 96, 98].

1.2.2 Local-Area Differential GPS

Local-area differential GPS (LAGPS) is a form of DGPS in which the user's GPS receiver also receives real-time pseudorange and, possibly, carrier phase corrections from a local reference receiver generally located within the line of sight. The corrections account for the combined effects of navigation message ephemeris and satellite clock errors (including the effects of SA) and, usually, atmospheric propagation delay errors at the reference station. With the assumption that these errors are also common to the measurements made by the user's receiver, the application of the corrections will result in more accurate coordinates.

1.2.3 Wide-Area Differential GPS

Wide-area DGPS (WADGPS) is a form of DGPS in which the user's GPS receiver receives corrections determined from a network of reference stations distributed over a wide geographical area. Separate corrections are usually determined for specific error sources—such as satellite clock, ionospheric propagation delay, and ephemeris. The corrections are applied in the user's receiver or attached computer in computing the receiver's coordinates. The corrections are typically supplied in real time by way of a geostationary communications satellite or through a network of ground-based transmitters. Corrections may also be provided at a later date for postprocessing collected data [63].

1.2.4 Wide-Area Augmentation System

Three space-based augmentation systems (SBASs) were under development at the beginning of the third millenium. These are the Wide Area Augmentation System (WAAS), European Geostationary Navigation Overlay System (EGNOS),

and Multifunctional Transport Satellite (MTSAT) Based Augmentation System (MSAS).

The WAAS enhances the GPS SPS over a wide geographical area. The U.S. Federal Aviation Administration (FAA), in cooperation with other agencies, is developing WAAS to provide WADGPS corrections, additional ranging signals from geostationary earth orbit (GEO) satellites, and integrity data on the GPS and GEO satellites.

1.2.5 Inmarsat Civil Navigation

The Inmarsat overlay is an implementation of a wide-area differential service. Inmarsat is the International Mobile Satellite Organization, an 80-nation international consortium, originally created in 1979 to provide maritime[2] mobile services on a global basis but now offering a much wider range of mobile satellite services. Inmarsat launched four geostationary satellites that provide complete coverage of the globe from $\pm 70^\circ$ latitude. The data broadcast by the satellites are applicable to users in regions having a corresponding ground station network. The U.S. region is the continental U.S. (CONUS) and uses Atlantic Ocean Region West (AOR-W) and Pacific Ocean Region (POR) geostationary satellites. This is called the WAAS and is being developed by the FAA. The ground station network is operated by the service provider, that is, the FAA, whereas Inmarsat is responsible for operation of the space segment. Inmarsat affiliates operate the uplink earth stations (e.g., COMSAT in the United States). WAAS is discussed further in Chapter 9.

1.2.6 Satellite Overlay

The Inmarsat Civil Navigation Geostationary Satellite Overlay extends and complements the GPS and GLONASS satellite systems. The overlay navigation signals are generated at ground based facilities. For example, for WAAS, two signals are generated from Santa Paula, California—one for AOR-W and one for POR. The back-up signal for POR is generated from Brewster, Washington. The backup signal for AOR-W is generated from Clarksburg, Maryland. Signals are uplinked to Inmarsat-3 satellites such as AOR-W and POR. These satellites contain special satellite repeater channels for rebroadcasting the navigation signals to users. The use of satellite repeater channels differs from the navigation signal broadcast techniques employed by GLONASS and GPS. GLONASS and GPS satellites carry their own navigation payloads that generate their respective navigation signals.

1.2.7 Future Satellite Systems

In Europe, activities supported by the European TRIPARTITE Group [European Space Agency (ESA), European Commission (EC), EUROCONTROL] are under-

[2] The “mar” in the name originally stood for “maritime.”

way to specify, install, and operate a future civil Global Navigation Satellite System (GNSS) (GNSS-2 or GALILEO).

Based on the expectation that GNSS-2 will be developed through an evolutionary process as well as long-term augmentations [e.g., GNSS-1 or European GNSS Navigation Overlay Service (EGNOS)], short- to midterm augmentation systems (e.g., differential systems) are being targeted.

The first steps toward GNSS-2 will be made by the TRIPARTITE Group. The augmentations will be designed such that the individual elements will be suitable for inclusion in GNSS-2 at a later date. This design process will provide the user with maximum continuity in the upcoming transitions.

In Japan, the Japanese Commercial Aviation Board (JCAB) is developing the MSAS.

1.3 APPLICATIONS

Both GPS and GLONASS have evolved from dedicated military systems into true dual-use systems. Satellite navigation technology is utilized in numerous civil and military applications, ranging from golf and leisure hiking to spacecraft navigation. Further discussion on applications can be found in Chapters 8 and 9.

1.3.1 Aviation

The aviation community has propelled the use of GNSS and various augmentations (e.g., WAAS, EGNOS, and MSAS). These systems provide guidance for en route through precision approach phases of flight. Incorporation of a data link with a GNSS receiver enables the transmission of aircraft location to other aircraft and/or to air traffic control (ATC). This function is called automatic dependent surveillance (ADS) and is in use in the POR. Key benefits are ATC monitoring for collision avoidance and optimized routing to reduce travel time and fuel consumption [98].

1.3.2 Spacecraft Guidance

The space shuttle utilizes GPS for guidance in all phases of its operation (e.g., ground launch, on-orbit and reentry, and landing). NASA's small satellite programs use and plan to use GPS, as does the military on SBIRLEO (space-based infrared low earth orbit) and GBI (ground-based interceptor) kill vehicles.

1.3.3 Maritime

GNSS has been used by both commercial and recreational maritime communities. Navigation is enhanced on all bodies of waters, from oceanic travel to river ways, especially in bad weather.

1.3.4 Land

The surveying community heavily depends on DGPS to achieve measurement accuracies in the millimeter range. Similar techniques are used in farming, surface mining, and grading for real-time control of vehicles and in the railroad community to obtain train locations with respect to adjacent tracks. GPS is a key component in Intelligent Transport Systems (ITS). In vehicle applications, GNSS is used for route guidance, tracking, and fleet management. Combining a cellular phone or data link function with this system enables vehicle tracing and/or emergency messaging.

1.3.5 Geographic Information Systems (GIS), Mapping, and Agriculture

Applications include utility and asset mapping and automated airborne mapping, with remote sensing and photogrammetry. Recently, GIS, GPS, and remote sensing have matured enough to be used in agriculture. GIS companies such as Environmental System Research Institute (Redlands, California) have developed software applications that enable growers to assess field conditions and their relationship to yield. Real time kinematic and differential GNSS applications for precision farming are being developed. This includes soil sampling, yield monitoring, chemical, and fertilizer applications. Some GPS analysts are predicting precision site-specific farming to become “the wave of the future.”

Global Positioning Systems, Inertial Navigation, and Integration,
Mohinder S. Grewal, Lawrence R. Weill, Angus P. Andrews

Print ISBN 0-471-35032-X Electronic ISBN 0-471-20071-9

2

Fundamentals of Satellite and Inertial Navigation

2.1 NAVIGATION SYSTEMS CONSIDERED

This book is about GPS and INS and their integration. An inertial navigation unit can be used anywhere on the globe, but it must be updated within hours of use by independent navigation sources such as GPS or celestial navigation. Thousands of self-contained INS units are in continuous use on military vehicles, and an increasing number are being used in civilian applications.

2.1.1 Systems Other Than GPS

GPS signals may be replaced by LORAN-C signals produced by three or more long-range navigation (LORAN) signal sources positioned at fixed, known locations for outside-the-building location determination. A LORAN-C system relies upon a plurality of ground-based signal towers, preferably spaced apart 100–300 km, that transmit distinguishable electromagnetic signals that are received and processed by a LORAN signal antenna and LORAN signal receiver/processor that are analogous to the Satellite Positioning System signal antenna and receiver/processor. A representative LORAN-C system is discussed in *LORAN-C User Handbook* [85]. LORAN-C signals use carrier frequencies of the order of 100 kHz and have maximum reception distances of hundreds of kilometers. The combined use of FM signals for location determination inside a building or similar structure can also provide a satisfactory location determination (LD) system in most urban and suburban communities.

There are other ground-based radiowave signal systems suitable for use as part of an LD system. These include Omega, Decca, Tacan, JTIDS Relnav (U.S. Air Force Joint Tactical Information Distribution System Relative Navigation), and PLRS (U.S. Army Position Location and Reporting System). See summaries in [84, pp. 6–7 and 35–60].

2.1.2 Comparison Criteria

The following criteria may be used in selecting navigation systems appropriate for a given application system:

1. navigation method(s) used,
2. coordinates provided,
3. navigational accuracy,
4. region(s) of coverage,
5. required transmission frequencies,
6. navigation fix update rate,
7. user set cost, and
8. status of system development and readiness.

2.2 FUNDAMENTALS OF INERTIAL NAVIGATION

This is an introductory-level overview of inertial navigation. Technical details are in Chapter 6 and [22, 75, 83, 118].

2.2.1 Basic Concepts of Inertial Navigation

Inertia is the propensity of bodies to maintain constant translational and rotational velocity, unless disturbed by forces or torques, respectively (Newton's first law of motion).

An *inertial reference frame* is a coordinate frame in which Newton's laws of motion are valid. Inertial reference frames are neither *rotating* nor *accelerating*.

Inertial sensors measure *rotation rate* and *acceleration*, both of which are *vector-valued* variables:

(a) *Gyroscopes* are sensors for measuring *rotation*: *rate gyroscopes* measure *rotation rate*, and *displacement gyroscopes* (also called *whole-angle* gyroscopes) measure *rotation angle*.

(b) *Accelerometers* are sensors for measuring *acceleration*. However, accelerometers *cannot* measure *gravitational acceleration*. That is, an accelerometer in free fall (or in orbit) has no detectable input.

The *input axis* of an inertial sensor defines which vector component it measures. *Multiaxis sensors* measure more than one component.

Inertial navigation uses gyroscopes and accelerometers to maintain an estimate of the position, velocity, attitude, and attitude rates of the vehicle in or on which the INS is carried, which could be a spacecraft, missile, aircraft, surface ship, submarine, or land vehicle.

An *inertial navigation system* (INS) consists of the following:

(a) an *inertial measurement unit* (*IMU*) or *inertial reference unit* (*IRU*) containing a cluster of sensors: *accelerometers* (two or more, but usually three) and *gyroscopes* (three or more, but usually three). These sensors are rigidly mounted to a common base to maintain the same relative orientations.

(b) *Navigation computers* (one or more) calculate the gravitational acceleration (not measured by accelerometers) and doubly integrate the net acceleration to maintain an estimate of the position of the host vehicle.

There are many different designs of inertial navigation systems with different performance characteristics, but they fall generally into two categories:

- gimbaled and
- strapdown.

These are illustrated in Fig. 2.1 and described in the following subsections.

2.2.2 Gimbaled Systems

2.2.2.1 Gimbals A *gimbal* is a rigid frame with rotation bearings for isolating the inside of the frame from external rotations about the bearing axes. If the bearings could be made perfectly frictionless and the frame could be made perfectly balanced (to eliminate unbalance torques due to acceleration), then the rotational inertia of the frame would be sufficient to isolate it from rotations of the supporting body. This level of perfection is generally not achievable in practice, however.

Alternatively, a gyroscope can be mounted inside the gimbal frame and used to detect any rotation of the frame due to torques from bearing friction or frame unbalance. The detected rotational disturbance can then be used in a feedback loop to provide restoring torques on the gimbal bearings to null out all rotations of the frame about the respective gimbal bearings.

At least three gimbals are required to isolate a subsystem from host vehicle rotations about three axes, typically labeled *roll*, *pitch*, and *yaw* axes.

The gimbals in an INS are mounted inside one another, as illustrated in Fig. 2.1*b*. We show *three* gimbals here because that is the *minimum* number required. Three gimbals will suffice for host vehicles with limited ranges of rotation in pitch and roll, such as surface ships and land vehicles. In those applications, the outermost axis is typically aligned with the host vehicle yaw axis (nominally vertical), so that all three

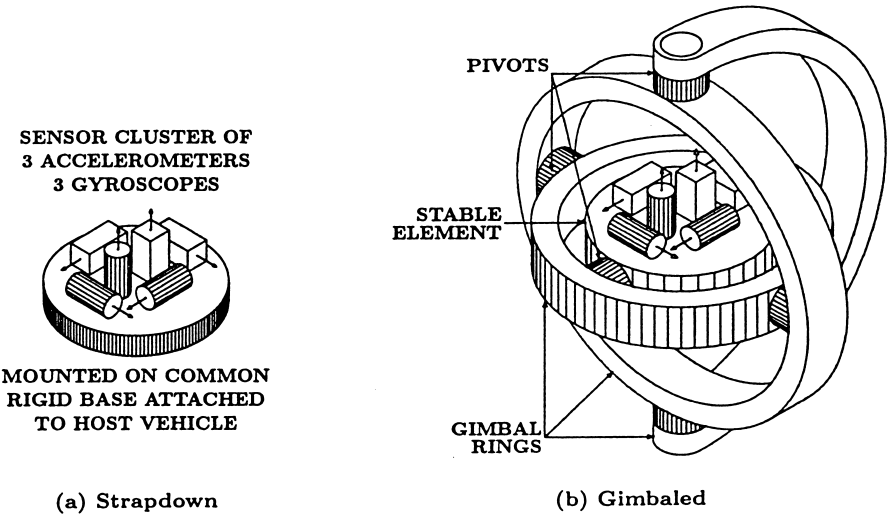


Fig. 2.1 *Inertial measurement units.*

gimbal rotation axes will remain essentially orthogonal when the inner gimbal axes are kept level and the vehicle rotates freely about its yaw axis only.

A fourth gimbal is required for vehicles with full freedom of rotation about all three axes—such as high-performance aircraft. Otherwise, rotations of the host vehicle can align two of the three gimbal axes parallel to one another in a condition called *gimbal lock*. In gimbal lock with only three gimbals, the remaining single "unlocked" gimbal can only isolate the platform from rotations about a second rotation axis. Rotations about the third axis of the "missing" gimbal will slew the platform unless a fourth gimbal axis is provided for this contingency.

2.2.2.2 Stable Platforms The earliest INSs were developed in the mid-twentieth century, when flight-qualified computers were not fast enough for integrating the full (rotational and translational) equations of motion. As an alternative, gimbals and torque servos were used to null out the rotations of a *stable platform* or *stable element* on which the inertial sensors were mounted, as illustrated in Fig. 2.1*b*.

The stable element of a gimbaled system is also called an *inertial platform* or "stable table." it contains a *sensor cluster* of accelerometers and gyroscopes, similar to that of the "strapdown" INS illustrated in Fig. 2.1*a*.

2.2.2.3 Signal Processing Essential software functions for a gimbaled INS are shown in signal flow form in Fig. 2.2, with blocks representing the major software functions, and x_1, x_2, x_3 representing position components.

The essential outputs of the gimbaled IMU are the sensed accelerations and rotation rates. These are first compensated for errors detected during sensor- or

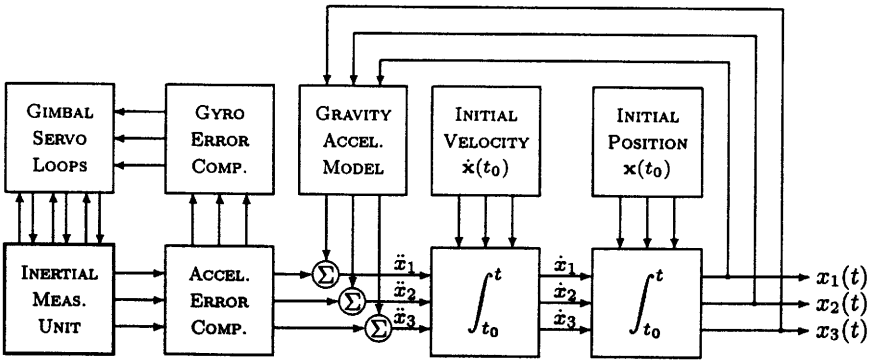


Fig. 2.2 *Essential signal processing for gimbaled INS.*

system-level calibrations. This includes compensation for gyro drift rates due to acceleration.

The compensated gyro signals are used for controlling the gimbals to keep the platform in the desired orientation, independent of the rotations of the host vehicle. This "desired orientation" can be (and usually is) *locally level*, with two of the accelerometer input axes horizontal and one accelerometer input axis vertical. *This is not an inertial orientation*, because the earth rotates, and because the host vehicle can change its longitude and latitude. Compensation for these effects is included in the gyro error compensation.

The accelerometer outputs are also compensated for known errors, including compensation for gravitational accelerations which cannot be sensed and must be modeled. The gravity model used in this compensation depends on vehicle position. This coupling of position and acceleration creates recognized dynamical behavior of position errors, including the following:

1. Schuler oscillation of horizontal position and velocity errors, in which the INS behaves like a pendulum with period equal to the orbital period (about 84.4 min at sea level). Any horizontal INS velocity errors will excite the Schuler oscillation, but the amplitude of the oscillations will be bounded so long as the INS velocity errors remain bounded.
2. Vertical-channel instability, in which positive feedback of altitude errors through the gravity model makes INS altitude errors unstable. For INS applications in surface ships, the vertical channel can be eliminated. External water pressure is used for estimating depth for submarines, and barometric altitude is commonly used to stabilize the vertical channel for aircraft.

After compensation for sensor errors and gravity, the accelerometer outputs are integrated once and twice to obtain velocity and position, respectively. The position estimates are usually converted to longitude, latitude, and altitude.

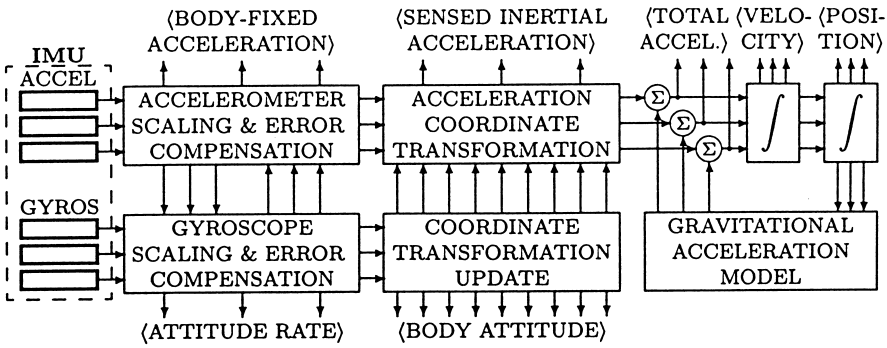


Fig. 2.3 *Essential signal processing for strapdown INS.*

2.2.3 Strapdown Systems

2.2.3.1 Sensor Cluster In strapdown systems, the inertial sensor cluster is "strapped down" to the frame of the host vehicle, without using intervening gimbals for rotational isolation, as illustrated in Fig. 2.1*a*. The system computer must then integrate the full (six-degree-of-freedom) equations of motion.

2.2.3.2 Signal Processing The major software functions performed by navigation computers for strapdown systems are shown in block form in Fig. 2.3. The additional processing functions, beyond those required for gimbaled inertial navigation, include the following:

1. The blocks labeled "Coordinate transformation update" and "Acceleration coordinate transformation" in Fig. 2.3, which essentially take the place of the gimbal servo loops in Fig. 2.2. In effect, the strapdown software maintains *virtual gimbals* in the form of a coordinate transformation from the unconstrained, body-fixed sensor coordinates to the equivalent sensor coordinates of an inertial platform.
2. Attitude rate compensation for accelerometers, which was not required for gimbaled systems but may be required for some applications of strapdown systems. The gyroscopes and gimbals of a gimbaled IMU were used to isolate the accelerometers from body rotation rates, which can introduce errors such as centrifugal accelerations in rotating accelerometers.

2.3 SATELLITE NAVIGATION

The GPS is widely used in navigation. Its augmentation with other space-based satellites is the future of navigation.

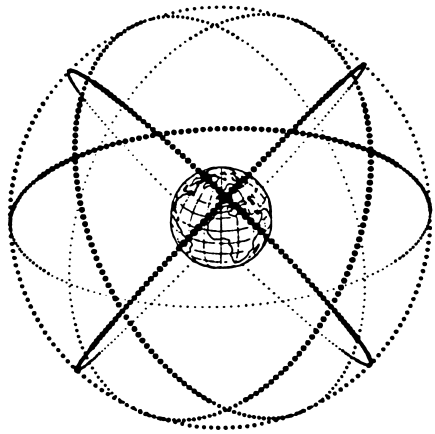


Fig. 2.4 GPS orbit planes.

2.3.1 Satellite Orbits

GPS satellites occupy six orbital planes inclined 55° from the equatorial plane, as illustrated in Fig. 2.4, with four or more satellites per plane, as illustrated in Fig. 2.5.

2.3.2 Navigation Solution (Two-Dimensional Example)

Receiver location in two dimensions can be calculated by using range measurements [45].

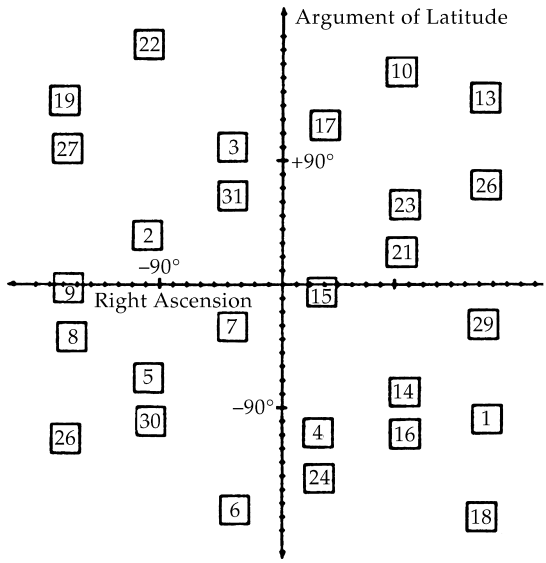


Fig. 2.5 GPS orbit phasing.

2.3.2.1 Symmetric Solution Using Two Transmitters on Land In this case, the receiver and two transmitters are located in the same plane, as shown in Fig. 2.6, with known positions x_1, y_1 and x_2, y_2. Ranges R_1 and R_2 of two transmitters from the user position are calculated as

$$R_1 = c\,\Delta T_1, \tag{2.1}$$

$$R_2 = c\,\Delta T_2 \tag{2.2}$$

where c = speed of light (0.299792458 m/ns)
ΔT_1 = time taken for the radio wave to travel from transmitter 1 to the user
ΔT_2 = time taken for the radio wave to travel from transmitter 2 to the user
(X, Y) = user position

The range to each transmitter can be written as

$$R_1 = [(X - x_1)^2 + (Y - y_1)^2]^{1/2}, \tag{2.3}$$

$$R_2 = [(X - x_2)^2 + (Y - y_2)^2]^{1/2}. \tag{2.4}$$

Expanding R_1 and R_2 in Taylor series expansion with small perturbation in X by Δx and Y by Δy, yields

$$\Delta R_1 = \frac{\partial R_1}{\partial X}\Delta x + \frac{\partial R_1}{\partial Y}\Delta y + u_1, \tag{2.5}$$

$$\Delta R_2 = \frac{\partial R_2}{\partial X}\Delta x + \frac{\partial R_2}{\partial Y}\Delta y + u_2, \tag{2.6}$$

where u_1 and u_2 are higher order terms. The derivatives of Eqs. 2.3 and 2.4 with respect to X, Y are substituted into Eqs. 2.5 and 2.6, respectively.

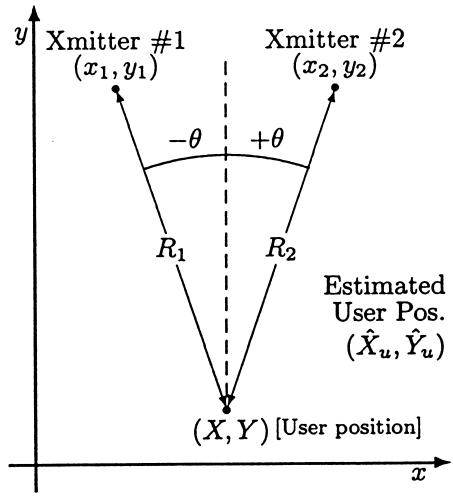


Fig. 2.6 *Two transmitters with known two-dimensional positions.*

Thus, for the symmetric case,

$$\Delta R_1 = \frac{X - x_1}{[(X - x_1)^2 + (Y - y_1)^2]^{1/2}} \Delta x + \frac{Y - y_1}{[(X - x_1)^2 + (Y - y_1)^2]^{1/2}} \Delta y + u_1, \quad (2.7)$$

$$= \sin\theta \Delta x + \cos\theta \Delta y + u_1, \quad (2.8)$$

$$\Delta R_2 = -\sin\theta \Delta x + \cos\theta \Delta y + u_2. \quad (2.9)$$

To obtain the least-squares estimate of (X, Y), we need to minimize the quantity

$$J = u_1^2 + u_2^2, \quad (2.10)$$

which is

$$J = \left(\underbrace{\Delta R_1 - \sin\theta \Delta x - \cos\theta \Delta y}_{u_1} \right)^2 + \left(\underbrace{\Delta R_2 + \sin\theta \Delta x - \cos\theta \Delta y}_{u_2} \right)^2. \quad (2.11)$$

The solution for the minimum can be found by setting $\partial J / \partial \Delta x = 0 = \partial J \partial \Delta y$, then solving for Δx and Δy:

$$0 = \frac{\partial J}{\partial \Delta x} \quad (2.12)$$

$$= 2(\Delta R_1 - \sin\theta \Delta x - \cos\theta \Delta y)(-\sin\theta) + 2(\Delta R_2 + \sin\theta \Delta x - \cos\theta \Delta y)(\sin\theta) \quad (2.13)$$

$$= \Delta R_2 - \Delta R_1 + 2\sin\theta \Delta x, \quad (2.14)$$

with solution

$$\Delta x = \frac{\Delta R_1 - \Delta R_2}{2\sin\theta}. \quad (2.15)$$

The solution for Δy may be found in similar fashion as

$$\Delta y = \frac{\Delta R_1 + \Delta R_2}{2\cos\theta}. \quad (2.16)$$

Navigation Solution Procedure Transmitter positions x_1, y_1, x_2, y_2 are given. Signal travel times ΔT_1, ΔT_2 are given. Estimated user position $\hat{X}_u$, $\hat{Y}_u$ are assumed.

Set position coordinates X, Y equal to their initial estimates:

$$X = \hat{X}_u, \qquad Y = \hat{Y}_u.$$

Compute the range errors,

$$\Delta R_1 = \overbrace{[(\hat{X}_u - x_1)^2 + (\hat{Y}_u - y_1)^2]^{1/2}}^{\text{Geometric ranges}} - \overbrace{C\Delta T_1}^{\text{Measured pseudoranges}}, \tag{2.17}$$

$$\Delta R_2 = [(\hat{X}_u - x_2)^2 + (\hat{Y}_u - y_2)^2]^{1/2} - C\Delta T_2. \tag{2.18}$$

Compute the theta angle,

$$\theta = \tan^{-1}\frac{x_1}{y_1} \tag{2.19}$$

or

$$\theta = \sin^{-1}\frac{x_1}{\sqrt{x_1^2 + y_1^2}}. \tag{2.20}$$

Compute user position corrections,

$$\Delta x = \frac{1}{2\sin\theta}(\Delta R_1 - \Delta R_2), \tag{2.21}$$

$$\Delta y = \frac{1}{2\cos\theta}(\Delta R_1 + \Delta R_2). \tag{2.22}$$

Compute a new estimate of position,

$$X = \hat{X}_u + \Delta x, \qquad Y = \hat{Y}_u + \Delta y. \tag{2.23}$$

Results are shown in Fig. 2.7:

Correction equations	Iteration equations
$\Delta X_{\text{best}} = \frac{1}{2\sin\theta}(\Delta R_1 - \Delta R_2),$	$X_{\text{new}} = X_{\text{old}} + \Delta X_{\text{best}},$
$\Delta Y_{\text{best}} = \frac{1}{2\cos\theta}(\Delta R_1 + \Delta R_2),$	$Y_{\text{new}} = Y_{\text{old}} + \Delta Y_{\text{best}}.$

2.3.3 Satellite Selection and Dilution of Precision

Just as in a land-based system, better accuracy is obtained by using reference points well separated in space. For example, the range measurements made to four reference points clustered together will yield nearly equal values. Position calculations involve range differences, and where the ranges are nearly equal, small relative errors are greatly magnified in the difference. This effect, brought about as a result of satellite geometry is known as *dilution of precision* (DOP). This means that range errors that occur from other causes such as clock errors are also magnified by the geometric effect.

To find the best locations of the satellites to be used in the calculations of the user position and velocity, the dilution of precision calculations (DOP) are needed.

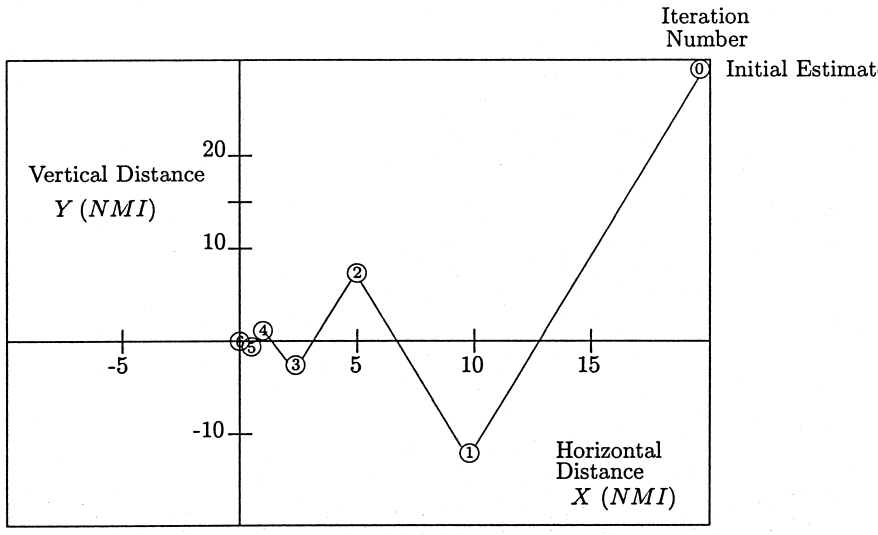


Fig. 2.7 *Results of the iteration.*

The observation equations in three dimensions for each satellite with known coordinates (x_i, y_i, z_i) and unknown user coordinates (X, Y, Z) are given by

$$Z_{\rho i} = \text{range} = \sqrt{(x_i - X)^2 + (y_i - Y)^2 + (z_i - Z)^2}. \tag{2.24}$$

These are nonlinear equations that can be linearized using Taylor series. (See, e.g., Chapter 5 of [46].)

Let the vector of ranges be $Z_\rho = \mathbf{h}(\mathbf{x})$, a nonlinear function $\mathbf{h}(\mathbf{x})$ of the four-dimensional vector $\mathbf{x}$ representing user position and receiver clock bias, and expand the left-hand side of this equation in a Tayor series about some nominal solution $\mathbf{x}^{\text{nom}}$ for the unknown vector

$$\mathbf{x} = [x^1 x^2 x^3 x^4]^{\mathrm{T}} \tag{2.25}$$

of variables

$x^1 \stackrel{\text{def}}{=}$ east component of the user's antenna location
$x^2 \stackrel{\text{def}}{=}$ north component of the user's antenna location
$x^3 \stackrel{\text{def}}{=}$ upward vertical component of the user's antenna location
$x^4 \stackrel{\text{def}}{=}$ receiver clock bias (C_b)

for which

$$
\begin{aligned}
Z_\rho &= \mathbf{h}(\mathbf{x}) = \mathbf{h}(\mathbf{x}^{\text{nom}}) + \left.\frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^{\text{nom}}} \delta\mathbf{x} + \text{H.O.T.},\\
\delta\mathbf{x} &= \mathbf{x} - \mathbf{x}^{\text{nom}}, \qquad \delta Z_\rho = \mathbf{h}(\mathbf{x}) - \mathbf{h}(\mathbf{x}^{\text{nom}}).
\end{aligned}
\tag{2.26}
$$

where H.O.T. is higher order term

These equations become

$$
\begin{aligned}
\delta Z_\rho &= \frac{\partial \mathbf{h}\ h(\mathbf{x})}{\partial \mathbf{x}}|_{\mathbf{x}=\mathbf{x}^{\text{nom}}}\delta\mathbf{x},\\
&= H^{[1]}\delta\mathbf{x},\\
\delta_x &= X - X_{\text{nom}}, \qquad \delta_y = Y - Y_{\text{nom}}, \qquad \delta_z = Z - Z_{\text{nom}},
\end{aligned}
\tag{2.27}
$$

where $H^{[1]}$ is the first-order term in the Taylor series expansion

$$
\begin{aligned}
\delta Z_\rho &= \rho(X, Y, Z) - \rho_r(X_{\text{nom}}, Y_{\text{nom}}, Z_{\text{nom}})\\
&\approx \underbrace{\frac{\partial \rho_r}{\partial X}|_{X_{\text{nom}}, Y_{\text{nom}}, Z_{\text{nom}}}}_{H^{[1]}}\delta\mathbf{x} + v_\rho
\end{aligned}
\tag{2.28}
$$

for $v_\rho =$ noise in receiver measurements. This vector equation can be written in scalar form where $i =$ satellite number as

$$
\begin{aligned}
\frac{\partial \rho_r^i}{\partial X} &= \frac{-(x_i - X)}{\sqrt{(x_i - X)^2 + (y_i - Y)^2 + (z_i - Z)^2}}|_{X=X_{\text{nom}}, Y_{\text{nom}}, Z_{\text{nom}}}\\
&= \frac{-(x_i - X_{\text{nom}})}{\sqrt{(x_i - X_{\text{nom}})^2 + (y_i - Y_{\text{nom}})^2 + (z_i - Z_{\text{nom}})^2}}\\
\frac{\partial \rho_r^i}{\partial Y} &= \frac{-(y_i - Y_{\text{nom}})}{\sqrt{(x_i - X_{\text{nom}})^2 + (y_i - Y_{\text{nom}})^2 + (z_i - Z_{\text{nom}})^2}}\\
\frac{\partial \rho_r^i}{\partial Z} &= \frac{-(z_i - Z_{\text{nom}})}{\sqrt{(x_i - X_{\text{nom}})^2 + (y_i - Y_{\text{nom}})^2 + (z_i - Z_{\text{nom}})^2}}
\end{aligned}
\tag{2.29}
$$

for

$$
i = 1, 2, 3, 4 \quad \text{(i.e., four satellites).} \tag{2.30}
$$

We can combine Eqs. 2.28 and 2.29 into the matrix equation

$$\underbrace{\begin{bmatrix}\delta z_\rho^1\\ \delta z_\rho^2\\ \delta z_\rho^3\\ \delta z_\rho^4\end{bmatrix}}_{4\times1} = \underbrace{\begin{bmatrix}\frac{\partial\rho_r^1}{\partial x} & \frac{\partial\rho_r^1}{\partial y} & \frac{\partial\rho_r^1}{\partial z} & 1\\ \frac{\partial\rho_r^2}{\partial x} & \frac{\partial\rho_r^2}{\partial y} & \frac{\partial\rho_r^2}{\partial z} & 1\\ \frac{\partial\rho_r^3}{\partial x} & \frac{\partial\rho_r^3}{\partial y} & \frac{\partial\rho_r^3}{\partial z} & 1\\ \frac{\partial\rho_r^4}{\partial x} & \frac{\partial\rho_r^4}{\partial y} & \frac{\partial\rho_r^4}{\partial z} & 1\end{bmatrix}}_{4\times4} \underbrace{\begin{bmatrix}\delta x\\ \delta y\\ \delta z\\ C_b\end{bmatrix}}_{4\times1} + \underbrace{\begin{bmatrix}v_\rho^1\\ v_\rho^2\\ v_\rho^3\\ v_\rho^4\end{bmatrix}}_{4\times1},$$

which we can write in symbolic form as

$$\overbrace{\delta Z_\rho}^{4\times1} = \overbrace{H^{[1]}}^{4\times4}\ \overbrace{\delta\mathbf{x}}^{4\times1} + \overbrace{v_k}^{4\times1}.$$

(See Table 5.3 in [46].)

To calculate $H^{[1]}$, one needs satellite positions and the nominal value of the user's position,

$$\delta Z_\rho = H^{[1]}\delta\mathbf{x} + v_\rho.$$

To calculate the geometric dilution of precision (GDOP) (approximately),

$$\overbrace{\delta Z_\rho}^{4\times1} = \overbrace{H^{[1]}}^{4\times4}\ \overbrace{\delta\mathbf{x}}^{4\times1}. \tag{2.31}$$

Known are δZ_ρ and $H^{[1]}$ from the pseudorange, satellite position, and nominal value of the user's position. The correction $\delta\mathbf{x}$ is the unknown vector.

If we premultiply both sides of Eq. 2.31 by $H^{[1]\mathrm{T}}$, the result will be

$$H^{[1]\mathrm{T}}\ \delta Z_\rho = \underbrace{\overbrace{H^{[1]\mathrm{T}}}^{4\times4}\ \overbrace{H^{[1]}}^{4\times4}}_{4\times4}\delta\mathbf{x} \tag{2.32}$$

Then we premultiply Eq. 2.32 by $(H^{[1]\mathrm{T}}\ H^{[1]})^{-1}$,

$$\delta x = (H^{[1]\mathrm{T}}\ H^{[1]})^{-1}H^{[1]\mathrm{T}}\ \delta Z_\rho. \tag{2.33}$$

If $\delta\mathbf{x}$ and δZ_ρ are assumed random with zero mean, the error covariance

$$E\langle\delta\mathbf{x}(\delta\mathbf{x})^{\mathrm{T}}\rangle \tag{2.34}$$

$$= E\langle(H^{[1]\mathrm{T}}\,H^{[1]})^{-1}H^{[1]\mathrm{T}}\delta Z_\rho[(H^{[1]\mathrm{T}}\,H^{[1]})^{-1}H^{[1]\mathrm{T}}\,\delta Z_\rho]^{\mathrm{T}}\rangle \tag{2.35}$$

$$= (H^{[1]\mathrm{T}}\,H^{[1]})^{-1}H^{[1]\mathrm{T}}\underbrace{E\langle\delta Z_\rho\delta Z_\rho^{\mathrm{T}}\rangle}H^{[1]}(H^{[1]\mathrm{T}}\,H^{[1]})^{-1}. \tag{2.36}$$

The pseudorange measurement covariance is assumed uncorrelated satellite-to-satellite with variance (σ^2):

$$\underbrace{E\langle\delta Z_\rho\delta Z_\rho^{\mathrm{T}}\rangle = \sigma^2\mathbf{I}}_{4\times4} \tag{2.37}$$

Substituting Eq. 2.37 into Eq. 2.35 gives

$$\begin{aligned} E\langle\delta\mathbf{x}(\delta\mathbf{x})^{\mathrm{T}}\rangle &= \sigma^2(H^{[1]\mathrm{T}}\,H^{[1]})^{-1}\underbrace{(H^{[1]\mathrm{T}}\,H^{[1]})(H^{[1]\mathrm{T}}\,H^{[1]})^{-1}}_{\mathbf{I}} \\ &= \sigma^2(H^{[1]\mathrm{T}}\,H^{[1]})^{-1} \end{aligned} \tag{2.38}$$

for

$$\overbrace{\delta\mathbf{x}}^{4\times1} = \begin{bmatrix} \Delta E \\ \Delta N \\ \Delta U \\ C_b \end{bmatrix}$$

and

$$\begin{matrix} E = \text{east} \\ N = \text{north} \\ U = \text{up} \end{matrix} \qquad \begin{pmatrix} \text{locally} \\ \text{level} \\ \text{coordinate} \\ \text{frame} \end{pmatrix}$$

the covariance matrix becomes

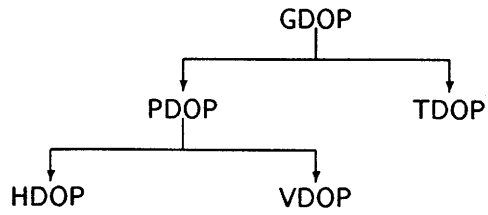
$$\underbrace{E\langle\delta\mathbf{x}(\delta\mathbf{x})^{\mathrm{T}}\rangle}_{4\times4} = \begin{bmatrix} E(\Delta E)^2 & E(\Delta E\Delta N) & E(\Delta E\Delta U) & E(\Delta E\Delta Cb) \\ E(\Delta N\Delta E) & E(\Delta N)^2 & \vdots & \vdots \\ \ddots & & E(\Delta U)^2 & \vdots \\ & & & E(C_b)^2 \end{bmatrix}. \tag{2.39}$$

We are principally interested in the diagonal elements of

$$(H^{[1]\mathrm{T}}\,H^{[1]})^{-1} = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix}, \tag{2.40}$$

which are

$$\begin{aligned}
\text{GDOP} &= \sqrt{A_{11} + A_{22} + A_{33} + A_{44}} && \text{(geometric DOP)} \\
\text{PDOP} &= \sqrt{A_{11} + A_{22} + A_{33}} && \text{(position DOP)} \\
\text{HDOP} &= \sqrt{A_{11} + A_{22}} && \text{(horizontal DOP)} \\
\text{VDOP} &= \sqrt{A_{33}} && \text{(vertical DOP)} \\
\text{TDOP} &= \sqrt{A_{44}} && \text{(time DOP)}
\end{aligned}$$



The unscaled covariance matrix $H^{[1]\mathrm{T}}H^{[1]}$ then has the form

$$\begin{bmatrix}
(\text{east DOP})^2 & & & \text{covariance terms} \\
& (\text{north DOP})^2 & & \\
& & (\text{vertical DOP})^2 & \\
& \text{covariance terms} & & (\text{time DOP})^2
\end{bmatrix},$$

where

$$\text{GDOP} = \sqrt{\text{trace}(H^{[1]\mathrm{T}}\, H^{[1]})^{-1}}\,, \; H^{[1]} = \left.\frac{\partial \rho}{\partial x}\right|_{X_{\text{nom}}, Y_{\text{nom}}, Z_{\text{nom}}}.$$

2.3.4 Typical Calculation of GDOP

2.3.4.1 Four Satellites The best accuracy is found with three satellites equally spaced on the horizon, at minimum elevation angle, with the fourth satellite directly overhead:

	By Satellite Location			
	1	2	3	4
Elevation (deg)	5	5	5	90
Azimuth (deg)	0	120	240	0

Typical example values of $H^{[1]}$ for this geometry are

$$H^{[1]} = \begin{bmatrix} 0.0 & 0.996 & 0.087 & 1.0 \\ 0.863 & -0.498 & 0.087 & 1.0 \\ -0.863 & -0.498 & 0.087 & 1.0 \\ 0.0 & 0.0 & 1.00 & 1.0 \end{bmatrix}.$$

The GDOP calculations for this example are

$$\underbrace{(H^{[1]\mathrm{T}} H^{[1]})^{-1}}_{4\times4} = \begin{bmatrix} 0.672 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.672 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.60 & -0.505 \\ 0.0 & 0.0 & -0.505 & 0.409 \end{bmatrix},$$

$$\begin{aligned} \mathrm{GDOP} &= \sqrt{0.672 + 0.672 + 1.6 + 0.409} \\ &= 1.83, \\ \mathrm{HDOP} &= 1.16, \\ \mathrm{VDOP} &= 1.26, \\ \mathrm{PDOP} &= 1.72, \\ \mathrm{TDOP} &= 0.64. \end{aligned}$$

2.4 TIME AND GPS

2.4.1 Coordinated Universal Time Generation

Coordinated Universal Time (UTC) is the time scale based on the atomic second, but occasionally corrected by the insertion of leap seconds, so as to keep it approximately synchronized with the earth's rotation. The leap second adjustments keep UTC within 0.9 s of UT1, which is a time scale based on the earth's axial spin. UT1 is a measure of the true angular orientation of the earth in space. Because the earth does not spin at exactly a constant rate, UT1 is not a uniform time scale [3].

2.4.2 GPS System Time

The time scale to which GPS signals are referenced is referred to as GPS time. GPS time is derived from a composite or "paper" clock that consists of all operational monitor station and satellite atomic clocks. Over the long run, it is steered to keep it within about 1 μs of UTC, as maintained by the master clock at the U.S. Naval Observatory, ignoring the UTC leap seconds. At the integer second level, GPS time equalled UTC in 1980. However, due to the leap seconds that have been inserted into UTC, GPS time was ahead of UTC by 10 s in April 2000.

2.4.3 Receiver Computation of UTC

The parameters needed to calculate UTC from GPS time are found in subframe 4 of the navigation data message. This data includes a notice to the user regarding the scheduled future or recent past (relative to the navigation message upload) value of the delta time due to leap seconds Δt_{LSF}, together with the week number $\mathrm{WN}_{\mathrm{LSF}}$ and the day number DN at the end of which the leap second becomes effective. The latter two quantities are known as the *effectivity time* of the leap second. "Day one" is defined as the first day relative to the end/start of a week and the $\mathrm{WN}_{\mathrm{LSF}}$ value consists of the eight least significant bits (LSBs) of the full week number.

Three different UTC/GPS time relationships exist, depending on the relationship of the effectivity time to the user's current GPS time:

1. *First Case*. Whenever the effectivity time indicated by the $\mathrm{WN}_{\mathrm{LSF}}$ and WN values is not in the past relative to the user's present GPS time, *and* the user's present time does not fall in the timespan starting at $\mathrm{DN} + \frac{3}{4}$ and ending at $\mathrm{DN} + \frac{5}{4}$, the UTC time is calculated as:

$$t_{\mathrm{UTC}} = (t_E - \Delta t_{\mathrm{UTC}}) \quad (\text{modulo } 86400)\text{s} \tag{2.41}$$

   where t_{UTC} is in seconds and

$$\Delta t_{\mathrm{UTC}} = \Delta t_{\mathrm{LS}} + A_0 + A_1[t_E - t_{0t} + 60{,}4800(\mathrm{WN} - \mathrm{WN}_t)]\text{s}, \tag{2.42}$$

   where t_E = user GPS time from start of week (s)
   Δt_{LS} = the delta time due to leap seconds
   A_0 = a constant polynomial term from the ephemeris message
   A_1 = a first-order polynomial term from the ephemeris message
   t_{0t} = reference time for UTC data
   WN = current week number derived from subframe 1
   WN_t = UTC reference week number

   The user GPS time t_E is in seconds relative to the end/start of the week, and the reference time t_{0t} for UTC data is referenced to the start of that week, whose number WN_t is given in word eight of page 18 in subframe 4. The WN_t value consists of the eight LSBs of the full week number. Thus, the user must account for the truncated nature of this parameter as well as truncation of WN, WN_t and $\mathrm{WN}_{\mathrm{LSF}}$ due to rollover of the full week number. These parameters are managed by the GPS control segment so that the absolute value of the difference between the untruncated WN and WN_t values does not exceed 127.

2. *Second Case*. Whenever the user's current GPS time falls within the timespan of $\mathrm{DN} + \frac{3}{4}$ to $\mathrm{DN} + \frac{5}{4}$, proper accommodation of the leap second event with a possible week number transition is provided by the following expression for UTC:

$$t_{\mathrm{UTC}} = W \quad [\text{modulo } (86{,}400 + \Delta t_{\mathrm{LSF}} - \Delta t_{\mathrm{LS}})] \qquad \text{s}, \tag{2.43}$$

where

$$W = (t_E - \Delta t_{\text{UTC}} - 43{,}200) \quad (\text{modulo } 86{,}400) + 43{,}200 \text{ s}, \tag{2.44}$$

and the definition of Δt_{UTC} previously given applies throughout the transition period.

3. *Third Case*. Whenever the effectivity time of the leap second event, as indicated by the WN_{LSF} and DN values, is in the past relative to the user's current GPS time, the expression given for t_{UTC} in the first case above is valid except that the value of Δt_{LSF} is used instead of Δt_{LS}. The GPS control segment coordinates the update of UTC parameters at a future upload in order to maintain a proper continuity of the t_{UTC} time scale.

2.5 USER POSITION CALCULATIONS WITH NO ERRORS

With the position of the satellites known, this section will discuss how to calculate the range (pseudorange) with no errors, including clock bias. Additional errors are receiver errors, selective availability, clock errors, and ionospheric errors.

Neglecting clock errors, let us first determine position calculation with no errors:

$$\begin{aligned} \rho_r &= \text{pseudorange (known)}, \\ x, y, z &= \text{satellite position coordinates (known)}, \\ X, Y, Z &= \text{user position coordinates (unknown)}, \end{aligned}$$

x, y, z and X, Y, Z are in the earth-centered, earth-fixed (ECEF) coordinate system.

Position calculation with no errors gives

$$\rho_r = \sqrt{(x - X)^2 + (y - Y)^2 + (z - Z)^2}. \tag{2.45}$$

Squaring both sides yields

$$\begin{aligned} \rho_r^2 &= (x - X)^2 + (y - Y)^2 + (z - Z)^2 \\ &= \underbrace{X^2 + Y^2 + Z^2}_{r^2} + x^2 + y^2 + z^2 - 2Xx - 2Yy - 2Zz, \end{aligned} \tag{2.46}$$

$$\rho_r^2 - (x^2 + y^2 + z^2) - r^2 = \text{Crr} - 2Xx - 2Yy - 2Zz, \tag{2.47}$$

where r = radius of earth

Crr = clock bias correction

The four unknowns are (X, Y, Z, Crr). Satellite position (x, y, z) is calculated from ephemeris data.

For four satellites, Eq. 2.47 becomes

$$\begin{aligned}
\rho_{r_1}^2 - (x_1^2 + y_1^2 + z_1^2) - r^2 &= \mathrm{Crr} - 2Xx_1 - 2Yy_1 - 2Zz_1, \\
\rho_{r_2}^2 - (x_2^2 + y_2^2 + z_2^2) - r^2 &= \mathrm{Crr} - 2Xx_2 - 2Yy_2 - 2Zz_2, \\
\rho_{r_3}^2 - (x_3^2 + y_3^2 + z_3^2) - r^2 &= \mathrm{Crr} - 2Xx_3 - 2Yy_3 - 2Zz_3, \\
\rho_{r_4}^2 - (x_4^2 + y_4^2 + z_4^2) - r^2 &= \mathrm{Crr} - 2Xx_4 - 2Yy_4 - 2Zz_4,
\end{aligned} \tag{2.48}$$

with unknown 4×1 state vector

$$\begin{bmatrix} X \\ Y \\ Z \\ \mathrm{Crr} \end{bmatrix}.$$

We can rewrite the four equations in matrix form as

$$\begin{bmatrix} \rho_{r_1}^2 - (x_1^2 + y_1^2 + z_1^2) - r^2 \\ \rho_{r_2}^2 - (x_2^2 + y_2^2 + z_2^2) - r^2 \\ \rho_{r_3}^2 - (x_3^2 + y_3^2 + z_3^2) - r^2 \\ \rho_{r_4}^2 - (x_4^2 + y_4^2 + z_4^2) - r^2 \end{bmatrix} = \begin{bmatrix} 2x_1 - 2y_1 - 2z_1 & 1 \\ 2x_2 - 2y_2 - 2z_2 & 1 \\ 2x_3 - 2y_3 - 2z_3 & 1 \\ 2x_4 - 2y_4 - 2z_4 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ \mathrm{Crr} \end{bmatrix}$$

or

$$\overbrace{R}^{4\times1} = \overbrace{M}^{4\times4}\ \overbrace{U_\rho}^{4\times1}, \tag{2.49}$$

where R = vector (known)
M = matrix (known)
U_ρ = vector (unknown)

Then we premultiply both sides of Eq. 2.49 by M^{-1}:

$$\begin{aligned}
M^{-1}R &= M^{-1}MU_\rho \\
&= U_\rho \\
&= \begin{bmatrix} X \\ Y \\ Z \\ \mathrm{Crr} \end{bmatrix}.
\end{aligned}$$

If the rank of M (defined in Section B.5.2), the number of linearly independent columns of the matrix M, is less than 4, then M will not be invertible. In that case, its determinant (defined in Section B.6.1) is given as

$$\det M = |M| = 0.$$

2.6 USER VELOCITY CALCULATION WITH NO ERRORS

The governing equation in this case is

$$\dot{\rho}_r = [(x - X)(\dot{x} - \dot{X}) + (y - Y)(\dot{y} - \dot{Y}) + (z - Z)(\dot{z} - \dot{Z})]/\rho_r \tag{2.50}$$

where $\dot{\rho}_r$ = range rate (known)
ρ_r = range (known)
(x, y, z) = satellite positions (known)
$(\dot{x}, \dot{y}, \dot{z})$ = satellite rate (known)
(X, Y, Z) = user position (known from position calculations)
$(\dot{X}, \dot{Y}, \dot{Z})$ = user velocity (unknown)

$$\dot{\rho}_r + \frac{1}{\rho_r}[\dot{x}(x - X) + \dot{y}(y - Y) + \dot{z}(z - Z)] = \left(\frac{x - X}{\rho_r}\dot{X} + \frac{y - Y}{\rho_r}\dot{Y} + \frac{z - Z}{\rho_r}\dot{Z}\right). \tag{2.51}$$

For three satellites

$$\begin{bmatrix} \dot{\rho}_{r_1} + \dfrac{1}{\rho_{r_1}}[\dot{x}_1(x_1 - X) + \dot{y}_1(y_1 - Y) + \dot{z}_1(z_1 - Z)] \\ \dot{\rho}_{r_2} + \dfrac{1}{\rho_{r_2}}[\dot{x}_2(x_2 - X) + \dot{y}_2(y_2 - Y) + \dot{z}_2(z_2 - Z)] \\ \dot{\rho}_{r_3} + \dfrac{1}{\rho_{r_3}}[\dot{x}_3(x_3 - X) + \dot{y}_3(y_3 - Y) + \dot{z}_3(z_3 - Z)] \end{bmatrix}$$
$$= \begin{bmatrix} \dfrac{(x_1 - X)}{\rho_{r_1}} & \dfrac{(y_1 - Y)}{\rho_{r_1}} & \dfrac{(z_1 - Z)}{\rho_{r_1}} \\ \dfrac{(x_2 - X)}{\rho_{r_2}} & \dfrac{(y_2 - Y)}{\rho_{r_2}} & \dfrac{(z_2 - Z)}{\rho_{r_2}} \\ \dfrac{(x_3 - X)}{\rho_{r_3}} & \dfrac{(y_3 - Y)}{\rho_{r_3}} & \dfrac{(z_3 - Z)}{\rho_{r_3}} \end{bmatrix} \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix}, \tag{2.52}$$

$$\overbrace{D_R}^{3\times 1} = \overbrace{N}^{3\times 3}\ \overbrace{U_v}^{3\times 1}, \tag{2.53}$$

$$\overbrace{U_v}^{3\times 1} = N^{-1} D_R. \tag{2.54}$$

However, if the rank of N (defined on Section B.5.2) is <3, N will not be invertible.

Problems

Refer to Eq. C.103 and C.104 in Appendix C for satellite orbit equations.

2.1 For the following GPS satellites, find the satellite position in ECEF coordinates at $t = 3$ s. (*Hint*: see Appendix C.)

	Ω_0 (deg)	θ_0 (deg)
(a)	326	68
(b)	26	34

2.2 Using the results of Problem 2.1, find the satellite positions in the local reference frame. Reference should be to the COMSAT facility in Santa Paula, California, located at 32.4° latitude, −119.2° longitude. Use coordinate shift matrix $S = 0$. (Ref. to Appendix C, section C.3.9.)

2.3 Given the following GPS satellites:

	Ω_0 (deg)	θ_0 (deg)	ρ (m)
Satellite 1	326	68	2.324e7
Satellite 2	26	340	2.0755e7
Satellite 3	146	198	2.1103e7
Satellite 4	86	271	2.3491e7

(a) Find the user's position in ECEF coordinates.

(b) Find the user's position in locally level coordinates referenced to 0° latitude, 0° longitude. Coordinate shift matrix $S = 0$.

2.4 Given two satellites in north and east coordinates,

$$x(1) = 6.1464\text{e}06 \qquad y(1) = 2.0172\text{e}07 \qquad \text{in meters,}$$
$$x(2) = 6.2579\text{e}06 \qquad y(2) = -7.4412\text{e}06 \qquad \text{in meters,}$$

with

$$c\,\Delta t(1) = \rho_r(1) = 2.324\text{e}07 \qquad \text{in meters,}$$
$$c\,\Delta t(2) = \rho_r(2) = 2.0755\text{e}07 \qquad \text{in meters,}$$

and starting with an initial guess of $(x_{\text{est}}, y_{\text{est}})$, find the user's position.

Global Positioning Systems, Inertial Navigation, and Integration,
Mohinder S. Grewal, Lawrence R. Weill, Angus P. Andrews

Print ISBN 0-471-35032-X Electronic ISBN 0-471-20071-9

3

Signal Characteristics and Information Extraction

Why is the GPS signal so complex? GPS was designed to be readily accessible to millions of military and civilian users. Therefore, it is a receive-only passive system for a user, and the number of users that can simultaneously use the system is unlimited. Because there are many functions that must be performed, the GPS signal has a rather complex structure. As a consequence, there is a correspondingly complex sequence of operations that a GPS receiver must carry out in order to extract desired information from the signal. In this chapter we characterize the signal mathematically, describe the purposes and properties of the important signal components, and discuss generic methods for extracting information from these components.

3.1 MATHEMATICAL SIGNAL WAVEFORM MODELS

Each GPS satellite simultaneously transmits on two L-band frequencies denoted by L_1 and L_2, which are 1575.42 and 1227.60 MHz, respectively. The carrier of the L_1 signal consists of an in-phase and a quadrature-phase component. The in-phase component is biphase modulated by a 50-bps data stream and a pseudorandom code called the C/A-code consisting of a 1023-chip sequence that has a period of 1 ms and a chipping rate of 1.023 MHz. The quadrature-phase component is also biphase modulated by the same 50-bps data stream but with a different pseudorandom code

called the P-code, which has a 10.23-MHz chipping rate and a one-week period. The mathematical model of the L_1 waveform is

$$s(t) = \sqrt{2P_I}d(t)c(t)\cos(\omega t + \theta) + \sqrt{2P_Q}d(t)p(t)\sin(\omega t + \theta), \tag{3.1}$$

where P_I and P_Q are the respective carrier powers for the in-phase and quadrature-phase carrier components, $d(t)$ is the 50-bps data modulation, $c(t)$ and $p(t)$ are the respective C/A and P pseudorandom code waveforms, ω is the L_1 carrier frequency in radians per second, and θ is a common phase shift in radians. The quadrature carrier power P_Q is approximately 3 dB less than P_I.

In contrast to the L_1 signal, the L_2 signal is modulated with only the 50-bps data and the P-code, although there is the option of not transmitting the 50-bps data stream. The mathematical model of the L_2 waveform is

$$s(t) = \sqrt{2P_Q}d(t)p(t)\sin(\omega t + \theta). \tag{3.2}$$

Figures 3.1 and 3.2 respectively show the structure of the in-phase and quadrature-phase components of the L_1 signal. The 50-bps data bit boundaries always occur at

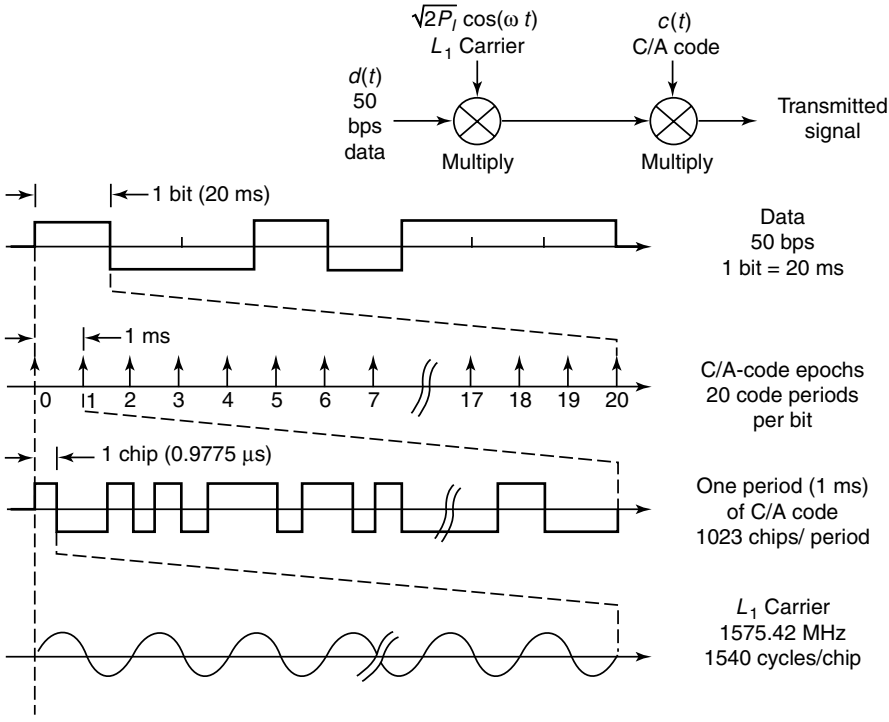


Fig. 3.1 *Structure of in-phase component of the L_1 signal.*

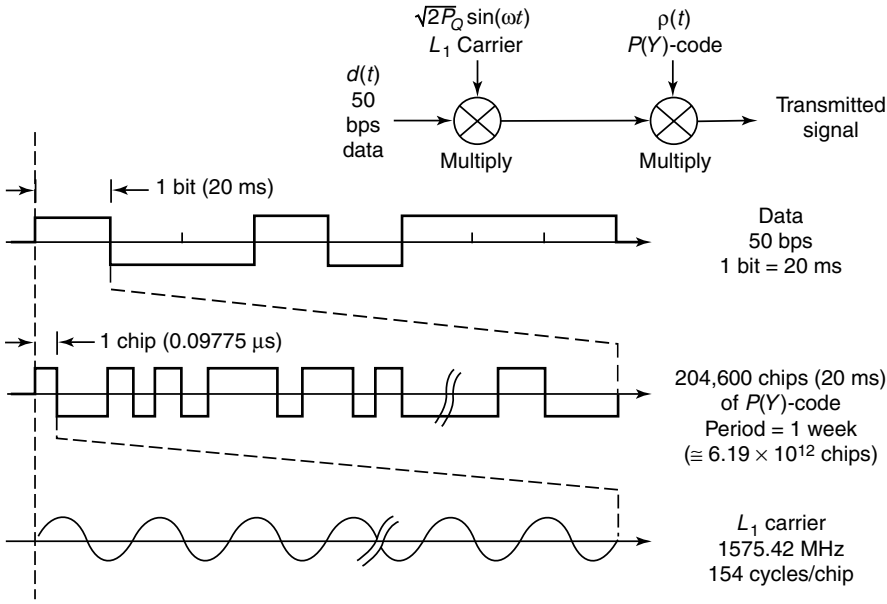


Fig. 3.2 *Structure of quadrature-phase component of the L_1 signal.*

an epoch of the C/A-code. The C/A-code epochs mark the beginning of each period of the C/A-code, and there are precisely 20 code epochs per data bit, or 20,460 C/A-code chips. Within each C/A-code chip there are precisely 1540 L_1 carrier cycles. In the quadrature-phase component of the L_1 signal there are precisely 204,600 P-code chips within each 50-bps data bit, and the data bit boundaries always coincide with the beginning of a P-code chip [42, 56].

3.2 GPS SIGNAL COMPONENTS, PURPOSES, AND PROPERTIES

3.2.1 50-bps Data Stream

The 50-bps data stream conveys the *navigation message*, which includes, but is not limited to, the following information:

1. *Satellite Almanac Data*. Each satellite transmits orbital data called the *almanac*, which enables the user to calculate the approximate location of every satellite in the GPS constellation at any given time. Almanac data is not accurate enough for determining position but can be stored in a receiver where it remains valid for many months. It is primarily used to determine which satellites are visible at a given location so that the receiver can search for those satellites when it is first turned on. It can also be used to determine the

approximate expected signal Doppler shift to aid in rapid acquisition of the satellite signals.

2. *Satellite Ephemeris Data*. Ephemeris data is similar to almanac data but enables a much more accurate determination of satellite position needed to convert signal propagation delay into an estimate of user position. In contrast to almanac data, ephemeris data for a particular satellite is only broadcast by that satellite, and the data is valid for only several hours.
3. *Signal Timing Data*. The 50-bps data stream includes time tagging, which is used to establish the transmission time of specific points on the GPS signal. This information is needed to determine the satellite-to-user propagation delay used for ranging.
4. *Ionospheric Delay Data*. Ranging errors due to ionospheric effects can be partially canceled by using estimates of ionospheric delay that are broadcast in the data stream.
5. *Satellite Health Message*. The data stream also contains information regarding the current health of the satellite, so that the receiver can ignore that satellite if it is not operating properly.

Structure of the Navigation Message The information in the navigation message has the basic frame structure shown in Fig. 3.3. A complete message consists of 25 frames, each containing 1500 bits. Each frame is subdivided into five 300-bit subframes, and each subframe consists of 10 words of 30 bits each, with the most significant bit (MSB) of the word transmitted first. Thus, at the 50-bps rate it takes 6 s to transmit a subframe and 30 s to complete one frame. Transmission of the complete 25-frame navigation message requires 750 s, or 12.5 min. Except for occasional updating, subframes 1, 2, and 3 are constant (i.e., repeat) with each frame at the 30-s frame repetition rate. On the other hand, subframes 4 and 5 are each subcommutated 25 times. The 25 versions of subframes 4 and 5 are referred to as pages 1–25. Hence, except for occasional updating, each of these pages repeats every 750 s, or 12.5 min.

A detailed description of all information contained in the navigation message is beyond the scope of this text. Therefore, we only give an overview of the fundamental elements. Each subframe begins with a *telemetry word* (TLM). The first 8 bits of the TLM is a preamble that makes it possible for the receiver to determine when a subframe begins. The remainder of the TLM contains parity bits and a telemetry message that is available only to authorized users and is not a fundamental item. The second word of each subframe is called the *hand-over word* (HOW).

Z-Count Information contained in the HOW is derived from a 29-bit quantity called the *Z-count*. The *Z*-count is not transmitted as a single word, but part of it is transmitted within the HOW. The *Z*-count counts *epochs* generated by the X_1 register of the P-code generator in the satellite, which occur every 1.5 s. The 19 LSBs of the *Z*-count, called the *time-of-week* (TOW) count, indicate the number of X_1 epochs

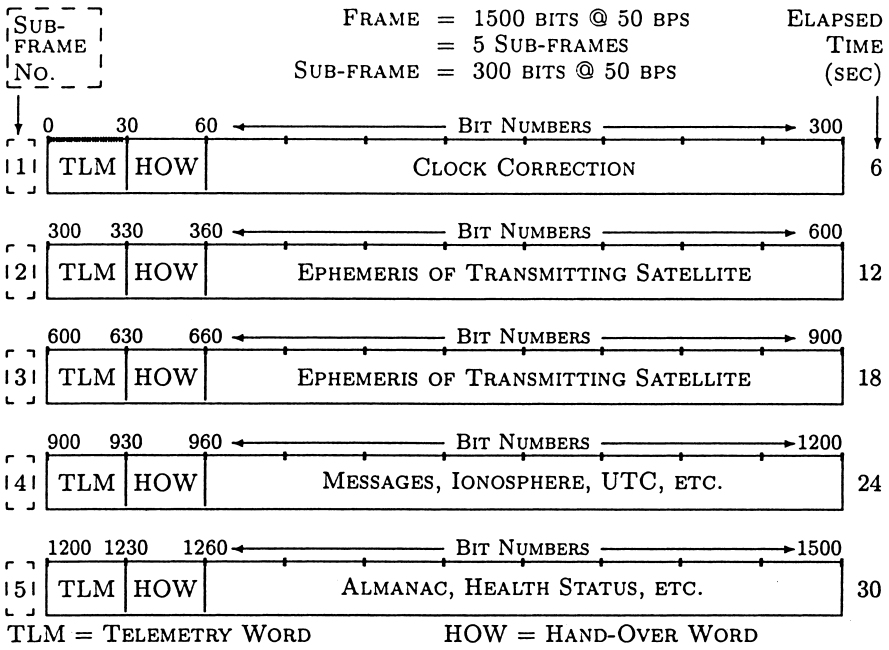


Fig. 3.3 *Navigation message frame structure.*

that have occurred since the start of the current week. The start of the current week occurs at the X_1 epoch, which occurs at approximately midnight of Saturday night/Sunday morning. The TOW count increases from zero at the start of the week to 403,199 and then rolls over to zero again at the start of the following week. A TOW count of zero always occurs at the beginning of subframe 1 of the first frame (the frame containing page 1 of subcommutated subframes 4 and 5). A truncated version of the TOW count, containing its 17 MSBs, comprises the first 17 bits of the HOW. Multiplication of this truncated count by 4 gives the TOW count at the start of the following subframe. Since the receiver can use the TLM preamble to determine precisely the time at which each subframe begins, a method for determining the time of transmission of any part of the GPS signal is thereby established. The relationship between the HOW counts and TOW counts is shown in Fig. 3.4.

GPS Week Number The 10 MSBs of the Z-count contain the GPS *week number* (WN), which is a modulo-1024 week count. The zero state is defined to be that week that started with the X_1 epoch occurring at approximately midnight on the night of January 5, 1980/morning of January 6, 1980. Because WN is a modulo-1024 count, an event called the *week rollover* occurs every 1024 weeks (a few months short of 20 years), and GPS receivers must be designed to accommodate it.[1] The WN is not part of the HOW but instead appears as the first 10 bits of the third word in subframe 1.

[1] The most recent rollover occurred at GPS time zero on August 22, 1999, with little difficulty.

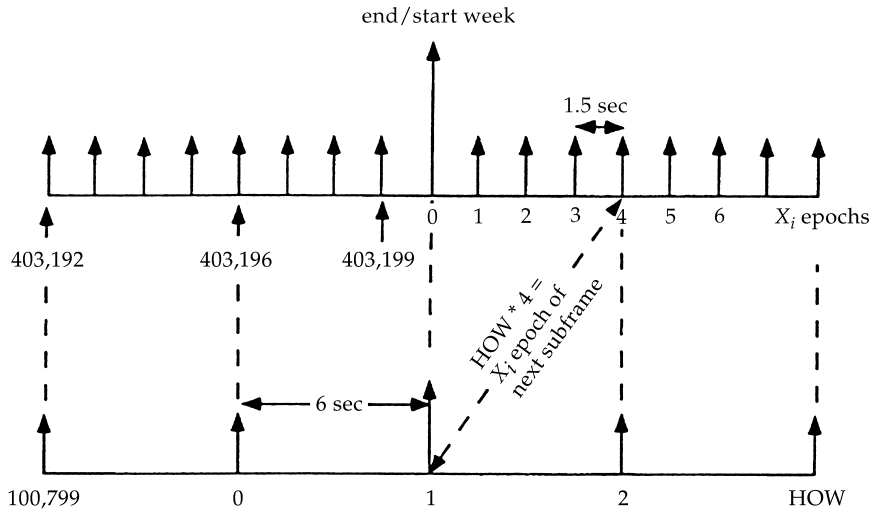


Fig. 3.4 *Relationship between HOW counts and TOW counts.*

Frame and Subframe Identification Three bits of the HOW are used to identify which of the five subframes is being transmitted. The frame being transmitted (corresponding to a page number from 1 to 25) can readily be identified from the TOW count computed from the HOW of subframe 5. This TOW count is the TOW at the start of the next frame. Since there are 20 TOW counts per frame, the frame number of that frame is simply (TOW/20) (mod 25).

Information by Subframe In addition to the TLM and HOW, which occur in every subframe, the following information is contained within the remaining eight words of subframes 1–5 (only fundamental information is described):

1. *Subframe 1*. The WN portion of the *Z*-count is part of word 3 in this subframe. Subframe 1 also contains GPS clock correction data for the satellite in the form of polynomial coefficients defining how the correction varies with time. Time defined by the clocks in the satellite is commonly called *SV time* (space vehicle time); the time after corrections have been applied is called *GPS time*. Thus, even though individual satellites may not have perfectly synchronized SV times, they do share a common GPS time. Additional information in subframe 1 includes the quantities t_{0c}, T_{GD}, and IODC. The clock reference time t_{0c} is used as a time origin to calculate satellite clock error, the ionospheric group delay T_{GD} is used to correct for ionospheric propagation delay errors, and IODC (issue of date, clock) indicates the issue number of the clock data set to alert users to changes in clock parameters.
2. *Subframes 2 and 3*. These subframes contain the ephemeris data, which is used to determine the precise satellite position and velocity required by the

navigation solution. Unlike the almanac data, this data is very precise, is valid over a relatively short period of time (several hours), and applies only to the satellite transmitting it. The components of the ephemeris data are listed in Table 3.1, and the algorithm that should be used to compute satellite position in WGS 84 coordinates is given in Table 3.2. The satellite position computation using these data is implemented in the Matlab m-file ephemeris.m on the accompanying diskette. The IODE (issue of date, ephemeris) informs users when changes in ephemeris parameters have occurred. Each time new parameters are uploaded from the GPS control segment, the IODE number changes.

3. *Subframe 4*. The 25 pages of this subframe contain the almanac for satellites with PRN (pseudorandom code) numbers 25 and higher, as well as special messages, ionospheric correction terms, and coefficients to convert GPS time to UTC time. There are also spare words for possible future applications. The components of an almanac are very similar to those of the ephemeris, and the calculation of satellite position is performed in essentially the same way.

Table 3.1 Components of Ephemeris Data

Name	Description	Units[a]
M_0	Mean anomaly at reference time	semicircle
Δn	Mean motion difference from computed value	semicircle/s
e	Eccentricity	dimensionless
$\sqrt{a}$	Square root of semimajor axis	$m^{1/2}$
Ω_0	Longitude of ascending node of orbit plane at weekly epoch	semicircle
i_0	Inclination angle at reference time	semicircle
ω	Argument of perigee	semicircle
$\dot{\Omega}$	Rate of right ascension	semicircle/s
IDOT	Rate of inclination angle	semicircle/s
C_{uc}	Amplitude of cosine harmonic correction term to the argument of latitude	rad
C_{us}	Amplitude of sine harmonic correction term to the argument of latitude	rad
C_{rc}	Amplitude of cosine harmonic correction term to the orbit radius	m
C_{rs}	Amplitude of sine harmonic correction term to the orbit radius	m
C_{ic}	Amplitude of cosine harmonic correction term to the angle of inclination	rad
C_{is}	Amplitude of sine harmonic correction term to the angle of inclination	rad
t_{0e}	Ephemeris reference time	s
IODE	Issue of data, ephemeris	dimensionless

[a] Units used in MATLAB m-file ephemeris are different.

Table 3.2 Algorithm for Computing Satellite Position

$\mu = 3.986005 \times 10^{14}\ \text{m}^3/\text{s}^2$	WGS 84 value of earth's universal gravitational parameter
$\dot{\Omega}_e = 7.292115167 \times 10^{-5}\ \text{rad/s}$	WGS 84 value of earth's rotation rate
$a = (\sqrt{a})^2$	Semimajor axis
$n_0 = \sqrt{\mu/a^3}$	Computed mean motion, rad/s
$t_k = t - t_{0e}$ [a]	Time from ephemeris reference epoch
$n = n_0 + \Delta_n$	Corrected mean motion
$M_k = M_0 + nt_k$	Mean anomaly
$M_k = E_k - e\sin E_k$	Kepler's equation for eccentric anomaly
$f_k = \cos^{-1}\left(\dfrac{\cos E_k - 1}{1 - e\cos E_k}\right)$	True anomaly from cosine
$f_k = \sin^{-1}\left(\dfrac{\sqrt{1-e^2}\sin E_k}{1 - e\cos E_k}\right)$	True anomaly from sine
$E_k = \cos^{-1}\left(\dfrac{e + \cos f_k}{1 + e\cos f_k}\right)$	Eccentric anomaly from cosine
$\phi_k = f_k + \omega$	Argument of latitude
$\delta\mu_k = C_{\mu c}\cos 2\phi_k + C_{\mu s}\sin 2\phi_k$	Second-harmonic correction to argument of latitude
$\delta r_k = C_{rc}\cos 2\phi_k + C_{rs}\sin 2\phi_k$	Second-harmonic correction to radius
$\delta i_k = C_{ic}\cos 2\phi_k + C_{is}\sin 2\phi_k$	Second-harmonic correction to inclination
$\mu_k = \phi_k + \delta\mu_k$	Corrected argument of latitude
$r_k = a(1 - e\cos E_k) + \delta r_k$	Corrected radius
$i_k = i_0 + \delta i_k + (\text{IDOT})t_k$	Corrected inclination
$x'_k = r_k\cos\mu_k$	X coordinate in orbit plane
$y'_k = r_k\sin\mu_k$	Y coordinate in orbit plane
$\Omega_k = \Omega_0 + (\dot{\Omega} - \dot{\Omega}_0)t_k - \dot{\Omega}_e t_{0e}$	Corrected longitude of ascending node
$x_k = x'_k\cos\Omega_k - y'_k\cos i_k\sin\Omega_k$	ECEF X coordinate
$y_k = x'_k\sin\Omega_k + y'_k\cos i_k\cos\Omega_k$	ECEF Y coordinate
$z_k = y'_k\sin i_k$	ECEF Z coordinate

[a] t is in GPS system time at time of transmission, i.e., GPS time corrected for transit time (range/speed of light). Furthermore, t_k shall be the actual total time difference between the time t and the time epoch t_{0e} and must account for beginning or end of week crossovers. That is, if t_k is greater than 302,400 s, subtract 604,800 s from t_k. If t_k is less than $-302,400$ s, add 604,800 s to t_k.

4. *Subframe 5*. The 25 pages of this subframe includes the almanac for satellites with PRN numbers from 1 to 24.

It should be noted that since each satellite transmits all 25 pages, almanac data for all satellites is transmitted by every satellite. Unlike ephemeris data, almanac data is valid for long periods (months) but is much less precise. Additional data contained in the navigation message is user range error (URE), which estimates the range error due to errors in satellite ephemeris, timing errors, and selective availability (SA) and flags to indicate the health status of the satellites.

3.2.2 C/A-Code and Its Properties

The C/A-code has the following functions:

1. *To enable accurate range measurements and resistance to errors caused by multipath.* To establish the position of a user to within 10–100 m, accurate user-to-satellite range estimates are needed. The estimates are made from measurements of signal propagation delay from the satellite to the user. To achieve the required accuracy in measuring signal delay, the GPS carrier must be modulated by a waveform having a relatively large bandwidth. The needed bandwidth is provided by the C/A-code modulation, which also permits the receiver to use correlation processing to effectively combat measurement errors due to thermal noise. Because the C/A-code causes the bandwidth of the signal to be much greater than that needed to convey the 50-bps data stream, the resulting signal is called a *spread-spectrum* signal.

   Using the C/A-code to increase the signal bandwidth also reduces errors in measuring signal delay caused by multipath (the arrival of the signal via multiple paths such as reflections from objects near the receiver antenna) since the ability to separate the direct path signal from the reflected signal improves as the signal bandwidth is made larger.
2. *To permit simultaneous range measurement from several satellites*. The use of a distinct C/A-code for each satellite permits all satellites to use the same L_1 and L_2 frequencies without interfering with each other. This is possible because the signal from an individual satellite can be isolated by correlating it with a replica of its C/A-code in the receiver. This causes the C/A-code modulation from that satellite to be removed so that the signal contains only the 50-bps data and is therefore narrow band. This process is called *despreading* of the signal. However, the correlation process does not cause the signals from other satellites to become narrow band, because the codes from different satellites are orthogonal. Therefore the interfering signals can be rejected by passing the desired despread signal through a narrow-band filter, a bandwith-sharing process called *code division multiplexing* (CDM) or *code division multiple access* (CDMA).
3. *To provide protection from jamming*. The C/A-code also provides a measure of protection from intentional or unintentional jamming of the received signal by another man-made signal. The correlation process that despreads the desired signal has the property of spreading any other signal. Therefore, the signal power of any interfering signal, even if it is narrow band, will be spread over a large frequency band, and only that portion of the power lying in the narrow-band filter will compete with the desired signal. The C/A-code provides about 20–30 dB of improvement in resistance to jamming from narrowband signals.

We next detail important properties of the C/A-code.

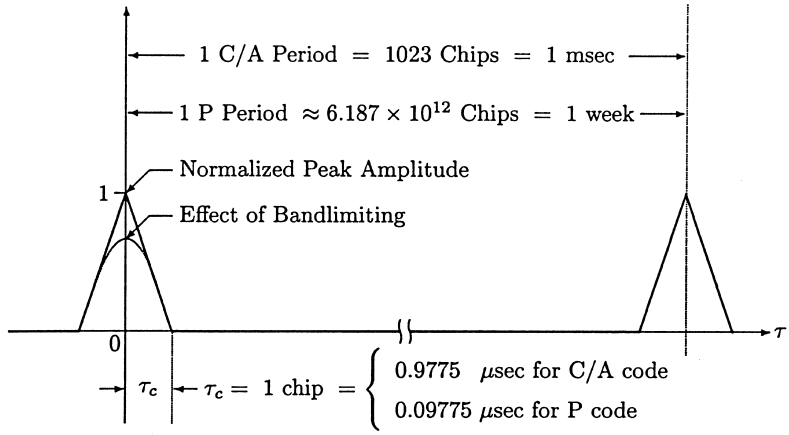


Fig. 3.5 *Autocorrelation functions of C/A- and P(Y)-codes.*

Temporal Structure Each satellite has a unique C/A-code, but all of the codes consist of a repeating sequence of 1023 chips occurring at a rate of 1.023 MHz with a period of 1 ms, as previously shown in Fig. 3.1. The leading edge of a specific chip in the sequence, called the *C/A-code epoch*, defines the beginning of a new period. Each chip is either positive or negative with the same magnitude. The polarities of the 1023 chips appear to be randomly distributed but are in fact generated by a deterministic algorithm implemented by shift registers. The algorithm produces maximal-length *Gold codes*, which have the property of low cross-correlation between different codes (orthogonality) as well as reasonably small autocorrelation sidelobes.

Autocorrelation Function The autocovariance[2] function of the C/A-code is

$$\psi(\tau) = \frac{1}{T}\int_0^T c(t)c(t-\tau)\,dt, \tag{3.3}$$

where $c(t)$ is the idealized C/A-code waveform (with chip values of ± 1), τ is the relative delay measured in seconds, and T is the code period (1 ms). The autocorrelation function is periodic in τ with a period of 1 ms. A single period is plotted in Fig. 3.5, which is basically a triangle two chips wide at its base with a peak located at $\tau = 0$ [in reality $\psi(\tau)$ contains small-sidelobe structures outside the triangular region, but these are of little consequence].

The C/A-code autocorrelation function plays a substantial role in GPS receivers, inasmuch as it forms the basis for code tracking and accurate user-to-satellite range

[2] Strictly speaking, the *autocorrelation* function $\psi(\tau) = \psi(\tau)/\psi(0)$ is the autocovariance function rescaled by the signal variance [$\psi(0)$], but the terms *autocorrelation* and *autocovariance* are often interchanged in engineering usage.

measurement. In fact, the receiver continually computes values of this function in which $c(t)$ in the above integral is the signal code waveform and $c(t-\tau)$ is an identical reference waveform (except for the relative delay τ) generated in the receiver. Special hardware and software enable the receiver to adjust the reference waveform delay so that the value of τ is zero, thus enabling determination of the time of arrival of the received signal.

Power Spectrum The power spectrum $\Psi(f)$ of the C/A-code describes how the power in the code is distributed in the frequency domain. It can be defined either in terms of a Fourier series expansion of the code waveform or equivalently in terms of the code autocorrelation function. Using the latter, we have

$$\Psi(f) = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} \psi(\tau) e^{-j2\pi f\tau}\, d\tau. \tag{3.4}$$

A plot of $\Psi(f)$ is shown as a smooth curve in Fig. 3.6; however, in reality $\Psi(f)$ consists of spectral lines with 1-kHz spacing due to the 1-ms periodic structure of $\psi(\tau)$. The power spectrum $\Psi(f)$ has a characteristic $\sin^2(x)/x^2$ shape with first nulls located 1.023 MHz from the central peak. Approximately 90% of the signal power is located between these two nulls, but the smaller portion lying outside the nulls is very important for accurate ranging. Also shown in the figure for comparative purposes is a typical noise power spectral density found in a GPS receiver after

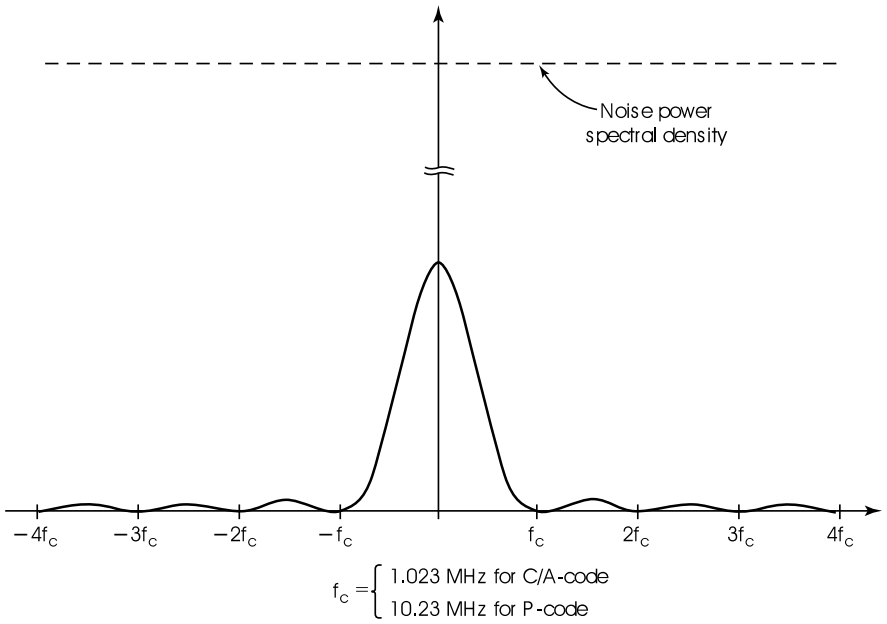


Fig. 3.6 *Power spectra of C/A- and P(Y)-codes.*

frequency conversion of the signal to baseband (i.e., with carrier removed). It can be seen that the presence of the C/A-code causes the entire signal to lie well below the noise level, because the signal power has been spread over a wide frequency range (approximately ±1 MHz).

Despreading of the Signal Spectrum The mathematical model of the signal modulated by the C/A-code is

$$s(t) = \sqrt{2P_I}d(t)c(t)\cos(\omega t + \theta) \tag{3.5}$$

where P_I is the carrier power, $d(t)$ is the 50-bps data modulation, $c(t)$ is the C/A-code waveform, ω is the L_1 carrier frequency in radians per second, and θ is the carrier phase shift in radians. When this signal is frequency shifted to baseband and tracked with a phase-lock loop, the carrier is removed and only the data modulation and the C/A-code modulation remain. The resulting signal, which in normalized form is

$$s(t) = d(t)c(t), \tag{3.6}$$

has a power spectrum similar to that of the C/A-code in Fig. 3.6. As previously mentioned, the signal in this form has a power spectrum lying below the receiver noise level, making it inacessible. However, if the signal is multiplied by a replica of $c(t)$ in exact alignment with it, the result is

$$s(t)c(t) = d(t)c(t)c(t) = d(t)c^2(t) = d(t), \tag{3.7}$$

where the last equality arises from the fact that the values of the ideal C/A-code waveform are ±1 (in reality the received C/A-code waveform is not ideal due to bandlimiting in the receiver; however, the effects are usually minor). This procedure, called *code despreading*, removes the C/A-code modulation from the signal. The resulting signal has a two-sided spectral width of approximately 100 Hz due to the 50-bps data modulation. From the above equation it can be seen that the total signal power has not been changed in this process, but it now is contained in a much narrower bandwith. Thus the magnitude of the power spectrum is greatly increased, as indicated in Fig. 3.7. In fact, it now exceeds that of the noise, and the signal can be recovered by passing it through a small-bandwidth filter (signal recovery filter) to remove the wide-band noise, as shown in the figure.

Role of Despreading in Interference Suppression At the same time that the spectrum of the desired GPS signal is narrowed by the despreading process, any interfering signal that is not modulated by the C/A-code will instead have its spectrum *spread* to a width of at least 2 MHz, so that only a small portion of the interfering power can pass through the signal recovery filter. The amount of interference suppression gained by using the C/A-code depends on the bandwidth of the recovery filter, the bandwidth of the interfering signal, and the bandwidth of

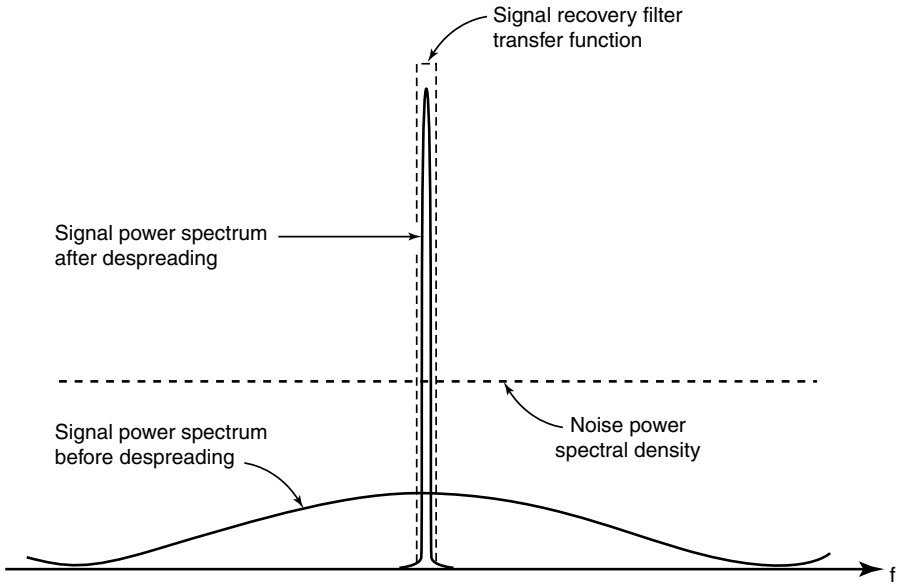


Fig. 3.7 *Despreading of the C/A-code.*

the C/A-code. For narrow-band interferors whose signal can be modeled by a nearly sinusoidal waveform and a signal recovery filter bandwidth of 1000 Hz or more, the amount of interference suppression in decibels is given approximately by

$$\eta = 10 \log\left(\frac{W_c}{W_f}\right) \qquad \text{dB}, \tag{3.8}$$

where W_c and W_f are respectively the bandwidths of the C/A-code (2.046 MHz) and the signal recovery filter. If $W_f = 2000$ Hz, about 30 dB of suppression can be obtained for narrow-band interferors. When the signal recovery filter has a bandwidth smaller than 1000 Hz, the situation is more complicated, since the despread interfering sinusoid will have discrete spectral components with a 1000-Hz spacing. As the bandwidth of the interfering signal increases, the C/A-code despreading process provides a decreasing amount of interference suppression. For interferors having a bandwidth greater than that of the signal recovery filter, the amount of suppression in decibels provided by the C/A-code is approximately

$$\eta = 10 \log\left(\frac{W_I + W_c}{W_I}\right) \qquad \text{dB}, \tag{3.9}$$

where W_I is the bandwidth of the interferor. When $W_I \gg W_c$, the C/A-code provides essentially no interference suppression at all compared to the use of an unspread carrier.

Code Division Multiplexing Property The C/A-codes from different satellites are *orthogonal*, which means that for any two codes $c_1(t)$ and $c_2(t)$ from different satellites, the cross-covariance

$$\frac{1}{T}\int_0^T c_1(t)c_2(t-\tau)\,dt \cong 0 \quad \text{for all } \tau. \tag{3.10}$$

Thus, when a selected satellite signal is despread using a replica of its code, the signals from other satellites look like wide-band interferors which are below the noise level. This permits a GPS receiver to extract a multiplicity of individual satellite signals and process them individually, even though all signals are transmitted at the same frequency. This process is called *code division multiplexing* (CDM).

3.2.3 P-Code and Its Properties

The P-code, which is used primarily for military applications, has the following functions:

1. *Increased Jamming Protection*. Because the bandwidth of the P-code is 10 times greater than that of the C/A-code, it offers approximately 10 dB more protection from narrow-band interference. In military applications the interference is likely to be a deliberate attempt to jam (render useless) the received GPS signal.
2. *Provision for Antispoofing*. In addition to jamming, another military tactic that an enemy can employ is to radiate a signal that appears to be a GPS signal (*spoofing*), but in reality is designed to confuse the GPS receiver. This is prevented by encrypting the P-code. The would-be spoofer cannot know the encryption process and cannot make the contending signal look like a properly encrypted signal. Thus the receiver can reject the false signal and decrypt the desired one.
3. *Denial of P-Code Use*. The structure of the P-code is published in the open literature, so than anyone may generate it as a reference code for despreading the signal and making range measurements. However, encryption of the P-code by the military will deny its use by unauthorized parties.
4. *Increased Code Range Measurement Accuracy.* All other parameters being equal, accuracy in range measurement improves as the signal bandwidth increases. Thus, the P-code provides improved range measurement accuracy as compared to the C/A-code. Simultaneous range measurements using both codes is even better. Due to its increased bandwidth, the P-code is also more resistant to range errors caused by multipath.

P-Code Characteristics Unlike the C/A-code, the P-code modulates both the L_1 and L_2 carriers. Its chipping rate is 10.23 MHz, which is precisely 10 times the

C/A rate, and it has a period of one week. It is transmitted synchronously with the C/A- code in the sense that each chip transition of the C/A-code always corresponds to a chip transition in the P-code. Like the C/A-code, the P-code autocorrelation function has a triangular central peak centered at $\tau = 0$, but with one-tenth the base width, as shown in Fig. 3.5. The power spectrum also has a $\sin^2(x)/x^2$ characteristic, but with 10 times the bandwidth, as indicated in Fig. 3.6. Because the period of the P-code is so long, the power spectrum may be regarded as continuous for practical purposes. Each satellite broadcasts a unique P-code. The technique used to generate it is similar to that of the C/A-code, but somewhat more complicated, and will not be covered in this book.

Y-Code The encrypted form of the P-code used for antispoofing and denial of the P-code to unauthorized users is called the *Y-code*. The Y-code is formed by multiplying the P-code by an encrypting code called the *W-code*. The W-code is a random-looking sequence of chips that occur at a 511.5-kHz rate. Thus there are 20 P-code chips for every W-code chip. Since both the P-code and the W-code have chip values of ± 1, the resulting Y-code has the same appearance as the P-code, that is, it also has a 10.23-MHz chipping rate. However, the Y-code cannot be despread by a receiver replica P-code unless it is decrypted. Decryption consists of multiplying the Y-code by a receiver-generated replica of the W-code which is made available only to authorized users. Since the encrypting W-code is also not known by the creators of spoofing signals, it is easy to verify that such signals are not legitimate.

3.2.4 L_1 and L_2 Carriers

The L_1 (or L_2) carrier is used for the following purposes:

1. *To provide very accurate range measurements for precision applications using carrier phase*.
2. *To provide accurate Doppler measurements*. The phase rate of the received carrier can be used for accurate determination of user velocity. The integrated Doppler, which can be obtained by counting the cycles of the received carrier, is often used as a precise delta range observable that can materially aid the performance of code tracking loops. The integrated Doppler history is also used as part of the carrier phase ambiguity resolution process.

Dual-Frequency Operation The use of *both* the L_1 and L_2 frequencies provides the following benefits:

1. *Provides accurate measurement of ionospheric signal delay.* A major source of ranging error is caused by changes in both the phase velocity and group velocity of the signal as it passes through the ionosphere. Range errors of 10–20 m are commonplace and sometimes much larger. Because the delay

induced by the ionosphere is known to be inversely proportional to the square of frequency, ionospheric range error can be estimated accurately by comparing the times of arrival of the L_1 and L_2 signals. Details on the calculations appear in Chapter 5.

2. *Facilitates carrier phase ambiguity resolution*. In high-accuracy GPS differential positioning, the range estimates using carrier phase measurements are precise but highly ambiguous due to the periodic structure of the carrier. The ambiguity is more easily resolved (by various methods) as the carrier frequency decreases. By using L_1 and L_2 carrier frequencies, the ambiguity resolution can be based on their frequency difference (1575.42 − 1227.6 MHz), which is smaller than either carrier frequency alone, and hence will result in better ambiguity resolution performance.
3. *Provides system redundancy (primarily for the military user)*.

3.3 SIGNAL POWER LEVELS

The L_1 C/A-code signal is transmitted at a minimum level of 478.63 W (26.8 dBW) effective isotropic radiated power (EIRP), which means that the minimum received power is the same as that which would be obtained if the satellite radiated 478.63 W from an isotropic antenna. This effective power level is reached by radiating a smaller total power in a beam approximately 30° wide toward the earth. The radiated power level was chosen to provide a signal-to-noise ratio sufficient for tracking of the signal by a receiver on the Earth with an unobstructed view of the satellite. However, the chosen power has been criticized as being inadequate in light of the need to operate GPS receivers under less desirable conditions, such as in heavy vegetation or in urban canyons where considerable signal attenuation often occurs. For this reason, future satellites may have higher transmitted power.

As the signal propagates toward the earth, it loses power density due to spherical spreading. The loss is accounted for by a quantity called the free-space loss factor (FSLF), given by

$$\text{FSLF} = \left(\frac{\lambda}{4\pi R}\right)^2. \tag{3.11}$$

The FSLF is the fractional power density at a distance R meters from the transmitting antenna compared to a value normalized to unity at the distance $\lambda/4\pi$ meters from the antenna phase center. Using $R = 2 \times 10^7$ and $\lambda = 0.19$ m at the L_1 frequency, the FSLF is about 5.7×10^{-19}, or −182.4 dB.

An additional atmospheric loss factor (ALF) of about 2.0 dB occurs as the signal gets attenuated by the atmosphere. If the receiving antenna is assumed to be isotropic, the received signal power is EIRP − FSLF − ALF = 26.8 − 182.4 − 2.0 = −157.6 dBW. Since a typical GPS antenna with right-hand circular polarization and a hemispherical pattern has about 3.0 dB of gain relative to an isotropic

Table 3.3 Calculation of Minimum Received Signal Power

Minimum transmitted signal power (EIRP)	26.8[a] dBW
Free-space loss factor (FSLF)	−182.4 dB
Atmospheric loss factor (ALF)	−2.0 dB
Receiver antenna gain relative to isotropic (RAG)	3.0 dB
Minimum received signal power (EIRP − FSLF − ALF + RAG)	−154.6 dBW

[a] Including antenna gain.

antenna, the minimum received signal power for such an antenna is about 3.0 dB larger. These results are summarized in Table 3.3.

3.4 SIGNAL ACQUISITION AND TRACKING

When a GPS receiver is turned on, a sequence of operations must ensue before information in a GPS signal can be accessed and used to provide a navigation solution. In the order of execution, these operations are as follows:

1. Determine which satellites are visible to the antenna.
2. Determine the approximate Doppler of each visible satellite.
3. Search for the signal both in frequency and C/A-code phase.
4. Detect the presence of a signal and confirm detection.
5. Lock onto and track the C/A-code.
6. Lock onto and track the carrier.
7. Perform data bit synchronization.
8. Demodulate the 50-bps navigation data.

3.4.1 Determination of Visible Satellites

In many GPS receiver applications it is desirable to minimize the time from receiver turn-on until the first navigation solution is obtained. This time interval is commonly called *time to first fix* (TTFF). Depending on receiver characteristics, the TTFF might range from 30 s to several minutes. An important consideration in minimizing the TTFF is to avoid a fruitless search for those satellite signals that are blocked by the earth, that is, below the horizon. A receiver can restrict its search to only those satellites that are visible if it knows its approximate location (within several hundred miles) and approximate time (within approximately 10 min) and has satellite almanac data obtained within the last several months. The approximate location can be manually entered by the user or it can be the position obtained by GPS when the receiver was last in operation. The approximate time can also be entered manually, but most receivers have a sufficiently accurate real-time clock that operates continuously, even when the receiver is off.

Using the approximate time, approximate position, and almanac data, the receiver calculates the elevation angle of each satellite and identifies the visible satellites as those whose elevation angle is greater than a specified value, called the *mask angle*, which has typical values of 5° to 15°. At elevation angles below the mask angle, tropospheric attenuation and delays tend to make the signals unreliable.

Most receivers automatically update the almanac data when in use, but if the receiver is just "out of the box" or has not been used for many months, it will need to search "blind" for a satellite signal to collect the needed almanac. In this case the receiver will not know which satellites are visible, so it simply must work its way down a predetermined list of satellites until a signal is found. Although such a "blind" search may take an appreciable length of time, it is infrequently needed.

3.4.2 Signal Doppler Estimation

The TTFF can be further reduced if the approximate Doppler shifts of the visible satellite signals are known. This permits the receiver to establish a frequency search pattern in which the most likely frequencies of reception are searched first. The expected Doppler shifts can be calculated from knowledge of approximate position, approximate time, and valid almanac data. The greatest benefit is obtained if the receiver has a reasonably accurate clock reference oscillator.

However, once the first satellite signal is found, a fairly good estimate of receiver clock frequency error can be determined by comparing the predicted Doppler shift with the measured Doppler shift. This error can then be subtracted out while searching in frequency for the remaining satellites, thus significantly reducing the range of frequencies that need to be searched.

3.4.3 Search for Signal in Frequency and C/A-Code Phase

Why is a Signal Search Necessary? Since GPS signals are radio signals, one might assume that they could be received simply by setting a dial to a particular frequency, as is done with AM and FM broadcast band receivers. Unfortunately, this is not the case.

1. GPS signals are *spread-spectrum* signals in which the C/A or P-codes spread the total signal power over a wide bandwidth. The signals are therefore virtually undetectable unless they are *despread* with a replica code in the receiver which is precisely aligned with the received code. Since the signal cannot be detected until alignment has been achieved, a search over the possible alignment positions (code search) is required.
2. A relatively narrow post-despreading bandwidth (perhaps 100–1000 Hz) is required to raise the signal-to-noise ratio to detectable and/or usable levels. However, because of the high carrier frequencies and large satellite velocities used by GPS, the received signals can have large Doppler shifts (as much as ± 5 kHz) which may vary rapidly (as much as 1 Hz/s). The observed Doppler

shift also varies with location on earth, so that the received frequency will generally be unknown *a priori*. Furthermore, the frequency error in typical receiver reference oscillators will typically cause several kilohertz or more of frequency uncertainty at L-band. Thus, in addition to the code search, there is also the need for a search in frequency.

Therefore, a GPS receiver must conduct a two-dimensional search in order to find each satellite signal, where the dimensions are C/A-code delay and carrier frequency. A search must be conducted across the full delay range of the C/A-code for each frequency searched. A generic method for conducting the search is illustrated in Fig. 3.8 in which the received waveform is multiplied by delayed replicas of the C/A-code, translated by various frequencies, and then passed through a baseband correlator containing a low-pass filter which has a relatively small bandwidth (perhaps 100–1000 Hz). The output energy of the detection filter serves as a signal detection statistic and will be significant only if both the selected code delay and frequency translation match that of the signal. When the energy exceeds a predetermined threshold β, a tentative decision is made that a signal is being received, subject to later confirmation. The value chosen for the threshold β is a compromise between the conflicting goals of maximizing the probability P_D of detecting the signal when it is actually present at a given Doppler and code delay and minimizing the probability P_{FA} of false alarm when it is not.

Searching in Code Delay For each frequency searched, the receiver generates the same PRN code as that of the satellite and moves the delay of this code in discrete steps (typically 0.5 chips) until approximate alignment with the received code (and also a match in Doppler) is indicated when the correlator output energy exceeds

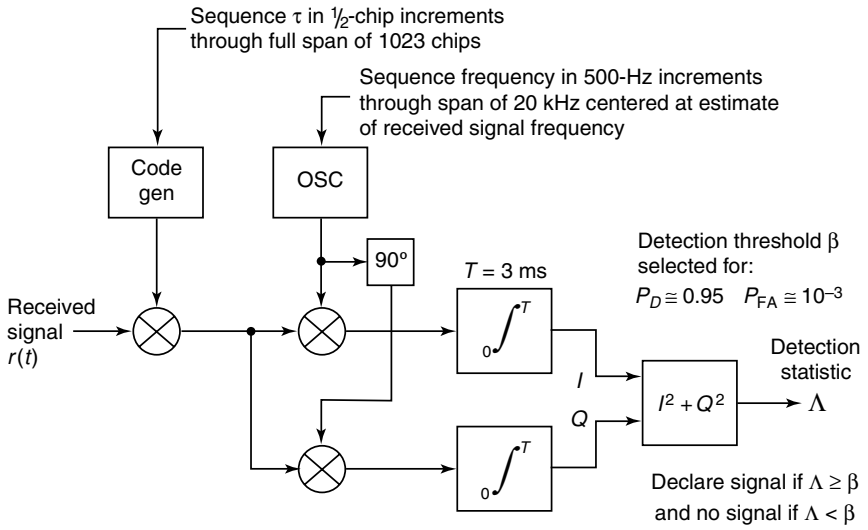


Fig. 3.8 *Signal search method.*

threshold β. A step size of 0.5 code chip, which is used by many GPS receivers, is an acceptable compromise between the conflicting requirements of search speed (enhanced by a larger step size) and guaranteeing a code delay that will be located near the peak value of the code correlation function (enhanced by a smaller step size). The best situation occurs when one of the delay positions is at the correlation function peak, and the worst one occurs when there are two delay positions straddling the peak, as indicated in Fig. 3.9. In the latter case, the effective SNR is reduced by as much as 6 dB. However, the effect is ameliorated because, instead of only one delay position with substantial correlation, there are two that can be tested for the presence of signal.

An important parameter in the code search is the dwell time used for each code delay position, since it influences both the search speed and the detection/false-alarm performance. The dwell time should be an integral multiple of 1 ms to assure that the correct correlation function, using the full range of 1023 code states, is obtained. Satisfactory performance is obtained with dwell times from 1 to 4 ms in most GPS receivers, but longer dwell times are sometimes used to increase detection capability in weak-signal environments. However, if the dwell time for the search is a substantial fraction of 20 ms (the duration of one data bit), it becomes increasingly probable that a bit transition of the 50-Hz data modulation will destroy the coherent processing of the correlator during the search and lead to a missed detection. This imposes a practical limit for a search using coherent detection.

The simplest type of code search uses a fixed dwell time, a single detection threshold value β, and a simple yes/no binary decision as to the presence of a signal. Many receivers achieve considerable improvement in search speed by using a sequential detection technique in which the overall dwell time is conditioned on a ternary decision involving an upper and a lower detection threshold. Details on this approach can be found in [125].

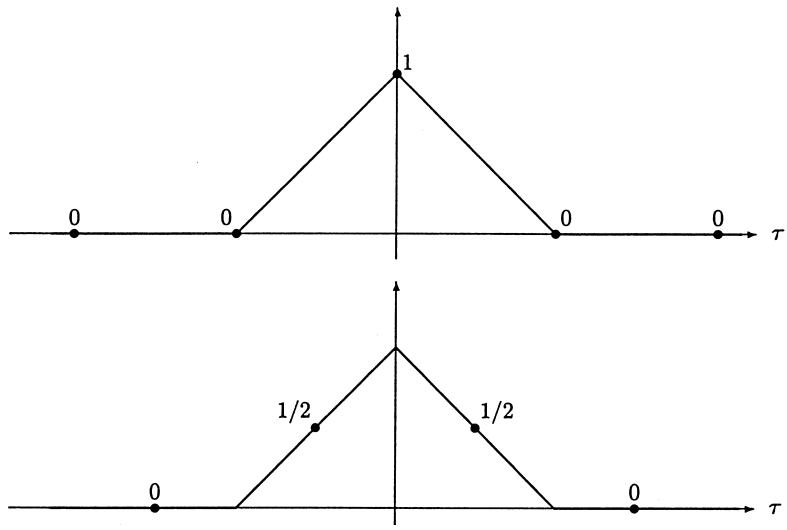


Fig. 3.9 *Effect of $\frac{1}{2}$ chip step size in code search.*

Searching in Frequency The range of frequency uncertainty that must be searched is a function of the accuracy of the receiver reference oscillator, how well the approximate user position is known, and the accuracy of the receiver's built-in real-time clock. The first step in the search is to use stored almanac data to obtain an estimate of the Doppler shift of the satellite signal. An interval $[f_1, f_2]$ of frequencies to be searched is then established. The center of the interval is located at $f_c + f_d$, where f_c is the L_1 (or L_2) carrier frequency and f_d is the estimated carrier Doppler shift. The width of the search interval is made large enough to account for worst-case errors in the receiver reference oscillator, in the estimate of user postion, and in the real-time clock. A typical range for the frequency search interval is $f_c + f_d \pm 5$ kHz.

The frequency search is conducted in N discrete frequency steps that cover the entire search interval. The value of N is $(f_2 - f_1)/\Delta f$, where Δf is the spacing between adjacent frequencies (bin width). The bin width is determined by the effective bandwidth of the correlator. For the coherent processing used in many GPS receivers, the frequency bin width is approximately the reciprocal of the search dwell time. Thus, typical values of Δf are 250–1000 Hz. Assuming a $\pm$5-kHz frequency search range, the number N of frequency steps to cover the entire search interval would typically be 10–40.

Frequency Search Strategy Because the received signal frequency is more likely to be near, rather than far from, the Doppler estimate, the expected time to detect the signal can be minimized by starting the search at the estimated frequency and expanding in an outward direction by alternately selecting frequencies above and below the estimate, as indicated in Fig. 3.10. On the other hand, the unknown code delay of the signal can be considered to be uniformly distributed over its range so that each delay value is equally likely. Thus, the delays used in the code search can simply sequence from 0 to 1023.5 chips in 0.5-chip increments.

Sequential Versus Parallel Search Methods Almost all current GPS receivers are multichannel units in which each channel is assigned a satellite and processing in the channels is carried out simultaneously. Thus, simultaneous searches can be made for all usable satellites when the receiver is turned on. Because the search in each channel consists of sequencing through all possible frequency and code delay steps, it is called a *sequential search*. In this case, the expected time required to acquire as many as eight satellites is typically 30–100 s, depending on the specific search parameters used.

Certain applications (mostly military) demand that the satellites be acquired much more rapidly (perhaps within a few seconds). This can be accomplished by using a *parallel search* technique in which extra hardware permits many frequencies and code delays to be searched at the same time. However, this approach is seldom used in commercial receivers because of its high cost.

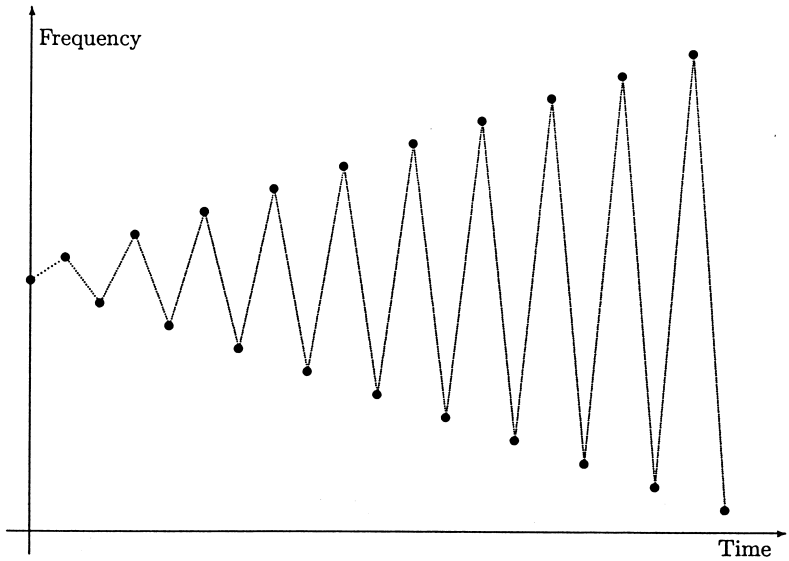


Fig. 3.10 *Frequency search strategy.*

3.4.4 Signal Detection and Confirmation

As previously mentioned, there is a trade-off between the probability of detection P_D and false alarm P_{FA}. As the detection threshold β is decreased, P_D increases but P_{FA} also increases, as illustrated in Fig. 3.11. Thus, the challenge in receiver design is to achieve a sufficiently large P_D so that a signal will not be missed but at the same time keep P_{FA} small enough to avoid difficulties with false detections. When a false detection occurs, the receiver will try to lock onto and track a nonexistent signal. By the time the failure to track becomes evident, the receiver will have to initiate a completely new search for the signal. On the other hand, when a detection failure occurs, the receiver will waste time continuing to search remaining search cells that contain no signal, after which a new search must be initiated.

Detection Confirmation One way to achieve both a large P_D and a small P_{FA} is to increase the dwell time so that the relative noise component of the detection statistic is reduced. However, to reliably acquire weak GPS signals, the required dwell time may result in unacceptably slow search speed. An effective way around this problem is to use some form of *detection confirmation*.

To illustrate the detection confirmation concept, suppose that to obtain the detection probability $P_D = 0.95$ with a typical medium-strength GPS signal, we obtain the false-alarm probability $P_{\text{FA}} = 10^{-3}$. (These are typical values for a fixed search dwell time of 3 ms.) This means that on the average, there will be one false detection in every 1000 frequency/code cells searched. A typical two-dimensional GPS search region might contain as many as 40 frequency bins and 2046 code delay positions, for a total of $40 \times 2046 = 81,840$ such cells. Thus we could expect about

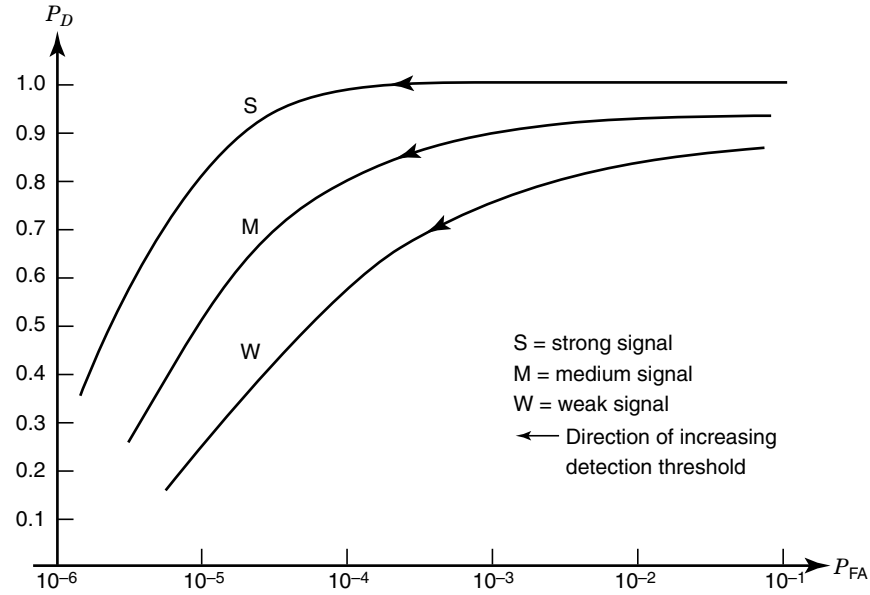


Fig. 3.11 *Illustration of trade-off between P_D and P_{FA}.*

82 false detections in the full search region. Given the implications of a false detection discussed previously, this is clearly unacceptable.

However, suppose we change the rules for what happens when a detection (false or otherwise) occurs by performing a confirmation of detection before turning the signal over to the tracking loops. Because a false detection takes place only once in 1000 search cells, it is possible to use a much longer dwell (or a sequence of repeated dwells) for purposes of confirmation without markedly increasing the overall search speed, yet the confirmation process will have an extremely high probability of being correct. In the event that confirmation indicates no signal, the search can continue without interruption by the large time delay inherent in detecting the failure to track. In addition to using longer dwell times, the confirmation process can also perform a *local search* in which the frequency/code cell size is smaller than that of the main, or *global*, search, thus providing a more accurate estimate of signal frequency and code phase when a detection is confirmed. Figure 3.12 depicts this scheme. The global search uses a detection threshold β that provides a high P_D and a moderate value of P_{FA}. Whenever the detection statistic Λ exceeds β at a frequency/delay cell, a confirmation search is performed in a local region surrounding that cell. The local region is subdivided into smaller cells to obtain better frequency delay resolution, and a longer dwell time is used in forming the detection statistic Λ. The longer dwell time makes it possible to use a value of β that provides both a high P_D and a low P_{FA}.

Adaptive Signal Searches Some GPS receivers use a simple adaptive search in which shorter dwell times are first used to permit rapid acquisition of moderate to

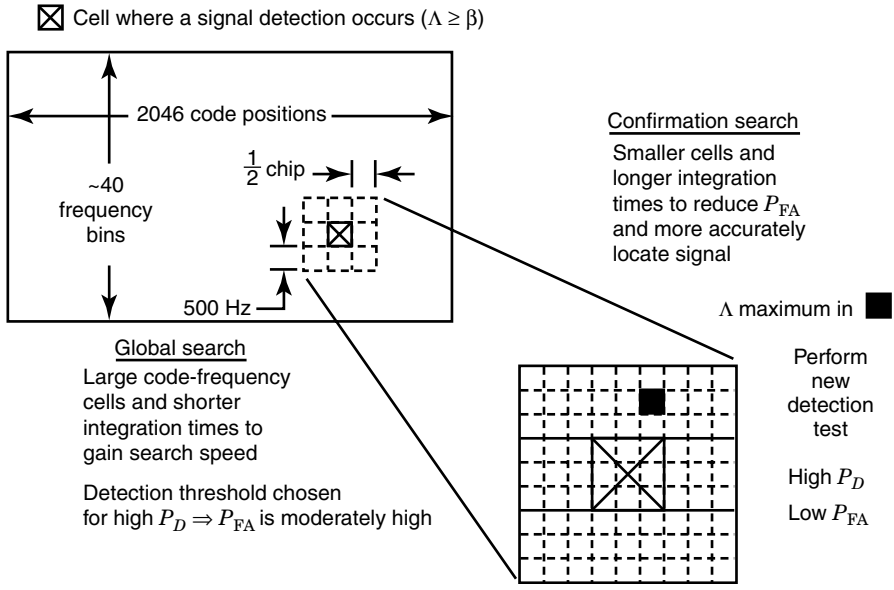


Fig. 3.12 *Global and confirmation search regions.*

strong signals. Whenever a search for a particular satellite is unsuccessful, it is likely that the signal from that satellite is relatively weak, so the receiver increases the dwell time and starts a new search that is slower but has better performance in acquiring weak signals.

Coordination of Frequency Tuning and Code Chipping Rate As the receiver is tuned in frequency during search, it is advantageous to precess the chipping rate of the receiver generated code so that it is in accordance with the Doppler shift under consideration. The relationship between Doppler shift and the precession rate of the C/A-code is given by $p(t) = f_d/1540$, where $p(t)$ is the code precession rate in chips per second, f_d is the Doppler shift in Hertz, and a positive precession rate is interpreted as an increase in the chipping rate. Precession is not required while searching because the dwell times are so short. However, when detection of the signal occurs, it is important to match the incoming and reference code rates during the longer time required for detection confirmation and/or initiation of code tracking to take place.

3.4.5 Code Tracking Loop

At the time of detection confirmation the receiver-generated reference C/A-code will be in approximate alignment with that of the signal (usually within one-half chip), and the reference code chipping rate will be approximately that of the signal. Additionally, the frequency of the signal will be known to within the frequency bin width Δf. However, unless further measures are taken, the residual Doppler on the

signal will eventually cause the received and reference codes to drift out of alignment and the signal frequency to drift outside the frequency bin at which detection occurred. If the code alignment error exceeds one chip in magnitude, the incoming signal will no longer despread and will disappear below the noise level. The signal will also disappear if it drifts outside the detection frequency bin. Thus there is the need to continually adjust the timing of the reference code so that it maintains accurate alignment with the received code, a process called *code tracking*. The process of maintaining accurate tuning to the signal carrier, called *carrier tracking*, is also necessary and will be discussed in following sections.

Code tracking is initiated as soon as signal detection is confirmed, and the goal is to make the receiver-generated code line up with incoming code as precisely as possible. There are two objectives in maintaining alignment:

1. *Signal Despreading*. The first objective is to fully despread the signal so that it is no longer below the noise and so that information contained in the carrier and the 50-bps data modulation can be recovered.
2. *Range Measurements*. The second objective is to enable precise measurement of the time of arrival (TOA) of received code for purposes of measuring range. Such measurements cannot be made directly from the received signal, since it is below the noise level. Therefore, a code tracking loop, which has a large processing gain, is employed to generate a reference code precisely aligned with that of the received signal. This enables range measurements to be made using the reference code instead of the much noisier received signal code waveform.

Figure 3.13 illustrates the concept of a code tracking loop. It is assumed that a numerically controlled oscillator (NCO) has translated the signal to complex baseband form (i.e., zero frequency). Each component (I and Q) of the baseband signal is multiplied by three replicas of the C/A-code that are formed by delaying the output of a single code generator by three delay values called *early*, *punctual*, *and late*. In typical GPS receivers the early and late codes respectively lead and lag the punctual code by 0.05 to 0.5 code chips and always maintain these relative positions. Following each multiplier is a low-pass filter (LPF) or integrator that, together with its associated multiplier, forms a correlator. The output magnitude of each correlator is proportional to the cross-correlation of its received and reference codes, where the cross-correlation function has the triangular shape previously shown in Fig. 3.5. In normal operation the punctual code is aligned with the code of the incoming signal so that the squared magnitude $I_P^2 + Q_P^2$ of the punctual correlator output is at the peak of the cross-correlation function, and the output magnitudes of the early and late correlators have smaller but equal values on each side of the peak. To maintain this condition, a loop error signal

$$e_c(\tau) = I_L^2 + Q_L^2 - (I_E^2 + Q_E^2) \tag{3.12}$$

is formed, which is the difference between the squared magnitudes of the late and early correlators. The loop error signal as a function of received code delay is shown

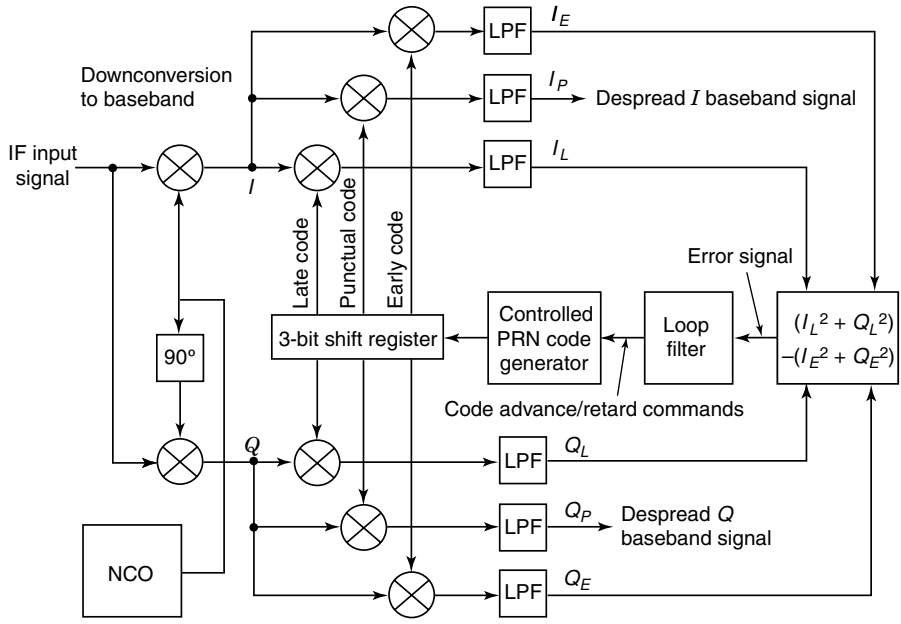


Fig. 3.13 *Code tracking loop concept.*

in Fig. 3.14. Near the tracking point the error is positive if the received code is delayed relative to the punctual code and negative if it is advanced. Alignment of the punctual code with the received code is maintained by using the error signal to delay the reference code generator when the error signal is positive and to advance it when the error signal is negative. Since $e_c(\tau)$ is generally quite noisy, it is sent through a low-pass *loop filter* before it controls the timing of the reference code generator, as indicated in Fig. 3.13. The bandwidth of this filter is usually quite small, resulting in a closed-loop bandwidth typically less than 1 Hz. This is the source of the large processing gain that can be realized in extracting the C/A-code from the signal.

When the code tracking loop is first turned on, the integration time T for the correlators is usually no more than a few milliseconds, in order to minimize corruption of the correlation process by data bit transitions of the 50-bps data stream whose locations in time are not yet known. However, after bit synchronization has located the data bit boundaries, the integration interval can span a full data bit (20 ms) in order to achieve a maximum contribution to processing gain.

Coherent Versus Noncoherent Code Tracking If the error signal is formed from only the squared magnitudes of the (complex) early and late correlator outputs as described above, the loop is called a *noncoherent* code tracking loop. A distinguishing feature of such a loop is its insensitivity to the phase of the received signal. Insensitivity to phase is desirable when the loop is first turned on, since at that time the signal phase is random and not yet under any control.

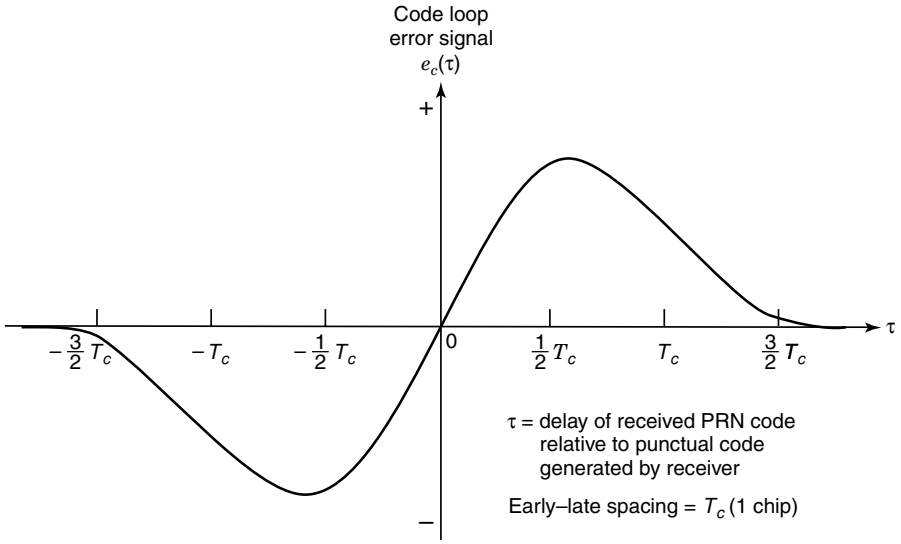


Fig. 3.14 *Code tracking loop error signal.*

On the other hand, once the phase of the signal is being tracked, a *coherent* code tracker can be employed, in which the outputs of the early and late correlators are purely real. In this situation the loop error signal can be formed directly from the difference of the early and late squared magnitudes from only the I correlator. By avoiding the noise in the Q correlator outputs, a 3-dB SNR advantage is thereby gained in tracking the code. However, a price is paid in that the code loop error signal becomes sensitive to phase error in tracking the carrier. If phase tracking is ever lost, complete failure of the code tracking loop could occur. This is a major disadvantage, especially in mobile applications where the signal can vary rapidly in magnitude and phase. Since noncoherent operation is much more robust in this regard and is still needed when code tracking is initiated, most GPS receivers use only noncoherent code tracking.

Factors Affecting Code Tracking Performance The bandwidth of the code tracking loop is determined primarily by the loop filter and needs to be narrow for best ranging accuracy but wide enough to avoid loss of lock if the receiver is subject to large accelerations that can suddenly change the apparent chipping rate of the received code. Excessive accelerations cause loss of lock by moving the received and reference codes too far out of alignment before the loop can adequately respond. Once the alignment error exceeds approximately 1 code chip, the loop loses lock because it no longer has the ability to form the proper error signal.

In low-dynamics applications with lower cost receivers, code tracking loop bandwidths on the order of 1 Hz permit acceptable performance in hand-held

units and in receivers with moderate dynamics (e.g., in automobiles). For high-dynamics applications, such as missile platforms, loop bandwidths might be on the order of 10 Hz or larger. In surveying applications, which have no appreciable dynamics, loop bandwidths can be as small as 0.01 Hz to obtain the required ranging accuracy. Both tracking accuracy and the ability to handle dynamics are greatly enhanced by means of *carrier aiding* from the receiver's carrier phase tracking loop, which will be be discussed subsequently.

3.4.6 Carrier Phase Tracking Loops

The purposes of tracking carrier phase are

1. to obtain a phase reference for coherent detection of the GPS biphase modulated data,
2. to provide precise velocity measurements (via phase rate),
3. to obtain integrated Doppler for rate aiding of the code tracking loop, and
4. to obtain precise carrier phase pseudorange measurements in high-accuracy receivers.

Tracking of carrier phase is usually accomplished by a phase-lock loop (PLL). A Costas-type PLL or its equivalent must be used to prevent loss of phase coherence induced by the biphase data modulation on the GPS carrier. The origin of the Costas PLL is described in [24]. A typical Costas loop is shown in Fig. 3.15. In this design the output of the receiver last intermediate-frequency (IF) amplifier is converted to a complex baseband signal by multiplying the signal by both the in-phase and quadrature-phase outputs of an NCO and integrating each product over each 20-ms data bit interval to form a sequence of phasors. The phase angle of each phasor is the phase difference between the signal carrier and the NCO output during the 20-ms integration. A loop phase error signal is formed by multiplying together the I and Q components of each phasor. This error signal is unaffected by the biphase data modulation because the modulation appears on both I and Q and is removed in forming the $I \times Q$ product. After passing through a low-pass loop filter the error signal controls the NCO phase to drive the loop error signal $I \times Q$ to zero (the phase-locked condition). In some receivers the error signal is generated by forming twice the four-quadrant arctangent of the I and Q phasor components, as indicated in the figure.

Because the Costas loop is unaffected by the data modulation, it will achieve phase lock at two stable points where the NCO output phase differs from that of the signal carrier by either 0° or 180°, respectively. This can be seen by considering $I = A\cos\theta$ and $Q = A\sin\theta$, where A is the phasor amplitude and θ is its phase. Then,

$$I \times Q = A^2 \cos\theta \sin\theta = \tfrac{1}{2} A^2 \sin 2\theta. \tag{3.13}$$

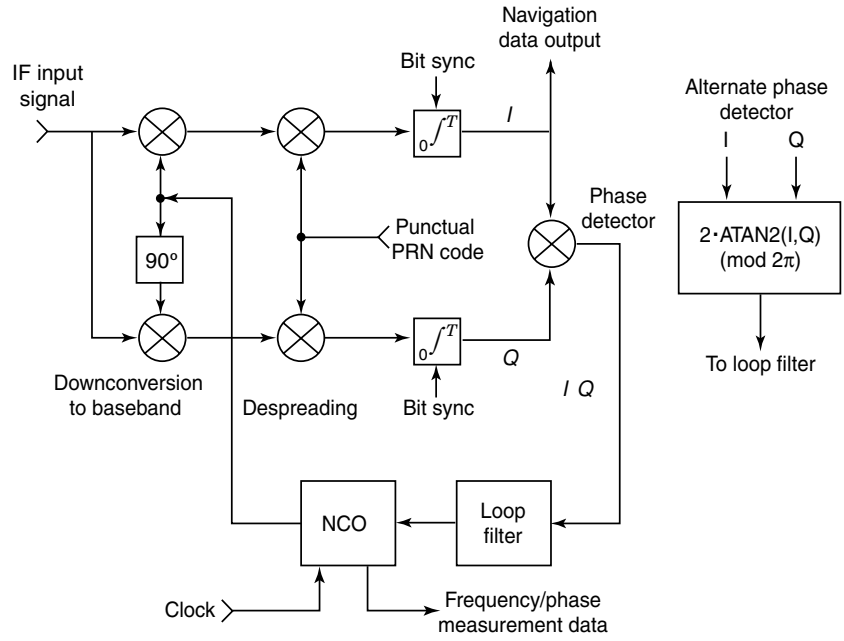


Fig. 3.15 *Costas PLL.*

There are four values of θ in $[0, 2\pi)$ where the error signal $I \times Q = 0$. Two of these are the stable points, namely $\theta = 0$ and $\theta = 180^\circ$, toward which the loop tends to return if perturbed. Since $\sin 2\theta$ is unchanged by 180° changes in θ caused by the data bits, the data modulation will have no effect. At either of the two stable points the Q integrator output is nominally zero and the I integrator output contains the demodulated data stream, but with a polarity ambiguity that can be removed by observing frame preamble data. Thus the Costas loop has the additional feature of serving as a data demodulator.

In the Costas loop design shown the phase of the signal is measured by comparing the phase of the NCO output with a reference signal. Normally the reference signal frequency is a rational multiple of the same crystal-controlled oscillator that is used in frequency shifting the GPS signal down to the last IF. When the NCO is locked to the phase of the incoming signal, the measured phase rate will typically be in the range of ± 5 kHz due to signal Doppler shift. Two types of phase measurement are usually performed on a periodic basis (the period might be every 20 ms). The first is an accurate measurement of the phase modulo 2π, which is used in precision carrier phase ranging. The second is the number of cycles (including the fractional part) of phase change that have occurred from a defined point in time up to the present time. The latter measurement is often called integrated Doppler and is used for aiding the code tracking loop. By subtracting consecutive integrated Doppler measurements, extremely accurate average frequency measurements can

be made, which can be used by the navigation filter to accurately determine user velocity.

Although the Costas loop is not disturbed by the presence of data modulation, at low SNR its performance degrades considerably from that of a loop designed for a pure carrier. The degradation is due to the noise × noise component of the $I \times Q$ error signal. Furthermore, the 20-ms duration of the I and Q integrations represents a limit to the amount of coherent processing that can be achieved. If it is assumed that the maximum acceptable bit error rate for the 50-bps data demodulation is 10^{-5}, GPS signals become unusable when C/N_0 falls below about 25 dB-Hz.

The design bandwidth of the PLL is determined by the SNR, desired tracking accuracy, signal dynamics, and ability to "pull in" when acquiring the signal or when lock is momentarily lost.

PLL Capture Range An important characteristic of the PLL is the ability to "pull-in" to the frequency of a received signal. When the PLL is first turned on following code acquisition, the difference between the signal carrier frequency and the NCO frequency must be sufficiently small or the PLL will not lock. In typical GPS applications, the PLL must have a relatively small bandwidth (1–10 Hz) to prevent loss of lock due to noise. However, this results in a small pull-in (or capture) range (perhaps only 3–30 Hz), which would require small (hence many) frequency bins in the signal acquisition search algorithm.

Use of Frequency-Lock Loops for Carrier Capture Some receivers avoid the conflicting demands of the need for a small bandwidth and a large capture range in the PLL by using a frequency-lock loop (FLL). The capture range of a FLL is typically much larger than that of a PLL, but the FLL cannot lock to phase. Therefore, a FLL is often used to pull the NCO frequency into the capture range of the PLL, at which time the FLL is turned off and the PLL is turned on. A typical FLL design is shown in Fig. 3.16. The FLL generates a loop error signal e_{FLL} that is approximately proportional to the rotation rate of the baseband signal phasor and is derived from the vector cross-product of successive baseband phasors $[I(t-\tau), Q(t-\tau)]$ and $[I(t), Q(t)]$, where τ is a fixed delay, typically 1–5 ms. More precisely,

$$e_{\mathrm{FLL}} = Q(t)I(t-\tau) - I(t)Q(t-\tau). \tag{3.14}$$

PLL Order The *order* of a PLL refers to its capability to track different types of signal dynamics. Most GPS receivers use second- or third-order PLLs. A second-order loop can track a constant rate of phase change (i.e., constant frequency) with zero average phase error and a constant rate of frequency change with a nonzero but constant phase error. A third-order loop can track a constant rate of frequency change with zero average phase error and a constant acceleration of frequency with nonzero but constant phase error. Low-cost receivers typically use a second-order PLL with fairly low bandwidth because the user dynamics are minimal and the rate of change of the signal frequency due to satellite motion is sufficiently low

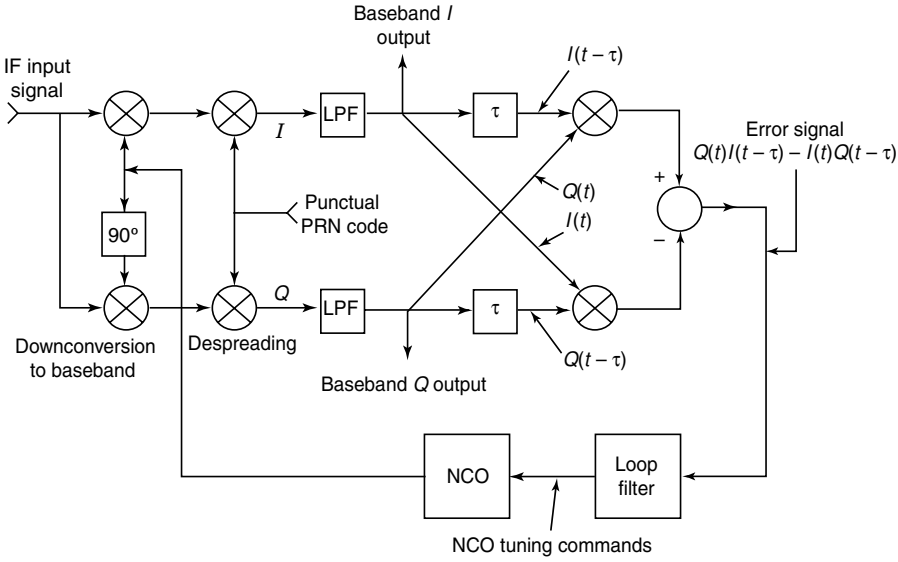


Fig. 3.16 *Frequency-lock loop.*

(< 1 Hz/s) that phase tracking error is negligible. On the other hand, receivers designed for high dynamics (i.e., missiles) will sometimes use third-order or even higher order PLLs to avoid loss of lock due to the large accelerations encountered.

The price paid for using higher order PLLs is a somewhat less robust performance in the presence of noise. If independent measurements of platform dynamics are available (such as accelerometer or INS outputs), they can be used to aid the PLL by reducing stress on the loop. This can be advantageous because it often makes the use of higher order loops unnecessary.

3.4.7 Bit Synchronization

Before bit synchronization can occur, the PLL must be locked to the GPS signal. This is accomplished by running the Costas loop in a 1-ms integration mode where each interval of integration is over one period of the C/A-code, starting and ending at the code epoch. Since the 50-Hz biphase data bit transitions can occur only at code epochs, there can be no bit transitions while integration is taking place. When the PLL achieves lock, the output of the I integrator will be a sequence of values occurring once per millisecond or 20 times per data bit. With nominal signal levels the processing gain of the integrator is sufficient to guarantee with high probability that the polarity of the 20 integrator outputs will remain constant during each data bit interval and will change polarity when a data bit transition occurs.

A simple method of bit synchronization is to clock a modulo 20 counter with the epochs of the receiver-generated reference C/A-code and record the count each time the polarity of the I integrator output changes. A histogram of the frequency of each

count is constructed, and the count having the highest frequency identifies the epochs that mark the data bit boundaries.

3.4.8 Data Bit Demodulation

Once bit synchronization has been achieved, demodulation of the data bits can occur. As previously described, many GPS receivers demodulate the data by integrating the in-phase (I) component of the baseband phasor generated by a Costas loop, which tracks the carrier phase. Each data bit is generated by integrating the I component over a 20-ms interval from one data bit boundary to the next. The Costas loop causes a polarity ambiguity of the data bits that can be resolved by observation of the subframe preamble in the navigation message data.

3.5 EXTRACTION OF INFORMATION FOR NAVIGATION SOLUTION

After data demodulation has been performed, the essential information in the signal needed for the navigation solution is at hand. This information can be classified into the following three categories:

1. the information needed to determine signal transmission time,
2. the information needed to establish the position and velocity of each satellite, and
3. the various pseudorange and Doppler measurements made by the receiver.

3.5.1 Signal Transmission Time Information

In our previous discussion of the Z-count, we saw that the receiver can establish the time of transmission of the beginning of each subframe of the signal and of the corresponding C/A-code epoch that coincides with it. Since the epochs are transmitted precisely 1 ms apart, the receiver labels subsequent C/A-code epochs merely by counting them. This enables the determination of the transmission time of *any* part of the signal by a process to be described later.

3.5.2 Ephemeris Data

The ephemeris data permits the position and velocity of each satellite to be computed at the signal transmission time. The calculations have previously been outlined in Table 3.2.

3.5.3 Pseudorange Measurements Using C/A-Code

In its basic form, finding the three-dimensional position of a user would consist of determining the *range*, that is, the distance of the user from each of three or more

satellites having known positions in space, and mathematically solving for a point in space where that set of ranges would occur. The range to each satellite can be determined by measuring how long it takes for the signal to propagate from the satellite to the receiver and multiplying the propagation time by the speed of light.

Unfortunately, however, this method of computing range would require very accurate synchronization of the satellite and receiver clocks used for the time measurements. GPS satellites use very accurate and stable atomic clocks, but it is economically infeasible to provide a comparable clock in a receiver. The problem of clock synchronization is circumvented in GPS by treating the receiver clock error as an additional unknown in the navigation equations and using measurements from an additional satellite to provide enough equations for a solution for time as well as for position. Thus the receiver can use an inexpensive clock for measuring time. Such an approach leads to perhaps the most fundamental measurement made by a GPS receiver: the *pseudorange* measurement, computed as

$$\rho = c(t_{\text{rcve}} - t_{\text{xmit}}), \tag{3.15}$$

where t_{rcve} is the time at which a specific, identifiable portion of the signal is received, t_{xmit} is the time at which that same portion of the signal is transmitted, and c is the speed of light (2.99792458×10^8 m/s). It is important to note that t_{rcve} is measured according to the receiver clock, which may have a large time error, but t_{xmit} is in terms of GPS time, which in turn is SV (spacecraft vehicle) time plus a time correction transmitted by the satellite. If the receiver clock were synchronized to GPS time, then the pseudorange measurement would in fact be the range to the satellite.

Figure 3.17 shows the pseudorange measurement concept with four satellites, which is the minimum number needed for a three-dimensional position solution without synchronized clocks. The raw measurements are simultaneous snapshots at time t_{rcve} of the states of the received C/A-codes from all satellites. This is accomplished indirectly by observation of the receiver-generated code state from each code tracking loop. For purposes of simplicity we define the *state* of the C/A-code to be the number of chips (including the fractional part) that have occured since the last code epoch. Thus the state is a real number in the interval [0,1023).

As previously discussed, the receiver has been able to tag each code epoch with its GPS transmission time. Thus, it is a relatively simple matter to compute the time of transmission of the code state that is received at time t_{rcve}. For a given satellite let t_e denote the GPS transmission time of the last code epoch received prior to t_{rcve}, let X denote the code state observed at t_{rcve}, and let c_r denote the C/A-code chipping rate (1.023×10^6 chips/s). Then the transmission time of that code state is

$$t_{\text{xmit}} = t_e + \frac{X}{c_r}. \tag{3.16}$$

Basic Positioning Equations If pseudorange measurements can be made from at least four satellites, enough information exists to solve for the unknown position (X, Y, Z) of the GPS user and for the receiver clock error b_c (often called the clock *bias*). The equations are set up by equating the measured pseudorange to each

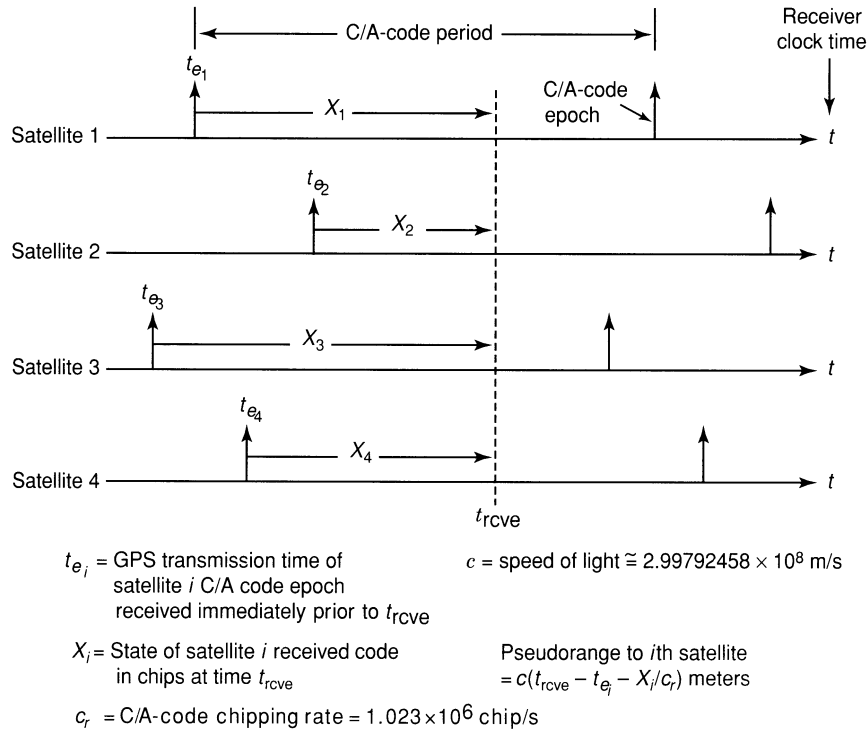


Fig. 3.17 *Pseudorange measurement concept.*

satellite with the corresponding unknown user-to-satellite distance plus the distance error due to receiver clock bias:

$$\begin{aligned}
\rho_1 &= \sqrt{(x_1 - X)^2 + (y_1 - Y)^2 + (z_1 - Z)^2} + C_b, \\
\rho_2 &= \sqrt{(x_2 - X)^2 + (y_2 - Y)^2 + (z_2 - Z)^2} + C_b, \\
&\vdots \\
\rho_n &= \sqrt{(x_n - X)^2 + (y_n - Y)^2 + (z_n - Z)^2} + C_b,
\end{aligned} \tag{3.17}$$

where ρ_i denotes the measured pseudorange of the ith satellite whose position in ECEF coordinates at t_{xmit} is (x_i, y_i, z_i) and $n \geq 4$ is the number of satellites observed. The unknowns in this nonlinear system of equations are the user position (X, Y, Z) in ECEF coordinates and the receiver clock bias C_b.

3.5.4 Pseudorange Measurements Using Carrier Phase

Although pseudorange measurements using the C/A-code are the most commonly employed, a much higher level of measurement precision can be obtained by measuring the received phase of the GPS L_1 or L_2 carrier. Because the carrier waveform has a very short period (6.35×10^{-10} s at the L_1 frequency), the noise-induced error in measuring signal delay by means of phase measurements is typically 10–100 times smaller than that encountered in code delay measurements.

However, carrier phase measurements are highly ambiguous because phase measurements are simply modulo 2π numbers. Without further information such measurements determine only the fractional part of the pseudorange when measured in carrier wavelengths. Additional measurements are required to effect *ambiguity resolution*, in which the integer number of wavelengths in the pseudorange measurement can be determined. The relation between the measured signal phases ϕ_i and the unambiguous pseudoranges ρ_i can be expressed as

$$\begin{aligned} \rho_1 &= \lambda\left(\frac{\phi_1}{2\pi} + k_1\right), \\ \rho_2 &= \lambda\left(\frac{\phi_2}{2\pi} + k_2\right), \\ &\vdots \\ \rho_n &= \lambda\left(\frac{\phi_n}{2\pi} + k_n\right), \end{aligned} \tag{3.18}$$

where n is the number of satellites observed, λ is the carrier wavelength, and k_i is the unknown integral number of wavelengths contained in the pseudorange. The additional measurements required for determination of the k_i may include C/A- and/or P(Y)-code pseudorange measurements from the same satellites used for the phase measurements. Since the code measurements are unambiguous, they significantly narrow the range of admissible integer values for the k_i. Additionally, phase measurements made on both the L_1 and L_2 signals can be used to obtain a virtual carrier frequency equal to the difference of the two carrier frequencies ($1575.42 - 1227.60 = 347.82$ MHz). The 86.3-cm wavelength of this virtual carrier thins out the density of pseudorange ambiguities by a factor of about 4.5, making the ambiguity resolution process much easier. Redundant code and phase measurements from extra satellites can also be used to aid the process; the extra code measurements further narrow the range of admissible integer values for the k_i, and the extra phase measurements thin out the phase ambiguity density by virtue of satellite geometry.

Because of unpredictable variations in propagation delay of the code and carrier due to the ionosphere and other error sources, it is all but impossible to obtain ambiguity resolution with single-receiver positioning. Therefore, carrier phase measurements are almost always relegated to high-accuracy applications in which

such errors are canceled out by differential operation with an additional receiver (base station).

In GPS receivers, carrier phase is usually measured by sampling the phase of the reference oscillator of the carrier tracking loop. In most receivers this oscillator is an NCO that tracks the phase of the incoming signal at a relatively low intermediate frequency. The signal phase is preserved when the incoming signal is frequency downconverted. The NCO is designed to provide a digital output of its instantaneous phase in response to a sampling signal. Phase-based pseudorange measurements are made by simultaneously sampling at time t_{rcve} the phases of the NCOs tracking the various satellites. As with all receiver measurements, the reference for the phase measurements is the receiver's clock reference oscillator.

3.5.5 Carrier Doppler Measurement

Measurement of the received carrier frequency provides information that can be used to determine the velocity vector of the user. Although this could be done by forming differences of code-based position estimates, frequency measurement is inherently much more accurate and has faster response time in the presence of user dynamics. The equations relating the measurements of Doppler shift to the user velocity are

$$\begin{aligned} f_{d1} &= \frac{1}{\lambda}(\mathbf{v}\cdot\mathbf{u}_1 - \mathbf{v}_1\cdot\mathbf{u}_1) + f_b, \\ f_{d2} &= \frac{1}{\lambda}(\mathbf{v}\cdot\mathbf{u}_2 - \mathbf{v}_2\cdot\mathbf{u}_2) + f_b, \\ &\vdots \\ f_{dn} &= \frac{1}{\lambda}(\mathbf{v}\cdot\mathbf{u}_{\text{n}} - \mathbf{v}_{\text{n}}\cdot\mathbf{u}_{\text{n}}) + f_b, \end{aligned} \tag{3.19}$$

where the unknowns are the user velocity vector $\mathbf{v} = (v_x, v_y, v_z)$ and the receiver reference clock frequency error f_b in hertz and the known quantities are the carrier wavelength λ and the measured Doppler shifts f_{di} in hertz, satellite velocity vectors $\mathbf{v}_i$, and unit satellite direction vectors $\mathbf{u}_i$ (pointing from the receiver antenna toward the satellite antenna) for each satellite index i. The unit vectors $\mathbf{u}_i$ are determined by computing the user-to-ith satellite displacement vectors $\boldsymbol{\rho}_i$ and normalizing them to unit length:

$$\begin{aligned} \boldsymbol{\rho}_i &= [(x_i - X), (y_i - Y), (z_i - Z)]^T, \\ \mathbf{u}_i &= \frac{\boldsymbol{\rho}_i}{|\boldsymbol{\rho}_i|}, \end{aligned} \tag{3.20}$$

In these expressions the ith satellite position (x_i, y_i, z_i) at time t_{xmit} is computed from the ephemeris data and the user position (X, Y, Z) can be determined from solution of the basic positioning equations using the C/A- or P(Y)-codes.

In GPS receivers, the Doppler measurements f_{di} are usually derived by sampling the frequency setting of the NCO (Fig. 3.15) that tracks the phase of the incoming signal. An alternate method is to count the output cycles of the NCO over a relatively short time period, perhaps 1 s or less. However, in either case, the measured Doppler shift is not the raw measurement itself, but the deviation from what the raw NCO measurement would be without any signal Doppler shift, assuming that the receiver reference clock oscillator had no error.

3.5.6 Integrated Doppler Measurements

Integrated Doppler can be defined as the number of carrier cycles of Doppler shift that have occurred in a given interval $[t_0, t]$. For the ith satellite the relation between integrated Doppler F_{di} and Doppler shift f_{di} is given by

$$F_{di}(t) = \int_{t_0}^{t} f_{di}(t)\, dt. \tag{3.21}$$

However, accurate calculation of integrated Doppler according to this relation would require that the Doppler measurement be a continuous function of time. Instead, GPS receivers take advantage of the fact that by simply observing the output of the NCO in the carrier tracking loop (Fig. 3.15), the number of cycles that have occurred since initial time t_0 can be counted directly.

Integrated Doppler measurements have several uses:

1. *Accurate Measurement of Receiver Displacement over Time*. Motion of the receiver causes a change in the Doppler shift of the incoming signal. Thus, by counting carrier cycles to obtain integrated Doppler, precise estimates of the *change* in position (*delta position*) of the user over a given time interval can be obtained. The error in these estimates is much smaller than the error in establishing the absolute position using the C/A- or P(Y)-codes. The capability of accurately measuring changes in position is used extensively in *real-time kinematic* surveying with differential GPS. In such applications the user needs to determine the locations of many points in a given land area with great accuracy (perhaps to within a few centimeters). When the receiver is first turned on, it may take a relatively long time to acquire the satellites, to make both code and phase pseudorange measurements, and to resolve phase ambiguities so that the location of the first surveyed point can be determined. However, once this is done, the relative displacements of the remaining points can be found very rapidly and accurately by transporting the receiver from point to point while it continues to make integrated Doppler measurements.
2. *Positioning Based on Received Signal Phase Trajectories*. In another form of differential GPS, a fixed receiver is used to measure the integrated Doppler function, or *phase trajectory curve*, from each satellite over relatively long periods of time (perhaps 5–20 min). The position of the receiver can be

determined by solving a system of equations relating the shape of the trajectories to the receiver location. The accuracy of this positioning technique, typically within a few decimeters, is not as good as that obtained with carrier phase pseudoranging but has the advantage that there is no phase ambiguity. Some hand-held GPS receivers employ this technique to obtain relatively good positioning accuracy at low cost.

3. *Carrier Rate Aiding for the Code Tracking Loop*. In the code tracking loop, proper code alignment is achieved by using observations of the loop error signal to determine whether to advance or retard the state of the otherwise free-running receiver-generated code replica. Because the error signal is relatively noisy, a narrow loop bandwidth is desirable to maintain good pseudoranging accuracy. However, this degrades the ability of the loop to maintain accurate tracking in applications where the receiver is subject to substantial accelerations. The difficulty can be substantially mitigated with *carrier rate aiding*, in which the primary code advance/retard commands are not derived from the code discriminator (early–late correlator) error signal but instead are derived from the Doppler-induced accumulation of carrier cycles in the integrated Doppler function. Since there are 1540 carrier cycles per C/A-code chip, the code will therefore be advanced by precisely one chip for every 1540 cycles of accumulated count of integrated Doppler. The advantage of this approach is that, even in the presence of dynamics, the integrated Doppler can track the received code *rate* very accurately. As a consequence, the error signal from the code discriminator is "decoupled" from the dynamics and can be used for very small and infrequent adjustments to the code generator.

3.6 THEORETICAL CONSIDERATIONS IN PSEUDORANGE AND FREQUENCY ESTIMATION

In a well-designed GPS receiver the major source of measurement error within the receiver is thermal noise, and it is useful to know the best performance that is theoretically possible in its presence. Theoretical bounds on errors in estimating code-based and carrier-based pseudorange, as well as in Doppler frequency estimates, have been developed within an interesting branch of mathematical statistics called *estimation theory*. There it is seen that a powerful estimation approach called the *method of maximum likelihood* (ML) can often approach theoretically optimum performance (see Section 7.2.4). ML estimates of pseudorange (using either the code or the carrier) and frequency are *unbiased*, which means that the expected value of the error due to random noise is zero.

An important lower bound on the error variance of any unbiased estimator is provided by the *Cramer–Rao bound*, and any estimator that reaches this lower limit is called a *minimum-variance unbiased estimator* (MVUE). It can be shown that at the typical SNRs encountered in GPS, ML estimates of code pseudorange, carrier

pseudorange, and carrier frequency are all MVUEs. Thus, these estimators are optimal in the sense that no unbiased estimator has a smaller error variance [123].

3.6.1 Theoretical Versus Realizable Code-Based Pseudoranging Performance

It can be shown that a ML estimate τ_{ML} of signal delay based on code measurements is obtained by maximizing the cross-correlation of the received code $c_r(t)$ with a reference code $c_{\mathrm{ref}}(t)$ that is an identical replica (including bandlimiting) of the received code:

$$\tau_{\mathrm{ML}} = \max_{\tau} \int_0^T c_r(t) c_{\mathrm{ref}}(t-\tau), \tag{3.22}$$

where $[0, T]$ is the signal observation interval. Here we assume coherent processing for purposes of simplicity. This estimator is a MVUE, and it can be shown that the error variance of τ_{ML} (which equals the Cramer–Rao bound) is

$$\sigma^2_{\tau_{\mathrm{ML}}} = \frac{N_0}{2\int_0^T [c_r'(t)]^2\, dt}. \tag{3.23}$$

This is a fundamental relation that in temporal terms states that the error variance is proportional to the power spectral density N_0 of the noise and inversely proportional to the integrated square of the derivative of the received code waveform. It is generally more convenient to use an expression for the standard deviation, rather than the variance, of delay error, in terms of the bandwidth of the C/A-code. The following is derived in [126]:

$$\sigma_{\tau_{\mathrm{ML}}} = \frac{3.444 \times 10^{-4}}{\sqrt{(C/N_0)WT}}. \tag{3.24}$$

In this expression it is assumed that the received code waveform has been bandlimited by an ideal low-pass filter with one-sided bandwidth W. The signal observation time is still denoted by T, and C/N_0 is the ratio of power in the code waveform to the one-sided power spectral density of the noise. A similar expression is obtained for the error variance using the P(Y)-code, except the numerator is $\sqrt{10}$ times smaller.

Figure 3.18 shows the theoretically achievable pseudoranging error using the C/A-code as a function of signal observation time for various values of C/N_0. The error is surprisingly small if the code bandwidth is sufficiently large. As an example, for a moderately strong signal with $C/N_0 = 31,623$ (45 dB-Hz), a bandwidth $W = 10$ MHz, and a signal observation time of 1 s, the standard deviation of the ML delay estimate obtained from the above formula is about 6.2×10^{-10} s, corresponding to 18.6 cm after multiplying by the speed of light.

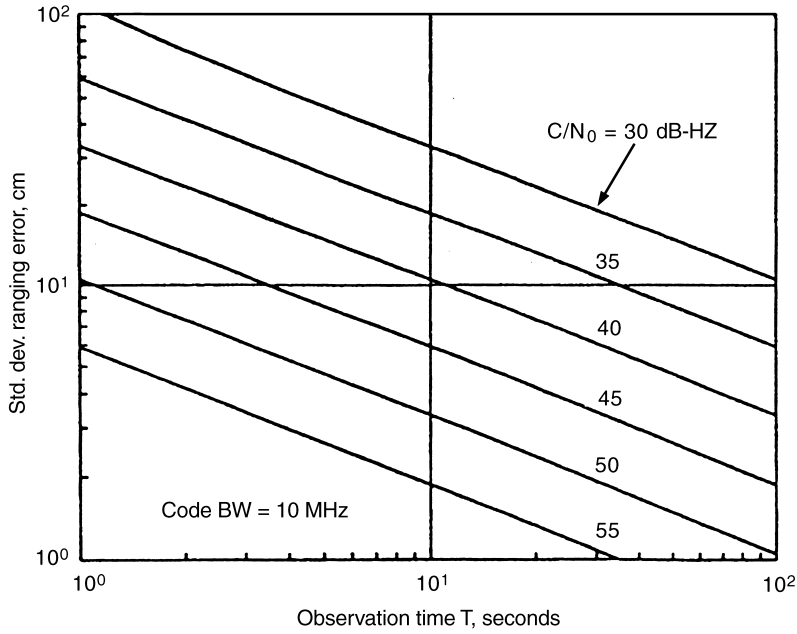


Fig. 3.18 *Theoretically achievable C/A-code pseudoranging error.*

Code Pseudoranging Performance of Typical Receivers Most GPS receivers approximate the ML estimator by correlating the incoming signal with an ideal code waveform that does not include bandlimiting effects and use early and late correlators in the code tracking loop that straddle the location of the correlation function peak rather than find its actual location. As a result, the code tracking error can be significantly larger than the theoretical minimum discussed above. One-chip early–late spacing of the tracking correlators was common practice for the several decades preceding the early 1990s. It is somewhat surprising that the substantial amount of performance degradation resulting from this approach went unnoticed for so long. Not until 1992 was it widely known that significant error reduction could be obtained by narrowing the spacing down to 0.1–0.2 C/A-code chips in combination with a large precorrelation bandwidth. Details of this approach, dubbed *narrow correlator technology*, can be found in [121]. With narrow early–late spacing the random noises on the early and late correlator outputs become highly correlated and therefore tend to cancel when the difference error signal is formed. A large precorrelation bandwidth sharpens the peak of the correlation function so that the closely spaced early and late correlators can still operate on the high-slope portion of the correlation function, thus preserving SNR in the loop.

It can be shown that the variance of the code tracking error continues to decrease as the early–late spacing approaches zero but approaches a limiting value. Some researchers are aware that forming a difference signal with early and late correlators

is mathematically equivalent to a single correlation with the difference of the early and late codes, which in the limit (as the early–late spacing goes to zero) becomes equivalent to polarity modulated sampling of the received code at the punctual reference code transitions and summing the sample values to produce the loop error signal. Some GPS receivers already put this principle into practice.

Figure 3.19, found in [126], compares the performance of several correlator schemes, including the narrow correlator, with theoretical limits. It is seen that the narrow correlator approaches the theoretical performance limit given by the Cramer–Rao bound as the early–late spacing $2e$ approaches zero.

3.6.2 Theoretical Error Bounds for Carrier-Based Pseudoranging

At typical GPS signal-to-noise ratios the ML estimate τ_{ML} of signal delay using carrier phase is a MVUE, and it can be shown that the error standard deviation is

$$\sigma_{\tau_{\mathrm{ML}}} = \frac{1}{2\pi f_c \sqrt{2(C/N_0)T}} \tag{3.25}$$

where f_c is the GPS carrier frequency, and C/N_0 and T have the same meaning as in Eq. 3.24. This result is also reasonably accurate for a carrier tracking loop if T is set equal to the reciprocal of the loop bandwidth. As an example of the much greater

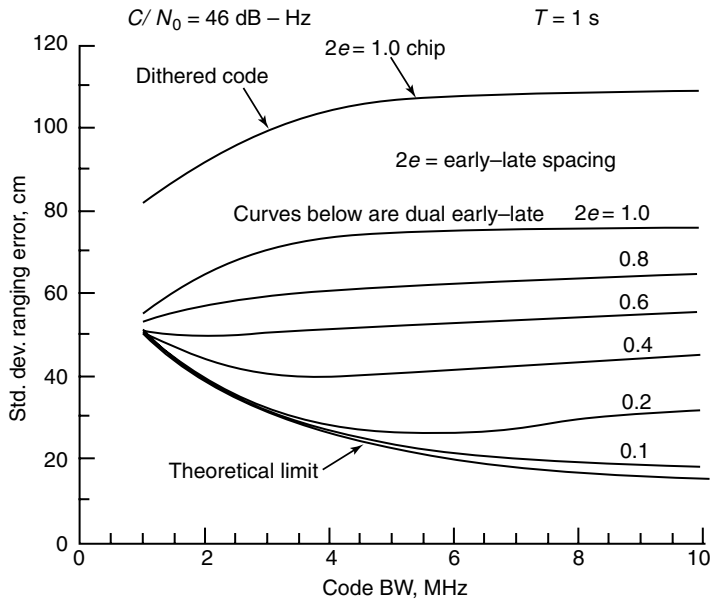


Fig. 3.19 *Performance of various correlators.*

accuracy of carrier phase pseudoranging compared with code pseudoranging, a signal at $C/N_0 = 45$ dB-Hz observed for 1 s can theoretically yield an error standard deviation of 4×10^{-13} s, which corresponds to only 0.12 mm. However, typical errors of 1–3 mm are experienced in most receivers due to random phase jitter in the reference oscillator.

3.6.3 Theoretical Error Bounds for Frequency Measurement

The ML estimate f_{ML} of the carrier frequency is also a MVUE, and the expression for its error standard deviation is

$$\sigma_{f_{\mathrm{ML}}} = \sqrt{\frac{3}{2\pi^2 (C/N_0) T^3}} \tag{3.26}$$

A 1-s observation of a despread GPS carrier with $C/N_0 = 45$ dB-Hz yields a theoretical error standard deviation of about 0.002 Hz, which could also be obtained with a phase tracking loop having a bandwidth of 1 Hz. As in the case of phase estimation, however, phase jitter in the receiver reference oscillator yields frequency error standard deviations from 0.05 to 0.1 Hz.

3.7 MODERNIZATION OF GPS

Since it was declared fully operational in April 1995, the GPS has been operating continuously with 24 or more operational satellites, and user equipment has evolved rapidly, especially in the civil sector. As a result, radically improved levels of performance have been reached in positioning, navigation, and time transfer. However, the availability of GPS has also spawned new and demanding applications that reveal certain shortcomings of the present system. Therefore, within the last decade numerous governmental and civilian committees have investigated the needs and deficiencies of the existing system in order to conceive a plan for GPS modernization.

The modernization of GPS is a difficult and complex task that requires trade-offs in many areas. Major issues include spectrum needs and availability, military and civil performance, signal integrity and availability, financing and cost containment, and potential competition from Europe's Galileo system. However, after many years of hard work it now appears that critical issues have been resolved. Major decisions have been made for the incorporation of new civil frequencies, new civil and military signals, and higher transmitted power levels.

3.7.1 Deficiencies of the Current System

The changes that are planned for GPS address the following needs:

1. *Civil users need two-frequency ionospheric correction capability in autonomous operation*. Since only the encrypted P-code appears at the L_2 frequency, civil users are denied the benefit of dual-frequency operation to remove ionospheric range error in autonomous (i.e., nondifferential) operation. Although special techniques such as signal squaring can be used to recover the L_2 carrier, the P-code waveform is lost and the SNR is dramatically reduced. Consequently, such techniques are of little value to the civil user in reducing ionospheric range error.
2. *Signal blockage and attenuation are often encountered*. In some applications heavy foliage in wooded areas can attenuate the signal to an unusable level. In certain locations, such as in urban canyons, a satellite signal can be completely blocked by buildings or other features of the terrain. In such situations there will not be enough visible satellites to obtain a navigation solution. New applications, such as emergency 911 position location by GPS receivers embedded in cellular telephone handsets, will require reliable operation of GPS receivers inside buildings, despite heavy signal attenuation due to roof, floors, and walls. Weak signals are difficult to acquire and track.
3. *Ability to resolve ambiguities in phase measurements needs improvement*. High-accuracy differential positioning at the centimeter level by civil users requires rapid and reliable resolution of ambiguities in phase measurements. Ambiguity resolution with single-frequency (L_1) receivers generally requires a sufficient length of time for the satellite geometry to change significantly. Performance is improved with dual-frequency receivers. However, the effective SNR of the L_2 signal is dramatically reduced because the encrypted P-code cannot be despread by the civil user.
4. *Selective Availability is detrimental to performance in civil applications*. SA has been suspended as of 8 pm. EDT on May 1, 2000.The degradation in autonomous positioning performance by SA (about 50 m RMS error) is of concern in many civil applications requiring the full accuracy of which GPS is capable. A prime example is vehicle tracking systems in which an accuracy of 5–10 m RMS is needed to establish the correct city street on which a vehicle is located. Moreover, many civil and military committees have found that a military adversary can easily mitigate errors due to SA by using differential positioning. In the civil sector, a large and costly infrastructure has developed to overcome its effects.
5. *Improvements in system integrity and robustness are needed*. In applications involving public safety the integrity of the current system is judged to be marginal. This is particularly true in aviation landing systems that demand the presence of an adequate number of healthy satellite signals and functional cross-checks during precision approaches. Additional satellites and higher transmitted power levels are desirable in this context.
6. *Improvement is needed in multipath mitigation capability.* Multipath remains a dominant source of GPS positioning error and cannot be removed by differential techniques. Although certain mitigation techniques, such as multi-

path mitigation technology (MMT), approach theoretical performance limits for in-receiver processing, the required processing adds to receiver costs. In contrast, effective multipath rejection could be made available to all receivers by using new GPS signal designs.

7. *The military needs improved acquisition capability and jamming immunity.* Because the P(Y)-code has an extremely long period (seven days), it is difficult to acquire unless some knowledge of the code timing is known. In the current system P(Y) timing information is supplied by the HOW. However, to read the HOW, the C/A-code must first be acquired to gain access to the navigation message. Unfortunately, the C/A-code is relatively susceptible to jamming, which would seriously impair the ability of a military receiver to acquire the P(Y) code. It would be far better if direct acquisition of a high-performance code were possible.

3.7.2 Elements of the Modernized GPS

Civil Spectrum Modernization The upper part of Fig. 3.20 outlines the current civil GPS signal spectrum and the additional codes and signal frequencies in the plans for modernization. The major elements are as follows:

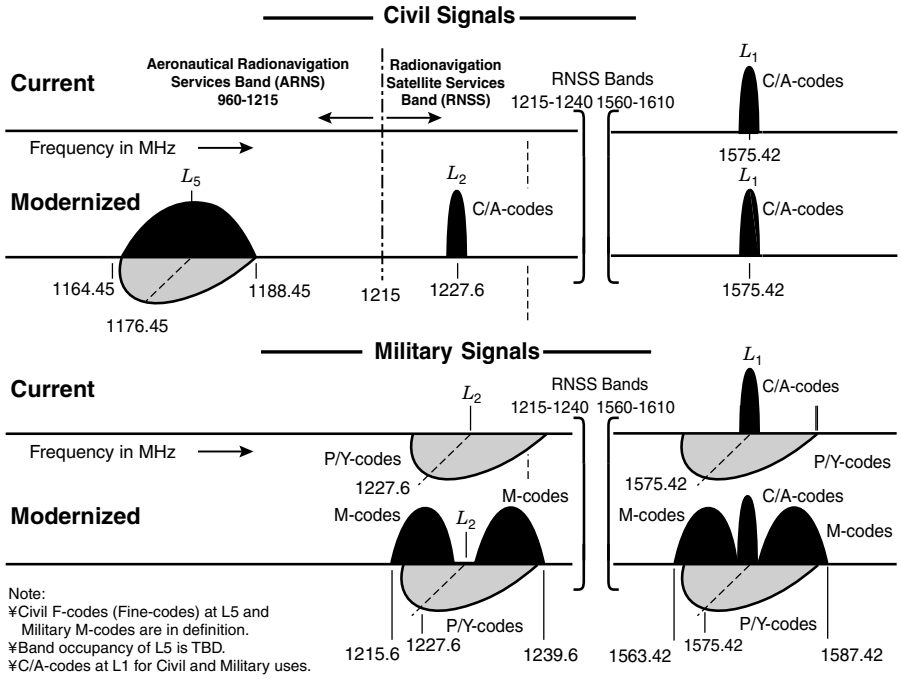


Fig. 3.20 *Existing and modernized GPS signal spectrum.*

1. *C/A Codes on the L_2 Frequency.* Each satellite will transmit the same C/A-code on the L_2 frequency (1227.6 MHz) as it currently does on the L_1 frequency (1575.42 MHz), which will result in the following improvements for civil users:

   - *Two-frequency ionospheric error correction becomes possible*. The $1/f^2$ dispersive delay characteristic of the ionosphere can be used to accurately estimate errors in propagation delay.
   - *Carrier phase ambiguity resolution will be significantly improved*. The accessibility of the L_1 and L_2 carriers provides "wide lane" phase measurements having ambiguities that are much easier to resolve.
   - *The additional L_2 signal will improve robustness in acquisition and tracking and improve C/A-code positioning accuracy.*

   Retention of the current C/A-codes on the L_1 frequency will satisfy a legacy requirement for older civil GPS receivers.

2. *A New L_5 Signal Modulated by a New Code Structure*. Although the use of the L_1 and L_2 frequencies can satisfy most civil users, there are concerns that the L_2 frequency band may be subject to unacceptable levels of interference for applications involving public safety, such as aviation. The potential for interference arises because the International Telecommunications Union (ITU) has authorized this band on a coprimary basis with radiolocation services, such as high-power radars. As a result of FAA requests, the Department of Transportation and Department of Defense have determined a new civil GPS frequency, called L_5, in the Aeronautical Radio Navigation System band at 1176.45 MHz. To gain maximum performance, the L_5 spread-spectrum codes will have a higher chipping rate and longer period than the C/A-codes. Proposed codes have a 10.23-megachip/s chipping rate and a period of 10230 chips. Additionally, the plan is to transmit two signals in phase quadrature, one of which will not carry data modulation. The L_5 signal will provide the following system improvements:

   - *Ranging accuracy will improve*. Pseudorange errors due to random noise will be reduced below levels obtainable with the C/A-codes due to the larger bandwidth of the proposed codes. As a consequence, both code-based positioning accuracy and phase ambiguity resolution performance will improve.
   - *Errors due to multipath will be reduced*. The larger bandwidth of the new codes will sharpen the peak of the code autocorrelation function, thereby reducing the shift in the peak due to multipath signal components.
   - *Carrier phase tracking will improve*. Weak-signal phase tracking performance of GPS receivers is severely limited by the necessity of using a Costas (or equivalent type) PLL to remove carrier phase reversals of the

data modulation. Such loops rapidly degrade below a certain threshold (about 25–30 dB-Hz) because truly coherent integration of the carrier phase is limited to the 20-ms data bit length. In contrast, the "data-free" quadrature component of the L_5 signal will permit coherent integration of the carrier for arbitrarily long periods, which will permit better phase tracking accuracy and lower tracking thresholds.

- *Weak-signal code acquisition and tracking will be enhanced*. The "data-free" component of the L_5 signal will also permit new levels of positioning capability with very weak signals. Acquisition will be improved because fully coherent integration times longer than 20-ms will be possible. Code tracking will also improve by virtue of better carrier phase tracking for the purpose of code rate aiding.
- *The L_5 signal will further support rapid and reliable carrier phase ambiguity resolution*. Because the difference between the L_5 and L_2 frequencies is only 51.15 MHz as opposed to the 347.82 MHz difference between the L_1 and L_2 frequencies, carrier phase ambiguity will be possible using an extra-wide lane width of about 5.9 m instead of 0.86 m. The inevitable result will be virtually instantaneous ambiguity resolution, a critical issue in high-performance real-time kinematic modes of GPS positioning.
- *The codes will be better isolated from each other*. The longer length of the L_5 codes will reduce the size of cross-correlation between codes from different satellites, thus minimizing the probability of locking onto the wrong code during acquisition, even at the increased power levels of the modernized signals.

3. *Higher Transmitted Power Levels*. For safety, cost, and performance, many in the GPS community are advocating a general increase of 3–6 dB in the signal power at at all three civil frequencies.

Military Spectrum Modernization The lower part of Fig. 3.20 shows the current and modernized spectrum used by the military community. The current signals consist of C/A-codes and P/Y-codes transmitted in quadrature in the L_1 band and only P/Y-codes in the L_2 band. The primary elements of the modernized spectrum are as follows:

1. *All existing signals will be retained for legacy purposes*.
2. *New M-codes will also be transmitted in both the L_1 and L_2 bands*. At the time of this writing, these codes have not been finalized, but they will have a bit rate from 3–8 megachips/s and modulate a subcarrier whose frequency will lie somewhere between 6 and 9 MHz. The resulting spectrum has two lobes, one on each side of the band center, and for this reason the M-codes are sometimes called "split-spectrum" codes. They will be transmitted in the same quadrature channel as the C/A-codes, that is, in phase quadrature with the P(Y)

codes. Civil use of these codes will be denied by as yet unannounced encryption techniques. The M-codes will provide the following advantages to military users:

- *Direct acquisition of the M-codes will be possible*. The design of these codes will eliminate the need to first acquire the L_1 C/A-code with its relatively high vulnerability to jamming.
- *Better ranging accuracy will result*. As can be seen in Fig. 3.20, the M-codes have significantly more energy near the edges of the bands, with a relatively small amount of energy near band center. Since most of the C/A-code power is near band center, potential interference between the codes is mitigated. The effective bandwidth of the M-codes is much larger than that of the P(Y)-codes, which concentrate most of their power near the L_1 or L_2 carrier. Because of the modulated subcarrier, the autocorrelation function of the M-codes has, not just one peak, but several peaks spaced one subcarrier period apart, with the largest at the center. The modulated subcarrier will cause the central peak to be significantly sharpened, significantly reducing pseudorange measurement error.
- *Error due to multipath will be reduced*. The sharp central peak of the M-code autocorrelation function is less susceptible to shifting in the presence of multipath correlation function components.

3.7.3 Modernization and System Performance Timetables

The schedule for modernization of the GPS has not been fully developed, but it appears that it will occur over a period of 10–20 years. Figure 3.21 is an estimate of how positioning accuracy might improve for civil and military receivers in various modes of operation over the 2000–2010 time period.

3.8 GPS SATELLITE POSITION CALCULATIONS

The ephemeris parameters and algorithms used for computing satellite positions are given in Tables 3.1 and 3.2, respectively.

The interface between the GPS space and user segments consists of two radio-frequency (RF) links L_1 and L_2. The carriers of the L-band links are modulated by up to two bit trains, each of which normally is a composite generated by the modulo 2 addition of a PRN ranging code and the downlink system data. Utilizing these links, the space vehicles of the GPS space segment should provide continuous earth coverage for signals that provide to the user segment the ranging codes and system data needed to accomplish the GPS navigation mission. These signals are available to a suitably equipped user with RF visibility to a space vehicle. Therefore, the GPS users continuously receive navigation information from the space vehicles in the form of modulated data bits.

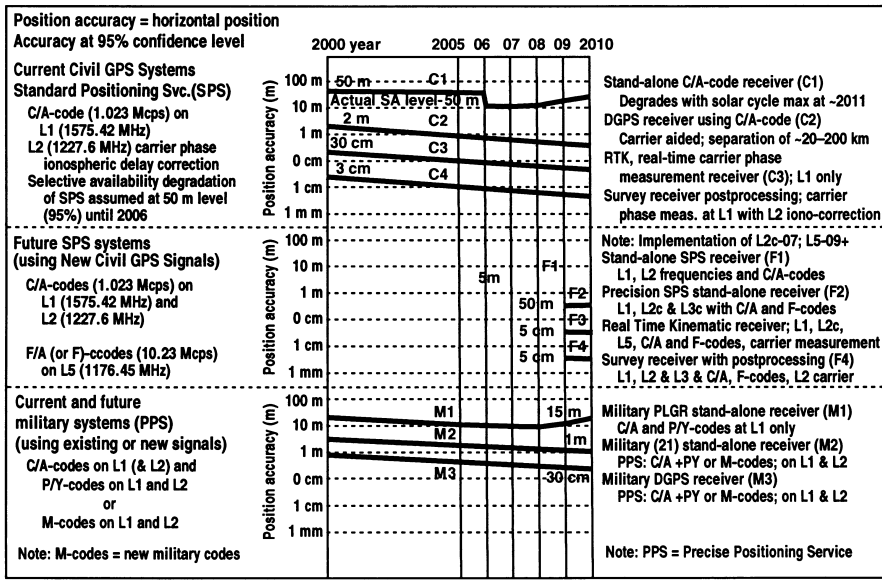


Fig. 3.21 *Estimated accuracy improvements and schedule.*

The received information is computed and controlled by the control segment and includes the satellite's time, its clock correction and ephemeris parameters, almanacs and health for all GPS space vehicles, and text messages. The precise position and clock offset of the space vehicle antenna phase center in the ECEF coordinates can be computed by receiving this information.

The ephemeris parameters describe the orbit during the interval of time (at least 1 h) for which the parameters are transmitted. This representation model is characterized by a set of parameters that is an extension (including drag) to the Keplerian orbital parameters. They also describe the ephemeris for an additional interval of time (at least one-half hour) to allow time for the user to receive the parameters for the new interval of time. The definitions of the parameters are given in Table 3.1.

The age of data word (AODE) provides a confidence level in the ephemeris representation parameters. The AODE represents the time difference (age) between the reference time (t_{0e}) and the time of the last measurement update (t_L) used to estimate the representation parameters.

The ECEF coordinates for the phase center of the satellite's antennas can be calculated using a variation of the equations shown in Table 3.2. In this table, the time t is the GPS system time at the time of transmission, that is, GPS time corrected for transit time (range/speed of light). Further, t_k is the actual total time difference between the time t and the epoch time t_{0e} and must account for beginning- or end-of-week crossovers. That is, if t_k is greater than 302,400 s, subtract 604,800 s from t_k. If t_k is less than $-302{,}400$ s, add 604,800 s to t_k. Also, note that Kepler's equation for eccentric anomaly is nonlinear in E_k. It is impractical to solve for E_k any way except

approximation. Standard practice is to solve this equation by the Newton–Raphson second-order method explicitly and then use the resulting value of E_k to calculate true anomaly. The satellite's antenna phase center position is very sensitive to small perturbations in most ephemeris parameters. The sensitivity of position to the parameters $\sqrt{a}$, C_{rc}, and C_{rs} is about 1 m/m. This sensitivity to angular parameters is on the order of 10^8 m/semicircle and to the angular rate parameters on the order of 10^{12} m/semicircle/s. Because of this extreme sensitivity to angular perturbations, the required value of π (a mathematical constant, the ratio of a circle's circumference to its diameter) used in the curve fit is given here:

$$\pi = 3.1415926535898.$$

The user must correct the time received from the space vehicle in seconds with the equation

$$t = t_{\text{sv}} - \Delta t_{\text{sv}}, \tag{3.27}$$

where t = GPS system time (s)
t_{sv} = effective SV PRN code phase time at message transmission time (s)
Δt_{sv} = SV PRN code phase time offset (s)

The SV PRN code phase offset is given by

$$\Delta t_{\text{sv}} = a_{f0} + a_{fl}(t - t_{0c}) + a_{f2}(t - t_{0c})^2 + \Delta t_r, \tag{3.28}$$

where a_{f0}, a_{f1}, a_{f2} = polynomial coefficients given in the ephemeris data file
t_{0c} = clock data reference time (s)
Δt_r = relativistic correction term (s) given by

$$\Delta t_r = Fe\sqrt{a}\sin E_k. \tag{3.29}$$

In Eq. 3.29, F is a constant whose value is given as

$$F = \frac{-2\sqrt{\mu}}{c^2} = -4.442807633 \times 10^{-10}\ \text{s/m}^{1/2} \tag{3.30}$$

where the speed of light $c = 2.99792458 \times 10^8$ m/s. Note that Eqs. 3.27 and 3.28 are coupled. While the coefficients a_{f0}, a_{f1}, and a_{f2} are generated by using GPS time as indicated in Eq. 3.28, sensitivity of t_{sv} to t is negligible. This negligible sensitivity will allow the user to approximate t by t_{sv} in Eq. 3.28. The value of t must account for beginning- or end-of-week crossovers. That is, if the quantity $t - t_{0c}$ is greater than 302,400 s, subtract 604,800 s from t. If the quantity $t - t_{0c}$ is less than $-302{,}400$ s, add 604,800 s to t.

By using the value of the ephemeris parameters for satellite PRN 2 in the set of equations in Table 3.1 and Eqs. 3.27–3.30, we can calculate the space vehicle time offset and the ECEF coordinates of the satellite position [30]. Computer software (`ephemeris.m`) is on the accompanying diskette.

Problems

3.1 An important signal parameter is the maximum Doppler shift due to satellite motion, which must be accommodated by a receiver. Find its approximate

value by assuming that a GPS satellite has a circular orbit with a radius of 27,000 km, an inclination angle of 55°, and a 12-h period. Is the rotation rate of the earth significant? At what latitude(s) would one expect to see the largest possible Doppler shift?

3.2 Another important parameter is the maximum *rate* of Doppler shift in hertz per second that a phase-lock loop must be able to track. Using the orbital parameters of the previous problem, calculate the maximum rate of Doppler shift of a GPS signal one would expect, assuming that the receiver is stationary with respect to the earth.

3.3 Find the power spectrum of the 50-bps data stream containing the navigation message. Assume that the bit values are -1 and 1 with equal probability of occurrence, that the bits are uncorrelated random variables, and that the location of the bit boundary closest to $t = 0$ is a uniformly distributed random variable on the interval $[-0.01\,\text{s},\ 0.01\,\text{s}]$. *Hint:* First find the autocorrelation function $R(\tau)$ of the bit stream and then take its Fourier transform.

3.4 In two-dimensional positioning, the user's altitude is known, so only three satellites are needed. Thus, there are three pseudorange equations containing two position coordinates (e.g., latitude and longitude) and the receiver clock bias term B. Since the equations are nonlinear, there will generally be more than one position solution, and all solutions will be at the same altitude. Determine a procedure that isolates the correct solution.

3.5 Some civil receivers attempt to extract the L_2 carrier by squaring the received waveform after it has been frequency shifted to a lower IF. Show that the squaring process removes the P(Y)-code and the data modulation, leaving a sinusoidal signal component at twice the frequency of the original IF carrier. If the SNR in a 20-MHz IF bandwidth is -30 dB before squaring, find the SNR of the double-frequency component after squaring if it is passed through a 20-MHz bandpass filter. How narrow would the bandpass filter have to be to increase the SNR to 0 dB?

Global Positioning Systems, Inertial Navigation, and Integration,
Mohinder S. Grewal, Lawrence R. Weill, Angus P. Andrews

Print ISBN 0-471-35032-X Electronic ISBN 0-471-20071-9

4

Receiver and Antenna Design

4.1 RECEIVER ARCHITECTURE

Although there are many variations in GPS receiver design, all receivers must perform certain basic functions. We will now discuss these functions in detail, each of which appears as a block in the diagram of the generic receiver shown in Fig. 4.1.

4.1.1 Radio-Frequency Stages (Front End)

The purpose of the receiver front end is to filter and amplify the incoming GPS signal. As was pointed out earlier, the GPS signal power available at the receiver antenna output terminals is extremely small and can easily be masked by interference from more powerful signals adjacent to the GPS passband. To make the signal usable for digital processing at a later stage, RF amplification in the receiver front end provides as much as 35–55 dB of gain. Usually the front end will also contain passband filters to reduce out-of-band interference without degradation of the GPS signal waveform. The nominal bandwith of both the L_1 and L_2 GPS signals is 20 MHz ($\pm$10 MHz on each side of the carrier), and sharp cutoff bandpass filters are required for out-of-band signal rejection. However, the small ratio of passband width to carrier frequency makes the design of such filters infeasible. Consequently, filters with wider skirts are commonly used as a first stage of filtering, which also helps to prevent front-end overloading by strong interference, and the sharp cutoff filters are used later after downconversion to intermediate frequencies (IFs).

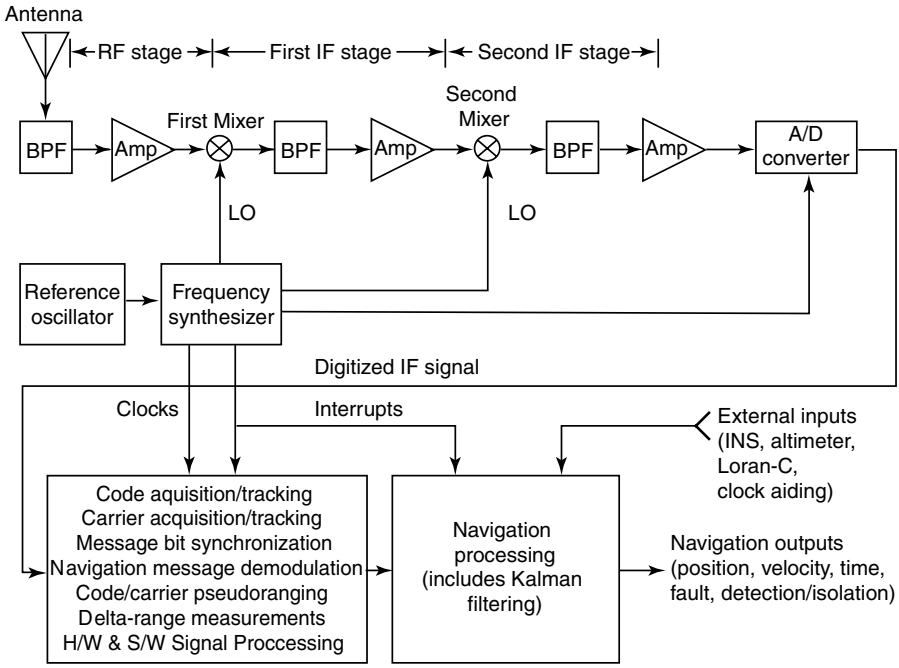


Fig. 4.1 *Generic GPS receiver.*

4.1.2 Frequency Downconversion and IF Amplification

After amplification in the receiver front end, the GPS signal is converted to a lower frequency called an intermediate frequency for further amplification and filtering. Downconversion accomplishes several objectives:

1. The total amount of signal amplification needed by the receiver exceeds the amount that can be performed in the receiver front end at the GPS carrier frequency. Excessive amplification can result in parasitic feedback oscillation, which is difficult to control. In addition, since sharp cutoff filters with a GPS signal bandwidth are not feasible at the L-band, excessive front-end gain makes the end-stage amplifiers vulnerable to overloading by strong nearby out-of-band signals. By providing additional amplification at an IF different from the received signal frequency, a large amount of gain can be realized without the tendency toward oscillation.
2. By converting the signal to a lower frequency, the signal bandwidth is unaffected, and the increased ratio of bandwidth to center frequency permits the design of sharp-cutoff bandpass filters. These filters can be placed ahead of the IF amplifiers to prevent saturation by strong out-of-band signals. The filtering is often by means of surface acoustic wave (SAW) devices.

3. Conversion of the signal to a lower frequency makes the sampling of the signal required for digital processing much more feasible.

Downconversion is accomplished by multiplying the GPS signal by a sinusoid called the local oscillator signal in a device called a mixer. The local oscillator frequency is either larger or smaller than the GPS carrier frequency by an amount equal to the IF. In either case the IF signal is the difference between the signal and local oscillator frequencies. Sum frequency components are also produced, but these are eliminated by a simple band-pass filter following the mixer. An incoming signal either above or below the local oscillator frequency by an amount equal to the IF will produce an IF signal, but only one of the two signals is desired. The other signal, called the image, can be eliminated by bandpass filtering of the desired signal prior to downconversion. However, since the frequency separation of the desired and image signals is twice the IF, the filtering becomes difficult if a single downconversion to a low IF is attempted. For this reason downconversion is often accomplished in more than one stage, with a relatively high first IF (30–100 MHz) to permit image rejection.

Whether it is single stage or multistage, downconversion typically provides a final IF that is low enough to be digitally sampled at feasible sampling rates without frequency aliasing. In low-cost receivers typical final IFs range from 4 to 20 MHz with bandwidths that have been filtered down to several MHz. This permits a relatively low digital sampling rate and at the same time keeps the lower edge of the signal spectrum well above 0 Hz to prevent spectral foldover. However, for adequate image rejection either multistage downconversion or a special single-stage image rejection mixer is required. In more advanced receivers there is a trend toward single conversion to a signal at a relatively high IF (30–100 MHz), because advances in technology permit sampling and digitizing even at these high frequencies.

Signal-to-Noise Ratio An important aspect of receiver design is the calculation of signal quality as measured by the signal-to-noise ratio (SNR) in the receiver IF bandwith. Typical IF bandwidths range from about 2 MHz in low-cost receivers to the full GPS signal bandwidth of 20 MHz in high-end units, and the dominant type of noise is the thermal noise in the first RF amplifier stage of the receiver front end (or the antenna preamplifier if it is used). The noise power in this bandwidth is given by

$$N = kT_eB \tag{4.1}$$

where $k = 1.3806 \times 10^{-23}$ J/K, B is the bandwidth in Hz, and T_e is the effective noise temperature in degrees Kelvin. The effective noise temperature is a function of sky noise, antenna noise temperature, line losses, receiver noise temperature, and ambient temperature. A typical effective noise temperature for a GPS receiver is 513 K, resulting in a noise power of about -138.5 dBW in a 2-MHz bandwidth and -128.5 dBW in a 20-MHz bandwidth. The SNR is defined as the ratio of signal power to noise power in the IF bandwidth, or the difference of these powers when

expressed in decibels. Using −154.6 dBW for the received signal power obtained in Section 3.3, the SNR in a 20-MHz bandwidth is seen to be $-154.6 - (-128.5) = -26.1$ dB. Although the GPS signal has a 20-MHz bandwidth, about 90% of the C/A-code power lies in a 2-MHz bandwith, so there is only about 0.5 dB loss in signal power. Consequently the SNR in a 2-MHz bandwidth is $(-154.6 - 0.5) - (-138.5) = -16.6$ dB. In either case it is evident that the signal is completely masked by noise. Further processing to elevate the signal above the noise will be discussed subsequently.

4.1.3 Digitization

In modern GPS receivers digital signal processing is used to track the GPS signal, make pseudorange and Doppler measurements, and demodulate the 50-bps data stream. For this purpose the signal is sampled and digitized by an analog-to-digital converter (ADC). In most receivers the final IF signal is sampled, but in some the final IF signal is converted down to an analog baseband signal prior to sampling. The sampling rate must be chosen so that there is no spectral aliasing of the sampled signal; this generally will be several times the final IF bandwidth (2–20 MHz).

Most low-cost receivers use 1-bit quantization of the digitized samples, which not only is a very low cost method of analog-to-digital conversion, but has the additional advantage that its performance is insensitive to changes in voltage levels. Thus, the receiver needs no automatic gain control (AGC). At first glance it would appear that 1-bit quantization would introduce severe signal distortion. However, the noise, which is Gaussian and typically much larger than the signal at this stage, introduces a dithering effect that, when statistically averaged, results in an essentially linear signal component. One-bit quantization does introduce some loss in SNR, typically about 2 dB, but in low-cost receivers this is an acceptable trade-off. A major disadvantage of 1-bit quantization is that it exhibits a capture effect in the presence of strong interfering signals and is therefore quite susceptible to jamming.

Typical high-end receivers use anywhere from 1.5-bit (three-level) to 3-bit (eight-level) sample quantization. Three-bit quantization essentially eliminates the SNR degradation found in 1-bit quantization and materially improves performance in the presence of jamming signals. However, to gain the advantages of multibit quantization, the ADC input signal level must exactly match the ADC dynamic range. Thus the receiver must have AGC to keep the ADC input level constant. Some military receivers use even more than 3-bit quantization to extend the dynamic range so that jamming signals are less likely to saturate the ADC.

4.1.4 Baseband Signal Processing

Baseband signal processing refers to a collection of high-speed real-time algorithms implemented in dedicated hardware and controlled by software that acquire and track the GPS signal, extract the 50-bps navigation data, and provide measurements of code and carrier pseudoranges and Doppler.

Carrier Tracking Tracking of the carrier phase and frequency is accomplished by using feedback control of a numerically controlled oscillator (NCO) to frequency shift the signal to precisely zero frequency and phase. Because the shift to zero frequency results in spectral foldover of the signal sidebands, both in-phase (I) and a quadrature (Q) baseband signal components are formed in order to prevent signal information loss. The I component is generated by multiplying the digitized IF by the NCO output and the Q component is formed by first introducing a 90° phase lag in the NCO output before multiplication. Feedback is accomplished by using the measured baseband phase to control the NCO so that this phase is driven toward zero. When this occurs, signal power is entirely in the I component, and the Q component contains only noise. However, both components are necessary both in order to measure the phase error for feedback and to provide full signal information during acquisition when phase lock has not yet been achieved. The baseband phase θ_{baseband} is defined by

$$\theta_{\text{baseband}} = \text{atan2}(I, Q) \tag{4.2}$$

where atan2 is the four-quadrant arctangent function. The phase needed for feedback is recovered from I and Q after despreading of the signal. When phase lock has been achieved, the output of the NCO will match the incoming IF signal in both frequency and phase but will generally have much less noise due to low-pass filtering used in the feedback loop. Comparing the NCO phase to a reference derived from the receiver reference oscillator provides the phase measurements needed for carrier phase pseudoranging. Additionally, the cycles of the NCO output can be accumulated to provide the raw data for Doppler, delta-range, and integrated Doppler measurements.

Code Tracking and Signal Spectral Despreading The digitized IF signal, which has a wide bandwidth due to the C/A- (or P-) code modulation, is completely obscured by noise. The signal power is raised above the noise power by *despreading*, in which the digitized IF signal is multiplied by a receiver-generated replica of the code precisely time aligned with the code on the received signal. Typically the individual baseband I and Q signals from the controlled NCO mixer are despread in parallel, as previously shown in Fig. 3.13. The despreading process removes the code from the signal, thus concentrating the full signal power into the approximately 50-Hz baseband bandwidth of the data modulation. Subsequent filtering (usually in the form of integration) can now be employed to dramatically raise the SNR to values permitting observation and measurement of the signal. As an example, recall that in a GPS receiver a typical SNR in a 2-MHz IF bandwidth is −16.6 dB. After despreading and 50-Hz low-pass filtering the total signal power is still about the same, but the bandwidth of the noise has been reduced from 2 MHz to about 50 Hz, which increases the SNR by the ratio $2 \times 10^6/50$, or 46 dB. The resulting SNR is therefore $-16.6 + 46.0 = 29.4$ dB.

4.2 RECEIVER DESIGN CHOICES

4.2.1 Number of Channels and Sequencing Rate

GPS receivers must observe the signal from at least four satellites to obtain three-dimensional position and velocity estimates. If the user altitude is known, three satellites will suffice. There are several choices as to how the signal observations from a multiplicity of satellites can be implemented. In early designs, reduction of hardware cost and complexity required that the number of processing channels be kept at a minimum, often being smaller than the number of satellites observed. In this case, each channel must sequentially observe more than one satellite. As a result of improved lower cost technology, most modern GPS receivers have a sufficient number of channels to permit one satellite to be continuously observed on each channel.

4.2.1.1 Receivers with Channel Time Sharing

Single-Channel Receivers In a single-channel receiver, all processing, such as acquisition, data demodulation, and code and carrier tracking, is performed by a single channel in which the signals from all observed satellites are time shared. Although this reduces hardware complexity, the software required to manage the time-sharing process can be quite complex, and there are also severe performance penalties. The process of acquiring satellites can be very slow and requires a juggling act to track already-acquired satellites while trying to acquire others. The process is quite tricky when receiving ephemeris data from a satellite, since about 30 s of continuous reception is required. During this time the signals from other satellites are eclipsed, and resumption of reliable tracking can be difficult.

After all satellites have been acquired and their ephemeris data stored, two basic techniques can be used to track the satellite signals in a single-channel receiver. In *slow-sequencing* designs the signal from each satellite is observed for a duration (dwell time) on the order of 1 s. Since a minimum of four satellites must typically be observed, the signal from each satellite is eclipsed for an appreciable length of time. For this reason, extra time must be allowed for signal reacquisition at the beginning of each dwell interval. Continually having to reacquire the signal generally results in less reliable operation, since the probability of losing a signal is considerably greater as compared to the case of continuous tracking. This is especially critical in the presence of dynamics, in which unpredictable user platform motion can take place during signal eclipse. Generally the positioning and velocity accuracy is also degraded in the presence of dynamics.

If a single-channel receiver does not have to accurately measure velocity, tracking can be accomplished with only a frequency-lock loop (FLL) for carrier tracking. Since a FLL typically has a wider pull-in range and a shorter pull-in time than a phase-lock loop (PLL), reacquisition of the signal is relatively fast and the sequencing dwell time can be as small as 0.25 s per satellite. Because loss of phase lock is not an issue, this type of receiver is also more robust in the presence of dynamics. On the other hand, if accurate velocity determination is required, a PLL

must be used and the extra time required for phase lock during signal reacquisition pushes the dwell time up to about 1–1.5 s per satellite, with an increased probability of reacquisition failure due to dynamics.

A single-channel receiver requires relatively complex software for managing the satellite time-sharing process. A typical design employs only one pseudonoise (PN) code generator and one PPL in hardware. Typical tasks that the software must perform during the dwell period for a specific satellite are as follows:

1. Select the PN code corresponding to the satellite observed.
2. Compute the current state of the code at the start of the dwell based on the state at the end of the last dwell, the signal Doppler, and the eclipse time since the last dwell.
3. Load the code state into the code generator hardware.
4. Compute the initial Doppler frequency of the FLL/PLL reference.
5. Load the Doppler frequency into the FLL/PLL hardware.
6. Initiate the reacquisition process by turning on the code and carrier tracking loops.
7. Determine when reacquisition (code/frequency/phase lock) has occurred.
8. Measure pseudorange/carrier phase/carrier phase rate during the remainder of the dwell.

In addition to these tasks, the software must be capable of ignoring measurements from a satellite if the signal is momentarily lost and must permanently remove the satellite from the sequencing cycle when its signal becomes unusable, such as when the satellite elevation angle is below the mask angle. The software must also have the capability of acquiring new satellites and obtaining their ephemeris data as their signals become available while at the same time not losing the satellites already being tracked. A satellite whose ephemeris data is being recorded must have a much longer dwell time (about 30 s) than the dwell times of other satellites that are only being tracked, which causes a much longer eclipse time for the latter. The software must therefore modify the calculations listed above to take this into account.

Because current technology makes the hardware costs of a multichannel receiver almost as small as that for a single channel, the single-channel approach has been almost entirely abandoned in modern designs.

Another method of time sharing that can be used in single-channel receivers is *multiplexing*, in which the dwell time is much shorter, typically 5–10 ms per satellite. Because the eclipse time is so short, the satellites do not need to be reacquired at each dwell. However, a price is paid in that the effective SNR is significantly reduced in proportion to the number of satellites being tracked. Resistance to jamming is also degraded by values of 7 dB or more. Additionally, the process of acquiring new satellites without disruption is made more demanding because the acquisition search must be broken into numerous short time intervals. Due to the rapidity with which satellites are sequenced, a common practice with a two-channel receiver is to use a

full complement of PN code generators that run all the time, so that high-speed multiplexing of a single code generator can be avoided.

Two-Channel Receivers The use of two channels permits the second channel to be a "roving" channel, in which new satellites can be acquired and ephemeris data collected while on the first channel satellites can be tracked without slowdown in position/velocity updates. However, the satellites must still be time shared on the first channel. Thus the software must still perform the functions listed above and in addition must be capable of inserting/deleting satellites from the sequencing cycle. As with single-channel designs, either slow sequencing or multiplexing may be used.

Receivers with Three to Five Channels In either slow-sequencing or multiplexed receivers, additional channels will generally permit better accuracy and jamming immunity as well as more robust performance in the presence of dynamics. A major breakthrough in receiver performance occurs with five or more channels, because four satellites can be simultaneously tracked without the need for time sharing. The fifth channel can be used to acquire a new satellite and collect its ephemeris data before using it to replace one of the satellites being tracked on the other four channels.

Multichannel All-in-View Receivers The universal trend in receiver design is to use enough channels to receive all satellites that are visible. In most cases eight or fewer useful satellites are visible at any given time; for this reason modern receivers typically have no more than 10–12 channels, with perhaps several channels being used for acquisition of new satellites and the remainder for tracking. Position/velocity accuracy is materially improved because satellites do not have to be continually reacquired as is the case with slow sequencing, there is no reduction in effective SNR found in multiplexing designs, and the use of more than the minimum number of satellites results in an overdetermined solution. In addition, software design is much simpler because each channel has its own tracking hardware that tracks only one satellite and does not have to be time shared.

4.2.2 L_2 Capability

GPS receivers that can utilize the L_2 frequency (1227.60 MHz) gain several advantages over L_1-only receivers. Currently the L_2 carrier is modulated only with a military-encrypted P-code, called the Y-code, and the 50-bps data stream. Because of the encryption, civilians are denied the use of the P-code. However, it is still possible to recover the L_2 carrier, which can provide significant performance gains in certain applications.

Dual-Frequency Ionospheric Correction Because the pseudorange error caused by the ionosphere is inversely proportional to the square of frequency, it

can be calculated in military receivers by comparing the P-code pseudorange measurements obtained on the L_1 and L_2 frequencies. After subtraction of the calculated error from the pseudorange measurements, the residual error due to the ionosphere is typically no more than a few meters as compared to an uncorrected error of 5–30 m. Although civilians do not have access to the P-code, in differential positioning applications the L_2 carrier phase can be extracted without decryption, and the ionospheric error can then be estimated by comparing the L_1 and L_2 phase measurements.

Improved Carrier Phase Ambiguity Resolution in High-Accuracy Differential Positioning High-precision receivers, such as those used in surveying, use carrier phase measurements to obtain very precise pseudoranges. However, the periodic nature of the carrier makes the measurements highly ambiguous. Therefore, solution of the positioning equations yields a grid of possible positions separated by distances on the order of one to four carrier wavelengths, depending on geometry. Removal of the ambiguity is accomplished by using additional information in the form of code pseudorange measurements, changes in satellite geometry, or the use of more satellites than is necessary. In general, ambiguity resolution becomes less difficult as the frequency of the carrier decreases. By using both the L_1 and L_2 carriers, a virtual carrier frequency of $L_1 - L_2 = 1575.42 - 1227.60 = 347.82$ MHz can be obtained, which has a wavelength of about 86 cm as compared to the 19 cm wavelength of the L_1 carrier. Ambiguity resolution can therefore be made faster and more reliable by using the difference frequency.

4.2.3 Code Selections: C/A, P, or Codeless

All GPS receivers are designed to use the C/A-code, since it is the only code accessible to civilians and is used by the military for initial signal acquisition. Most military receivers also have P-code capability to take advantage of the improved performance it offers. On the other hand, commercial receivers seldom have P-code capability because the government does not make the needed decryption equipment available to the civil sector. Some receivers, notably those used for precision differential positioning application, also incorporate a codeless mode that permits recovery of the L_2 carrier without knowledge of the code waveform.

The C/A-Code The C/A-code, with its 1.023-MHz chipping rate and 1-ms period, has a bandwidth that permits a reasonably small pseudorange error due to thermal noise. The code is easily generated by a few relatively small shift registers. Because the C/A-code has only 1023 chips per period, it is relatively easy to acquire. In military receivers direct acquisition of the P-code would be extremely difficult and time consuming. For this reason these receivers first acquire the C/A-code on the L_1 frequency, allowing the 50-bps data stream to be recovered. The data contains a hand-over word that tells the military receiver a range in which to search for the P-code.

The P-Code The unencrypted P-code has a 10.23-MHz chipping rate and is known to both civilian and military users. It has a very long period of one week. The Y-code is produced by biphase modulation of the P-code by an encrypting code known as the W-code. The W-code has a slower chipping rate than the P-code; there are precisely 20 P-code chips per W-code chip. Normally the W-code is known only to military users who can use decryption to recover the P-code, so that the civilian community is denied the full use of the L_2 signal. However, as will be indicated shortly, useful information can still be extracted from the L_2 signal in civilian receivers without the need for decryption. Advantages of the P-code include the following:

Improved Navigation Accuracy. Because the P-code has 10 times the chipping rate of the C/A-code, its spectrum occupies a larger portion of the full 20-MHz GPS signal bandwidth. Consequently, military receivers can typically obtain three times better pseudoranging accuracy than that obtained with the C/A-code.

Improved Jamming Immunity. The wider bandwidth of the P-code gives about 40 dB suppression of narrow-band jamming signals as compared to about 30 dB for the C/A-code, which is of obvious importance in military applications.

Better Multipath Rejection. In the absence of special multipath mitigation techniques, the P-code provides significantly smaller pseudorange errors in the presence of multipath as compared to the C/A-code. Because the P-code correlation function is approximately one-tenth as wide as that of the C/A-code, there is less opportunity for a delayed-path component of the receiver-generated signal correlation function to cause range error by overlap with the direct-path component.

Codeless Techniques Commercial receivers can recover the L_2 carrier without knowledge of the code modulation simply by squaring the received signal waveform or by taking its absolute value. Because the a priori SNR is so small, the SNR of the recovered carrier will be reduced by as much as 33 dB because the squaring of the signal greatly increases the noise power relative to that of the signal. However, the squared signal has extremely small bandwidth (limited only by Doppler variations), so that narrow-band filtering can make up the difference.

4.2.4 Access to SA Signals

Selective Availability (SA) refers to errors that may be intentionally introduced into the satellite signals by the military to prevent full-accuracy capability by the civilian community. SA was suspended on May 1, 2000 but can be turned on again at the discretion of the DoD. The errors appear to be random, have a zero long-term average value, and typically have a standard deviation of 30 m. Instantaneous position errors of 50–100 m occur fairly often and are magnified by large position

dilution of precision (PDOP) values. Part of the SA error is in the ephemeris data transmitted by the satellite, and the rest is accomplished by dithering of the satellite clock that controls the timing of the carrier and code waveforms. Civil users with a single receiver generally have no way to eliminate errors due to SA, but authorized users (mostly military) have the key to remove them completely. On the other hand, civilians can remove SA errors by employing differential operation, and a large network of differential reference stations has been spawned by this need.

4.2.5 Differential Capability

Differential GPS (DGPS) is a powerful technique for improving the performance of GPS positioning. This concept involves the use of not only the user's receiver (sometimes called *the remote* or *roving* unit) but also a *reference receiver* at an accurately known location within perhaps 200 km of the user. Because the location of the reference receiver is known, pseudorange errors common to the user and reference receivers can be measured and removed in the user's positioning calculations.

Errors Common to Both Receivers The major sources of errors common to the reference and remote receivers, which can be removed (or mostly removed) by differential operation, are the following:

1. *Selective Availability Error.* As previously mentioned, these are typically about 30 m, 1σ.
2. *Ionospheric Delays*. Ionospheric signal propagation group delay, which is discussed further in Chapter 5, can be as much as 20–30 m during the day to 3–6 m at night. Receivers that can utilize both the L_1 and L_2 frequencies can largely remove these errors by applying the inverse square-law dependence of delay on frequency.
3. *Tropospheric Delays*. These delays, which occur in the lower atmosphere, are usually smaller and more predictable than ionospheric errors, and typically are in the 1–3 m range but can be significantly larger at low satellite elevation angles.
4. *Ephemeris Errors*. Ephemeris errors, which are the difference between the actual satellite location and the location predicted by satellite orbital data, are typically less than 3 m and will undoubtedly become smaller as satellite tracking technology improves.
5. *Satellite Clock Errors*. These are the difference between the actual satellite clock time and that predicted by the satellite data.

Differential operation can almost completely remove satellite clock errors, errors due to SA, and ephemeris errors. For these quantities the quality of correction has little dependence on the separation of the reference and roving receivers. However, because SA errors vary quite rapidly, care must be taken in time synchronizing the corrections to the pseudorange measurements of the roving receiver. The degree of

correction that can be achieved for ionospheric and tropospheric delays is excellent when the two receivers are in close proximity, say, up to 20 km. At larger separations the ionospheric/tropospheric propagation delays to the receivers become less correlated, and residual errors after correction are correspondingly larger. Nonetheless, substantial corrections can often be made with receiver separations as large as 100–200 km.

Differential operation is ineffective against errors due to multipath, because these errors are strictly local to each of the two receivers.

Corrections in the Measurement Domain Versus the Solution Domain In the broadest sense there are two ways that differential corrections can be made. In the *measurement domain*, corrections are determined for pseudorange measurements to each satellite in view of the reference receiver, and the user simply applies the corrections corresponding to the satellites the roving receiver is observing. On the other hand, in the *solution domain* approach, the reference station computes the position error that results from pseudorange measurements to a set of satellites, and this is applied as a correction to the user's computed position. A significant drawback to the solution domain approach is that the user and reference station must use exactly the same set of satellites if the position correction is to be valid. In most cases the reference station does not know which satellites can be received by the roving receiver (e.g., some might be blocked by obstacles) and therefore would have to transmit the position corrections for many possible sets of satellites. The impracticality of doing this strongly favors the use of the measurement domain method.

Real-Time Versus Postprocessed Corrections In some applications, such as surveying, it is not necessary to obtain differentially corrected position solutions in real time. In these applications it is common practice to obtain corrected positions at a later time by bringing together recorded data from both receivers. No reference-to-user data link is necessary if the recorded data from both receivers can be physically transported to a common computer for processing.

However, in the vast majority of cases it is imperative that corrections be applied as soon as the user has enough pseudorange measurements to obtain a position solution. When the user needs to know his or her corrected position in real time, current pseudorange corrections can be transmitted from the reference receiver to the user via a radio or telephone link, and the user can use them in the positioning calculations. This capability requires a user receiver input port for receiving and using differential correction messages. A standardized format of these messages has been recommended by Special Committee 104 (SC-104), established by the Radio Technical Commission for Maritime Service (RTCM) in 1983. Details on this format appear in [70].

4.2.6 Pseudosatellite Compatibility

Although differential GPS can improve the reliability, integrity, and accuracy of GPS navigation, it cannot overcome inherent limitations that are critical to successful

operation in specific applications. A major limitation is poor satellite geometry, which can be caused by signal failure of one or more satellites, signal blockage by local objects and/or terrain, and occasional periods of high PDOP, which can occur even with a full constellation of satellites. Vertical positioning error is usually more sensitive to this effect, which is bad news for aviation applications. In some cases a navigation solution may not exist because not enough satellite signals can be received.

The use of *pseudolites* can solve these problems within a local area. A pseudolite is simply a ground-based transmitter that acts as an additional GPS satellite by transmitting a GPS-like signal. This signal can be utilized by a receiver for pseudoranging and can also convey messages to the receiver to improve reliability and signal integrity. The RTCM SC-104 was formed in 1983 to study pseudolite system and receiver design issues. The recommendations of SC-104 can be found in [112]. The major improvements offered by pseudolites are the following:

1. *Improvement in Geometry.* Pseudolites, acting as additional satellites, can provide major improvements in geometry, hence in positioning accuracy, within their region of coverage. Vertical (VDOP) as well as horizontal (HDOP) dilution of precision can be dramatically reduced, which is of major importance to aviation. Experiments have shown that PDOP of about 3 over a region having a radius of 20–40 km can be obtained by using several pseudolites even when there are fewer than the minimum of four satellites that would otherwise be needed for a navigation solution.
2. *Improvement in Signal Availability.* Navigation solutions with fewer than the minimum required number of GPS satellites are made possible by using the additional signals provided by pseudolites.
3. *Inherent Transmission of Differential Corrections*. The GPS-like signals transmitted by a pseudolite include messaging capability that can be received directly by the GPS receiver, thus allowing the user to receive differential corrections without the need for a separate communications link.
4. *Self-Contained Failure Notification*. The additional signals provided by pseudolites permit the user to perform his or her own failure assesment. For example, if pseudorange measurements from four satellites and one pseudolite are available, a problem can be detected by examining the consistency of the measurements. If two pseudolites are available, not only can the failure of a single signal be detected, but the offending signal can be identified as well. These advantages are especially important in aviation, where pilot notification of signal failures must occur very rapidly (within 1–10 s).
5. *Solution of Signal Blockage Problems*. The additional signals from pseudolites can virtually eliminate problems due to blockage of the satellite signals by objects, terrain, or the receiving platform itself.

Pseudolite Signal Structure Ideally the pseudolite signal structure would permit reception by a standard GPS receiver with little or no modification of the

receiver design. Thus it would seem that the pseudolite signal should have a unique C/A-code with the same characteristics as the C/A-codes used by the satellites. However, with this scheme it would be difficult to prevent a pseudolite signal from interfering with the reception of the satellite signals, even if its C/A-code were orthogonal to the satellite codes. The fundamental difficulty, which is called the *near–far problem*, occurs because of the inverse square-law dependence of received signal power with range. The near–far problem does not occur with the GPS satellite signals because variation in the user-to-satellite range is relatively small compared to its average value. However, with pseudolites this is not the case. The problem is illustrated by considering that the received signal strength of a pseudolite must be at least approximately that of a satellite. If the pseudolite signal equals that of a satellite when the user is, say, 50 km from the pseudolite, then that same signal will be 60 dB stronger when the user is 50 m from the pseudolite. At this close range the pseudolite signal would be so strong that it would jam the weaker GPS satellite signals.

Several solutions to the near–far problem involving both pseudolite signal design and receiver design have been proposed in [112] for the 60-dB received signal dynamic range discussed above.

Pseudolite Signal Design Approaches

1. *Use of High-Performance Pseudorandom Codes*. The 60 dB of jamming protection would require the pseudolite to transmit a code much longer than a C/A-code and clocked at a much higher rate. This has been judged to be an impractical solution because it would reduce compatibility with the GPS signal structure and significantly increase receiver costs.
2. *Pseudolite Frequency Offset*. By moving the frequency of the pseudolite signal sufficiently far away from the 1575.42-MHz L_1 frequency, filters in the receiver could prevent the pseudolite signals from interfering with the satellite signals. Again, however, this approach would significantly increase receiver costs and reduce compatibility with the GPS signal structure.
3. *Low-Duty-Cycle Time Division Multiplexing*. A preferred approach is for the pseudolite to transmit at the L_1 frequency using short, low-duty-cycle pulses that interfere with the satellite signals only a small fraction of the time. The impact on receiver design is minimal because modifications are primarily digital and low in cost. This approach retains compatibility with the GPS signal structure by using a new set of 51 pseudolite Gold codes with the same chipping rate, period, and number of chips per period as the satellite C/A-codes and a 50-bps data stream. Although the codes run continuously in both the pseudolite and the user receiver, the pseudolite signal is gated on only during eleven 90.91-μs intervals in each 10-ms (half-data-bit) interval. Each of the 11 gate intervals transmits 93 new chips of the code, so that all 1023 chips get transmitted in 10 ms. However, the timing of the gate intervals is randomized in order to randomize the signal spectrum. Further details of the signal structure can be found in [112].

Pseudolite Characteristics

1. *Pseudolite Identification*. Identification of a pseudolite is accomplished by both its unique Gold code and its physical location, which appears in its 50-bps message. Since pseudolite signals are low power and thus can be received only within a relatively small coverage area, it is possible for pseudolites spaced sufficiently far apart to use the same Gold code. In this case correct identification is effected by noting the location transmitted by the pseudolite.
2. *Pseudolite Clock Offset*. Since the pseudolite can monitor GPS signals over extended time periods, it can determine GPS time. This permits the transmitted epochs of the pseudolite signal to be correct in GPS time and avoids the necessity of transmitting pseudolite clock corrections. The time reference for the differential pseudorange corrections transmitted by the pseudolite is also GPS time.
3. *Transmitted Signal Power*. The primary use of pseudolite signals is for aircraft in terminal areas, so that a typical maximum reception range is 50 km. At this range a half-hemisphere omnidirectional transmitting antenna fed with approximately 30 mW of signal power will provide a signal level comparable to that typical of a GPS satellite (−116 dBm). At a range of 50 m the signal level will be 60 dB larger (−56 dBm).
4. *Pseudolite Message Structure*. Although the pseudolite data stream is 50 bps to assure compatibility with GPS receivers, its structure must be modified to transmit information that differs somewhat from that transmitted by the GPS satellites. A proposed structure can be found in [112].
5. *Minimum Physical Spacing of Pseudolites*. Placement of pseudolites involves considerations that depend on whether the pseudolites use the same or different Gold codes.

Separation of Pseudolites Using the Same Code One approach when two pseudolites use the same code is to synchronize the timing of the gated signals of the pseudolites and separate the pseudolites by a distance that guarantees that received transmissions from different pseudolites will not overlap. This requires that the pseudolites be separated by at least 130 km, which guarantees that a user 50 km from the desired pseudolite will be at least 80 km from the undesired pseudolite. The pulses from the latter will then travel at least 30 km further than those from the desired pseudolite, thus arriving at least 100 μs later. Since the width of pulses is 90.91 μs, pulses from two pseudolites will not overlap and interference is thereby avoided.

However, a more conservative approach is to separate two pseudolites by a distance that is sufficient to guarantee that when the user is at the maximum usable range from one pseudolite, the signal from the other is too weak to interfere. Suppose each pseudolite is set to achieve a received signal level of −126 dBm at a maximum service radius of 50 km and that an undesired pseudolite signal must be at least 14 dB below the desired signal to avoid interference. A simple calculation

involving the inverse square power law shows that this can be achieved with a minimum spacing of 300 km between the two pseudolites, so that the minimum distance to the undesired pseudolite will be 250 km when the user is 50 km from the desired pseudolite.

Separation of Pseudolites Using Different Codes When the user must receive several pseudolites simultaneously, separation of the signals from different pseudolites might be possible by using different timing offsets of the transmitted pulses. However, this would substantially complicate system design. A preferred approach is to use synchronous transmissions but space the pseudolites so that when the received pulses do overlap, they can still be recovered by using a suitable low-cost receiver design. The situation is clarified by considering the two pseudolites shown in Fig. 4.2, which are separated by at least 27.25 km, the distance traveled by a signal in the time required to transmit a single pulse. With synchronous pulse transmissions from the pseudolites there exists a central region bounded on the left and right by two hyperbolic curves 27.25 km apart along the baseline connecting the pseudolites. This distance is independent of the separation of the pseudolites, but the curvature of the hyperbolas decreases as the pseudolite separation increases. Outside the central region the received pulses will not overlap and can easily be recovered by the receiver. The difficulty of separating the overlapping pulses within the central region is a function of the pseudolite separation. Separation is most difficult when the receiver is located at the intersection of a hyperbola and the baseline where the stronger of the two signals has its largest value, thus having the potential to overpower the weaker signal. This problem can be avoided by adequate separation of the pseusolites, but the separation required is a function of receiver design.

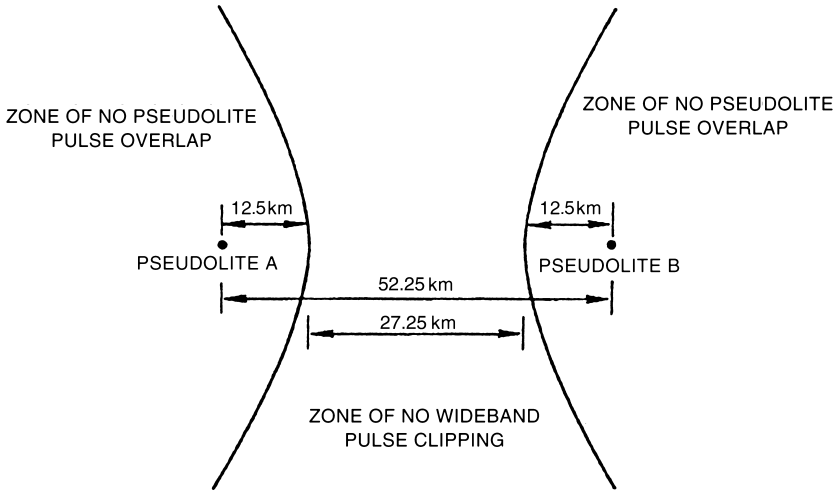


Fig. 4.2 *Minimum spacing of pseudolites.*

It will be seen later that a typical receiver designed for pseudolite operation might clip the incoming signal at $\pm 2\sigma$ of the precorrelation noise power in order to limit the received power of strong pseudolite signals. Under this assumption and an assumed ± 1-MHz precorrelation bandwidth, the clipping threshold in a receiver with a 4-dB noise figure would be -104 dBm. Assuming that the pseudolites are designed to produce a -116-dBm power level at 50 km, a receiver receiving overlapping pulses would need to be at least 12.5 km from both pseudolites to avoid the capture effect in the clipping process. Thus, the two pseudolites in Fig. 4.2 should each be moved 12.5 km from the boundaries of the central region, resulting in a minimum distance of 52.25 km between them.

Receiver Design for Pseudosatellite Compatibility Major design issues for a GPS receiver that receives pseudosatellite signals (often called a *participating receiver*) are as follows:

1. *Continuous Reception*. Because the receiver must continuously recover the pseudolite data message, a channel must be dedicated to this task. For this reason a single-channel slow-sequencing receiver could not be used. This is really not a problem, since almost all modern receivers use parallel channels.
2. *Ability to Track Pseudolite Gold Codes*. The receiver must be capable of generating and tracking each of the 51 special C/A-codes specified for the pseudolite signals. These codes and their method of generation can be found in [42]. Although the codes can be tracked with standard GPS tracking loops, optimum performance demands that the noise between pseudolite pulses be blanked to obtain a 10-dB improvement in SNR.
3. *Reduction of Pseudosatellite Interference to GPS Signal Channels*. In a GPS satellite channel a pseudolite signal appears as pulsed interference that can be 60 dB greater above the satellite signal level. The resulting degradation of the GPS satellite signal can be reduced to acceptable levels by properly designed wide-band precorrelation signal clipping in the receiver. This approach, which generally improves with increasing precorrelation bandwidth and decreasing clipping level, typically results in a reduction in the GPS SNR of 1–2 dB. A somewhat more effective approach is to blank the GPS signal ahead of the correlator during the reception of a pseudolite pulse, which results in a GPS SNR reduction of about 0.5 dB.
4. *Ability to Receive Overlapping Pseudolite Pulses*. A group of pseudolites designed to be utilized simultaneously must be located relatively close together, inevitably causing received pulse overlap in certain portions of the coverage area. Consequently, receiver design parameters must be chosen carefully to assure that overlapping pulses from different pseudolites can be separated. The signal level from a nearby pseudolite often can be strong enough to overcome the approximately 24 dB of interference suppression provided by the cross-correlation properties of distinct Gold codes and also can obliterate a second overlapping signal by saturating the receiver amplifiers.

Both of these problems can be solved by properly designed wide-band precorrelation signal clipping, in which there are two conflicting requirements. Deep (severe) clipping significantly reduces the amount of interfering power from a strong signal but gives the stronger signal more ability to blank out the weaker one (capture effect). On the other hand, more modest clipping levels reduce the capture effect at the expense of passing more power from the stronger signal into the correlators. As a result, more stress is put on the Gold codes to separate the weaker pulses from the stronger ones in the correlation process. An acceptable compromise for most purposes is to clip the received signal at about ± 2 standard deviations of the precorrelation noise power.

4.2.7 Immunity to Pseudolite Signals

A receiver that is not designed to receive pseudolite signals (a so-called *nonparticipating receiver*) must be designed so that a pseudolite signal, which might be 60 dB stronger than a satellite signal, will not interfere with the latter. The importance of this requirement cannot be overstated, since it is expected that use of pseudolites will grow dramatically, especially near airports. Therefore, purchasers of nonparticipating receivers would be well advised to obtain assurances of immunity to jamming by pseudolites.

Pseudolite immunity in a nonparticipating receiver can be effected by designing the front-end amplifier circuits for quick recovery from overload in combination with precorrelation hard limiting of the signal. This approach is suitable for low-cost receivers such as hand-held units. More sophisticated receivers using more than 1 bit of digital quantization to avoid quantization loss may still be designed to operate well if the clipping level is the same as that used in participating receivers. The design issues for obtaining immunity to pseudolite interference have been analyzed by RTCM SC 104 and can be found in [112].

4.2.8 Aiding Inputs

Although GPS can operate as a stand-alone system, navigation accuracy and coverage can be materially improved if additional information supplements the received GPS signals. Basic sources include the following:

1. *INS Aiding*. Although GPS navigation is potentially very accurate, periods of poor signal availability, jamming, and high-dynamic platform environments often limit its capability. INSs are relatively immune to these situations and thus offer powerful leverage in performance under these conditions. On the other hand, the fundamental limitation of INS long-term drift is overcome by the inherent calibration capability provided by GPS. Incorporation of INS measurements is readily achieved through Kalman filtering.
2. *Aiding with Additional Navigation Inputs*. Kalman filtering can also use additional measurement data from navigation systems such as LORAN C,

vehicular wheel sensors, and magnetic compasses, to improve navigation accuracy and reliability.

3. *Altimeter Aiding*. A fundamental property of GPS satellite geometry causes the greatest error in GPS positioning to be in the vertical direction. Vertical error can be significantly reduced by inputs from barometric, radar, or laser altimeter data. Coupling within the system of positioning equations tends to reduce the horizontal error as well.
4. *Clock Aiding*. An external clock with high stability and accuracy can materially improve navigation performance. It can be continuously calibrated when enough satellite signals are available to obtain precise GPS time. During periods of poor satellite visibility it can be used to reduce the number of satellites needed for positioning and velocity determination.

4.3 ANTENNA DESIGN

Although there is a wide variety of GPS antennas, most are normally right-hand circularly polarized to match the incoming signal and the spatial reception pattern is nominally a hemisphere. Such a pattern permits reception of satellites in any azimuthal direction from zenith down to the horizon. The short wavelengths at the L_1 and L_2 frequencies permit very compact designs. In low-cost hand-held receivers the antenna is often integrated with the receiver electronics in a rugged case. In more sophisticated applications it is often desirable that the antenna be separate from the receiver in order to site it more advantageously. In these situations the signal is fed from the antenna to the receiver via a low-loss coaxial cable. At L-band frequencies the cable losses are still quite large and can reach 1 dB for every 10 f of cable. Thus, it is often necessary to use a low-noise preamplifier at the antenna (active antenna). Preamplifier gain is usually in the range of 20–40 dB, and DC power is commonly fed to the preamplifier via the coaxial cable itself, with appropriate decoupling filters to isolate the signal from the DC power voltage. The preamplifier sets the noise figure for the entire receiver system and typically has a noise figure of 3–4 dB.

4.3.1 Various Types of Antennas

Figure 4.3 shows various types of GPS antennas:

Patch Antennas. The patch antenna, the most common antenna type, is often used in low-cost hand-held receivers. In typical designs the antenna elements are formed by etching the copper foil on a printed circuit board, which forms a very rugged low-profile unit. This is advantageous in some aviation applications, because it is relatively easy to integrate the antenna into the skin of the aircraft.

Dome Antennas. These antennas are housed in a bubblelike housing.

Blade Antennas. The blade antenna, also commonly used in aviation applications, resembles a small airfoil protruding from its base.

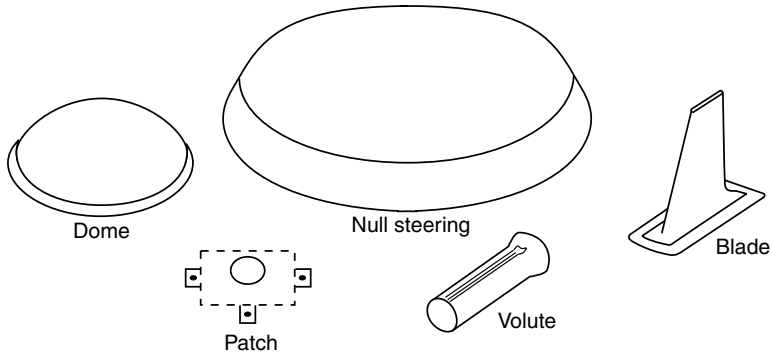


Fig. 4.3 *Types of GPS antennas.*

Helical (Volute) Antennas. Helical antennas contain elements that spiral along an axis that typically points toward the zenith. In some designs the helical elements are etched from a cylindrical copper-clad laminate to reduce cost. Helical antennas are generally more complex and costly to manufacture than patch antennas but tend to be somewhat more efficient. Some hand-held receivers use this type of antenna as an articulated unit that can be adjusted to point skyward while the case of the receiver can be oriented for comfortable viewing by the user. A popular design is the quadrifilar helix, which consists of four helixes symmetrically wound around a circular insulating core.

Choke-Ring Designs. In precision applications, such as surveying, choke-ring antennas are used to reduce the effects of multipath signal components reflected from the ground. These antennas are usually of the patch or helical type with a groundplane containing a series of concentric circular troughs one-quarter wavelength deep that act as transmission lines shorted at the bottom ends so that their top ends exhibit a very high impedance at the GPS carrier frequency. Low-elevation angle signals, including ground-reflected components, are nulled by the troughs, reducing the antenna gain in these directions. The size, weight, and cost of a choke-ring antenna are significantly greater than that of simpler designs.

Phased-Array (Null-Steering) Antennas. Although most applications of GPS require a nominally hemispherical antenna pattern, certain military applications require that the antenna be capable of inserting nulls in specified directions to reduce the effect of intentional jamming of the received GPS signal. Antenna designs that accomplish this consist of numerous elements arranged in an array, as depicted in Fig. 4.4. By introducing dynamically controlled phase shifts into the signal output of each element and then summing the phase-shifted outputs over all elements, specified nulls in the antenna pattern can be created that are capable of adapting, in real time, to changing jamming threats. Needless to say, phased-array antennas are much more costly than simpler designs and historically have only been used by the

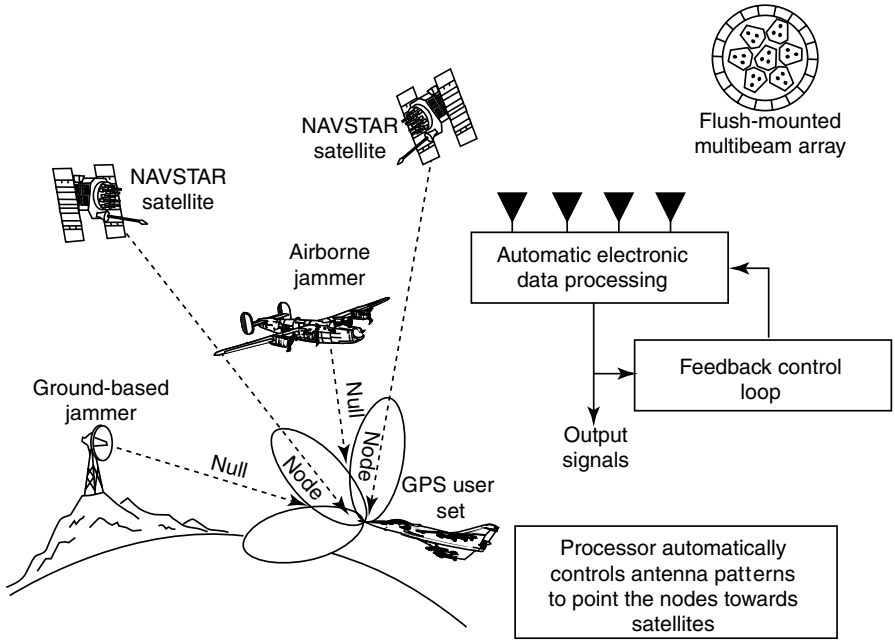


Fig. 4.4 *Electronic antenna nulling.*

military. However, civilian applications have recently begun to emerge, primarily for the purpose of improving positioning performance in the presence of multipath. An introduction to multipath-mitigation antennas and a design example can be found in [27].

Problems

4.1 An ultimate limit on the usability of weak GPS signals occurs when the bit error rate (BER) in demodulating the 50-bps navigation message becomes unacceptably large. Find the signal level in dBm at the output of the receiver antenna that will give a BER of 10^{-5}. Assume an effective receiver noise temperature of 513°K, and that all signal power has been translated to the baseband I-channel with optimal demodulation (integration over the 20-ms bit duration followed by polarity detection).

4.2 Support the claim that a 1-bit analog-to-digital converter (ADC) provides an essentially linear response to a signal deeply buried in Gaussian noise by solving the following problem: Suppose that the input signal s_{in} to the ADC is a DC voltage embedded in zero-mean additive Gaussian noise $n(t)$ with standard deviation σ_{in} and that the power spectral density of $n(t)$ is flat in

the frequency interval $[-W, W]$ and zero outside the interval. Assume that the 1-bit ADC is modeled as a hard limiter that outputs a value $v_{\text{out}} = 1$ if the polarity of the signal plus noise is positive and $v_{\text{out}} = -1$ if the polarity is negative. Define the output signal s_{out} by

$$s_{\text{out}} = E[v_{\text{out}}], \tag{4.3}$$

where E denotes expectation, and let σ_{out} be the standard deviation of the ADC output.

The ADC input signal-to-noise ratio SNR_{in} can then be defined by

$$\text{SNR}_{\text{in}} = \frac{s_{\text{in}}}{\sigma_{\text{in}}} \tag{4.4}$$

and the ADC output signal-to-noise ratio SNR_{out} by

$$\text{SNR}_{\text{out}} = \frac{s_{\text{out}}}{\sigma_{\text{out}}}, \tag{4.5}$$

where s_{out} and σ_{out} are respectively the expected value and the standard deviation of the ADC output. Show that if $s_{\text{in}} \ll \sigma_{\text{in}}$, then $s_{\text{out}} = Ks_{\text{in}}$, where K is a constant, and

$$\frac{\text{SNR}_{\text{out}}}{\text{SNR}_{\text{in}}} = \frac{2}{\pi}. \tag{4.6}$$

Thus, the signal component of the ADC output is linearly related to the input signal component, and the output SNR is about 2 dB less than that of the input.

4.3 Some GPS receivers directly sample the signal at an IF instead of using mixers for the final frequency shift to baseband. Suppose you wish to sample a GPS signal with a bandwidth of 1 MHz centered at an IF of 3.5805 MHz. What sampling rates will not result in frequency aliasing? If a sampling rate of 2.046 MHz were used, show how a digitally sampled baseband signal could be obtained from the samples.

4.4 Instead of forming a baseband signal with I and Q components, a single-component baseband signal can be created simply by multiplying the incoming L_1 (or L_2) carrier by a sinusoid of the same nominal frequency, followed by low-pass filtering. Discuss the problems inherent in this approach. *Hint:* Form the product of a sinusoidal carrier with a sinusoidal local oscillator signal, use trigonometric identities to reveal the sum and difference frequency components, and consider what happens to the difference frequency as the phase of the incoming signal assumes various values.

4.5 Write a computer program using C or another high-level language that produces the 1023-chip C/A-code used by satellite SV1. The code for this satellite is generated by two 10-stage shift registers called the G1 and G2 registers, each of which is initialized with all 1's. The input to the first stage of the G1 register is the exclusive OR of its 3rd and 10th stages. The input to the first stage of the G2 register is the exclusive OR of its 2nd, 3rd, 6th, 8th, 9th, and 10th stages. The C/A-code is the exclusive OR of stage 10 of G1, stage 2 of G2, and stage 6 of G2.

Global Positioning Systems, Inertial Navigation, and Integration,
Mohinder S. Grewal, Lawrence R. Weill, Angus P. Andrews

Print ISBN 0-471-35032-X Electronic ISBN 0-471-20071-9

5

GPS Data Errors

5.1 SELECTIVE AVAILABILITY ERRORS

Prior to May 1, 2000, Selective Availability (SA) was a mechanism adopted by the Department of Defense (DoD) to control the achievable navigation accuracy by nonmilitary GPS receivers. In the GPS SPS mode, the SA errors were specified to degrade navigation solution accuracy to 100 m (2D RMS) horizontally and 156 m (RMS) vertically.

In a press release on May 1, 2000, the President of the United States announced the decision to discontinue this intentional degradation of GPS signals available to the public. The decision to discontinue SA was coupled with continuing efforts to upgrade the military utility of systems using GPS and supported by threat assessments which concluded that setting SA to zero would have minimal impact on United States national security. The decision was part of an ongoing effort to make GPS more responsive to civil and commercial users worldwide.

The transition as seen from Colorado Springs, CO., U.S.A. at the GPS Support Center is shown in Figure 5.1. The figure shows the horizontal and vertical errors with SA, and after SA was suspended, midnight GMT (8 PM EDT), May 1, 2000. Figure 5.2 shows mean errors with and without SA, with satellite PRN numbers.

Aviation applications will probably be the most visible user group to benefit from the discontinuance of SA. However, precision approach will still require some form of augmentation to ensure that integrity requirements are met. Even though setting SA to zero reduces measurement errors, it does not reduce the need for and design of WAAS and LAAS ground systems and avionics.

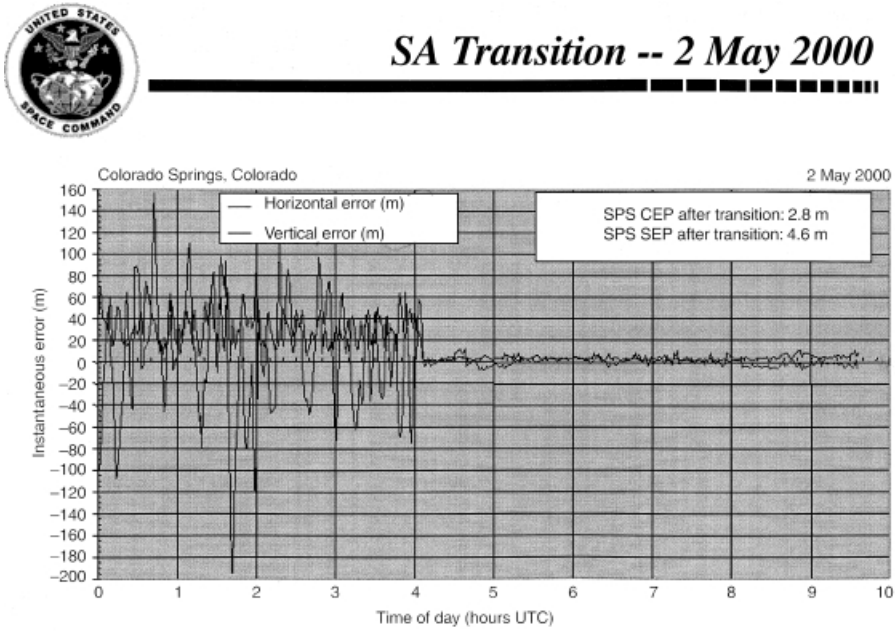


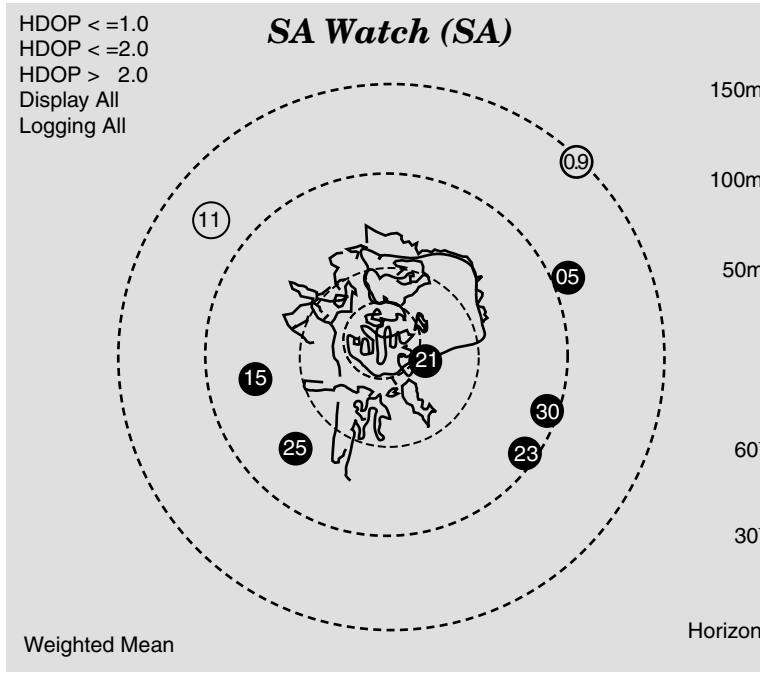
Fig. 5.1 *Pseudorange residuals with SA.*

Time and frequency users may see greater effects in the long term via communication systems that can realize significant future increases in effective bandwidth use due to tighter synchronization tolerances. The effect on vehicle tracking applications will vary. Tracking in the trucking industry requires accuracy only good enough to locate in which city the truck is, whereas public safety applications can require the precise location of the vehicle. Maritime applications have the potential for significant benefits. The personal navigation consumer will benefit from the availability of simpler and less expensive products, resulting in more extensive use of GPS worldwide.

Because SA could be resumed at any time, for example, in time of military alert, one needs to be aware of how to minimize these errors.

There are at least two mechanisms to implement SA. Mechanisms involve the manipulation of GPS ephemeris data and dithering the satellite clock (carrier frequency). The first is referred to as epsilon-SA (ε-SA), and the second as clock-dither SA. The clock-dither SA may be implemented by physically dithering the frequency of the GPS signal carrier or by manipulating the satellite clock correction data or both.

Though the mechanisms to implement SA and the true SA waveform are classified, a variety of SA models exist in the literature (e.g., [4, 15, 21, 134]). These references show various models. One proposed by Braasch [15] appears to be the most promising and suitable. Another used with some success for predicting SA is a Levinson predictor [6].



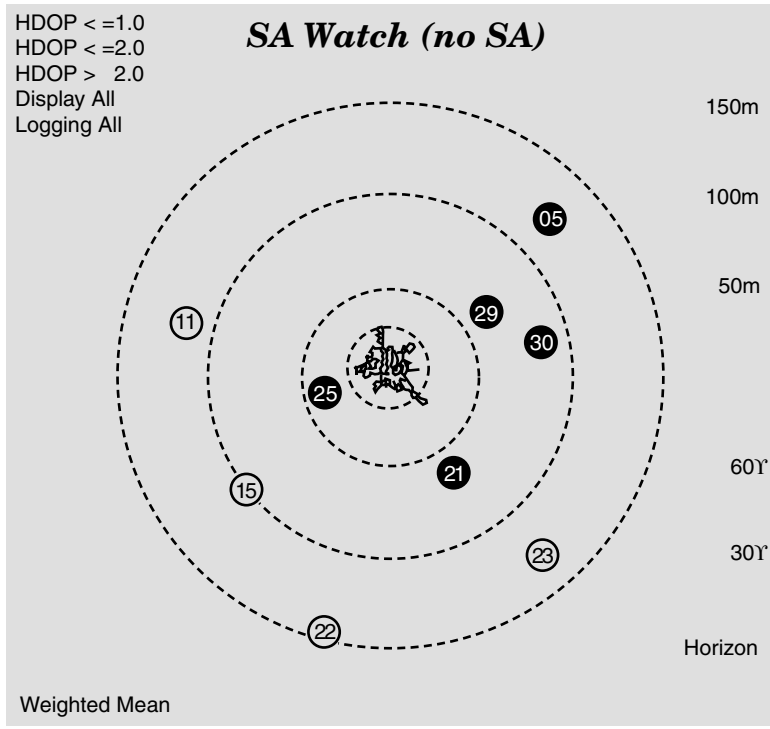


Fig. 5.2 *Pseudorange residuals with and without SA.*

The Braasch model assumes that all SA waveforms are driven by normal white noise through linear system (autoregressive moving-average, ARMA) models (see Chapter 3 of [46]). Using the standard techniques developed in system and parameter identification theory, it is then possible to determine the structure and parameters of the optimal linear system that best describes the statistical characteristics of SA. The problem of modeling SA is estimating the model of a random process (SA waveform) based on the input/output data.

The technique to find an SA model involves three basic elements:

- the observed SA,
- a model structure, and
- a criterion to determine the best model from the set of candidate models.

There are three choices of model structures:

1. an ARMA model of order $(p,\ q)$, which is represented as ARMA$(p,\ q)$;
2. an ARMA model of order $(p,\ 0)$ known as the moving-average MA(p) model; and
3. an ARMA model of order $(q,\ 0)$, the auto regression AR(q) model.

Selection from these three models is performed with physical laws and past experience.

5.1.1 Time Domain Description

Given observed SA data, the identification process repeatedly selects a model structure and then calculates its parameters. The process is terminated when a satisfactory model, according to a certain criterion, is found.

We start with the general ARMA model. Both the AR and MA models can be viewed as special cases of an ARMA model. An ARMA$(p,\ q)$ model is mathematically described by

$$a_1 y_k + a_2 y_{k-1} + \cdots + a_q y_{k-q+1} = b_1 x_k + b_2 x_{k-1} + \cdots + b_p x_{k-p+1} + e_k \qquad (5.1)$$

or in a concise form by

$$\sum_{i=1}^{q} a_i y_{k-i+1} = \sum_{j=1}^{p} b_j x_{k-j+1} + e_k, \qquad (5.2)$$

where $a_i,\ i = 1,\ 2, \ldots, q$, and $b_j,\ j = 1,\ 2, \ldots, p$, are the sets of parameters that describe the model structure, x_k and y_k are the input and output to the model at any time k for $k = 1,\ 2, \ldots,$ and e_k is the noise value at time k. Without loss of generality, it is always assumed that $a_l = 1$.

Once the model parameters a_i and b_j are known, the calculation of y_k for an arbitrary k can be accomplished by

$$y_k = -\sum_{i=2}^{q} a_i y_{k-i+1} + \sum_{j=1}^{p} b_j x_{k-j+1} + e_k. \tag{5.3}$$

It is noted that when all of the a_i in Eq. 5.3 take the value of 0, the model is reduced to the MA (p, 0) model or simply MA(p). When all of the b_j take the value of 0, the model is reduced to the AR(0, q) model or AR(q). In the latter case, y_k is calculated by

$$y_k = -\sum_{i=2}^{q} a_i y_{k-i+1} + e_k\,. \tag{5.4}$$

5.1.1.1 Model Structure Selection Criteria Two techniques, known as Akaike's final prediction error (FPE) criterion and the closely related Akaike's information theoretic criterion (AIC), may be used to aid in the selection of model structure. According to Akaike's theory, in the set of candidate models, the one with the smallest values of FPE or AIC should be chosen. The FPE is calculated as

$$\text{FPE} = \frac{1 + n/N}{1 - n/N}\ V, \tag{5.5}$$

where n is the total number of parameters of the model to be estimated, N is the length of the data record, and V is the loss function for the model under consideration. Here, V is defined as

$$V = \sum_{i=1}^{n}\ e_i^2, \tag{5.6}$$

where e is as defined in Eq. 5.2. The AIC is calculated as

$$\text{AIC} = \log[(1 + 2n/N)V] \tag{5.7}$$

In the following, an AR(12) model was chosen to characterize SA. This selection was based primarily on Braasch's recommendation [14]. As such, the resulting model should be used with caution before the validity of this model structure assumption is further studied using the above criteria.

5.1.1.2 Frequency Domain Description The ARMA models can be equivalently described in the frequency domain, which provides further insight

into model behavior. Introducing a one-step delay operator Z^{-1}, Eq. 5.2 can be rewritten as

$$A(Z^{-1})y_k = B(Z^{-l})x_k + e_k, \tag{5.8}$$

where

$$A(Z^{-1}) = \sum_{i=1}^{q} a_i Z^{-i+l}, \tag{5.9}$$

$$B(Z^{-1}) = \sum_{i=1}^{p} b_i Z^{i+1}, \tag{5.10}$$

and

$$Z^{-1}y_k = y_{k-1}. \tag{5.11}$$

It is noted that $A(Z^{-1})$ and $B(Z^{-1})$ are polynomials of the time-shift operator Z^{-1} and normal arithmetic operations may be carried out under certain conditions. Defining a new function $H(Z^{-1})$ as $B(Z^{-1})$ divided by $A(Z^{-1})$ and expanding the resulting $H(Z^{-1})$ in terms of operator Z^{-1}, we have

$$H(Z^{-1}) = \frac{B(Z^{-1})}{A(Z^{-1})} = \sum_{i=1}^{\infty} h_i Z^{i+1}. \tag{5.12}$$

The numbers of $\{h_i\}$ are the impulse responses of the model. It can be shown that h_i is the output of the ARMA model at time $i = 1,\ 2, \ldots$ when the model input x_i takes the value of zero at all times except for $i = 1$. The function $H(Z^{-1})$ is called the frequency function of the system. By evaluating its value for $Z^{-1} = e^{j\omega}$, the frequency response of the model can be calculated directly. Note that this process is a direct application of the definition of the discrete Fourier transform (DFT) of h_i.

5.1.1.3 AR Model Parameter Estimation

The parameters of an AR model with structure

$$A(Z^{-1})y_k = e_k \tag{5.13}$$

may be estimated using the least-squares (LS) method. If we rewrite Eq. 5.13 in matrix format for $k = q,\ q+1, \ldots, n$, we get

$$\begin{bmatrix} y_n & y_{n-l} & \cdots & y_{n-q+1} \\ y_{n-l} & y_{n-2} & \cdots & y_{n-q} \\ \vdots & \vdots & \vdots & \vdots \\ y_q & y_{q-1} & \cdots & y_1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_q \end{bmatrix} = \begin{bmatrix} e_n \\ e_{n-1} \\ \vdots \\ e_q \end{bmatrix} \tag{5.14}$$

or

$$H \cdot A = E, \tag{5.15}$$

where

$$H = \begin{bmatrix} y_n & y_{n-l} & \cdots & y_{n-q+1} \\ y_{n-l} & y_{n-2} & \cdots & y_{n-q} \\ \vdots & \vdots & \vdots & \vdots \\ y_q & y_{q-1} & \cdots & y_1 \end{bmatrix}, \tag{5.16}$$

$$A = [a_1\ a_2\ a_3\ \cdots\ a_q]^{\mathrm{T}}, \tag{5.17}$$

and

$$E = [e_n \quad e_{n-1} \quad e_{n-2} \quad \cdots \quad e_q]^{\mathrm{T}}. \tag{5.18}$$

The LS estimation of the parameter matrix A can then be obtained by

$$A = (H^T H)^{-1} H^{\mathrm{T}} E. \tag{5.19}$$

5.1.2 Collection of SA Data

To build effective SA models, samples of true SA data must be available. This requirement cannot be met directly as the mechanism of SA generation and the actual SA waveform are classified. The approach we take is to extract SA from flight test data. National Satellite Test Bed (NSTB) flight tests recorded the pseudorange measurements at all 10 RMS (Reference Monitoring Station) locations. These pseudorange measurements contain various clock, propagation, and receiver measurement errors, and they can, in general, be described as

$$\mathrm{PR}_M = \rho + \Delta T_{\mathrm{sat}} + \Delta T_{\mathrm{rcvr}} + \Delta T_{\mathrm{iono}} + \Delta T_{\mathrm{trop}} + \Delta T_{\mathrm{multipath}} + \mathrm{SA} + \Delta t_{\mathrm{noise}} \tag{5.20}$$

where ρ is the true distance between the GPS satellite and the RMS receiver; ΔT_{sat} and ΔT_{rcvr} are the satellite and receiver clock errors; ΔT_{iono} and ΔT_{trop} are the ionosphere and troposphere propagation delays, $\Delta T_{\mathrm{multipath}}$ is the multipath error; SA is the SA error; and $\Delta t_{\mathrm{noise}}$ is the receiver measurement noise.

To best extract SA from PR_M, values of the other terms were estimated. The true distance ρ is calculated by knowing the RMS receiver location and the precise orbit data available from the National Geodetic Survey (NGS) bulletin board. GIPSY/OASIS analysis (GOA) was used for this calculation, which re-created the precise orbit and converted all relevant data into the same coordinate system. Models for propagation and satellite clock errors have been built into GOA, and these were used

to estimate ΔT_{sat}, ΔT_{iono}, and ΔT_{trop}. The receiver clock errors were estimated by the NSTB algorithm using data generated from GOA for the given flight test conditions. From these, a simulated pseudorange PR_{sim} was formed.:

$$\text{PR}_{\text{sim}} = \rho_{\text{sim}} + \Delta T_{\text{sat}_{\text{sim}}} + \Delta T_{\text{rcvr}_{\text{sim}}} + \Delta T_{\text{iono}_{\text{sim}}} + \Delta T_{\text{trop}_{\text{sim}}} \tag{5.21}$$

where $\Delta T_{\text{sat}_{\text{sim}}}$, $\Delta T_{\text{rcvr}_{\text{sim}}}$, $\Delta T_{\text{iono}_{\text{sim}}}$, and $\Delta T_{\text{trop}_{\text{sim}}}$ are, respectively, the estimated values of ΔT_{sat}, ΔT_{rcvr}, ΔT_{iono}, and ΔT_{trop} in the simulation.

From Eqs. 5.20 and 5.21, pseudorange residuals are calculated:

$$\Delta\text{PR} = \text{PR}_M - \text{PR}_{\text{sim}} = \text{SA} + \Delta T_{\text{multipath}} + \Delta t_{\text{noise}} + \Delta T_{\text{models}}, \tag{5.22}$$

where ΔT_{models} stands for the total modeling error, given by

$$\begin{aligned}\Delta T_{\text{models}} = {} & (\rho - \rho_{\text{sim}}) + \left(\Delta T_{\text{sat}} - \Delta T_{\text{sat}_{\text{sim}}}\right) + \left(\Delta T_{\text{rcvr}} - \Delta T_{\text{rcvr}_{\text{sim}}}\right) \\ & + \left(\Delta T_{\text{iono}} - \Delta T_{\text{iono}_{\text{sim}}}\right) + \left(\Delta T_{\text{trop}} - \Delta T_{\text{trop}_{\text{sim}}}\right)\end{aligned} \tag{5.23}$$

It is noted that the terms $\Delta T_{\text{multipath}}$ and Δt_{noise} should be significantly smaller than SA, though it is not possible to estimate their values precisely. The term ΔT_{models} should also be negligible compared to SA. It is, therefore, reasonable to use ΔPR as an approximation to the actual SA term to estimate SA models. Examination of all available data show that their values vary between ± 80 m. These are consistent with previous reports on observed SA and with the DoD's specification of SPS accuracy.

5.2 IONOSPHERIC PROPAGATION ERRORS

The ionosphere, which extends from approximately 50 to 1000 km above the surface of the earth, consists of gases that have been ionized by solar radiation. The ionization produces clouds of free electrons that act as a dispersive medium for GPS signals in which propagation velocity is a function of frequency. A particular location within the ionosphere is alternately illuminated by the sun and shadowed from the sun by the earth in a daily cycle; consequently the characteristics of the ionosphere exhibit a diurnal variation in which the ionization is usually maximum late in midafternoon and minimum a few hours after midnight. Additional variations result from changes in solar activity.

The primary effect of the ionosphere on GPS signals is to change the signal propagation speed as compared to that of free space. A curious fact is that the signal modulation (the code and data stream) is delayed, while the carrier phase is advanced by the same amount. Thus the measured pseudorange using the code is larger than the correct value, while that using the carrier phase is equally smaller. The magnitude of either error is directly proportional to the total electron count (TEC) in a tube of 1 m^2 cross section along the propagation path. The TEC varies spatially

due to spatial nonhomogeneity of the ionosphere. Temporal variations are caused not only by ionospheric dynamics, but also by rapid changes in the propagation path due to satellite motion. The path delay for a satellite at zenith typically varies from about 1 m at night to 5–15 m during late afternoon. At low elevation angles the propagation path through the ionosphere is much longer, so the corresponding delays can increase to several meters at night and as much as 50 m during the day.

Since ionospheric error is usually greater at low elevation angles, the impact of these errors could be reduced by not using measurements from satellites below a certain elevation mask angle. However, in difficult signal environments, including blockage of some satellites by obstacles, the user may be forced to use low-elevation satellites. Mask angles of $5^\circ - 7.5^\circ$ offer a good compromise between the loss of measurements and the likelihood of large ionospheric errors.

The L_1-only receivers in nondifferential operation can reduce ionospheric pseudorange error by using a model of the ionosphere broadcast by the satellites, which reduces the uncompensated ionospheric delay by about 50% on the average. During the day errors as large as 10 m at midlatitudes can still exist after compensation with this model and can be much worse with increased solar activity. Other recently developed models offer somewhat better performance. However, they still do not handle adequately the daily variability of the TEC, which can depart from the modeled value by 25% or more.

The L_1/L_2 receivers in nondifferential operation can take advantage of the dependency of delay on frequency to remove most of the ionospheric error. A relatively simple analysis shows that the group delay varies inversely as the square of the carrier frequency. This can be seen from the following model of the code pseudorange measurements at the L_1 and L_2 frequencies:

$$\rho_i = \rho \pm \frac{k}{f_i^2}, \tag{5.24}$$

where ρ is the error-free pseudorange, ρ_i is the measured pseudorange, and k is a constant that depends on the TEC along the propagation path. The subscript $i = 1, 2$ identifies the measurement at the L_1 or L_2 frequencies, respectively, and the plus or minus sign is identified with respective code and carrier phase pseudorange measurements. The two equations can be solved for both ρ and k. The solution for ρ for code pseudorange measurements is

$$\rho = \frac{f_1^2}{f_1^2 - f_2^2}\rho_1 - \frac{f_2^2}{f_1^2 - f_2^2}\rho_2, \tag{5.25}$$

where f_1 and f_2 are the L_1 and L_2 carrier frequencies, respectively, and ρ_1 and ρ_2 are the corresponding pseudorange measurements.

An equation similar to the above can be obtained for carrier phase pseudorange measurements. However, in nondifferential operation the residual carrier phase

pseudorange error can be greater than either an L_1 or L_2 carrier wavelength, making ambiguity resolution difficult.

With differential operation ionospheric errors can be nearly eliminated in many applications, because ionospheric errors tend to be highly correlated when the base and roving stations are in sufficiently close proximity. With two L_1-only receivers separated by 25 km, the unmodeled differential ionospheric error is typically at the 10–20-cm level. At 100 km separation this can increase to as much as a meter. Additional error reduction using an ionospheric model can further reduce these errors by 25–50%.

5.2.1 Ionospheric Delay Model

J. A. Klobuchar's model [37,76] for ionospheric delay in seconds is given by

$$T_g = \text{DC} + A\left[1 - \frac{x^2}{2} + \frac{x^4}{24}\right] \quad \text{for } |x| \le \frac{\pi}{2} \tag{5.26}$$

where $x = \dfrac{2\pi(t - T_p)}{P}$ (rad)

$\text{DC} = 5$ ns (constant offset)

$T_p = \text{phase} = 50{,}400$ s

$A = \text{amplitude}$

$P = \text{period}$

$t =$ local time of the earth subpoint of the signal intersection with mean ionospheric height (s)

The algorithm assumes this latter height to be 350 km. The DC and phasing T_p are held constant at 5 ns and 14 h (50,400 s) local time.

Amplitude (A) and period (P) are modeled as third-order polynomials:

$$A = \sum_{n=0}^{3} \alpha_n \phi_m^n \quad \text{(s)}, \qquad P = \sum_{n=0}^{3} \beta_n \phi_m^n \quad \text{(s)},$$

where ϕ_m is the geomagnetic latitude of the ionospheric subpoint and α_n, β_n are coefficients selected (from 370 such sets of constants) by the GPS master control station and placed in the satellite navigation upload message for downlink to the user.

For Southbury, Connecticut,

$$\alpha_n = [0.8382 \times 10^{-8},\ -0.745 \times 10^{-8},\ -0.596 \times 10^{-7},\ 0.596 \times 10^{-7}],$$
$$\beta_n = [0.8806 \times 10^{5},\ -0.3277 \times 10^{5},\ -0.1966 \times 10^{6},\ 0.1966 \times 10^{6}]$$

The parameter ϕ_m is calculated as follows:

1. Subtended earth angle (EA) between user and satellite is given by the approximation

$$\text{EA} = \frac{445}{\text{el} + 20^{-4}} \quad \text{(deg)}$$

where el is the elevation of the satellite and with respect to the user equals 15.5°.

2. Geodetic latitude and longitude of the ionospheric subpoint are found using the approximations

$$\text{Iono lat. } \phi_I = \phi_{\text{user}} + \text{EA} \cos \text{AZ} \qquad \text{(deg)}$$

$$\text{Iono long. } \lambda_I = \lambda_{\text{user}} + \frac{\text{EA} \cos \text{AZ}}{\cos \phi_I} \qquad \text{(deg)}$$

where ϕ_{user} = geodetic latitude = 41°
λ_{user} = geodetic longitude = −73°
AZ = azimuth of the satellite with respect to the user, = 112.5°

3. The geodetic latitude is converted to a geomagnetic coordinate system using the approximation

$$\phi_m = \phi_I + 11.6^\circ \cos(\lambda_I - 291^\circ) \qquad \text{(deg)}.$$

The final step in the algorithm is to account for elevation angle effect by scaling with an obliquity factor (SF):

$$\text{SF} = 1 + 2\left[\frac{96^\circ - \text{el}}{90^\circ}\right]^3 \qquad \text{(unitless)}.$$

With scaling, time delay due to ionospheric becomes

$$T_g = \begin{cases} \text{SF(DC)} + A\left(1 - \dfrac{x^2}{2} + \dfrac{x^4}{24}\right), & |x| \le \dfrac{\pi}{2}, \\ \text{SF(DC)}, & |x| \le \dfrac{\pi}{2}, \end{cases}$$

$$T_G = CT_g$$

C = speed of light

where T_g is in seconds and T_G in meters. The MATLAB program Klobuchar_fix.m for computing ionospheric delay is described in Appendix A.

5.2.2 GPS Iono Algorithm

The problem of calculating ionospheric propagation delay from P-code and C/A-code can be formulated in terms of the following measurement equalities:

$$P_{RL1} = \rho + L_{1\text{iono}} + c\tau_{RX1} + c\tau_{\text{GD}}, \tag{5.27}$$

$$P_{RL2} = \rho + \frac{L_{1\text{iono}}}{(f_{L2}/f_{L1})^2} + c\tau_{RX2} + c\frac{\tau_{\text{GD}}}{(f_{L2}/f_{L1})^2}, \tag{5.28}$$

where $P_{RL1} = L_1$ pseudorange
$P_{RL2} = L_2$ pseudorange
ρ = geometric distance between GPS satellite transmitter and GPS receiver, including nondispersive contributions such as tropospheric refraction and clock drift
$f_{L1} = L_1$ frequency = 1575.42 MHz
$f_{L2} = L_2$ frequency = 1227.6 MHz
τ_{RX1} = receiver noise as manifested in code (receiver and calibration biases) at L_1 (ns)
τ_{RX2} = receiver noise as manifested in code (receiver and calibration biases) at L_2 (ns)
τ_{GD} = satellite group delay (interfrequency bias)
c = speed of light = 0.299792458 m/ns

Subtracting Eq. 5.28 from Eq. 5.27, we get

$$L_{1\text{iono}} = \frac{P_{RL1} - P_{RL2}}{1 - (f_{L1}/f_{L2})^2} - \frac{c(\tau_{RX1} - \tau_{RX2})}{1 - (f_{L1}/f_{L2})^2} - c\tau_{GD}. \tag{5.29}$$

What is actually measured in the ionospheric delay is the sum of receiver bias and interfrequency bias. The biases are determined and taken out from the ionospheric delay calculation. These biases may be up to 10 ns (3 m) [33, 92].

However, the presence of ambiguities N_1 and N_2 in carrier phase measurements of L_1 and L_2 preclude the possibility of using these in the daytime by themselves. At night, these ambiguities can be calculated from the pseudoranges and carrier phase measurements may be used for ionospheric calculations.

5.3 TROPOSPHERIC PROPAGATION ERRORS

The lower part of the earth's atmosphere is composed of dry gases and water vapor, which lengthen the propagation path due to refraction. The magnitude of the resulting signal delay depends on the refractive index of the air along the propagation path and typically varies from about 2.5 m in the zenith direction to 10–15 m at low satellite elevation angles. The troposphere is nondispersive at the GPS frequencies, so that delay is not frequency dependent. In contrast to the ionosphere,

tropospheric path delay is consequently the same for code and carrier signal components. Therefore, this delay cannot be measured by utilizing both L_1 and L_2 pseudorange measurements, and either models and/or differential positioning must be used to reduce the error.

The refractive index of the troposphere consists of that due to the dry gas component and the water vapor component, which respectively contribute about 90% and 10% of the total. Knowledge of the temperature, pressure, and humidity along the propagation path can determine the refractivity profile, but such measurements are usually not available to the user. However, using standard atmospheric models for dry delay permits determination of the zenith delay to within about 0.5 m and with an error at other elevation angles that approximately equals the zenith error times the cosecant of the elevation angle. These standard atmospheric models are based on the laws of ideal gases and assume spherical layers of constant refractivity with no temporal variation and an effective atmospheric height of about 40 km. Estimation of dry delay can be improved considerably if surface pressure and temperature measurements are available, bringing the residual error down to within 2–5% of the total.

The component of tropospheric delay due to water vapor (at altitudes up to about 12 km) is much more difficult to model, because there is considerable spatial and temporal variation of water vapor in the atmosphere. Fortunately, the wet delay is only about 10% of the total, with values of 5–30 cm in continental midlatitudes. Despite its variability, an exponential vertical profile model can reduce it to within about 2–5 cm.

In practice, a model of the standard atmosphere at the antenna location would be used to estimate the combined zenith delay due to both wet and dry components. Such models use inputs such as the day of the year and the latitude and altitude of the user. The delay is modeled as the zenith delay multiplied by a factor that is a function of the satellite elevation angle. At zenith, this factor is unity, and it increases with decreasing elevation angle as the length of the propagation path through the troposphere increases. Typical values of the multiplication factor are 2 at 30° elevation angle, 4 at 15°, 6 at 10°, and 10 at 5°. The accuracy of the model decreases at low elevation angles, with decimeter level errors at zenith and about 1 m at 10° elevation.

Much research has gone into the development and testing of various tropospheric models. Excellent summaries of these appear in [56, 65, 110].

Although a GPS receiver cannot measure pseudorange error due to the troposphere, differential operation can usually reduce the error to small values by taking advantage of the high spatial correlation of tropospheric errors at two points within 100–200 km on the earth's surface. However, exceptions often occur when storm fronts pass between the receivers, causing large gradients in temperature, pressure, and humidity.

5.4 THE MULTIPATH PROBLEM

Multipath propagation of the GPS signal is a dominant source of error in differential positioning. Objects in the vicinity of a receiver antenna (notably the ground) can

easily reflect GPS signals, resulting in one or more secondary propagation paths. These secondary-path signals, which are superimposed on the desired direct-path signal, always have a longer propagation time and can significantly distort the amplitude and phase of the direct-path signal.

Errors due to multipath cannot be reduced by the use of differential GPS, since they depend on local reflection geometry near each receiver antenna. In a receiver without multipath protection, C/A-code ranging errors of 10 m or more can be experienced. Multipath can not only cause large code ranging errors, but can also severely degrade the ambiguity resolution process required for carrier phase ranging such as that used in precision surveying applications.

Multipath propagation can be divided into two classes: static and dynamic. For a stationary receiver, the propagation geometry changes slowly as the satellites move across the sky, making the multipath parameters essentially constant for perhaps several minutes. However, in mobile applications there can be rapid fluctuations in fractions of a second. Therefore, different multipath mitigation techniques are generally employed for these two types of multipath environments. Most current research has been focused on static applications, such as surveying, where greater demand for high accuracy exists. For this reason, we will confine our attention to the static case.

5.5 HOW MULTIPATH CAUSES RANGING ERRORS

To facilitate an understanding of how multipath causes ranging errors, several simplifications can be made that in no way obscure the fundamentals involved. We will assume that the receiver processes only the C/A-code and that the received signal has been converted to complex (i.e., analytic) form at baseband (nominally zero frequency), where all Doppler shift has been removed by a carrier tracking phase-lock loop. It is also assumed that the 50-bps GPS data modulation has been removed from the signal, which can be achieved by standard techniques. When no multipath is present, the received waveform is represented by

$$r(t) = ae^{j\phi}c(t-\tau) + n(t), \tag{5.30}$$

where $c(t)$ is the normalized, undelayed C/A-code waveform as transmitted, τ is the signal propagation delay, a is the signal amplitude, ϕ is the carrier phase, and $n(t)$ is Gaussian receiver thermal noise having flat power spectral density. Pseudoranging consists of estimating the delay parameter τ. As we have previously seen, an optimal estimate (i.e., a minimum-variance unbiased estimate) of τ can be obtained by forming the cross-correlation function

$$R(\tau) = \int_{T_1}^{T_2} r(t)c_r(t-\tau)\,dt \tag{5.31}$$

of $r(t)$ with a replica $c_r(t)$ of the transmitted C/A-code and choosing as the delay estimate that value of τ that maximizes this function. Except for an error due to

receiver thermal noise, this occurs when the received and replica waveforms are in time alignment. A typical cross-correlation function without multipath for C/A-code receivers having a 2-MHz precorrelation bandwidth is shown by the solid lines Fig. 5.3 (these plots ignore the effect of noise, which would add small random variations to the curves).

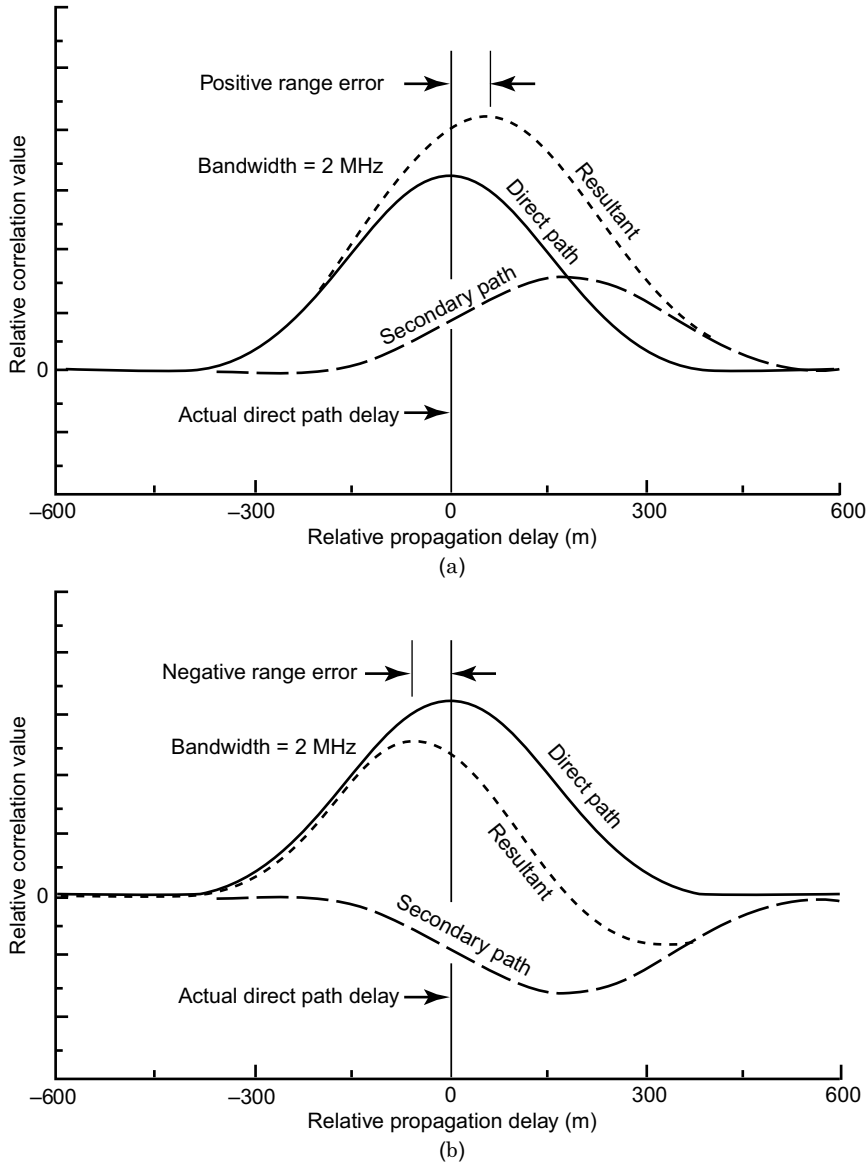


Fig. 5.3 *Effect of multipath on C/A-code cross-correlation function.*

If multipath is present with a single secondary path, the waveform of Eq. 5.30 changes to

$$r(t) = ae^{j\phi_1}c(t-\tau_1) + be^{j\phi_2}c(t-\tau_2) + n(t), \tag{5.32}$$

where the direct and secondary paths have respective propagation delays τ_1 and τ_2, amplitudes a and b, and carrier phases ϕ_1 and ϕ_2. In a receiver not designed expressly to handle multipath, the resulting cross-correlation function will now have two superimposed components, one from the direct path and one from the secondary path. The result is a function with a distortion depending on the relative amplitude, delay, and phase of the secondary-path signal, as illustrated at the top of Fig. 5.3 for an in-phase secondary path and at the bottom of the figure for an out-of-phase secondary path. Most importantly, the location of the peak of the function has been displaced from its correct position, resulting in a pseudorange error.

In vintage receivers employing standard code tracking techniques (early and late codes separated by one C/A-code chip), the magnitude of pseudorange error caused by multipath can be quite large, reaching 70–80 m for a secondary-path signal one-half as large as the direct-path signal and having a relative delay of approximately 250 m. Further details can be found in [51].

5.6 METHODS OF MULTIPATH MITIGATION

Processing against slowly changing multipath can be broadly separated into two classes: spatial processing and time domain processing. Spatial processing uses antenna design in combination with known or partially known characteristics of signal propagation geometry to isolate the direct-path received signal. In contrast, time domain processing achieves the same result by operating only on the multipath-corrupted signal within the receiver.

5.6.1 Spatial Processing Techniques

Antenna Location Strategy Perhaps the simplest form of spatial processing is to locate the antenna where it is less likely to receive reflected signals. For example, to obtain the position of a point near reflective objects, one can first use GPS to determine the position of a nearby point "in the clear" and then calculate the relative position of the desired point by simple distance and/or angle measurement techniques. Another technique that minimizes ever-present ground signal reflections is to place the receiver antenna directly at ground level. This causes the point of ground reflection to be essentially coincident with the antenna location so that the secondary path very nearly has the same delay as the direct path. Clearly such antenna location strategies may not always be possible but can be very effective when feasible.

Groundplane Antennas The most common form of spatial processing is an antenna designed to attenuate signals reflected from the ground. A simple design uses a metallic groundplane disk centered at the base of the antenna to shield the antenna from below. A deficiency of this design is that when the signal wavefronts arrive at the disk edge from below, they induce surface waves on the top of the disk that then travel to the antenna. The surface waves can be eliminated by replacing the groundplane with a *choke ring*, which is essentially a groundplane containing a series of concentric circular troughs one-quarter wavelength deep. These troughs act as transmission lines shorted at the bottom ends so that their top ends exhibit a very high impedance at the GPS carrier frequency. Therefore, induced surface waves cannot form, and signals that arrive from below the horizontal plane are significantly attenuated. However, the size, weight, and cost of a choke-ring antenna is significantly greater than that of simpler designs. Most importantly, the choke ring cannot effectively attenuate secondary-path signals arriving from above the horizontal, such as those reflecting from buildings or other structures. Nevertheless, such antennas have proven to be effective when signal ground bounce is the dominant source of multipath, particularly in GPS surveying applications.

Directive Antenna Arrays A more advanced form of spatial processing uses antenna arrays to form a highly directive spatial response pattern with high gain in the direction of the direct-path signal and attenuation in directions from which secondary-path signals arrive. However, inasmuch as signals from different satellites have different directions of arrival and different multipath geometries, many directivity patterns must be simultaneously operative, and each must be capable of adapting to changing geometry as the satellites move across the sky. For these reasons, highly directive arrays seldom are practical or affordable for most applications.

Long-Term Signal Observation If a GPS signal is observed for sizable fractions of an hour to several hours, one can take advantage of changes in multipath geometry caused by satellite motion. This motion causes the relative delays between the direct and secondary paths to change, resulting in measurable variations in the received signal. For example, a periodic change in signal level caused by alternate phase reinforcement and cancellation by the reflected signals is often observable. Although a variety of algorithms have been proposed for extracting the direct-path signal component from measurements of the received signal, the need for long observation times rules out this technique for most applications. However, it can be an effective method of multipath mitigation at a fixed site, such as at a differential GPS base station. In this case, it is even possible to observe the same satellites from one day to the next, looking for patterns of pseudorange or phase measurements that repeat daily.

5.6.2 Time Domain Processing

Despite the fact that time domain processing against GPS multipath errors has been the subject of active research for at least two decades, there is still much to be learned, both at theoretical and practical levels. Most of the practical approaches have been developed by receiver manufacturers, who are often reluctant to explicitly reveal their methods. Nevertheless, enough information about multipath processing exists to gain insight into its recent evolution.

Narrow-Correlator Technology (1990–1993) The first significant means to reduce GPS multipath effects by receiver processing made its debut in the early 1990s. Until that time, most receivers had been designed with a 2-MHz precorrelation bandwidth that encompassed most, but not all, of the GPS spread-spectrum signal power. These receivers also used one-chip spacing between the early and late reference C/A-codes in the code tracking loops. However, the 1992 paper [121] makes it clear that using a significantly larger bandwidth combined with much closer spacing of the early and late reference codes would dramatically improve the ranging accuracy both with and without multipath. It is somewhat surprising that these facts were not recognized earlier by the GPS community, given that they had been well known in radar circles for many decades.

A 2-MHz precorrelation bandwidth causes the peak of the direct-path cross-correlation function to be severely rounded, as illustrated in Fig. 5.3. Consequently, the sloping sides of a secondary-path component of the correlation function can significantly shift the location of the peak, as indicated in the figure. The result of using an 8-MHz bandwidth is shown in Fig. 5.4, where it can be noted that the sharper peak of the direct-path cross-correlation function is less easily shifted by the secondary-path component. It can also be shown that at larger bandwidths the sharper peak is more resistant to disturbance by receiver thermal noise, even though the precorrelation signal-to-noise ratio is increased.

Another advantage of a larger precorrelation bandwidth is that the spacing between the early and late reference codes in a code tracking loop can be made smaller without significantly reducing the gain of the loop, hence the name *narrow correlator*. It can be shown that this causes the noises on the early and late correlator outputs to become more highly correlated, resulting in less noise on the loop error signal. An additional benefit is that the code tracking loop will be affected only by the multipath-induced distortions near the peak of the correlation function.

Leading-Edge Techniques Because the direct-path signal always precedes secondary-path signals, the leading (left-hand) portion of the correlation function is uncontaminated by multipath, as is illustrated in Fig. 5.4. Therefore, if one could measure the location of just the leading part, it appears that the direct-path delay could be determined with no error due to multipath. Unfortunately, this seemingly happy state of affairs is illusory. With a small direct- to secondary-path separation, the uncontaminated portion of the correlation function is a miniscule piece at the extreme left, where the curve just begins to rise. In this region, not only is the signal-

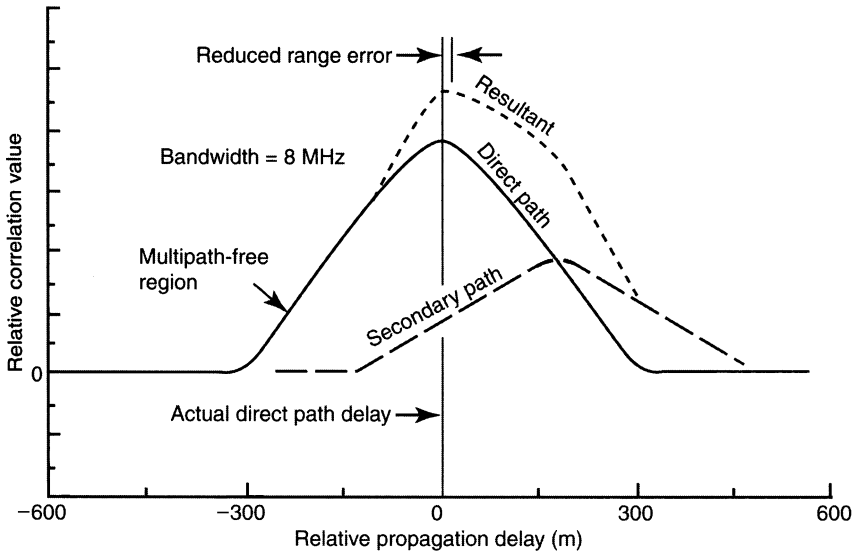


Fig. 5.4 *Reduced multipath error with larger precorrelation bandwidth.*

to-noise ratio relatively poor, but the slope of the curve is also relatively small, which severely degrades the accuracy of delay estimation.

For these reasons, the leading-edge approach best suits situations with a moderate to large direct- to secondary-path separation. However, even in these cases there is the problem of making the delay measurement insensitive to the slope of the correlation function leading edge, which can vary with signal strength. Such a problem does not occur when measuring the location of the correlation function peak.

Correlation Function Shape-Based Methods Some GPS receiver designers have attempted to determine the parameters of the multipath model from the shape of the correlation function. The idea has merit, but for best results many correlations with different values of reference code delay are required to obtain a sampled version of the function shape. Another practical difficulty arises in attempting to map each measured shape into a corresponding direct-path delay estimate. Even in the simple two-path model (Eq. 5.32) there are six signal parameters, so that a very large number of correlation function shapes must be handled. An example of a heuristically developed shape-based approach called the early–late slope (ELS) method can be found in [119], while a method based on maximum-likelihood estimation called the multipath-estimating delay-lock loop (MEDLL) is described in [120].

Modified Correlator Reference Waveforms A relatively new approach to multipath mitigation alters the waveform of the correlator reference PRN code to provide a cross-correlation function with inherent resistance to errors caused by

multipath. Examples include the strobe correlator as described in [39], the use of special code reference waveforms to narrow the correlation function developed in [127, 128], and the gated correlator developed in [90]. These techniques take advantage of the fact that the range information in the received signal resides primarily in the chip transitions of the C/A-code. By using a correlator reference waveform that is not responsive to the flat portions of the C/A-code, the resulting correlation function can be narrowed down to the width of a chip transition, thereby being almost immune to multipath having a primary- to secondary-path separation greater than 30–40 m. An example of such a reference waveform and the corresponding correlation function are shown in Fig. 5.5.

MMT Technology At the time of this writing, a recently developed proprietary mitigation approach called multipath mitigation technology (MMT) is being marketed to receiver manufacturers. The MMT technique appears to reach a theoretical performance limit, described in the next section, for both code and carrier phase ranging. It also has the advantage that its performance improves as the signal observation time is lengthened. The method used by MMT can be found in [130].

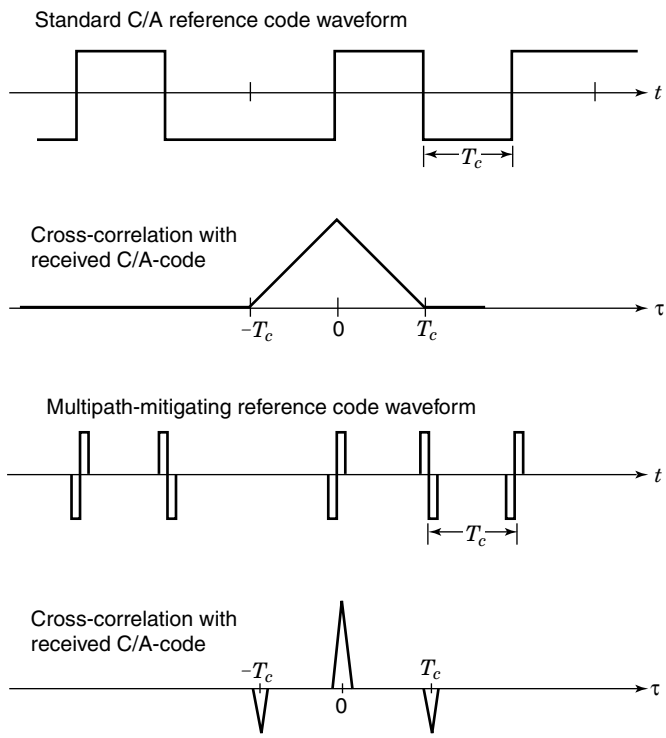


Fig. 5.5 *Multipath-mitigating reference code waveform.*

5.6.3 Performance of Time Domain Methods

Ranging with the C/A-Code Typical C/A-code ranging performance curves for several multipath mitigation approaches are shown in Fig. 5.6 for the case of an in-phase secondary path with amplitude one-half that of the direct path. Even with the best available methods, peak range errors of 3–6 m are not uncommon. It can be observed that the error tends to be largest for "close-in" multipath, where the separation of the two paths is less than 20–30 m. Indeed, this region poses the greatest challenge in multipath mitigation research because the extraction of direct-path delay from a signal with small direct- to secondary-path separation is an ill-conditioned parameter estimation problem.

A serious limitation of most existing multipath mitigation algorithms is that the residual error is mostly in the form of a bias that cannot be removed by further filtering or averaging. On the other hand, the previously mentioned MMT algorithm overcomes this limitation and also appears to have significantly better performance than other published algorithms, as is indicated by curve *F* of Fig. 5.6.

Carrier Phase Ranging The presence of multipath also causes errors in estimating carrier phase, which limits the performance in surveying and other precision applications, particularly with regard to carrier phase ambiguity resolution. Not all current multipath mitigation algorithms are capable of reducing multipath-induced phase error. The most difficult situation occurs at small separations between the direct and secondary paths (less than a few meters). It can be shown that under such conditions essentially no mitigation is theoretically possible. Typical phase error curves for the MMT algorithm, which appears to have the best performance of published methods, is shown in Fig. 5.7.

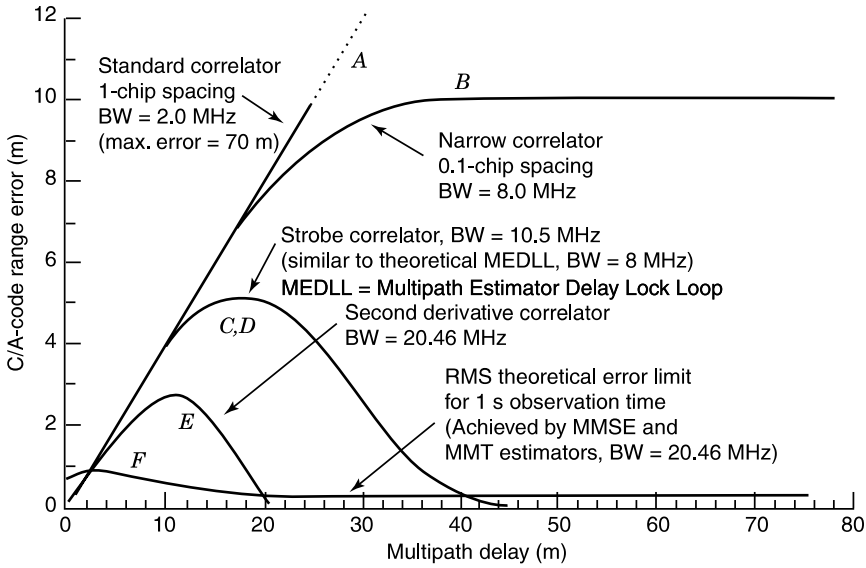


Fig. 5.6 *Performance of various multipath mitigation approaches.*

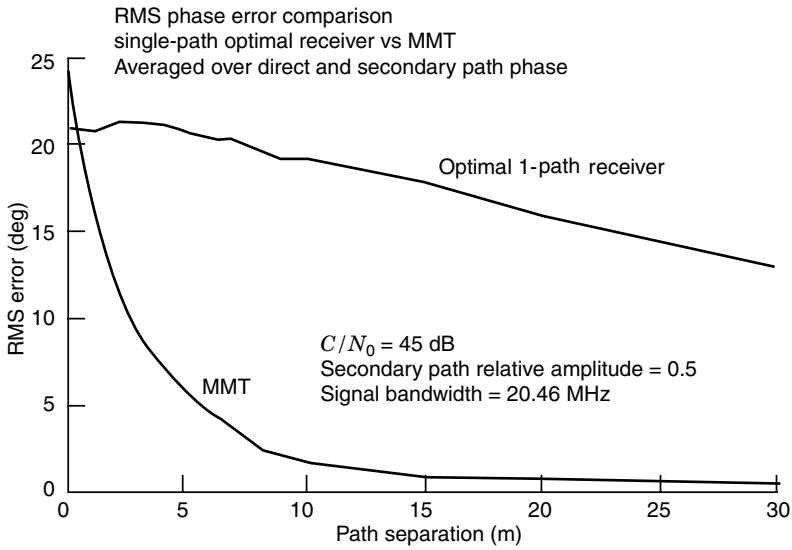


Fig. 5.7 *Residual multipath phase error using MMT algorithm.*

Testing Receiver Multipath Performance Conducting meaningful tests of receiver multipath mitigation performance on either an absolute or a comparative basis is no easy matter. There are often two conflicting goals. On the one hand, the testing should be under strictly controlled conditions, so that the signal levels and true multipath parameters are precisely known; otherwise the measured performance cannot be linked to the multipath conditions that actually exist. Generally this will require precision signal simulators and other ancillary equipment to generate accurately characterized multipath signals.

On the other hand, receiver end users place more credence on how well a receiver performs in the field. However, meaningful field measurements pose a daunting challenge. It is extremely difficult to know the amount and character of the multipath, and great difficulty can be experienced in isolating errors caused by multipath from those of other sources. To add to these difficulties, it is not clear that either the receiver manufacturers or the users have a good feel for the range of multipath parameter values that represent typical operation in the field.

5.7 THEORETICAL LIMITS FOR MULTIPATH MITIGATION

5.7.1 Estimation-Theoretic Methods

Relatively little has been published on multipath mitigation from the fundamental viewpoint of statistical estimation theory, despite the power of its methods and its ability to reach theoretical performance limits in many cases. Knowledge of such

limits provides a valuable benchmark in receiver design by permitting an accurate assessment of the potential payoff in developing techniques that are better than those in current use. Of equal importance is the revelation of the signal processing operations that can reach performance bounds. Although it may not be feasible to implement the processing directly, its revelation often leads to a practical method that achieves nearly the same performance.

Optimality Criteria In discussing theoretical performance limits, it is important to define the criterion of optimality. In GPS the optimal range estimator is traditionally considered to be the minimum-variance unbiased estimator (MVUE), which can be realized by properly designed receivers. However, it can be shown (see [129]) that the standard deviation of a MVUE designed for multipath becomes infinite as the primary- to secondary-path separation approaches zero. For this reason it seems that a better criterion of optimality would be the minimum RMS error, which can include both random and bias components. Unfortunately, it can be shown that *no* estimator exists having minimum RMS error for *every* combination of true multipath parameters.

5.7.3 MMSE Estimator

There is an estimator that can be claimed optimal in a weaker sense. The *minimum-mean-square-error* (MMSE) estimator has the property that no other estimator has a uniformly smaller RMS error. In other words, if some other estimator has smaller RMS error than the MMSE estimator for some set of true multipath parameter values, then that estimator must have a *larger* RMS error than the MMSE estimator for some *other* set of values.

The MMSE estimator also has an important advantage not possessed by most current multipath mitigation methods in that the RMS error decreases as the length of the signal observation interval is increased.

5.7.4 Multipath Modeling Errors

Although a properly designed estimation-theoretic approach such as the MMSE estimator will generally outperform other methods, the design of such estimators requires a mathematical model of the multipath-contaminated signal containing parameters to be estimated. If the actual signal departs from the assumed model, performance degradation can occur. For example, if the model contains only two signal propagation paths but in reality the signal is arriving via three or more paths, large bias errors in range estimation can result. On the other hand, poorer performance (usually in the form of random error cause by noise) can also occur if the model has too many degrees of freedom. Striking the right balance in the number of parameters in the model can be difficult if little information exists about the multipath reflection geometry.

5.8 EPHEMERIS DATA ERRORS

Small errors in the ephemeris data transmitted by each satellite cause corresponding errors in the computed position of the satellite (here we exclude the ephemeris error component of SA, which is regarded as a separate error source). Satellite ephemerides are determined by the master control station of the GPS ground segment based on monitoring of individual signals by four monitoring stations. Because the locations of these stations are known precisely, an "inverted" positioning process can calculate the orbital parameters of the satellites as if they were users. This process is aided by precision clocks at the monitoring stations and by tracking over long periods of time with optimal filter processing. Based on the orbital parameter estimates thus obtained, the master control station uploads the ephemeris data to each satellite, which then transmits the data to users via the navigation data message. Errors in satellite position when calculated from the ephemeris data typically result in range errors less than 1 m. Improvements in satellite tracking will undoubtedly reduce this error further.

5.9 ONBOARD CLOCK ERRORS

Timing of the signal transmission from each satellite is directly controlled by its own atomic clock without any corrections applied. This time frame is called *space vehicle (SV) time*. A schematic of a rubidium atomic clock is shown in Fig. 5.8. Although the atomic clocks in the satellites are highly accurate, errors can be large enough to require correction. Correction is needed partly because it would be difficult to directly synchronize the clocks closely in all the satellites. Instead, the clocks are allowed some degree of relative drift that is estimated by ground station observations and is used to generate clock correction data in the GPS navigation message. When SV time is corrected using this data, the result is called *GPS time*. The time of transmission used in calculating pseudoranges must be in GPS time, which is common to all satellites.

The onboard clock error is typically less than 1 ms and varies slowly. This permits the correction to be specified by a quadratic polynomial in time whose coefficients are transmitted in the navigation message. The correction has the form

$$\Delta t_{\mathrm{SV}} = a_{f0} + a_{f1}(t_{\mathrm{sv}} - t_{0c}) + a_{f2}(t_{\mathrm{SV}} - t_{0c}^2) + \Delta t_R \tag{5.33}$$

with

$$t_{\mathrm{GPS}} = t_{\mathrm{SV}} - \Delta t_{\mathrm{SV}} \tag{5.34}$$

where $a_{f0},\ a_{f1},\ a_{f2}$ are the correction coefficients, t_{SV} is SV time, and Δt_R is a small relativistic clock correction caused by the orbital eccentricity. The clock data reference time t_{0c} in seconds is broadcast in the navigation data message. The stability of the atomic clocks permits the polynomial correction given by Eq. 5.33 to

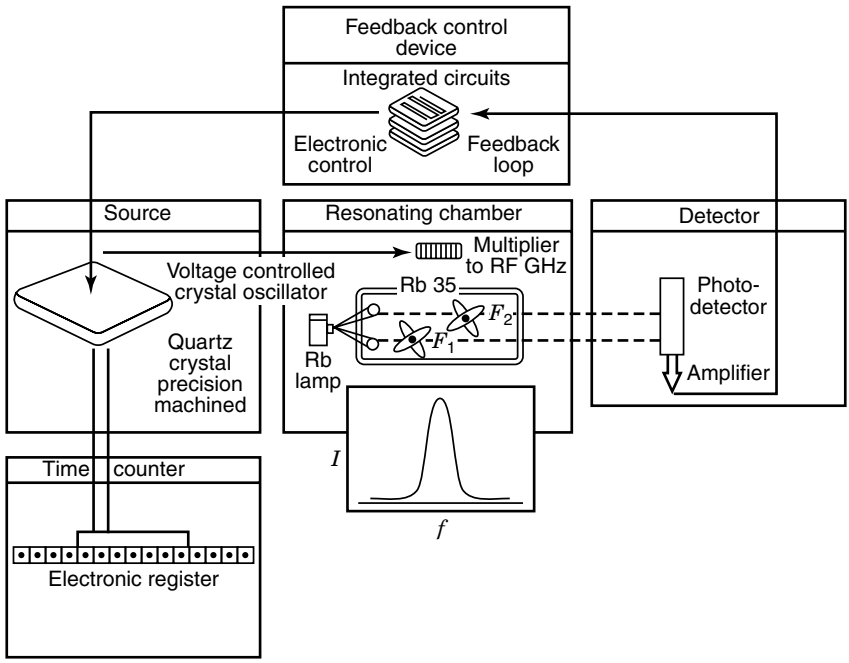


Fig. 5.8 *Schematic of a rubidium atomic clock.*

be valid over a time interval of 4–6 h. After the correction has been applied, the residual error in GPS time is typically less than a few nanoseconds, or about 1 m in range.

5.10 RECEIVER CLOCK ERRORS

Because the navigation solution includes a solution for receiver clock error, the requirements for accuracy of receiver clocks is far less stringent than for the GPS satellite clocks. In fact, for receiver clocks short-term stability over the pseudorange measurement period is usually more important than absolute frequency accuracy. In almost all cases such clocks are quartz crystal oscillators with absolute accuracies in the 1–10-ppm range over typical operating temperature ranges. When properly designed, such oscillators typically have stabilities of 0.01–0.05 ppm over a period of a few seconds.

Receivers that incorporate receiver clock error in the Kalman filter state vector need a suitable mathematical model of the crystal clock error. A typical model in the continuous-time domain is shown in Fig. 5.9, which is easily changed to a discrete version for the Kalman filter. In this model the clock error consists of a bias (frequency) component and a drift (time) component. The frequency error compo-

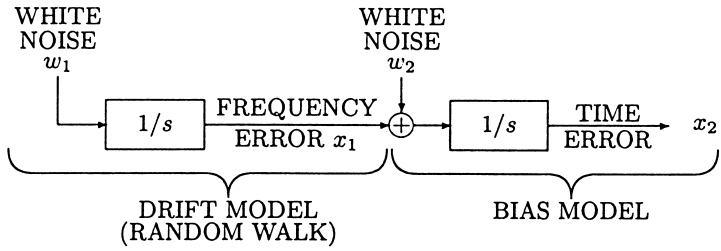


Fig. 5.9 *Crystal clock error model.*

nent is modeled as a random walk produced by integrated white noise. The time error component is modeled as the integral of the frequency error after additional white noise (statistically independent from that causing the frequency error) has been added to the latter. In the model the key parameters that need to be specified are the power spectral densities of the two noise sources, which depend on characteristics of the specific crystal oscillator used.

The continuous time model has the form

$$\dot{x}_1 = w_1,$$
$$\dot{x}_2 = x_1 + w_2,$$

where $w_1(t)$ and $w_2(t)$ are independent zero-mean white-noise processes with known variances.

The equivalent discrete-time model has the state vector

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

and the stochastic process model

$$\mathbf{x}_k = \begin{bmatrix} 1 & 0 \\ \Delta t & 1 \end{bmatrix} \mathbf{x}_{k-1} + \begin{bmatrix} w_{1,k-1} \\ w_{2,k-1} \end{bmatrix}, \tag{5.35}$$

where Δt is the discrete-time step and $\{w_{1,k-1}\}$, $\{w_{2,k-1}\}$ are independent zero-mean white-noise sequences with known variances.

5.11 ERROR BUDGETS

For purposes of analyzing the effects of the previously discussed errors, it is convenient to convert each error into an equivalent range error experienced by a user, which is called the *user-equivalent range error* (UERE). In general, the errors from different sources will have different statistical properties. For example, satellite

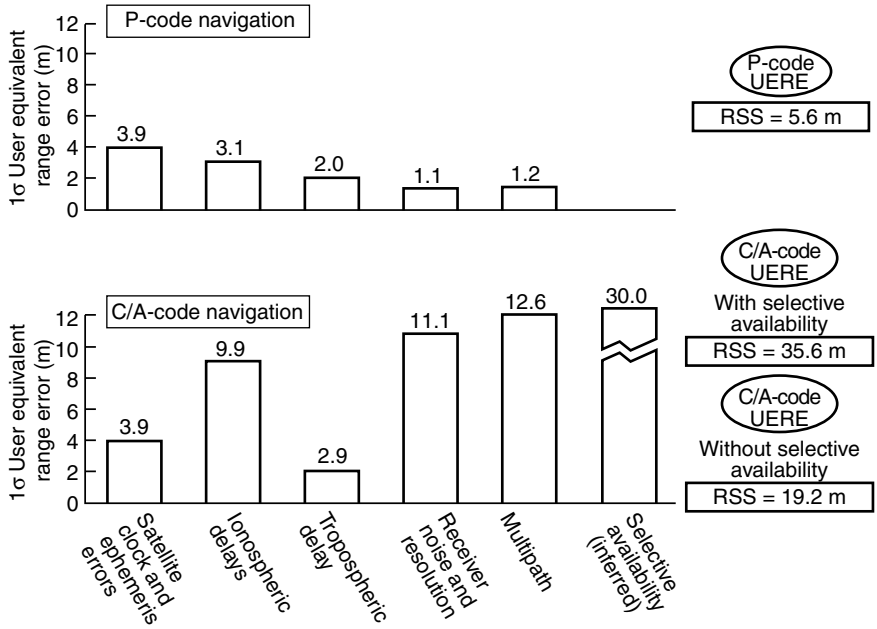


Fig. 5.10 *GPS UERE budget.*

clock and ephemeris errors tend to vary slowly with time and appear as biases over moderately long time intervals, perhaps hours. On the other hand, errors due to receiver noise and quantization effects may vary much more rapidly, perhaps within seconds. Nonetheless, if sufficiently long time durations over many navigation scenarios are considered, all errors can be considered as zero-mean random processes that can be combined to form a single UERE. This is accomplished by forming the root-sum-square of the UERE errors from all sources:

$$\text{UERE} = \sqrt{\sum_{i=1}^{n} (\text{UERE})_i^2}. \tag{5.36}$$

Figure 5.10 depicts the various GPS UERE errors and their combined effect for both C/A-code and P(Y)-code navigation at the 1-σ level.

When SA is on, the UERE for the C/A-code user is about 36 m and reduces to about 19 m when it is off. Aside from SA, it can be seen that for such a user the dominant error sources in nondifferential operations are multipath, receiver noise/resolution, and ionospheric delay (however, recent advances in receiver technology have in some cases significantly reduced receiver noise/resolution errors). On the other hand, the P(Y)-code user has a significantly smaller UERE of about 6 m, for the following reasons:

1. *Errors due to SA can be removed, if present*. The authorized user can employ a key to eliminate them.
2. *The full use of the L_1 and L_2 signals permits significant reduction of ionospheric error.*
3. *The wider bandwidth of the P(Y)-codes greatly reduces errors due to multipath and receiver noise.*

Problems

5.1 Using the values provided for the Klobuchar's model in Section 5.2, for Southbury, Connecticut, calculate the ionospheric delay and plot the results.

5.2 Assume that a direct-path GPS L_1 C/A-code signal arrives with a phase such that all of the signal power lies in the baseband I-channel, so that the baseband signal is purely real. Further assume an infinite signal bandwidth so that the cross-correlation of the baseband signal with an ideal C/A reference code waveform will be an isosceles triangle 600 m wide at the base.

(a) Suppose that in addition to the direct-path signal there is a secondary-path signal arriving with a relative time delay of precisely 250 L_1 carrier cycles (so that it is in phase with the direct-path signal) and with an amplitude one-half that of the direct path. Calculate the pseudorange error that would result, including its sign, under noiseless conditions. Assume that pseudorange is measured with a delay-lock loop using 0.1-chip spacing between the early and late reference codes. *Hint:* The resulting cross-correlation function is the superposition of the cross-correlation functions of the direct- and secondary-path signals.

(b) Repeat the calculations of part (a) but with a secondary-path relative time delay of precisely $250\frac{1}{2}$ carrier cycles. Note that in this case the secondary-path phase is 180° out of phase with the direct-path signal, but still lies entirely in the baseband I-channel.

5.3 (a) Using the discrete matrix version of the receiver clock model given by Eq. 5.35, find the standard deviation σ_{w_1} of the white-noise sequence $w_{1,k}$ needed in the model to produce a frequency standard deviation σ_{x_1} of 1 Hz after 10 min of continuous oscillator operation. Assume that the initial frequency error at $t = 0$ is zero and that the discrete-time step Δt is 1 s.

(b) Using the assumptions and the value of σ_{w_1} found in part (a), find the standard deviation σ_{x_2} of the bias error after 10 min. Assume that $\sigma_{w_2} = 0$.

(c) Show that σ_{x_1} and σ_{x_2} approach infinity as the time t approaches infinity. Will this cause any problems in the development of a Kalman filter that includes estimates of the clock frequency and bias error?

Global Positioning Systems, Inertial Navigation, and Integration,
Mohinder S. Grewal, Lawrence R. Weill, Angus P. Andrews

Print ISBN 0-471-35032-X Electronic ISBN 0-471-20071-9

6
Inertial Navigation

6.1 BACKGROUND

A more basic introduction to the fundamental concepts of inertial navigation can be found in Chapter 2, Section 2.2. Readers who are not already familiar with inertial navigation should review that section before starting this chapter.

6.1.1 History of Inertial Navigation

Inertial navigation has had a relatively short but intense history of development, much of it during the half-century of the Cold War, with contributions from thousands of engineers and scientists. The following is only an outline of developments in the United States. More details can be found, for example, in [22, 43, 75, 83, 88, 107, 135].

6.1.1.1 Gyroscopes The word "gyroscope" was first used by Jean Bernard Léon Foucault (1819–1868), who coined the term from the Greek words for turn (*γύρος*) and view (*σκόπς*). Foucault used one to demonstrate the rotation of the earth in 1852. Elmer Sperry (1860–1930) was one of the early pioneers in the development of gyroscope technology. Gyroscopes were applied to dead reckoning navigation for iron ships (which could not rely on a magnetic compass) around 1911, to automatic steering of ships in the 1920s, for steering torpedos in the 1920s, and for heading and artificial horizon displays for aircraft in the 1920s and 1930s. Rockets designed by Robert H. Goddard in the 1930s also used gyroscopes for steering, as

did the autopilots for the German V-1 cruise missiles and V-2 ballistic missiles of World War II.

6.1.1.2 Relation to Guidance and Control

Navigation is concerned with determining where you are relative to where you want to be, *guidance* with getting yourself to your destination, and *control* with staying on track. There has been quite a bit of synergism among these disciplines, especially in the development of missile technologies where all three could use a common set of sensors, computing resources, and engineering talent. As a consequence, the history of development of inertial navigation technology has a lot of overlap with that of guidance and control.

6.1.1.3 Gimbaled INS

Gimbals have been used for isolating gyroscopes from rotations of their mounting bases since the time of Foucault. They have been used for isolating an inertial sensor cluster in a gimbaled inertial measurement unit (IMU) since about 1950. Charles Stark Draper at the Instrumentation Laboratory at MIT (later the Charles Stark Draper Laboratory) played a major role in the development of gyroscope and INS technology for use on aircraft and ships. Much of the early INS development was for use on military vehicles. An early impetus for INS technology development for missiles was the Navaho Project, started soon after World War II by the U.S. Air Force for a supersonic cruise missile to carry a 15,000-lb payload (the atomic bomb of that period), cruising at Mach 3.25 at 90,000 ft for 5500 miles, and arriving with a navigation accuracy of about 1 nautical mile. The project was canceled in 1957 when nuclear devices had been shrunk to a size that could be carried by the rockets of the day, but by then the prime contractor, North American Aviation, had developed an operational INS for it. This technology was soon put to use in the intercontinental ballistic missiles that replaced Navaho, as well as in many military aircraft and ships. The navigation of the submarine *Nautilus* under the polar ice cap in 1958 would not have been possible without its INS. It was a gimbaled INS, as were nearly all such systems until the 1970s.

6.1.1.4 Early Strapdown Systems

A gimbaled INS was carried on each of nine Apollo command modules from the earth to the moon and back between December 1968 and December 1972, but a strapdown INS was carried on each of the six[1] Lunar Excursion Modules (LEMs) that shuttled two astronauts from lunar orbit to the lunar surface and back.

6.1.1.5 Navigation Computers

Strapdown INSs generally require more powerful navigation computers than their gimbaled counterparts. It was the development of silicon integrated circuit technology in the 1960s and 1970s that enabled strapdown systems to compete with gimbaled systems in all applications but those demanding extreme precision, such as ballistic missiles or submarines.

[1] Two additional LEMs were carried to the moon but did not land there. The Apollo 13 LEM did not make its intended lunar landing but played a far more vital role in crew survival.

6.1.2 Performance

Integration of acceleration sensing errors causes INS velocity errors to grow linearly with time, and Schuler oscillations (Section 2.2.2.3) tend to keep position errors proportional to velocity errors. As a consequence, INS position errors tend to grow linearly with time. These errors are generally not known, except in terms of their statistical properties. INS performance is also characterized in statistical terms.

6.1.2.1 CEP Rate A *circle of equal probability* (CEP) is a circle centered at the estimated location of an INS on the surface of the earth, with radius such that it is equally likely that the true position is either inside or outside that circle. The CEP radius is a measure of position uncertainty. *CEP rate* is a measure of how fast position uncertainty is growing.

6.1.2.2 INS Performance Ranges CEP rate has been used by the U.S. Air Force to define the three ranges of INS performance shown in Table 6.1, along with corresponding ranges of inertial sensor performance. These rough order-of-magnitude sensor performance requirements are for "cruise" applications, with acceleration levels on the order of 1 g.

6.1.3 Relation to GPS

6.1.3.1 Advantages/Disadvantages of INS The main advantages of inertial navigation over other forms of navigation are as follows:

1. It is *autonomous* and does not rely on any external aids or on visibility conditions. It can operate in tunnels or underwater as well as anywhere else.
2. It is inherently well suited for integrated navigation, guidance, and control of the host vehicle. Its IMU measures the derivatives of the variables to be controlled (e.g., position, velocity, and attitude).
3. It is immune to jamming and inherently stealthy. It neither receives nor emits detectable radiation and requires no external antenna that might be detectable by radar.

TABLE 6.1 INS and Inertial Sensor Performance Ranges

System or Sensor	Performance Units	Performance Ranges		
		High	Medium	Low
INS	CEP Rate (NMI/h)	$\leq 10^{-1}$	≈ 1	≥ 10
Gyros	deg/h	$\leq 10^{-3}$	$\approx 10^{-2}$	$\geq 10^{-1}$
Accelerometers	g[a]	$\leq 10^{-7}$	$\approx 10^{-6}$	$\geq 10^{-5}$

[a] $1\ g \approx 9.8$ m/s/s.

The disadvantages include the following:

1. Mean-squared navigation errors increase with time.
2. Cost, including:
   (a) Acquisition cost, which can be an order of magnitude (or more) higher than GPS receivers.
   (b) Operations cost, including the crew actions and time required for initializing position and attitude. Time required for initializing INS attitude by gyrocompass alignment is measured in minutes. Time-to-first-fix for GPS receivers is measured in seconds.
   (c) Maintenance cost. Electromechanical avionics systems (e.g., INS) tend to have higher failure rates and repair costs than purely electronic avionics systems (e.g., GPS).
3. Size and weight, which have been shrinking:
   (a) Earlier INS systems weighed tens to hundreds of kilograms.
   (b) Later "mesoscale" INSs for integration with GPS weighed a few kilograms.
   (c) Developing micro-electromechanical sensors are targeted for gram-size systems.

   INS weight has a multiplying effect on vehicle system design, because it requires increased structure and propulsion weight as well.
4. Power requirements, which have been shrinking along with size and weight but are still higher than those for GPS receivers.
5. Heat dissipation, which is proportional to and shrinking with power requirements.

6.1.3.2 Competition from GPS In the 1970s, U.S. commercial air carriers were required by FAA regulations to carry two INS systems on all flights over water. The cost of these two systems was on the order of 10^5 U.S. dollars at that time. The relatively high cost of INS was one of the factors leading to the development of GPS. After deployment of GPS in the 1980s, the few remaining applications for "stand-alone" (i.e., unaided) INS include submarines, which cannot receive GPS signals while submerged, and intercontinental ballistic missiles, which cannot rely on GPS availability in time of war.

6.1.3.3 Synergism with GPS GPS integration has not only made inertial navigation perform better, it has made it cost less. Sensor errors that were unacceptable for stand-alone INS operation became acceptable for integrated operation, and the manufacturing and calibration costs for removing these errors could be eliminated. Also, new low-cost manufacturing methods using micro-electromechanical systems (MEMSs) technologies could be applied to meet the less stringent sensor requirements for integrated operation.

The use of integrated GPS/INS for mapping the gravitational field near the earth's surface has also enhanced INS performance by providing more detailed and accurate gravitational models.

Inertial navigation also benefits GPS performance by carrying the navigation solution during loss of GPS signals and allowing rapid reacquisition when signals become available.

Integrated GPS/INS have found applications that neither GPS nor INS could perform alone. These include low-cost systems for precise automatic control of vehicles operating at the surface of the earth, including automatic landing systems for aircraft and autonomous control of surface mining equipment, surface grading equipment, and farm equipment.

6.2 INERTIAL SENSORS

The design of inertial sensors is limited only by human imagination and the laws of physics, and there are literally thousands of designs for gyroscopes and accelerometers. Not all of them are used for inertial navigation. Gyroscopes, for example, are used for steering and stabilizing ships, torpedoes, missiles, gunsights, cameras, and binoculars, and acceleration sensors are used for measuring gravity, sensing seismic signals, leveling, and measuring vibrations.

6.2.1 Sensor Technologies

A sampling of inertial sensor technologies used in inertial navigation is presented in Table 6.2. There are many more, but these will serve to illustrate the great diversity of technologies applied to inertial navigation. How these and other example devices function will be explained briefly. A more thorough treatment of inertial sensor designs is given in [118].

TABLE 6.2 Some Basic Inertial Sensor Technologies

Sensor	Gyroscope			Accelerometer		
Physical Effect Used[a]	Conservation of angular momentum	Coriolis effect	Sagnac effect	Gyroscopic precession	Electro-magnetic force	Strain under load
Sensor Implementation Methods	Angular displacement	Vibration	Ring laser	Angular displacement	Drag cup	Piezo-electric
	Torque rebalance	Rotation	Fiber optic	Torque rebalance	Electro-magnetic	Piezo-resistive

[a] All accelerometers use a proof mass. The physical effect is the manner in which acceleration of the proof mass is sensed.

6.2.2 Common Error Models

6.2.2.1 Sensor-Level Models Some of the more common types of sensor errors are illustrated in Fig. 6.1. These are

(a) bias, which is any nonzero sensor output when the input is zero;
(b) scale factor error, often resulting from aging or manufacturing tolerances;
(c) nonlinearity, which is present in most sensors to some degree;
(c) scale factor sign asymmetry, often from mismatched push–pull amplifiers;
(e) a dead zone, usually due to mechanical stiction or lock-in [for a ring laser gyroscope (RLG)]; and
(f) quantization error, inherent in all digitized systems.

Theoretically, one should be able to recover the input from the sensor output so long as the input/output relationship is known and invertible. Dead-zone errors and quantization errors are the only ones shown with this problem. The cumulative effects of both types (dead zone and quantization) often benefit from zero-mean input noise or dithering. Also, not all digitization methods have equal cumulative effects. Cumulative quantization errors for sensors with frequency outputs are bounded by $\pm\frac{1}{2}$LSB, but the variance of cumulative errors from independent sample-to-sample A/D conversion errors can grow linearly with time.

6.2.2.2 Cluster-Level Models For a cluster of three gyroscopes or accelerometers with nominally orthogonal input axes, the effects of individual scale factor

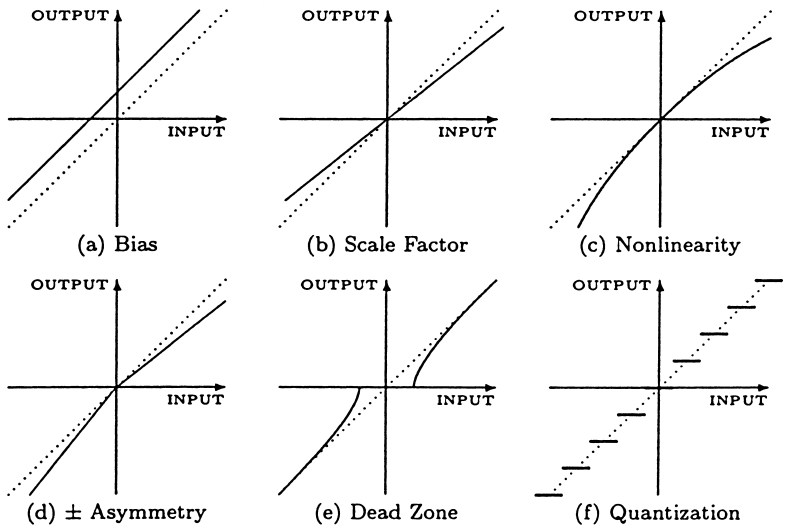


Fig. 6.1 *Common input/output error types.*

deviations and input axis misalignments from their nominal values can be modeled by the equation

$$\mathbf{z}_{\text{output}} = S_{\text{nominal}}\{\mathbf{I} + \mathbf{M}\}\mathbf{z}_{\text{input}} + \mathbf{b}_z, \tag{6.1}$$

where the components of the vector $\mathbf{b}_z$ are the three sensor output biases, the components of the $\mathbf{z}_{\text{input}}$ and $\mathbf{z}_{\text{output}}$ vectors are the sensed values (accelerations or angular rates) and output values from the sensors, respectively, S_{nominal} is the nominal sensor scale factor, and the elements m_{ij} of the "scale factor and misalignment matrix" $\mathbf{M}$ represent the individual scale factor deviations and input axis misalignments as illustrated in Fig. 6.2. The larger arrows in the figure represent the nominal input axis directions (labeled #1, #2, and #3) and the smaller arrows (labeled m_{ij}) represent the directions of scale factor deviations ($i = j$) and misalignments ($i \neq j$).

Equation 6.1 is in "error form." That is, it represents the outputs as functions of the inputs. The corresponding "compensation form" is

$$\mathbf{z}_{\text{input}} = \frac{1}{S_{\text{nominal}}}\{\mathbf{I} + \mathbf{M}\}^{-1}\{\mathbf{z}_{\text{output}} - \mathbf{b}_z\} \tag{6.2}$$

$$= \frac{1}{S_{\text{nominal}}}\{\mathbf{I} - \mathbf{M} + \mathbf{M}^2 - \mathbf{M}^3 + \cdots\}\{\mathbf{z}_{\text{output}} - \mathbf{b}_z\} \tag{6.3}$$

$$\approx \frac{1}{S_{\text{nominal}}}\{\mathbf{I} - \mathbf{M}\}\{\mathbf{z}_{\text{output}} - \mathbf{b}_z\} \tag{6.4}$$

if the sensor errors are sufficiently small (e.g., $<10^{-3}$ rad misalignments and $<10^{-3}$ parts/part scale factor deviations).

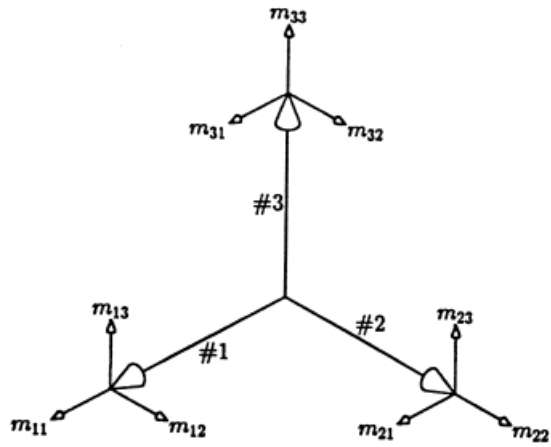


Fig. 6.2 *Directions of modeled sensor cluster errors.*

The compensation form is the one used in system implementation for compensating sensor outputs using a single constant matrix $\overline{\mathbf{M}}$ in the form

$$\mathbf{z}_{\text{input}} = \overline{\mathbf{M}}\{\mathbf{z}_{\text{output}} - \mathbf{b}_z\} \tag{6.5}$$

$$\overline{\mathbf{M}} \stackrel{\text{def}}{=} \frac{1}{S_{\text{nominal}}}\{\mathbf{I} + \mathbf{M}\}^{-1}. \tag{6.6}$$

6.2.3 Attitude Sensors

6.2.3.1 Nongyroscopic Attitude Sensors Gyroscopes are the attitude sensors used in nearly all INSs. There are other types of attitude sensors, but they are primarily used as aids to INSs with gyroscopes. These include the following:

1. Magnetic sensors, used primarily for coarse heading initialization.
2. Star trackers, used primarily for space-based or near-space applications. The U-2 spy plane, for example, used an inertial-platform-mounted star tracker to maintain INS alignment on long flights.
3. Optical ground alignment systems used on some space launch systems. Some of these systems used Porro prisms mounted on the inertial platform to maintain optical line-of-sight reference through ground-based theodolites to reference directions at the launch complex.
4. GPS receiver systems using antenna arrays and carrier phase interferometry. These have been developed for initializing artillery fire control systems, for example, but the same technology could be used for INS aiding. The systems generally have baselines in the order of several meters, which could limit their applicability to some vehicles.

6.2.3.2 Gyroscope Performance Grades Gyroscopes used in inertial navigation are called "inertial grade," which generally refers to a range of sensor performance, depending on INS performance requirements. Table 6.3 lists some

TABLE 6.3 Performance Grades for Gyroscopes

Performance Parameter	Units	Performance Grades		
		Inertial	Intermediate	Moderate
Maximum Input	deg/h deg/s	10^2–10^6 10^{-2}–10^2	10^2–10^6 10^{-2}–10^2	10^2–10^6 10^{-2}–10^2
Scale Factor	part/part	10^{-6}–10^{-4}	10^{-4}–10^{-3}	10^{-3}–10^{-2}
Bias Stability	deg/h deg/s	10^{-4}–10^{-2} 10^{-8}–10^{-6}	10^{-2}–10 10^{-6}–10^{-3}	10–10^2 10^{-3}–10^{-2}
Bias Drift	deg/$\sqrt{\text{h}}$ deg/$\sqrt{\text{s}}$	10^{-4}–10^{-3} 10^{-6}–10^{-5}	10^{-2}–10^{-1} 10^{-5}–10^{-4}	1–10 10^{-4}–10^{-3}

generally accepted performance grades used for gyroscopes, based on their intended applications but not necessarily including integrated GPS/INS applications.

These are only rough order-of-magnitude ranges for the different error characteristics. Sensor requirements are largely determined by the application. For example, gyroscopes for gimbaled systems generally require smaller input ranges than those for strapdown applications.

6.2.3.3 Sensor Types Gyroscope designers have used many different approaches to a common sensing problem, as evidenced by the following samples. There are many more, and probably more yet to be discovered.

Momentum Wheels Momentum wheel gyroscopes use a spinning mass patterned after the familiar child's toy gyroscope. If the spinning momentum wheel is mounted inside gimbals to isolate it from rotations of the body on which it is mounted, then its spin axis tends to remain in an inertially fixed direction and the gimbal angles provide a readout of the total angular displacement of that direction from body-fixed axis directions. If, instead, its spin axis is torqued to follow the body axes, then the required torque components provide a measure of the body angular rates normal to the wheel spin axis. In either case, this type of gyroscope can potentially measure two components (orthogonal to the momentum wheel axle) of angular displacement or rate, in which case it is called a *two-axis gyroscope*. Because the drift characteristics of momentum wheel gyroscopes are so strongly affected by bearing torques, these gyroscopes are often designed with innovative bearing technologies (e.g., gas, magnetic, or electrostatic bearings). If the mechanical coupling between the momentum wheel and its axle is flexible with just the right mechanical spring rate—depending on the rotation rate and angular momentum of the wheel—the effective torsional spring rate on the momentum wheel can be canceled. This type of dynamical "tuning" isolates the gyroscope from bearing torques and generally improves gyroscope performance.

Coriolis Effect The Coriolis effect is named after Gustave Gaspard de Coriolis (1792–1843), who described the apparent acceleration acting on a body moving with constant velocity in a rotating coordinate frame [26]. It can be modeled in terms of the vector cross-product (defined in Section B.2.10) as

$$\mathbf{a}_{\text{Coriolis}} = -\Omega \otimes \mathbf{v} \tag{6.7}$$

$$= -\begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix} \otimes \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \tag{6.8}$$

$$= \begin{bmatrix} -\Omega_2 v_3 + \Omega_3 v_2 \\ -\Omega_3 v_1 + \Omega_1 v_3 \\ -\Omega_1 v_2 + \Omega_2 v_1 \end{bmatrix}, \tag{6.9}$$

where $\mathbf{v}$ is the vector velocity of the body in the rotating coordinate frame, Ω is the inertial *rotation rate vector* of the coordinate frame (i.e., with direction parallel to the rotation axis and magnitude equal to the rotation rate), and $\mathbf{a}_{\text{Coriolis}}$ is the apparent acceleration acting on the body in the rotating coordinate frame.

Rotating Coriolis Effect Gyroscopes The gyroscopic effect in momentum wheel gyroscopes can be explained in terms of the Coriolis effect, but there are also gyroscopes that measure the Coriolis acceleration on the rotating wheel. An example of such a two-axis gyroscope is illustrated in Fig. 6.3. For sensing rotation, it uses an accelerometer mounted off-axis on the rotating member, with its acceleration input axis parallel to the rotation axis of the base. When the entire assembly is rotated about any axis normal to its own rotation axis, the accelerometer mounted on the rotating base senses a sinusoidal Coriolis acceleration.

The position and velocity of the rotated accelerometer with respect to inertial coordinates will be

$$\mathbf{x}(t) = \rho \begin{bmatrix} \cos(\Omega_{\text{drive}} t) \\ \sin(\Omega_{\text{drive}} t) \\ 0 \end{bmatrix}, \tag{6.10}$$

$$\mathbf{v}(t) = \frac{d}{dt}\mathbf{x}(t) \tag{6.11}$$

$$= \rho\Omega_{\text{drive}} \begin{bmatrix} -\sin(\Omega_{\text{drive}} t) \\ \cos(\Omega_{\text{drive}} t) \\ 0 \end{bmatrix}, \tag{6.12}$$

where Ω_{drive} is the drive rotation rate and ρ is the offset distance of the accelerometer from the base rotation axis.

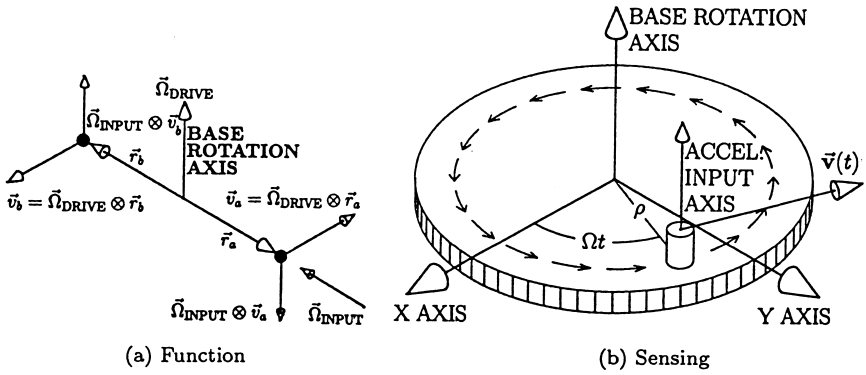


Fig. 6.3 *Rotating Coriolis effect gyroscope.*

The input axis of the accelerometer is parallel to the rotation axis of the base, so it is insensitive to rotations about the base rotation axis (z-axis). However, if this apparatus is rotated with components $\Omega_{x,\text{input}}$ and $\Omega_{y,\text{input}}$ orthogonal to the z-axis, then the Coriolis acceleration of the accelerometer will be the vector cross-product

$$\mathbf{a}_{\text{Coriolis}}(t) = -\begin{bmatrix} \Omega_{x,\text{input}} \\ \Omega_{y,\text{input}} \\ 0 \end{bmatrix} \otimes \mathbf{v}(t) \tag{6.13}$$

$$= -\rho\Omega_{\text{drive}} \begin{bmatrix} \Omega_{x,\text{input}} \\ \Omega_{y,\text{input}} \\ 0 \end{bmatrix} \otimes \begin{bmatrix} -\sin(\Omega_{\text{drive}}t) \\ \cos(\Omega_{\text{drive}}t) \\ 0 \end{bmatrix} \tag{6.14}$$

$$= \rho\Omega_{\text{drive}} \begin{bmatrix} & 0 & \\ & 0 & \\ -\Omega_{x,\text{input}} & \cos(\Omega_{\text{drive}}t) + \Omega_{y,\text{input}} & \sin(\Omega_{\text{drive}}t) \end{bmatrix} \tag{6.15}$$

The rotating z-axis accelerometer will then sense the z-component of Coriolis acceleration,

$$a_{z,\text{input}}(t) = \rho\Omega_{\text{drive}}[\Omega_{x,\text{input}} \cos(\Omega_{\text{drive}}t) - \Omega_{y,\text{input}} \sin(\Omega_{\text{drive}}t)], \tag{6.16}$$

which can be demodulated to recover the phase components $\rho\Omega_{\text{drive}}\Omega_x$ (in phase) and $\rho\Omega_{\text{drive}}\Omega_{y,\text{input}}$ (in quadrature), each of which is proportional to a component of the input rotation rate. Demodulation of the accelerometer output removes the DC bias, so this implementation is insensitive to accelerometer bias errors.

Rotating Multisensor Another accelerometer can be mounted on the moving base of the rotating Coriolis effect gyroscope, but with its input axis tangential to its direction of motion. Its ouputs can be demodulated in similar fashion to implement a two-axis accelerometer with zero effective bias error.

Torsion Resonator Gyroscope This is a micro-electromechanical systems (MEMS) device first developed at C. S. Draper Laboratories in the 1980s, then jointly with Rockwell, Boeing, and Honeywell. It is similar in some respects to the rotating Coriolis effect gyroscope, except that the wheel rotation is sinusoidal at the torsional resonance frequency and input rotations are sensed as the wheel tilting at that frequency. This gyroscope uses a momentum wheel coupled to a torsion spring and driven at resonance to create sinusoidal angular momentum in the wheel. If the device is turned about any axis in the plane of the wheel, the Coriolis effect will introduce sinusoidal tilting about the orthogonal axis in the plane of the wheel, as

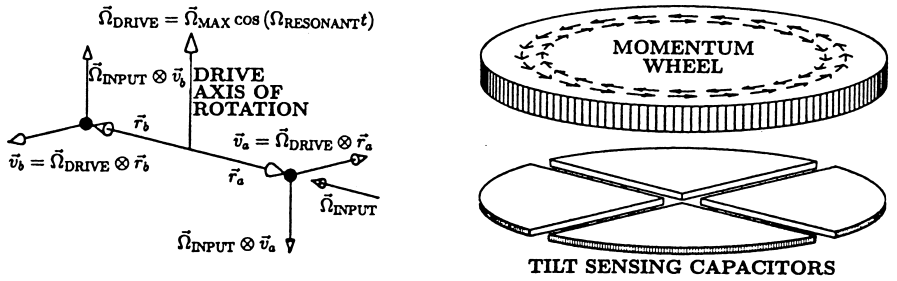


Fig. 6.4 *Torsion resonator gyroscope.*

illustrated in Fig. 6.4*a*. This sinusoidal tilting is sensed by four capacitor sensors in close proximity to the wheel underside, as illustrated in Fig. 6.4*b*.

Other Vibrating Coriolis Effect Gyroscopes These include vibrating wires, vibrating beams, tuning forks (effectively, paired vibrating beams), and "wine glasses" (using the vibrating modes thereof), in which a combination of turning rate and Coriolis effect couples one mode of vibration into another. The vibrating member is driven in one mode, the input is rotation rate, and the output is the sensed vibration in the undriven mode. All vibrating Coriolis effect gyroscopes measure a component of angular rate orthogonal to the vibrational velocity. The example shown in Fig. 6.5 is a tuning fork driven in a vibration mode with its tines coming together and apart in unison (Fig. 6.5*a*). Its sensitive axis is parallel to the tines. Rotation about this axis is orthogonal to the direction of tine velocity, and the resulting Coriolis acceleration will be in the direction of $\boldsymbol{\omega} \otimes \mathbf{v}$, which excites the output vibration mode shown in Fig. 6.5*b*. This "twisting" mode will create a torque couple through the handle, and some designs use a double-ended fork to transfer this mode to a second set of output tines.

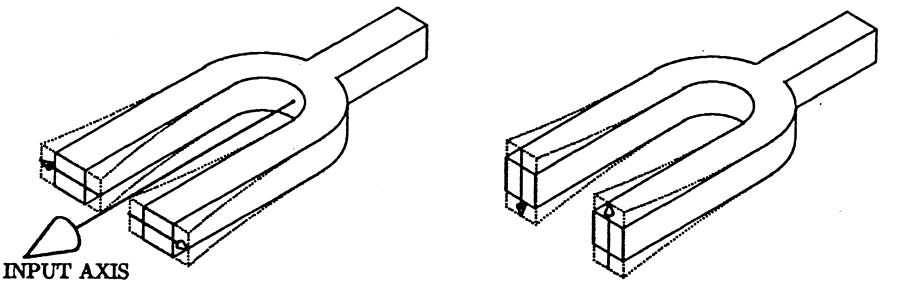


Fig. 6.5 *Vibration modes of tuning fork gyroscope.*

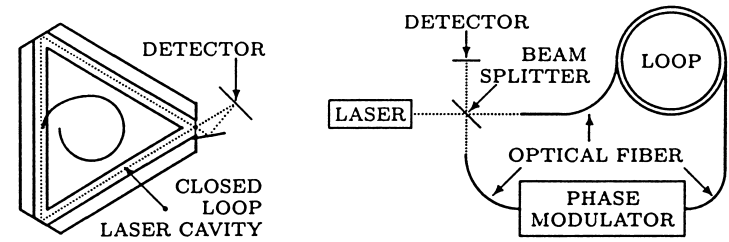


Fig. 6.6 *Basic optical components of laser gyroscopes.*

In some ways, performance of Coriolis effect sensors tends to get better as the device sizes shrink, because sensitivity scales with velocity, which scales with resonant frequency, which increases as the device sizes shrink.

Laser Gyroscopes Two fundamental laser gyroscope types are the *ring laser gyroscope* (RLG) and the *fiber optic gyroscope* (FOG), both of which use the Sagnac effect[2] on counterrotating laser beams and a interferometric phase detector to measure their relative phase changes. The basic optical components and operating principles of both types are illustrated in Fig. 6.6.

Ring Laser Gyroscope The principal optical components of a RLG are illustrated in Fig. 6.6*a*, which shows a triangular lasing cavity with mirrors at the three vertices. Lasing occurs in both directions, creating clockwise and counterclockwise laser beams. The lasing cavity length is controlled by servoing one mirror, and one mirror allows enough leakage so that the two counterrotating beams can form an interference pattern on a photodetector array. Inertial rotation of this device in the plane of the page will change the effective cavity lengths of the clockwise and counterclockwise beams (the Sagnac effect), causing an effective relative frequency change at the detector. The output is an interference fringe frequency proportional to the input rotation rate, making the ring laser gyroscope a *rate-integrating gyroscope*. The sensor scale factor is proportional to the area enclosed by the laser paths.

Fiber-Optic Gyroscope The principal optical components of a FOG are illustrated in Fig. 6.6*b*, which shows a common external laser source generating both clockwise and counterclockwise light waves traveling around a loop of optical fiber. Inertial rotation of this device in the plane of the page will change the effective path lengths of the clockwise and counterclockwise beams in the loop of fiber (Sagnac effect), causing an effective relative phase change at the detector. The interference phase between the clockwise and counterclockwise beams is measured at the output detector, but in this case the output phase difference is proportional to rotation rate. In effect, the FOG is a rate gyroscope, whereas the RLG is a rate-integrating gyroscope. Phase modulation in the optical path (plus some signal processing) can

[2] Essentially, the finite velocity of light.

be used to improve the effective output phase resolution. The FOG scale factor is proportional to the product of the enclosed loop area and the number of turns.

Temperature changes and accelerations can alter the strain distribution in the optical fiber, which could cause output errors. Minimizing these effects is a major concern in the art of FOG design.

6.2.3.4 Gyroscope Error Models Error models for gyroscopes are used primarily for two purposes:

1. In the design of gyroscopes, for predicting performance characteristics as functions of design parameters. The models used for this purpose are usually based on physical principles relating error characteristics to dimensions and physical properties of the gyroscope and its component parts, including electronics.
2. Calibration and compensation of output errors. Calibration is the process of observing the gyroscope outputs with known inputs and using that data to fit the unknown parameters of mathematical models for the outputs (including errors) as functions of the known inputs. This relationship is inverted for error compensation (i.e., determining the true inputs as functions of the corrupted outputs). The models used for this purpose generally come from two sources:
   (a) Models derived for design analysis and reused for calibration and compensation. However, it is often the case that there is some "model overlap" among such models, in that there can be more independent causes than observable effects. In such cases, all coefficients of the independent models will not be observable from test data, and one must resort to choosing a subset of the underdetermined models.
   (b) Mathematical models derived strictly from empirical data fitting. These models are subject to the same sorts of observability conditions as the models from design analysis, and care must be taken in the design of the calibration procedure to assure that all model coefficients can be determined sufficiently well to meet error compensation requirements. The covariance equations of Kalman filtering are very useful for this sort of calibration analysis (see Chapters 7 and 8).

   Integrated GPS/INS applications effectively perform sensor error model calibration "on the fly" using sensor error models, sensor data redundancy, and a Kalman filter.

In this chapter, we will be primarily concerned with error compensation and with the mathematical forms of the error models. Error modeling for GPS/INS integration is described in Chapter 8.

Bias Causes of output bias in gyroscopes include bearing torques (for momentum wheel types), drive excitation feedthrough, and output electronics offsets [46, Ch. 3]. There are generally three types of bias errors to worry about:

1. fixed bias, which only needs to be calibrated once;
2. bias stability from turn-on to turn-on, which may result from thermal cycling of the gyroscope and its electronics, among other causes; and
3. bias drift after turn-on, which is usually modeled as a random walk (defined in Section 7.5.1.2) and specified in such units as deg/h/$\sqrt{\text{h}}$ or other equivalent units suitable for characterizing random walks.

After each turn-on, the general-purpose gyroscope bias error model will have the form of a drift rate (rotation rate) about the gyroscope input axis:

$$\omega_{\text{output}} = \omega_{\text{input}} + \delta\omega_{\text{bias}} \tag{6.17}$$

$$\delta\omega_{\text{bias}} = \delta\omega_{\text{constant}} + \delta\omega_{\text{turn-on}} + \delta\omega_{\text{randomwalk}}, \tag{6.18}$$

where $\delta\omega_{\text{constant}}$ is a known constant, $\delta\omega_{\text{turn-on}}$ is an unknown constant, and $\delta\omega_{\text{randomwalk}}$ is modeled as a random-walk process:

$$\frac{d}{dt}\delta\omega_{\text{randomwalk}} = w(t), \tag{6.19}$$

where $w(t)$ is a zero-mean white-noise process with known variance.

Bias variability from turn-on is called *bias stability*, and bias variability after turn-on is called *bias drift*.

Scale Factor The gyroscope scale factor is usually specified in compensation form as

$$\omega_{\text{input}} = C_{\text{scalefactor}}\omega_{\text{output}}, \tag{6.20}$$

where $C_{\text{scalefactor}}$ can have components that are constant, variable from turn-on to turn-on, and drifting after turn-on:

$$C_{\text{scalefactor}} = C_{\text{constantscalefactor}} + C_{\text{scalefactorstability}} + C_{\text{scalefactordrift}}, \tag{6.21}$$

similar to the gyroscope bias model.

Input Axis Misalignments The input axis for a gyroscope defines the component of rotation rate that it senses. Its input axis is a direction fixed with respect to the gyroscope mount. It is usually not possible to manufacture the gyroscope such that its input axis is in the desired direction to the precision required, so some compensation is necessary. The first gimbaled systems used mechanical shimming to align the gyroscope input axes in orthogonal directions, because the navigation computers did not have the capacity to do it in software as it is done nowadays.

There are two orthogonal components of input axis misalignment. For small-angle misalignments, these components are approximately orthogonal to the desired

input axis direction and they make the misaligned gyroscope sensitive to the rotation rate components in these orthogonal directions. The small-angle approximation for the output error $\delta\omega_i$ will then be of the form

$$\delta\omega_i \approx \omega_j\alpha_{ij} + \omega_k\alpha_{ik}, \tag{6.22}$$

where ω_i = component of rotation rate the gyroscope is intended to read
ω_j = rotation rate component orthogonal to ω_i
ω_k = rotation rate component orthogonal to ω_i and ω_j
α_{ij} = misalignment angular component (in radians) toward to ω_j
α_{ik} = misalignment angular component (in radians) toward to ω_j

Combined Three-Gyroscope Compensation Cluster-level compensation for bias, scale factor, and input axis alignments for three gyroscopes with nominally orthogonal input axes is implemented in matrix form as shown in Eq. 6.5 (p. 188), which will have the form

$$\begin{bmatrix} \omega_{i,\text{input}} \\ \omega_{j,\text{input}} \\ \omega_{k,\text{input}} \end{bmatrix} = \overline{\mathbf{M}}_{\text{gyro}} \left\{ \begin{bmatrix} \omega_{i,\text{output}} \\ \omega_{j,\text{output}} \\ \omega_{k,\text{output}} \end{bmatrix} - \boldsymbol{\omega}_{\text{bias}} \right\}, \tag{6.23}$$

where $\boldsymbol{\omega}_{\text{bias}}$ is the bias compensation (a vector) and $\overline{\mathbf{M}}_{\text{gyro}}$ (a 3×3 matrix) is the combined scale factor and misalignment compensation. The diagonal elements of $\overline{\mathbf{M}}_{\text{gyro}}$ compensate for the three scale factor errors, and the off-diagonal elements of $\overline{\mathbf{M}}_{\text{gyro}}$ compensate for the six input axis misalignments.

Input/Output Nonlinearity The nonlinearities of sensors are typically modeled in terms of a MacLauren series expansion, with the first two terms being bias and scale factor. The next order term will be the squared term, and the expansion will have the forms

$$\omega_{\text{output}} = C_0 + C_1\omega_{\text{input}} + C_2\omega_{\text{input}}^2 + \cdots, \tag{6.24}$$

$$\omega_{\text{input}} = \mathscr{C}_0 + \mathscr{C}_1\omega_{\text{output}} + \mathscr{C}_2\omega_{\text{output}}^2 + \cdots, \tag{6.25}$$

depending on whether the input is modeled as a function of the output or vice versa. The output compensation form of Eq. 6.25 is more useful in implementation, however.

Acceleration Sensitivity Momentum wheel gyroscopes exhibit precession rates caused by relative displacement of the center of mass from the center of the mass-supporting force, as illustrated in Fig. 6.7. Gyroscope designers strive to make the relative displacement as small as possible, but, for illustrative purposes, we have used an extreme case of mass offset in Fig. 6.7. The paired couple of equal and

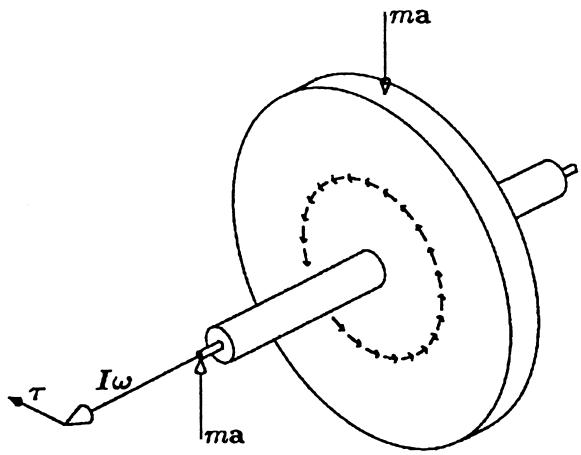


Fig. 6.7 *Precession due to mass unbalance.*

opposite acceleration and inertial forces ma, separated by a distance d, creates a torque of magnitude $\tau = dma$. The analog of Newton's second law for linear motion, $\mathbf{F} = m\mathbf{a}$, for angular motion is $\tau = \mathbf{I}\dot{\boldsymbol{\omega}}$, where $\mathbf{I}$ is the moment of inertia (the angular analog of mass) of the rotor assembly and $\boldsymbol{\omega}$ is its angular velocity. For the example shown, this torque is at right angles to the rotor angular velocity $\boldsymbol{\omega}$ and causes the angular velocity vector to precess.

Gyroscopes without momentum wheels may also exhibit acceleration sensitivity, although it may not have the same functional form. In some cases, it is caused by mechanical strain of the sensor structure.

6.2.3.5 g-squared Sensitivity (Anisoelasticity) Gyroscopes may also exhibit output errors proportional to the square of acceleration components. The causal mechanism in early momentum wheel designs could be traced to anisoelasticity (mismatched compliances of the gyroscope support under acceleration loading).

6.2.4 Acceleration Sensors

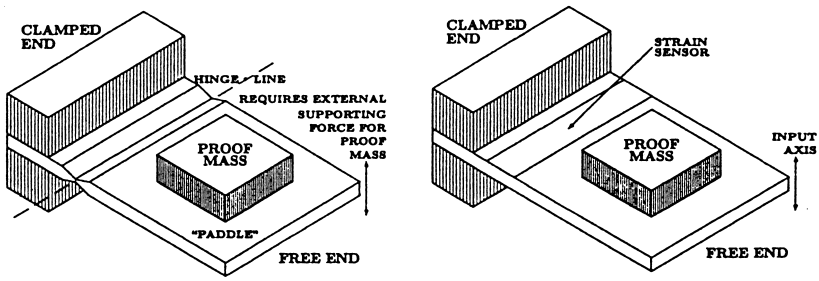
All acceleration sensors used in inertial navigation are called "accelerometers." Acceleration sensors used for other purposes include bubble levels (for measuring the direction of acceleration), gravimeters (for measuring gravity fields), and seismometers (used in seismic prospecting and for sensing earthquakes and underground explosions).

6.2.4.1 Accelerometer Types Accelerometers used for inertial navigation depend on Newton's second law (in the form $F = ma$) to measure acceleration (a) by measuring force (F), with the scaling constant (m) called "proof mass." These common origins still allow for a wide range of sensor designs, however.

Gyroscopic Accelerometers Gyroscopic accelerometers measure acceleration through its influence on the precession rate of a mass-unbalanced gyroscope, as illustrated in Fig. 6.7. If the gyroscope is allowed to precess, then the net precession angle change (integral of precession rate) will be proportional to velocity change (integral of acceleration). If the gyroscope is torqued to prevent precession, then the required torque will be proportional to the disturbing acceleration. A pulse-integrating gyroscopic accelerometer (PIGA) uses repeatable torque pulses, so that pulse rate is proportional to acceleration and each pulse is equivalent to a constant change in velocity (the integral of acceleration). Gyroscopic accelerometers are also sensitive to rotation rates, so they are used almost exclusively in gimbaled systems.

Pendulous Accelerometers Pendulous accelerometers use a hinge to support the proof mass in two dimensions, as illustrated in Fig. 6.8*a*, so that it is free to move only in the input axis direction, normal to the "paddle" surface. This design requires an external supporting force to keep the proof mass from moving in that direction, and the force required to do it will be proportional to the acceleration that would otherwise be disturbing the proof mass.

Force Rebalance Accelerometers *Electromagnetic accelerometers* (EMAs) are pendulous accelerometers using electromagnetic force to keep the paddle from moving. A common design uses a voice coil attached to the paddle and driven in an arrangement similar to the speaker cone drive in permanent magnet speakers, with the magnetic flux through the coils provided by permanent magnets. The coil current is controlled through a feedback servo loop including a paddle position sensor such as a capacitance pickoff. The current in this feedback loop through the voice coil will be proportional to the disturbing acceleration. For *pulse-integrating accelerometers*, the feedback current is supplied in discrete pulses with very repeatable shapes, so that each pulse is proportional to a fixed change in velocity. An up/down counter keeps track of the net pulse count between samples of the digitized accelerometer output.



(a) Pendulus Accelerometer (b) Beam Accelerometer

Fig. 6.8 *Single-axis accelerometers.*

Integrating Accelerometers The pulse-feedback electromagnetic accelerometer is an integrating accelerometer, in that each pulse output corresponds to a constant increment in velocity. The "drag cup" accelerometer illustrated in Fig. 6.9 is another type of integrating accelerometer. It uses the same physical principles as the drag cup speedometer used for half a century in automobiles, consisting of a rotating bar magnet and conducting envelope (the drag cup) mounted on a common rotation shaft but coupled only through the eddy current drag induced on the drag cup by the relative rotation of the magnet. (The design includes a magnetic circuit return ring outside the drag cup, not shown in this illustration.) The torque on the drag cup is proportional to the relative rotation rate of the magnet. The drag cup accelerometer has a deliberate mass unbalance on the drag cup, such that accelerations of the drag cup orthogonal to the mass unbalance will induce a torque on the drag cup proportional to acceleration. The bar magnet is driven by an electric motor, the speed of which is servoed to keep the drag cup from rotating. The rotation rate of the motor is then proportional to acceleration, and each revolution of the motor corresponds to a fixed velocity change. These devices can be daisychained to perform successive integrals. Two of them coupled in tandem, with the drag cup of one used to drive the magnet of the other, would theoretically perform double integration, with each motor drive revolution equivalent to a fixed increment of position.

Strain-Sensing Accelerometers The cantilever beam accelerometer design illustrated in Fig. 6.8*b* senses the strain at the root of the beam resulting from support of the proof mass under acceleration load. The surface strain near the root of the beam will be proportional to the applied acceleration. This type of accelerometer can be manufactured relatively inexpensively using MEMS technologies, with an ion-implanted piezoresistor pattern to measure surface strain.

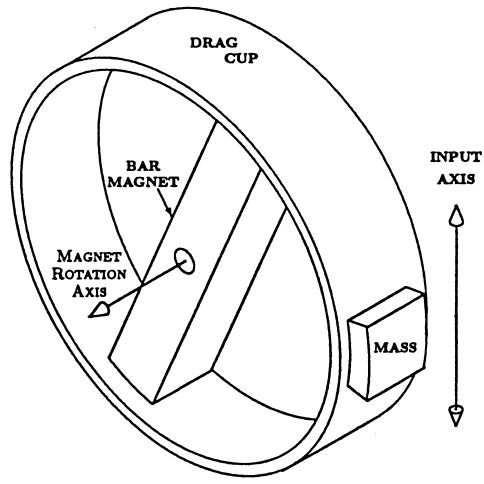


Fig. 6.9 *Drag cup accelerometer.*

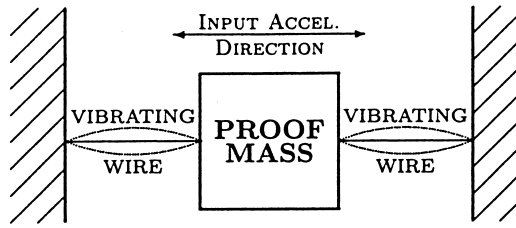


Fig. 6.10 *Single-axis vibrating wire accelerometer.*

Vibrating-Wire Accelerometers The resonant frequencies of vibrating wires (or strings) depend upon the length, density, and elastic constant of the wire and on the square of the tension in the wire. The motions of the wires must be sensed (e.g., by capacitance pickoffs) and forced (e.g., electrostatically or electromagnetically) to be kept in resonance. The wires can then be used as digitizing force sensors, as illustrated in Fig. 6.10. The configuration shown is for a single-axis accelerometer, but the concept can be expanded to a three-axis accelerometer by attaching pairs of opposing wires in three orthogonal directions.

In the "push–pull" configuration shown, any lateral acceleration of the proof mass will cause one wire frequency to increase and the other to decrease. Furthermore, if the preload tensions in the wires are servoed to keep the sum of their frequencies constant, then the difference frequency

$$\omega_{\text{left}} - \omega_{\text{right}} \propto \frac{ma}{\omega_{\text{left}} + \omega_{\text{right}}} \tag{6.26}$$

$$\propto a. \tag{6.27}$$

Both the difference frequency $\omega_{\text{left}} - \omega_{\text{right}}$ and the sum frequency $\omega_{\text{left}} + \omega_{\text{right}}$ (used for preload tension control) can be obtained by mixing and filtering the two wire position signals from the resonance forcing servo loop. Each cycle of the difference frequency then corresponds to a constant delta velocity, making the sensor inherently digital.

6.2.4.2 Error Models

Linear and Bias Models Many of the error models used for calibration and compensation of accelerometers have the same functional forms as those for gyroscopes, although the causal mechanisms may be quite different. The zero-order (bias) and first-order (scale factor and input axis misalignments), in particular, are functionally identical, as modeled in Eq. 6.5. For accelerometers, this model has the form

$$\begin{bmatrix} a_{i,\text{input}} \\ a_{j,\text{input}} \\ a_{k,\text{input}} \end{bmatrix} = \overline{\mathbf{M}}_{\text{acc}} \left\{ \begin{bmatrix} a_{i,\text{output}} \\ a_{j,\text{output}} \\ a_{k,\text{output}} \end{bmatrix} - \mathbf{a}_{\text{bias}} \right\}, \tag{6.28}$$

where $\mathbf{a}_{\text{bias}}$ is the bias compensation (a vector) and $\overline{\mathbf{M}}_{\text{acc}}$ (a 3×3 matrix) is the combined scale factor and misalignment compensation. Just as for the case with gyroscopes, the diagonal elements of $\overline{\mathbf{M}}_{\text{acc}}$ compensate for the three scale factor errors, and the off-diagonal elements of $\mathbf{M}_{\text{acc}}$ compensate for the six input axis misalignments.

Higher Order Models Nonlinearities of accelerometers are modeled the same as those of gyroscopes: as a MacLauren series expansion. The first two terms of the series model bias and scale factor, which we have just considered. The next order term is the so-called "*g-squared*" accelerometer error sensitivity, which is not uncommon in inertial grade accelerometers:

$$a_{\text{input}} = \underbrace{\mathscr{C}_0}_{\text{bias}} + \underbrace{\mathscr{C}_1 a_{\text{output}}}_{\text{scalefactor}} + \underbrace{\mathscr{C}_2 a_{\text{output}}^2}_{g\text{-squared}} + \cdots. \tag{6.29}$$

Some accelerometers also exhibit second-order output errors called *cross-axis coupling errors*, which are proportional to the product of the input acceleration component and an acceleration component orthogonal to the input axis:

$$\delta a_{i,\text{cross-axis}} \propto a_i a_j, \tag{6.30}$$

where a_i is the input acceleration along the input axis and a_j is a component orthogonal to the input axis.

Instability Models Accelerometers also exhibit the same sorts of parameter instabilities observed in gyroscopes (i.e., turn-on and drift), the composite model for which is given in Eq. 6.21.

Centrifugal Acceleration Effects Accelerometers have input axes defining the component(s) of acceleration that they measure. There is a not-uncommon superstition that these axes must intersect at a point to avoid some unspecified error source. That is generally not the case, but there can be some differential sensitivity to centrifugal accelerations due to high rotation rates and relative displacements between accelerometers. The effect is rather weak, but not always negligible. It is modeled by the equation

$$a_{i,\text{centrifugal}} = \omega^2 r_i, \tag{6.31}$$

where ω^2 is the rotation rate and r_i is the displacement component along the input axis from the axis of rotation to the effective center of the accelerometer. Even manned vehicles can rotate at $\omega \approx 3$ rad/s, which creates centrifugal accelerations of about $1\,g$ at $r_i = 1$ m and $0.001\,g$ at 1 mm. The problem is less significant, if not insignificant, for MEMS-scale accelerometers that can be mounted within millimeters of one another.

Center of Percussion Because ω can be measured, sensed centrifugal accelerations can be compensated, if necessary. This requires designating some reference point within the instrument cluster and measuring the radial distances and directions to the accelerometers from that reference point. The point within the accelerometer required for this calculation is sometimes called its "center of percussion." It is effectively the point such that rotations about all axes through the point produce no sensible centrifugal accelerations, and that point can be located by testing the accelerometer at differential reference locations on a rate table.

Angular Acceleration Sensitivity Pendulus accelerometers are sensitive to angular acceleration about their hinge lines, with errors equal to $\dot{\omega}\Delta_{\text{hinge}}$, where $\dot{\omega}$ is the angular acceleration in radians per second squared and Δ_{hinge} is the displacement of the accelerometer proof mass (at its center of mass) from the hinge line. This effect can reach the $1\,g$ level for $\Delta_{\text{hinge}} \approx 1\,\text{cm}$ and $\dot{\omega} \approx 10^3\ \text{rad/s}^2$, but these extreme conditions are usually not persistent enough to matter in most applications.

6.3 NAVIGATION COORDINATES

Navigation is concerned with determining where you are relative to your destination, and coordinate systems are used for specifying both locations. Definitions of the principal coordinate systems used in GPS/INS integration and navigation are given in Appendix C. These include coordinate systems used for representing the trajectories of GPS satellites and user vehicles in the near-earth environment and for representing the attitudes of host vehicles relative to locally level coordinates, including the following:

1. Inertial coordinates:
   (a) Earth-centered inertial (ECI), with origin at the center of mass of the earth and principal axes in the directions of the vernal equinox (defined in Section C.2.1) and the rotation axis of the earth.
   (b) Satellite orbital coordinates, as illustrated in Fig. C.4 and used in GPS ephemerides.
2. Earth-fixed coordinates:
   (a) Earth-centered, earth-fixed (ECEF), with origin at the center of mass of the earth and principal axes in the directions of the prime meridian (defined in Section C.3.5) at the equator and the rotation axis of the earth.
   (b) Geodetic coordinates, based on an ellipsoid model for the shape of the earth. Longitude in geodetic coordinates is the same as in ECEF coordinates, and geodetic latitude as defined as the angle between the equatorial plane and the normal to the reference ellipsoid surface. Geodetic latitude can differ from geocentric latitude by as much as 12 arc minutes, equivalent to about 20 km of northing distance.
   (c) Local tangent plane (LTP) coordinates, also called "locally level coordinates," essentially representing the earth as being locally flat. These

coordinates are particularly useful from a human factors standpoint for representing the attitude of the host vehicle and for representing local directions. They include

(i) east–north–up (ENU), shown in Fig. C.7;

(ii) north–east–down (NED), which can be simpler to relate to vehicle coordinates; and

(iii) alpha wander, rotated from ENU coordinates through an angle α about the local vertical (see Fig. C.8).

3. Vehicle-fixed coordinates:
   (a) Roll–pitch–yaw (RPY) (axes shown in Fig. C.9).

Transformations between these different coordinate systems are important for representing vehicle attitudes, for resolving inertial sensor outputs into inertial navigation coordinates, and for GPS/INS integration. Methods used for representing and implementing coordinate transformations are also presented in Appendix C, Section C.4.

6.4 SYSTEM IMPLEMENTATIONS

6.4.1 Simplified Examples

The following examples are intended as an introduction to INS technology for nonspecialists in INS technology. They illustrate some of the key properties of inertial sensors and inertial system implementations.

6.4.1.1 Inertial Navigation in One Dimension If we all lived in one-dimensional "Line Land," then there could be no rotation and no need for gyroscopes. In that case, an INS would need only one accelerometer and navigation computer, and its implementation would be as illustrated in Fig. 6.11, where the variable x denotes position in one dimension.

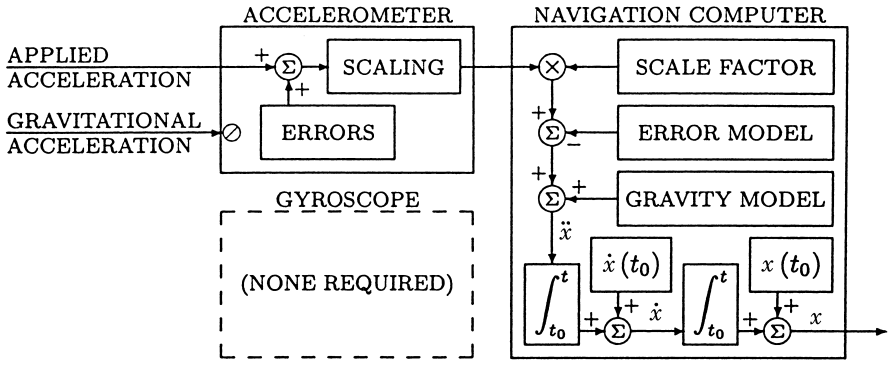


Fig. 6.11 *INS functional implementation for a one-dimensional world.*

This implementation for one dimension has many features common to implementations for three dimensions:

- *Accelerometers cannot measure gravitational acceleration*. An accelerometer effectively measures the force acting on its proof mass to make it follow its mounting base, which includes only nongravitational accelerations applied through physical forces acting on the INS through its host vehicle. Satellites, which are effectively in free fall, experience no sensible accelerations.
- Accelerometers have *scale factors*, which are the ratios of input acceleration units to output signal magnitude units (e.g., meters per second squared per volt). The signal must be rescaled in the navigation computer by multiplying by this scale factor.
- Accelerometers have *output errors*, including
  1. unknown constant *offsets*, also called *biases*;
  2. unknown constant *scale factor errors*;
  3. unknown *sensor input axis misalignments*;
  4. unknown *nonconstant variations* in bias and scale factor; and
  5. unknown zero-mean additive *noise* on the sensor outputs, including quantization noise and electronic noise. The noise itself is not predictable, but its statistical properties are used in Kalman filtering to estimate drifting scale factor and biases.
- *Gravitational accelerations must be modeled* and calculated in the navigational computer, then added to the sensed acceleration (after error and scale compensation) to obtain the net acceleration $\ddot{x}$ of the INS.
- The navigation computer must integrate acceleration to obtain velocity. *This is a definite integral and it requires an initial value*, $\dot{x}(t_0)$. That is, the INS implementation in the navigation computer must start with a known initial velocity.
- The navigation computer must also integrate velocity ($\dot{x}$) to obtain position (x). *This is also a definite integral and it also requires an initial value*, $x(t_0)$. The INS implementation in the navigation computer must start with a known initial location, too.

6.4.1.2 Inertial Navigation in Three Dimensions

Inertial navigation in three dimensions requires more sensors and more signal processing than in one dimension, and it also introduces more possibilities for implementation. The earliest successful INSs used gimbals to isolate the sensors from rotations of the host vehicle.

Gimbaled INS A *stable platform, inertial platform*, or "*stable table*" is a mechanically rigid unit isolated from the rotations of the host vehicle by a set of three or (preferably) four gimbals, as illustrated in Figs. 6.12*a*,*b*. Each gimbal is effectively a ring with orthogonal inside and outside pivot axes. These are nested

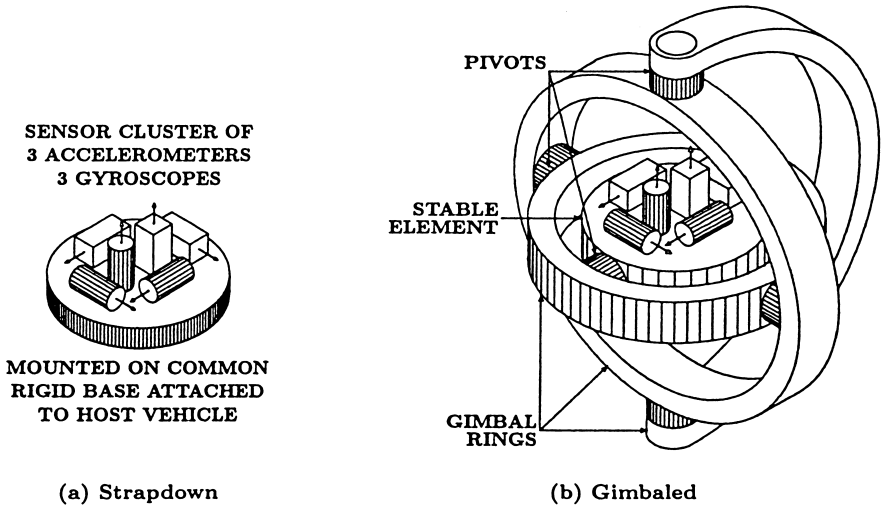


Fig. 6.12 *Inertial measurement units.*

inside one another, with the innermost gimbal attached through its inner pivot bearing to the stable platform and the outermost gimbal (half cut away in this illustration) attached to the vehicle. Gimbal pivot bearings include angle sensors and electromagnetic torquing coils for sensing and controlling the gimbal pivot angles.

A *sensor cluster* of three gyroscopes and three accelerometers is rigidly mounted to the stable platform, as illustrated in Fig. 6.12*a*. In the illustration, gyroscopes and accelerometers are represented by square or circular cylinders, with their input axes parallel to the cylinder axes. The gyroscopes on the stable platform are used to sense any rotation of the platform, and their outputs are used in servo feedback loops with gimbal pivot torque actuators to control the gimbals such that the platform remains stable (i.e., does not rotate).

A fourth gimbal is required for vehicles such as missiles or high-performance aircraft with full freedom of rotation about all three axes. Otherwise, rotations of the host vehicle can align two of the three gimbal axes parallel to one another in a condition called *gimbal lock*. In gimbal lock with only three gimbals, the remaining single "unlocked" gimbal can only isolate the platform from rotations about a second rotation axis. Rotations about the third axis of the "missing" gimbal will slew the platform unless a fourth gimbal axis is provided for this contingency.

Floated-Ball Systems. The function of gimbals is to isolate the stable platform from the rotations of the host vehicle. *Floated-ball systems* achieve the same effect by floating the platform (now shaped like a ball) inside a fluid-filled sphere using fluid thrusters attached to the stable ball to control its attitude and keep it centered in the fluid cavity. This approach requires that the density of the fluid make the ball neutrally buoyant, and some provisions are needed for getting power into the ball and getting heat and signals out.

Advantages and Disadvantages of Gimbaled Systems Gimbals are very sophisticated electromechanical assemblies that are expensive to manufacture. As a consequence, gimbaled systems tend to be more expensive than strapdown systems. However, the isolation of the inertial platform from rotations of the host vehicle can be exploited to eliminate many sensor error sources and achieve very high system accuracy. This is especially important for applications in which GPS aiding is not available, such as for submarine navigation.

6.4.1.3 Strapdown INS In strapdown systems, the gyroscopes and accelerometers are hard mounted ("strapped down") to a common base, as in Fig. 6.12*a*. The common sensor mounting base is no longer inertially stabilized as in Fig. 6.12*b*, but it may be attached to the vehicle frame with shock isolators designed to limit rotational vibration between the vehicle frame and the instrument base.

Strapdown system gyroscopes are not used to keep the accelerometer input axes stabilized, but they are used to maintain a coordinate transformation from the accelerometer input axes to *virtually stabilized directions*, in the form of *navigation coordinates*, as illustrated in Fig. 6.13. The navigation coordinates can be the same sorts of local tangent plane coordinates used by inertial platforms.

Inertial sensors for strapdown systems experience much higher rotation rates than their gimbaled counterparts. Rotation introduces error mechanisms that render some sensor types (e.g., gyroscopic accelerometers) unacceptable for strapdown implementation and require redesign or attitude rate-dependent error compensation for others. This is shown by the signal flow arrows shown in Fig. 6.13 between the accelerometer and gyroscope error compensation boxes. Acceleration-dependent error compensation for gyroscopes had been required for gimbaled systems.

6.4.2 Initialization and Alignment

6.4.2.1 Navigation Initialization INS initialization is the process of determining initial values for system position, velocity, and attitude in navigation coordinates. INS position initialization ordinarily relies on external sources such

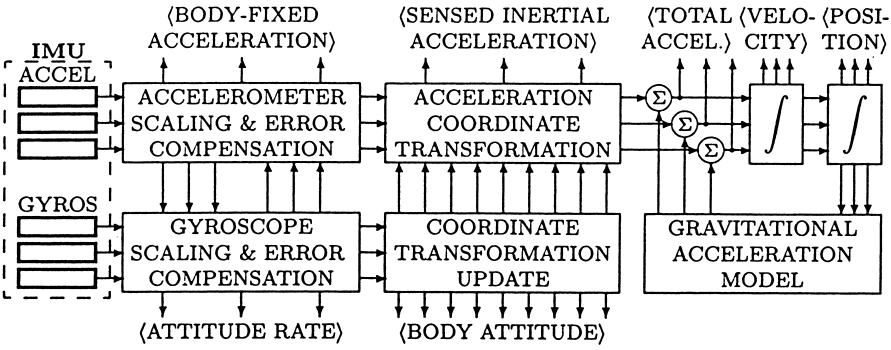


Fig. 6.13 *Outputs (in angular brackets) of simple strapdown INS.*

as GPS or manual entry by crew members. INS velocity initialization can be accomplished by starting when it is zero (i.e., the host vehicle is not moving) or (for vehicles carried in or on other vehicles) by reference to the carrier velocity. (See alignment method 3 below.) INS attitude initialization is called *alignment*.

6.4.2.2 Sensor Alignment

INS alignment is the process of aligning the stable platform axes parallel to navigation coordinates (for gimbaled systems) or that of determining the initial values of the coordinate transformation from sensor coordinates to navigation coordinates (for strapdown systems).

Alignment Methods Four basic methods for INS alignment are as follows:

1. *Optical alignment*, using either of the following:
   (a) Optical line-of sight reference to a ground-based direction (e.g., using a ground-based theodolite and a mirror on the platform). Some space boosters have used this type of optical alignment, which is much faster and more accurate than gyrocompass alignment. Because it requires a stable platform for mounting the mirror, it is only applicable to gimbaled systems.
   (b) An onboard star tracker, used primarily for alignment of gimbaled or strapdown systems in space.
2. *Gyrocompass alignment* of stationary vehicles, using the sensed direction of acceleration to determine the local vertical and the sensed direction of rotation to determine north. Latitude can be determined by the angle between the earth rotation vector and the horizontal, but longitude must be determined by other means and entered manually or electronically. This method is inexpensive, but the most time consuming (several minutes, typically).
3. *Transfer alignment* in a moving host vehicle, using velocity matching with an aligned and operating INS. This method is typically several times faster than gyrocompass alignment, but it requires another INS on the host vehicle and it may require special maneuvering of the host vehicle to attain observability of the alignment variables. It is commonly used for in-air INS alignment for missiles launched from aircraft and for on-deck INS alignment for aircraft launched from carriers. Alignment of carrier-launched aircraft may also use the direction of the velocity impulse imparted by the steam catapult.
4. *GPS-aided alignment*, using position matching with GPS to estimate the alignment variables. It is an integral part of integrated GPS/INS implementations. It does not require the host vehicle to remain stationary during alignment, but there will be some period of time after turn-on (a few minutes, typically) before system navigation errors settle to acceptable levels.

Gyrocompass alignment is the only one of these that requires no external aiding. Gyrocompass alignment is not necessary for integrated GPS/INS, although many INSs may already be configured for it.

INS Gyrocompass Alignment Accuracy A rough rule-of-thumb for gyrocompass alignment accuracy is

$$\sigma^2_{\text{gyrocompass}} > \sigma^2_{\text{acc}} + \frac{\sigma^2_{\text{gyro}}}{15^2 \cos^2(\phi_{\text{geodetic}})}, \tag{6.32}$$

where

$\sigma_{\text{gyrocompass}}$ is the minimum achievable RMS alignment error in radians,
σ_{acc} is the RMS accelerometer accuracy in g's,
σ_{gyro} is the RMS gyroscope accuracy in degrees per hour,
15 degrees per hour is the rotation rate of the earth, and
ϕ_{geodetic} is the latitude at which gyrocompassing is performed.

Alignment accuracy is also a function of the time allotted for it, and the time required to achieve a specified accuracy is generally a function of sensor error magnitudes (including noise) and the degree to which the vehicle remains stationary.

Gimbaled INS Gyrocompass Alignment Gyrocompass alignment for gimbaled systems is a process for aligning the inertial platform axes with the navigation coordinates using only the sensor outputs while the host vehicle is essentially stationary. For systems using ENU navigation coordinates, for example, the platform can be tilted until two of its accelerometer inputs are zero, at which time both input axes will be horizontal. In this locally leveled orientation, the sensed rotation axis will be in the north–up plane, and the platform can be slewed about the vertical axis to null the input of one of its horizontal gyroscopes, at which time that gyroscope input axis will point east–west. That is the basic concept used for gyrocompass alignment, but practical implementation requires filtering[3] to reduce the effects of sensor noise and unpredictable zero-mean vehicle disturbances due to loading activities and/or wind gusts.

Strapdown INS Gyrocompass Alignment Gyrocompass alignment for strapdown systems is a process for "virtual alignment" by determining the sensor cluster attitude with respect to navigation coordinates using only the sensor outputs while the system is essentially stationary.

If the sensor cluster could be firmly affixed to the earth and there were no sensor errors, then the sensed acceleration vector $\mathbf{a}_{\text{output}}$ in sensor coordinates would be in the direction of the local vertical, the sensed rotation vector $\boldsymbol{\omega}_{\text{output}}$ would be in the

[3] The vehicle dynamic model used for gyrocompass alignment filtering can be "tuned" to include the major resonance modes of the vehicle suspension.

direction of the earth rotation axis, and the unit column vectors

$$\mathbf{1}_U = \frac{\mathbf{a}_{\text{output}}}{|\mathbf{a}_{\text{output}}|}, \tag{6.33}$$

$$\mathbf{1}_N = \frac{\boldsymbol{\omega}_{\text{output}} - (\mathbf{1}_U^T \boldsymbol{\omega}_{\text{output}})\mathbf{1}_U}{|\boldsymbol{\omega}_{\text{output}} - (\mathbf{1}_U^T \boldsymbol{\omega}_{\text{output}})\mathbf{1}_U|}, \tag{6.34}$$

$$\mathbf{1}_E = \mathbf{1}_N \otimes \mathbf{1}_U \tag{6.35}$$

would define the initial value of the coordinate transformation matrix from sensor-fixed coordinates to ENU coordinates:

$$\mathbf{C}_{\text{sensor}\rightarrow\text{ENU}} = \left[\mathbf{1}_E | \mathbf{1}_N | \mathbf{1}_U\right]^{\text{T}}. \tag{6.36}$$

In practice, the sensor cluster is usually mounted in a vehicle that is not moving over the surface of the earth but may be buffeted by wind gusts and/or disturbed by fueling and payload handling. Gyrocompassing then requires some amount of filtering to reduce the effects of vehicle buffeting and sensor noise. The gyrocompass filtering period is typically on the order of several minutes for a medium-accuracy INS but may continue for hours or days for high-accuracy systems.

6.4.3 Earth Models

Inertial navigation and satellite navigation require models for the shape, gravity, and rotation of the earth.

6.4.3.1 Earth Rotation Rate Until the discovery of hyperfine transition atomic clocks in the mid-twentieth century, the rotation of the earth had been our most accurate clock. It has given us the time units of days, hours, minutes, and seconds we use to manage our lives, and we continue to use its rotation as our primary time reference, adding or subtracting leap seconds from our reference based on atomic clocks to keep them synchronized to the rotation of the earth.

Variations in Earthrate We have known for several centuries that the directions of the equinoxes are changing, but it was not until atomic clocks were developed that we could measure the variations in the magnitude of earth rotation rate attributed to dynamic forces within the earth and to interactions with the gravitational fields of the sun and moon. Magnitudes, time scales, and causal mechanisms are still being sorted out, but these are some current estimates:

1. On the scale of millions to billions of years, the rotation of the earth is slowing down due to tidal effects that essentially convert rotational kinetic energy to heat and transfer some of the angular momentum and energy of the earth to the moon in its orbit about the earth. The rate of slowdown is, itself, slowing

down, but it is currently estimated to be on the order of 0.2 parts per billion per year.

2. On the scale of millennia, redistribution of water during ice ages changes the moments of inertia of the earth, with rotation rate varying inversely as the polar moment of inertia. There is also a steady change in direction of the rotation axis, due principally to the solar gravity gradient acting on the earth's equatorial bulge, which causes the direction of the rotation axis of the earth to move at about 20 arc seconds per year with a precession period of about 26,000 years. This also changes the direction of the equinoxes, where the equatorial plane intersects the ecliptic (earth–sun plane). In terms of equivalent torque required, a change of 20 arc seconds in direction is comparable to a change of about 100 ppm in magnitude.
3. On the scale of years to millennia, there are changes in the internal flows in the hydrosphere and lithosphere that alter both the direction and magnitude of the apparent rotation axis of the earth and the pole axis within the earth. Apparent pole shifts on the earth surface observed in the last few decades are on the order of tens of meters.
4. On the scale of months to years, the global changes in weather patterns known as *El Niño* are associated with a slowdown of the earth rotation rate on the order of several parts per billion. Even annual effects such as snow accumulation in winter or leaves growing on trees in the spring and falling in autumn have calculable consequences. These variations can also excite polhode motions of the earth rotation axis, which can be on the order of several meters at the surface, with a period somewhat longer than a year. Seasonal shifts in the jet streams may contribute some to this motion, too.
5. On the scale of days to weeks, shifts in the north and south jet streams and associated weather patterns are suspected of altering the earth rotation rate on the order of parts per billion.

However, for navigation missions on time scales on the order of hours, these variations can be ignored.

WGS 84 Earthrate Model The value of earthrate in the World Geodetic System 1984 (WGS 84) earth model used by GPS is $7{,}292{,}115{,}167 \times 10^{-14}$ rad/sec, or about 15.04109 deg/h. This is its *sidereal* rotation rate with respect to distant stars. Its mean rotation rate with respect to the nearest star (our sun), as viewed from the rotating earth, is 15 deg/h.

6.4.3.2 GPS Gravity Models Accurate gravity modeling is important for maintaining ephemerides for GPS satellites, and models developed for GPS have been a boon to inertial navigation as well. However, spatial resolution of the earth gravitational field required for GPS operation may be a bit coarse compared to that for precision inertial navigation, because the GPS satellites are not near the surface and the mass concentration anomalies that create surface gravity anomalies. GPS orbits have very little sensitivity to surface-level undulations of the gravitational field

on the order of 100 km or less, but these can be important for high-precision inertial systems.

6.4.3.3 INS Gravity Models

Because an INS operates in a world with gravitational accelerations it is unable to sense and unable to ignore, it must use a reasonably faithful model of gravity.

Gravity models for the earth include centrifugal acceleration due to rotation of the earth as well as true gravitational accelerations due to the mass distribution of the earth, but they do not generally include oscillatory effects such as tidal variations.

Gravitational Potential Gravitational potential is defined to be zero at a point infinitely distant from all massive bodies and to decrease toward massive bodies such as the earth. That is, a point at infinity is the reference point for gravitational potential.

In effect, the gravitational potential at a point in or near the earth is defined by the potential energy lost by a unit of mass falling to that point from infinite altitude. In falling from infinity, kinetic energy is converted to kinetic energy, $mv_{\text{escape}}^2/2$, where v_{escape} is the *escape velocity*. Escape velocity at the surface of the earth is about 11 km/s.

Gravitational Acceleration Gravitational acceleration is the negative gradient of gravitational potential. Potential is a scalar function, and its gradient is a vector. Because gravitational potential increases with altitude, its gradient points upward and the negative gradient points downward.

Equipotential Surfaces An equipotential surface is a surface of constant gravitational potential. If the ocean and atmosphere were not moving, then the surface of the ocean at static equilibrium would be an equipotential surface. *Mean sea level* is a theoretical equipotential surface obtained by time averaging the dynamic effects.

Ellipsoid Models for Earth Geodesy is the process of determining the shape of the earth, often using ellipsoids as approximations of an equipotential surface (e.g., mean sea level), as illustrated in Fig. 6.14. The one shown in the figure is an ellipsoid of revolution, but there are many reference ellipsoids based on different survey data. Some are global approximations and some are local approximations. The global approximations deviate from a spherical surface by about ± 10 km, and locations on the earth referenced to different ellipsoidal approximations can differ from one another by 10^2–10^3 m.

Geodetic latitude on a reference ellipsoid is measured in terms of the angle between the equator and the normal to the ellipsoid surface (the local vertical reference), as illustrated in Fig. 6.14.

Orthometric height is measured normal to a reference ellipsoid surface.

WGS 84 Ellipsoid The World Geodetic System (WGS) is an international standard for navigation coordinates. WGS 84 is a reference earth model released

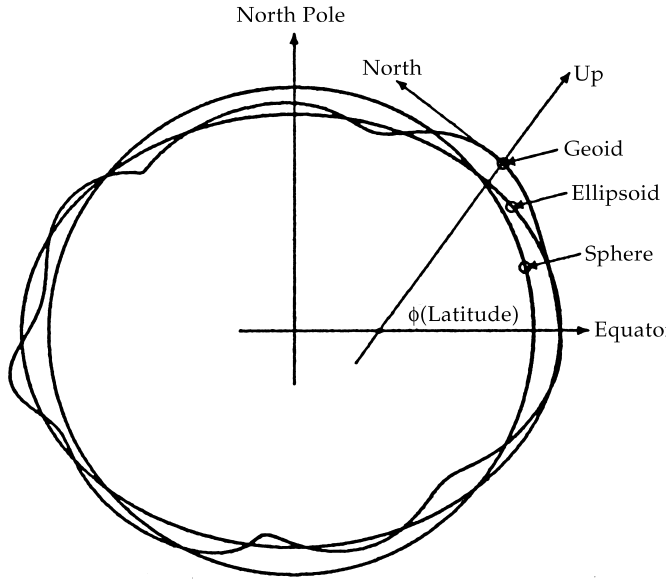


Fig. 6.14 *Equipotential surface models for earth.*

in 1984. It approximates mean sea level by an ellipsoid of revolution with its rotation axis coincident with the rotation axis of the earth, its center at the center of mass of the earth, and its prime meridian through Greenwich. Its semimajor axis (equatorial radius) is defined to be 6,378,137 m, and its semiminor axis (polar radius) is defined to be 6,356,752.3142 m.

Geoid Models Geoids are approximations of mean sea-level orthometric height with respect to a reference ellipsoid. Geoids are defined by additional higher order shapes, such as spherical harmonics of height deviations from an ellipsoid, as illustrated in Fig. 6.14. There are many geoid models based on different data, but the more recent, most accurate models depend heavily on GPS data. Geoid heights deviate from reference ellipsoids by tens of meters, typically.

The WGS 84 geoid heights vary about ± 100 m from the reference ellipsoid. As a rule, oceans tend to have lower geoid heights and continents tend to have higher geoid heights. Coarse 20-m contour intervals are plotted versus longitude and latitude in Fig. 6.15, with geoid regions above the ellipsoid shaded gray.

6.4.3.4 Longitude and Latitude Rates The second integral of acceleration in locally level coordinates should result in the estimated vehicle position. This integral is somewhat less than straightforward when longitude and latitude are the preferred horizontal location variables.

The rate of change of vehicle altitude equals its vertical velocity, which is the first integral of net (i.e., including gravity) vertical acceleration. The rates of change of

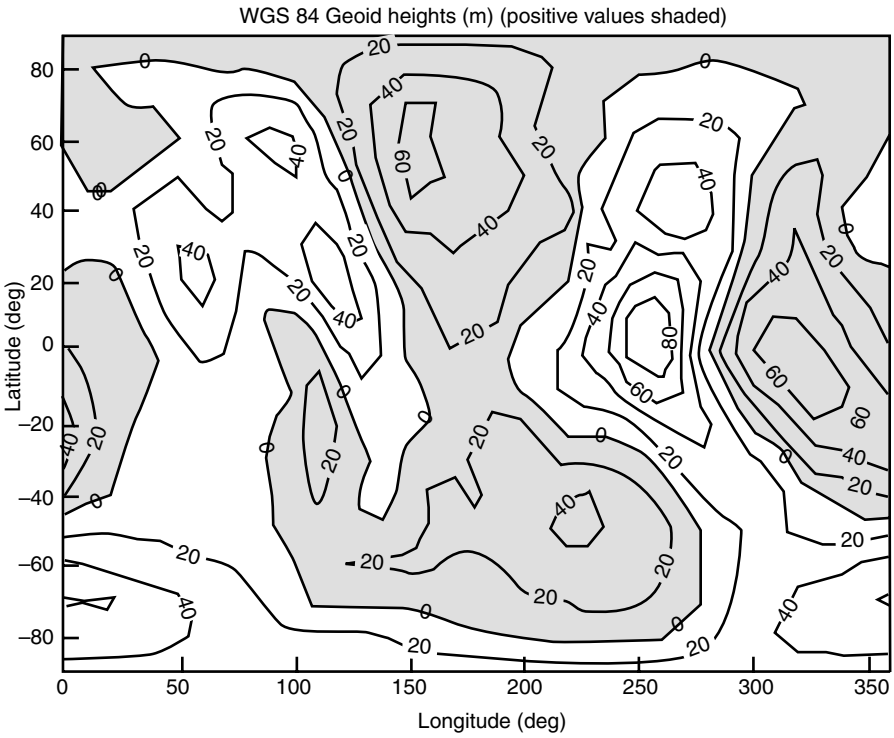


Fig. 6.15 *WGS 84 geoid heights.*

vehicle longitude and latitude depend on the horizontal components of vehicle velocity, but in a less direct manner. The relationship between longitude and latitude rates and east and north velocities is further complicated by the oblate shape of the earth.

The rates at which these angular coordinates change as the vehicle moves tangent to the surface will depend upon the radius of curvature of the reference surface model, which is an ellipsoid of revolution for the WGS 84 model. Radius of curvature can depend on the direction of travel, and for an ellipsoidal model there is one radius of curvature for north–south motion and another radius of curvature for east–west motion.

Meridional Radius of Curvature The radius of curvature for north–south motion is called the "meridional" radius of curvature, because north–south travel is along a meridian (i.e., line of constant longitude). For an ellipsoid of revolution (the WGS 84 model), all meridians have the same shape, which is that of the ellipse that was rotated to produce the ellipsoidal surface model. The tangent circle with the same radius of curvature as the ellipse is called the *"osculating" circle* (*osculating* means "kissing"). As illustrated in Fig. 6.16 for an oblate earth model, the radius of the meridional osculating circle is smallest where the geocentric radius is largest (at the

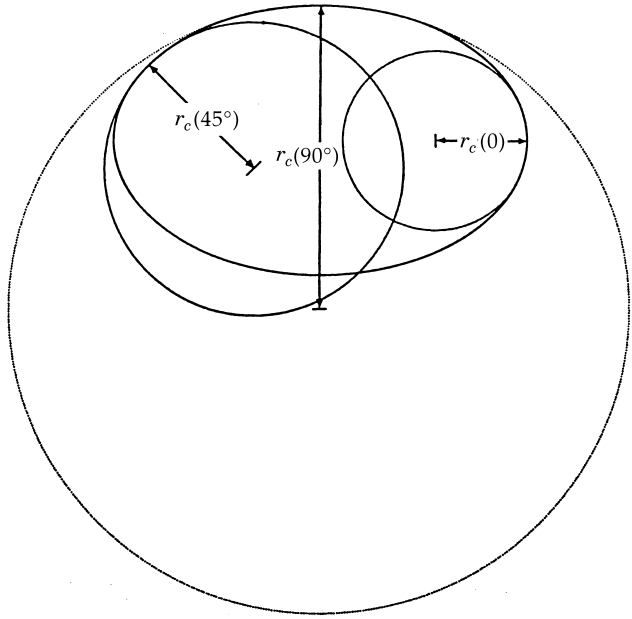


Fig. 6.16 *Ellipse and osculating circles.*

equator), and the radius of the osculating circle is largest where the geocentric radius is smallest (at the poles). The osculating circle lies inside or on the ellipsoid at the equator and outside or on the ellipsoid at the poles and passes through the ellipsoid surface for latitudes in between.

The formula for meridional radius of curvature as a function of geodetic latitude (ϕ_{geodetic}) is

$$r_M = \frac{b^2}{a[1 - e^2 \sin^2(\phi_{\text{geodetic}})]^{3/2}} \tag{6.37}$$

$$= \frac{a(1 - e^2)}{[1 - e^2 \sin^2(\phi_{\text{geodetic}})]^{3/2}}, \tag{6.38}$$

where a is the semimajor axis of the ellipse, b is the semiminor axis, and $e^2 = (a^2 - b^2)/a^2$ is the eccentricity squared.

Geodetic Latitude Rate The rate of change of geodetic latitude as a function of north velocity is then

$$\frac{d\phi_{\text{geodetic}}}{dt} = \frac{v_N}{r_M + h}, \tag{6.39}$$

and geodetic latitude can be maintained as the integral

$$\phi_{\text{geodetic}}(t_{\text{now}}) = \phi_{\text{geodetic}}(t_{\text{start}}) + \int_{t_{\text{start}}}^{t_{\text{now}}} \frac{v_N(t)\, dt}{a(1-e^2)/\{1-e^2 \sin^2[\phi_{\text{geodetic}}(t)]\}^{3/2} + h(t)]}, \tag{6.40}$$

where $h(t)$ is height above (+) or below (−) the ellipsoid surface and $\phi_{\text{geodetic}}(t)$ will be in radians if $v_N(t)$ is in meters per second and $r_M(t)$ and $h(t)$ are in meters.

Transverse Radius of Curvature The radius of curvature of the reference ellipsoid surface in the east–west direction (i.e., orthogonal to the direction in which the meridional radius of curvature is measured) is called the *transverse radius of curvature.* It is the radius of the osculating circle in the local east–up plane, as illustrated in Fig. 6.17, where the arrows at the point of tangency of the transverse osculating circle are in the local ENU coordinate directions. As this figure illustrates, on an oblate earth, the plane of a transverse osculating circle does not pass through the center of the earth, except when the point of osculation is at the equator. (All osculating circles at the poles are in meridional planes.) Also, unlike meridional osculating circles, transverse osculating circles generally lie outside the ellipsoidal surface, except at the point of tangency and at the equator, where the transverse osculating circle *is* the equator.

The formula for the transverse radius of curvature on an ellipsoid of revolution is

$$r_T = \frac{a}{\sqrt{1-e^2 \sin^2(\phi_{\text{geodetic}})}}, \tag{6.41}$$

where a is the semimajor axis of the generating ellipse and e is its eccentricity.

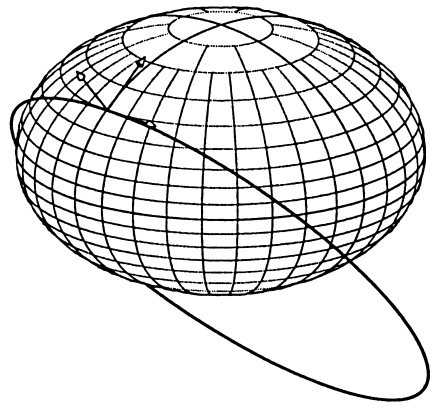


Fig. 6.17 *Transverse osculating circle.*

Longitude Rate The rate of change of longitude as a function of east velocity is then

$$\frac{d\theta}{dt} = \frac{v_E}{\cos(\phi_{\text{geodetic}})(r_T + h)} \tag{6.42}$$

and longitude can be maintained by the integral

$$\theta(t_{\text{now}}) = \theta(t_{\text{start}}) + \int_{t_{\text{start}}}^{t_{\text{now}}} \frac{v_E(t)\, dt}{\cos[\phi_{\text{geodetic}}(t)]\left(a/\sqrt{1 - e^2 \sin^2(\phi_{\text{geodetic}}(t))} + h(t)\right)}, \tag{6.42}$$

where $h(t)$ is height above (+) or below (−) the ellipsoid surface and θ will be in radians if $v_E(t)$ is in meters per second and $r_T(t)$ and $h(t)$ are in meters. Note that this formula has a singularity at the poles, where $\cos(\phi_{\text{geodetic}}) = 0$, a consequence of using latitude and longitude as location variables.

WGS 84 Reference Surface Curvatures The apparent variations in meridional radius of curvature in Fig. 6.16 are rather large because the ellipse used in generating Fig. 6.16 has an eccentricity of about 0.75. The WGS 84 ellipse has an eccentricity of about 0.08, with geocentric, meridional, and transverse radius of curvature as plotted in Fig. 6.18 versus geodetic latitude. For the WGS 84 model,

- mean geocentric radius is about 6371 km, from which it varies by −14.3 km (−0.22%) to +7.1 km (+0.11%);
- mean meridional radius of curvature is about 6357 km, from which it varies by −21.3 km (−0.33%) to 42.8 km (+0.67%); and
- mean transverse radius of curvature is about 6385 km, from which it varies by −7.1 km (−0.11%) to +14.3 km (+0.22%).

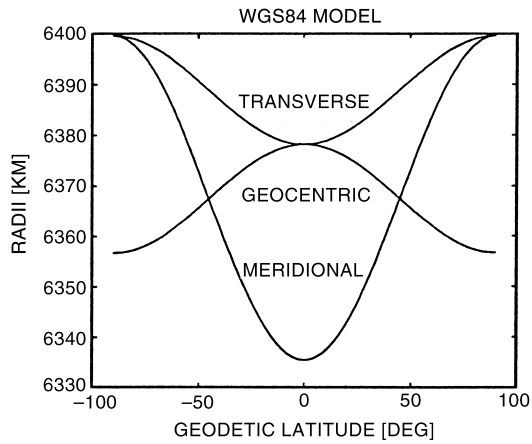


Fig. 6.18 *Radii of WGS 84 Reference Ellipsoid.*

Because these vary by several parts per thousand, one must take radius of curvature into account when integrating horizontal velocity components to obtain longitude and latitude.

6.4.4 Gimbaled System Implementations

6.4.4.1 Gimbal Design Issues The primary objective in the design of a gimbal system is to isolate the inertial platform from rotations of the host vehicle. Other issues addressed in design include

1. balancing the distribution of supported mass to avoid acceleration-dependent gimbal torques,
2. routing signals and electrical power between the inertial platform and the host vehicle,
3. controlling the temperature of the inertial platform and maintaining effective heat transfer from the platform in all operational gimbal attitudes,
4. minimizing mechanical deformation of the gimbals under acceleration loading, and
5. avoiding mechanical resonances of the gimbals that may cause vibration of the platform.

6.4.4.2 Gimbal Sensors and Actuators Gyroscopes on the stable element of a gimbaled INS are used to sense any inertial angular disturbances of the stable element, and torquers in the gimbal pivots are used to apply feedback corrections to null out the sensed disturbances. The feedback loops between the gyroscopes and gimbal torquers are not constant, however. Angle resolvers in the gimbal pivots are required for distributing the feedback torque signals among the different gimbal pivot axes, depending on the current gimbal angles. These gimbal pivot angle resolvers are also used to determine the attitude of the host vehicle with respect to platform coordinates.

6.4.4.3 Coordinate Rotation Corrections Platform angular rates for maintaining locally level alignment in local geodesic coordinates are

$$\omega_E = -\frac{v_N}{r_M + h}, \tag{6.44}$$

$$\omega_N = \omega_\otimes \cos(\phi_{\text{geodetic}}) + \frac{v_E}{r_T + h}, \tag{6.45}$$

$$\omega_{\text{up}} = \omega_\otimes \sin(\phi_{\text{geodetic)}}, \tag{6.46}$$

$$r_M = \frac{a(1 - e^2)}{[1 - e^2 \sin^2(\phi_{\text{geodetic}})]^{3/2}}, \tag{6.47}$$

$$r_T = \frac{a}{[1 - e^2 \sin^2(\phi_{\text{geodetic}})]^{3/2}}, \tag{6.48}$$

where all rotation rates are in units of radians per second and

v_N = north velocity (m/s)
v_E = east velocity (m/s)
$\omega_\otimes$ = rotation rate of the earth
ϕ_{geodetic} = current geodetic latitude
r_T = transverse (east–west) radius of curvature (m) of reference ellipsoid surface at current latitude (see Fig. 6.17)
r_M = meridional (north–south) radius of curvature (m) of reference ellipsoid surface at current latitude (see Fig. 6.16)
h = current altitude (m) above reference ellipsoid
a = polar radius (semimajor axis of reference ellipsoid)
e = reference ellipsoid model eccentricity, $= \sqrt{a^2 - b^2}/a$, where $b =$ polar radius

6.4.4.4 Coriolis Correction

Platform coordinates referenced to north and east are not inertial coordinates, and the resulting Coriolis effect must be compensated as an east acceleration correction,

$$\delta a_E \approx (72.92115 \times 10^{-6}) v_N \sin(\phi_{\text{geodetic}}), \tag{6.49}$$

where ϕ_{geodetic} is geodetic latitude and 72.92115×10^{-6} is the earth rotation rate in radians per second.

6.4.5 Strapdown System Implementations

6.4.5.1 Acceleration Integration

Components of acceleration measured by the accelerometers of a strapdown system are in body-fixed coordinates. These need to be transformed to navigation coordinates for integration into velocity and position components. If $\mathbf{C}_{\text{nav}}^{\text{body}}$ is the coordinate transformation matrix from body-fixed coordinates to navigation coordinates, and $\mathbf{a}_{\text{body}}$ is the vector of sensed (and error-compensated) accelerations in body-fixed coordinates, then

$$\mathbf{a}_{\text{nav}} = \mathbf{C}_{\text{nav}}^{\text{body}} \mathbf{a}_{\text{body}} \tag{6.50}$$

is the host vehicle acceleration in navigation coordinates, which would be equivalent to platform coordinates in a gimbaled system. The rest of the translational navigation implementation is similar to that of a gimbaled system. That is, this acceleration vector in navigation coordinates (along with gravitational accelerations) is integrated twice to maintain estimates of velocity and position in navigation coordinates.

6.4.5.2 Coordinate Transforms

There are many ways to represent and implement coordinate transformations. The more useful ones are coordinate transformation matrices, rotation vectors, and quaternions, with quaternions being the

preferred representation for strapdown coordinate transformations. All these methods are explained in Appendix C, and we explain here how these methods are applied to maintain the transformations between sensor-fixed coordinates and the navigation coordinates used for integrating the sensed accelerations.

6.4.5.3 Attitude Rate Integration To resolve sensed accelerations into navigation coordinates, strapdown systems need to keep track of where the navigation coordinates are pointed relative to body-fixed sensor coordinates. This can be done by rotating the row vectors of $\mathbf{C}_{\text{nav}}^{\text{body}}$ using the quaternion rotation formula of Eq. C.254 and the sensed and error-compensated angular rates in body-fixed coordinates.

The three row vectors of $\mathbf{C}_{\text{nav}}^{\text{body}}$ are the direction cosine vectors of the respective navigation coordinate axes in body-fixed coordinates.

The sampled rotation rate vectors $\boldsymbol{\omega}_k$, composed of the error-compensated gyroscope outputs, define a series of incremental rotation vectors $\boldsymbol{\rho}_k = \boldsymbol{\omega}_k \, \Delta t$ in body coordinates, where Δt is the sampling interval.

However, to an observer standing on a rotating earth, the sun appears to be rotating about the earth in the direction opposite earth rotation. The same is true of navigation coordinates as viewed in body-fixed coordinates. When the sensed rotation rate is $\boldsymbol{\omega}$, the directions of the navigation coordinates in body-fixed coordinates appear to be rotating in the direction of $-\boldsymbol{\omega}$. Therefore, the row vectors of $\mathbf{C}_{\text{nav}}^{\text{body}}$ need to be rotated through the incremental rotations $-\boldsymbol{\rho} = -\boldsymbol{\omega}_k \, \Delta t$ in body-fixed coordinates.

The implementation equations for maintaining the coordinate transformation between sensor coordinates and navigation coordinates are given in Eqs. C.221–C.225 (for implementation as direction cosine matrices) or Eqs. C.252–C.254 (for implementation as quaternions).

6.4.5.4 Troublesome Vibration Modes The stable elements of well-designed gimbaled systems are effectively isolated from the rotational components of internal vibration modes of the host vehicle. The gimbals do not isolate the stable element from translational accelerations due to vibration, but integration of the accelerations to velocities and positions reduces their influence to insignificant (or at least acceptable) levels. Strapdown acceleration integration occurs after multiplication of sensed acceleration and estimated direction cosines, both of which can contain vibration frequency components of the form $\sin(\omega_{\text{vib}} t)$ and $\cos(\omega_{\text{vib}} t)$, where ω_{vib} is the vibration frequency. The resulting products can then contain terms of the form $\sin^2(\omega_{\text{vib}} t)$ or $\cos^2(\omega_{\text{vib}} t)$, integration of which can cause cumulative errors in velocity and position.

Vibration Isolation of Strapdown Systems The design of shock and vibration isolators for strapdown systems in high-vibration environments can be critical to blocking vibration frequencies near or above the sensor sampling frequency and to avoiding certain detrimental vibration modes of the sensor cluster. The known "bad modes" of vibration include *sculling motion* and *coning motion*, which are

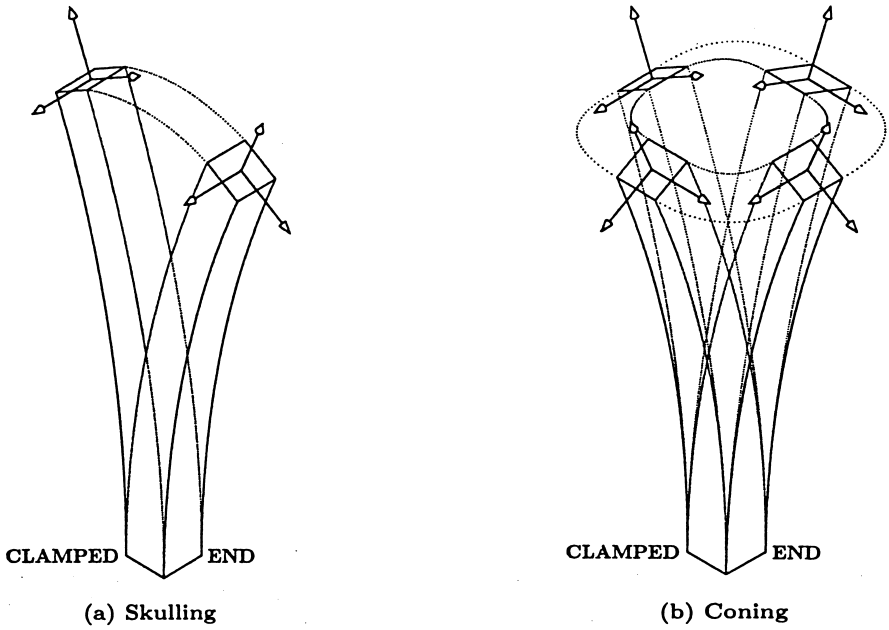


Fig. 6.19 *Detrimental vibration modes of cantilevered structures.*

illustrated in Fig. 6.19 as resonant modes of cantilever beams. A sensor cluster mounted away from the clamped end of such a structure will experience coupled translational and rotational motions at the vibration frequency, and mounting geometries with the center of mass of the instrument cluster offset from its effective center of support can induce vibrational modes of a similar character. Coning motion can also result in cumulative attitude errors due to mismatches in the frequency responses of sensors or neglected higher order terms in attitude rate integration.

6.5 SYSTEM-LEVEL ERROR MODELS

There is no single, all-encompassing design for INS/GPS integration, because there is no standard design for an INS. There may be minor differences between generations of GPS satellites, but the differences between INS types are anything but minor. There are some broad INS design types (e.g., gimbaled vs. strapdown), but there are literally thousands of different inertial sensor designs that can be used for each INS type.

What matters most from the standpoint of GPS/INS integration are the mathematical models for the different types of error sources. We present here a variety of inertial sensor error models, which will be sufficient for many of the sensors in common use but perhaps not for every conceivable inertial sensor. For applications with sensor characteristics different from those used here, the use of these error

models in GPS/INS integration will serve to illustrate the general integration *methodology*, so that users can apply the same methodology to GPS/INS integration with other sensor error models as well.

6.5.1 Error Sources

6.5.1.1 Initialization Errors Inertial navigators can only integrate sensed accelerations to propagate initial estimates of position and velocity. Systems without GPS aiding require other sources for their initial estimates of position and velocity. Initialization errors are the errors in these initial values.

6.5.1.2 Alignment Errors Most stand-alone INS implementations include an initial period for alignment of the gimbals (for gimbaled systems) or attitude direction cosines (for strapdown systems) with respect to the navigation axes. Errors remaining at the end of this period are the alignment errors. These include *tilts* (rotations about horizontal axes) and azimuth reference errors. Tilt errors introduce acceleration errors through the miscalculation of gravitational acceleration, and these propagate primarily as Schuler oscillations (i.e., zero-mean position and velocity errors with $\approx$ 84-min period) plus a non-zero-mean position error approximately equal to the tilt error in radians times the radius from the earth center. Initial azimuth errors primarily rotate the system trajectory about the starting point, but there are secondary effects due to Coriolis accelerations and excitation of Schuler oscillations.

6.5.1.3 Sensor Compensation Errors Sensor calibration is a procedure for estimating the parameters of models used in sensor error compensation. It is not uncommon for these modeled parameters to change over time and between turn-ons, and designing sensors to make the parameters sufficiently constant can also make the sensors relatively expensive. Costs resulting from stringent requirements for parameter stability can be reduced significantly for sensors that will be used in integrated GPS/INS applications, because Kalman-filter-based GPS/INS integration can use the differences between INS-derived position and GPS-derived position to make corrections to the calibration parameters.

These nonconstant sensor compensation parameters are not true parameters (i.e., constants), but "*slow variables*," which change slowly relative to the other dynamic variables. Other slow variables in the integrated system model include the satellite clock offsets for Selective Availability (SA).

The GPS/INS integration filter implementation requires models for how variations in the compensation parameters propagate into navigation errors. These models are derived in Section 6.5.3 for the more common types of sensors and their compensation parameters.

6.5.1.4 Gravity Model Errors

The influence of unknown gravity modeling errors on vehicle dynamics is usually modeled as a zero-mean exponentially correlated acceleration process,[4]

$$\delta \mathbf{a}_k = e^{-\Delta t/\tau_{\text{correlation}}} \delta \mathbf{a}_{k-1} + \mathbf{w}_k, \tag{6.51}$$

where Δt is the filter period, the correlation time

$$\tau_{\text{correlation}} \approx \frac{D_{\text{correlation}}}{|\mathbf{v}_{\text{horizontal}}|}, \tag{6.52}$$

$\mathbf{v}_{\text{horizontal}}$ is horizontal velocity, $D_{\text{correlation}}$ is the horizontal correlation distance of gravity anomalies (usually on the order of 10^4–10^5 m), $\mathbf{w}_k$ is a zero-mean white-noise process with covariance matrix

$$\mathbf{Q}_{\text{gravitymodel}} \stackrel{\text{def}}{=} E\langle \mathbf{w}_k \mathbf{w}_k^T \rangle \tag{6.53}$$

$$\approx a_{\text{RMS}}^2 (1 - e^{-2\Delta t/\tau}) \mathbf{I}, \tag{6.54}$$

a_{RMS}^2 is the variance of acceleration error, and $\mathbf{I}$ is an identity matrix. The correlation distance $D_{\text{correlation}}$ and RMS acceleration disturbance a_{RMS} will generally depend upon the local terrain. Here, $D_{\text{correlation}}$ tends to be larger and a_{RMS} smaller as terrain becomes more gentle or (for aircraft) as altitude increases.

6.5.2 Navigation Error Propagation

The dynamics of INS error propagation are strongly influenced by the fact that gravitational accelerations point toward the center of the earth and decrease in magnitude with altitude and is somewhat less influenced by the fact that the earth rotates.

6.5.2.1 Schuler Oscillation

Any horizontal location error ε will cause a proportional miscalculation of the horizontal component of the modeled gravitational acceleration $\hat{G}$, as illustrated in Fig. 6.20, and the acceleration error is in the direction opposite the location error. The net effect is an oscillation of the horizontal position error with a period τ_{Schuler} depending on the distance from the center of the earth and the acceleration due to gravity at that radius. At the surface of the earth,

$$\Omega_{\text{Schuler}} \approx \sqrt{\frac{9.8 \text{ m/s}^2}{6.4 \times 10^6 \text{ m}}} \tag{6.55}$$

$$\approx 0.00124 \text{ rad/s}, \tag{6.56}$$

$$\tau_{\text{Schuler}} \approx 84.4 \text{ min}. \tag{6.57}$$

The Schuler period is, in fact, the orbital period at that altitude.

[4] See Section 7.5.1.3.

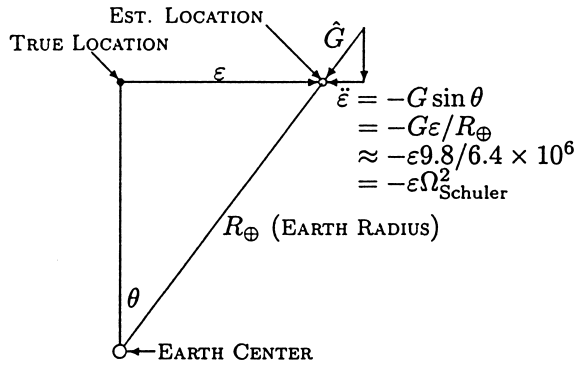


Fig. 6.20 *Schuler oscillation of position error.*

Schuler oscillation tends to make RMS horizontal position errors for INSs proportional to RMS horizontal velocity errors, with a proportionality constant

$$\frac{\dot{\varepsilon}_{\text{RMS}}}{\varepsilon_{\text{RMS}}} \approx 0.00124 \text{ s}^{-1}, \tag{6.58}$$

the Schuler frequency. For example, a Schuler oscillation with peak location errors on the order of 1 km will have peak velocity errors on the order of 1 m/s.

6.5.2.2 Vertical Channel Instability The vertical gradient of gravitational acceleration is G/R, where G is the gravitational acceleration at the radius R where the gradient is calculated. Consequently, any positive (upward) error in estimated altitude will result in an undercalculated downward gravitational acceleration, with the acceleration error in the same direction and proportional to the altitude error. The result is an ever-so-slightly unstable altitude error propagation equation, with an exponential time constant on the order of 180 h for operation at the surface of the earth. The vertical channel therefore requires some sort of auxiliary measurement, such as barometric altitude, to stabilize it.

6.5.2.3 Coriolis Coupling The Coriolis effect couples north velocity into east acceleration, with a proportionality constant equal to $\omega_\oplus \sin(\phi)$, where $\omega_\oplus$ is the earth rotation rate ($\approx 7.3 \times 10^{-5}$ rad/s) and ϕ is latitude. That is,

$$\ddot{\varepsilon}_E = -\Omega^2_{\text{Schuler}}\varepsilon_E + \omega_\oplus \sin(\phi)\dot{\varepsilon}_N, \tag{6.59}$$

$$\ddot{\varepsilon}_N = -\Omega^2_{\text{Schuler}}\varepsilon_N \tag{6.60}$$

where ε_E is east position error and ε_N is north position error.

6.5.3 Sensor Error Propagation

Errors made in compensating for inertial sensor errors will cause navigation errors. Here, we derive some approximating formulas for how errors in individual compensation parameters propagate into velocity and position errors.

6.5.3.1 Accelerometer Compensation Error Propagation The compensation equation for the most common accelerometer errors (scale factors, input axis misalignments, and biases) can be put in the form

$$\mathbf{a}_{\text{compensated}} \approx \mathbf{a}_{\text{input}} \tag{6.61}$$

$$= \overline{\mathbf{M}}_a\{\mathbf{a}_{\text{output}} - \mathbf{a}_{\text{bias}}\}, \tag{6.62}$$

where the 12 compensation parameters are the 9 elements of the scale factor misalignment matrix $\overline{\mathbf{M}}_a$ and the 3 components of the output bias vector $\mathbf{a}_{\text{bias}}$.

The first-order sensitivities of compensated acceleration to variations in these parameters can be calculated as the partial derivatives

$$\frac{\partial \mathbf{a}_{\text{compensated}}}{\mathbf{a}_{\text{bias}}} = -\overline{\mathbf{M}}_a, \tag{6.63}$$

$$\frac{\partial a_{i,\text{compensated}}}{\partial m_{akj}} = \begin{cases} 0, & k \neq i, \\ a_{j,\text{output}} - a_{j,\text{bias}}, & k = i, \end{cases} \tag{6.64}$$

where m_{akj} is the element in the kth row and jth column of $\overline{\mathbf{M}}_a$. If we let the order of compensation parameters be

$$\mathbf{p}_{\text{acc.comp}} = [a_{1,\text{bias}}\ a_{2,\text{bias}}\ a_{3,\text{bias}}\ m_{a11}\ m_{a12}\ m_{a13}\ m_{a21}\ m_{a22}\ m_{a23}\ m_{a31}\ m_{a32}\ m_{a33}]^{\text{T}}, \tag{6.65}$$

then the associated matrix of partial derivatives will be

$$\frac{\partial \mathbf{a}_{\text{compensated}}}{\partial \mathbf{p}_{\text{acc.comp}}} = \begin{bmatrix} -m_{a11} & -m_{a21} & -m_{a31} \\ -m_{a12} & -m_{a22} & -m_{a32} \\ -m_{a13} & -m_{a23} & -m_{a33} \\ a_{1,\text{output}} - a_{1,\text{bias}} & 0 & 0 \\ a_{2,\text{output}} - a_{2,\text{bias}} & 0 & 0 \\ a_{3,\text{output}} - a_{3,\text{bias}} & 0 & 0 \\ 0 & a_{1,\text{output}} - a_{1,\text{bias}} & 0 \\ 0 & a_{2,\text{output}} - a_{2,\text{bias}} & 0 \\ 0 & a_{3,\text{output}} - a_{3,\text{bias}} & 0 \\ 0 & 0 & a_{1,\text{output}} - a_{1,\text{bias}} \\ 0 & 0 & a_{2,\text{output}} - a_{2,\text{bias}} \\ 0 & 0 & a_{3,\text{output}} - a_{3,\text{bias}} \end{bmatrix}^{\text{T}}. \tag{6.66}$$

Acceleration errors due to accelerometer compensation errors in body coordinates and navigation coordinates will then be

$$\delta \mathbf{a}_{\text{body}} \approx \frac{\partial \mathbf{a}_{\text{compensated}}}{\partial \mathbf{p}_{\text{acc.comp}}} \delta \mathbf{p}_{\text{acc.comp}}, \tag{6.67}$$

$$\delta \mathbf{a}_{\text{nav}} = \mathbf{C}_{\text{nav}}^{\text{body}} \delta \mathbf{a}_{\text{body}} \tag{6.68}$$

$$\approx \mathbf{C}_{\text{nav}}^{\text{body}} \frac{\partial \mathbf{a}_{\text{compensated}}}{\partial \mathbf{p}_{\text{acc.comp}}} \delta \mathbf{p}_{\text{acc.comp}}, \tag{6.69}$$

where $\delta \mathbf{p}_{\text{acc.comp}}$ is the vector of compensation parameter errors and $\mathbf{C}_{\text{nav}}^{\text{body}}$ is the coordinate transformation matrix from body-fixed coordinates to navigation coordinates (e.g., $\mathbf{C}_{\text{ENU}}^{\text{RPY}}$ from Eq. C.93).

The velocity error sensitivities to each of the compensation parameters will be the integral over time of the acceleration sensitivities, and the position error sensitivities to each of the compensation parameters will be the integral over time of the velocity sensitivities. However, the accelerations must be transformed into navigation coordinates before integration:

$$\delta \mathbf{v}_{\text{nav}}(t) = \delta \mathbf{v}_{\text{nav}}(t_0) + \int_{t_0}^{t} \delta \mathbf{a}_{\text{nav}}(s)\, ds \tag{6.70}$$

$$= \delta \mathbf{v}_{\text{nav}}(t_0) + \int_{t_0}^{t} \mathbf{C}_{\text{nav}}^{\text{body}}(s)\, \delta \mathbf{a}_{\text{body}}(s)\, ds \tag{6.71}$$

$$\approx \delta \mathbf{v}_{\text{nav}}(t_0) + \int_{t_0}^{t} \mathbf{C}_{\text{nav}}^{\text{body}}(s) \frac{\partial \mathbf{a}_{\text{comp}}}{\partial \mathbf{p}_{\text{acc.comp}}}(s)\, \delta \mathbf{p}_{\text{acc.comp}} \tag{6.72}$$

$$\delta \mathbf{x}_{\text{nav}}(t) \approx \delta \mathbf{x}_{\text{nav}}(t_0) + (t - t_0)\, \delta \mathbf{v}_{\text{nav}}(t_0) + \int \int_{t_0}^{t} \mathbf{C}_{\text{nav}}^{\text{body}}(s) \frac{\partial \mathbf{a}_{\text{comp}}}{\partial \mathbf{p}_{\text{acc.comp}}}(s)\, \delta \mathbf{p}_{\text{acc.comp}}\, ds, \tag{6.73}$$

where $\mathbf{C}_{\text{nav}}^{\text{body}} \equiv \mathbf{I}$ for gimbaled systems and $\delta \mathbf{x}_{\text{nav}}$ is the navigation position error due to compensation parameter errors. The GPS navigation solution does not include $\delta \mathbf{x}_{\text{nav}}$, and it is the difference between the INS and GPS solutions that is used to estimate the compensation parameter errors.

6.5.3.2 Gyroscope Compensation Error Propagation

The principal effect of gyroscope compensation errors on inertial navigation position errors is from the miscalculation of gravitational acceleration due to the resulting tilt errors, as illustrated in Fig. 6.21, where

$$\delta a_E \approx -g\, \delta \theta_N, \tag{6.74}$$

$$\delta a_N \approx g\, \delta \theta_E, \tag{6.75}$$

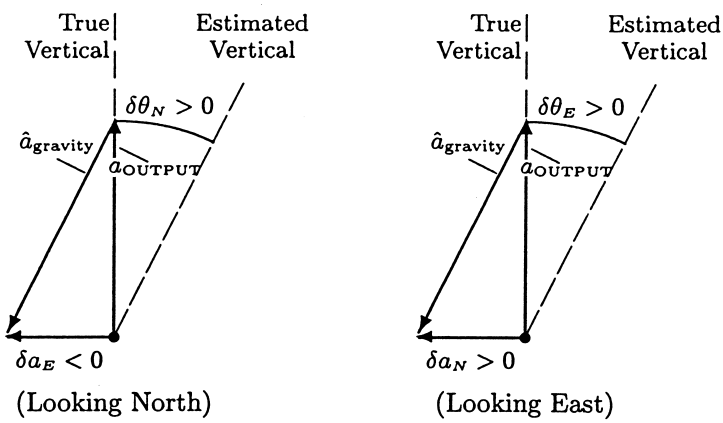


Fig. 6.21 *Acceleration errors due to tilts.*

for the tilt error angles $\delta\theta_E$, $\delta\theta_N$ in radians and $g \approx 9.8\,\text{m/s}^2$. The corresponding position errors will be the double integrals of the acceleration errors,

$$\delta x_E(t) \approx \delta x_E(t_0) + (t - t_0)\,\delta v_E(t_0) + g \int\int_{t_0}^{t} \delta\theta_N(s)\,ds \tag{6.76}$$

$$\delta x_N(t) \approx \delta x_N(t_0) + (t - t_0)\,\delta v_N(t_0) - g \int\int_{t_0}^{t} \delta\theta_E(s)\,ds. \tag{6.77}$$

The sensitivity to rotational error about the local vertical (i.e., heading error) is usually smaller, with

$$\delta x_E(t) \approx \delta x_E(t_0) - \delta\theta_U\,\Delta x_N, \tag{6.78}$$

$$\delta x_N(t) \approx \delta x_N(t_0) + \delta\theta_U\,\Delta x_E, \tag{6.79}$$

where δx_E and δx_N are the navigation error components due to heading error $\delta\theta_U$ (i.e., measured counterclockwise) in radians and Δx_E and Δx_N are the net position changes between time t_0 and t in the east and north directions, respectively.

The compensation equation for the most common gyroscope errors (scale factors, input axis misalignments, and biases) has the same form as those for accelerometer errors,

$$\boldsymbol{\omega}_{\text{input}} = \overline{\mathbf{M}}_g\{\boldsymbol{\omega}_{\text{output}} - \boldsymbol{\omega}_{\text{bias}}\}, \tag{6.80}$$

where the gyroscope compensation parameters are the nine elements of the gyroscope scale factor misalignment matrix $\overline{\mathbf{M}}_g$ and the three components of the output bias vector $\boldsymbol{\omega}_{\text{bias}}$. The first-order sensitivities of compensated rotation rate to variations in these parameters can be calculated as the partial derivatives

$$\frac{\partial \boldsymbol{\omega}_{\text{input}}}{\partial \boldsymbol{\omega}_{\text{bias}}} = -\overline{\mathbf{M}}_g, \tag{6.81}$$

$$\frac{\partial \omega_{i,\text{input}}}{\partial m_{gkj}} = \begin{cases} 0, & k \neq i, \\ \omega_{j,\text{input}} - \omega_{j,\text{bias}}, & k = i, \end{cases} \tag{6.82}$$

where m_{gkj} is the element in the kth row and jth column of $\overline{\mathbf{M}}_g$. If we let

$$\mathbf{p}_{\text{gyro.comp}} = [\omega_{1,\text{bias}}\ \omega_{2,\text{bias}}\ \omega_{3,\text{bias}}\ m_{g11}\ m_{g12}\ m_{g13}\ m_{g21}\ m_{g22}\ m_{g23},\ m_{g31}\ m_{g32}\ m_{g33}]^T, \tag{6.83}$$

then the matrix of partial derivatives becomes

$$\frac{\partial \boldsymbol{\omega}_{\text{comp}}}{\partial \mathbf{p}_{\text{gyro.comp}}} = \begin{bmatrix} -m_{g11} & -m_{g21} & -m_{g31} \\ m_{g12} & -m_{g22} & -m_{g32} \\ -m_{g13} & -m_{g23} & -m_{g33} \\ \omega_{1,\text{output}} - \omega_{1,\text{bias}} & 0 & 0 \\ \omega_{2,\text{output}} - \omega_{2,\text{bias}} & 0 & 0 \\ \omega_{3,\text{output}} - \omega_{3,\text{bias}} & 0 & 0 \\ 0 & \omega_{1,\text{output}} - \omega_{1,\text{bias}} & 0 \\ 0 & \omega_{2,\text{output}} - \omega_{2,\text{bias}} & 0 \\ 0 & \omega_{3,\text{output}} - \omega_{3,\text{bias}} & 0 \\ 0 & 0 & \omega_{1,\text{output}} - \omega_{1,\text{bias}} \\ 0 & 0 & \omega_{2,\text{output}} - \omega_{2,\text{bias}} \\ 0 & 0 & \omega_{3,\text{output}} - \omega_{3,\text{bias}} \end{bmatrix}^{\text{T}} \tag{6.84}$$

and the tilt errors

$$\delta\boldsymbol{\theta}_{\text{nav}}(t) = \delta\boldsymbol{\theta}_{\text{nav}}(t_0) + \int_{t_0}^{t} \delta\boldsymbol{\omega}_{\text{nav}}(s)\, ds \tag{6.85}$$

$$= \delta\boldsymbol{\theta}_{\text{nav}}(t_0) + \int_{t_0}^{t} \mathbf{C}_{\text{nav}}^{\text{body}}(s)\, \delta\boldsymbol{\omega}_{\text{body}}(s)\, ds \tag{6.86}$$

$$\approx \delta\boldsymbol{\theta}_{\text{nav}}(t_0) + \int_{t_0}^{t} \mathbf{C}_{\text{nav}}^{\text{body}}(s) \frac{\partial \boldsymbol{\omega}_{\text{comp}}}{\partial \mathbf{p}_{\text{gyro.comp}}}(s) \delta\mathbf{p}_{\text{gyro.comp}}\, ds. \tag{6.87}$$

The east and north tilt components can then be substituted into Eqs. 6.77 and 6.76 to obtain the equation for position error due to tilts. Schuler oscillations (Sections 2.2.2.3 and 6.5.2.1) are excited when these position errors, in turn, cause tilts.

6.5.3.3 Carouseling and Gimbal Flipping These methods are commonly used for mitigating the effects of sensor compensation errors in gimbaled systems. The ability to rotate a gimbaled inertial platform was soon exploited as a means of averaging out the effects of many types of sensor errors on navigational accuracy. The simplest schemes rotate the platform containing the sensors about the local vertical with a rotation period significantly shorter than the Schuler period of 84 min (carouseling) or in discrete 90° or 180° steps (gimbal flipping). The effects of many platform-fixed sensor errors (e.g., accelerometer and gyroscope biases and input axis misalignments) can be effectively cancelled by such rotations.

Problems

6.1 In the one-dimensional Line Land world of Section 6.4.1.1, an INS requires no gyroscopes. How many gyroscopes would be required for two-dimensional navigation in Flat Land?

6.2 Derive the equivalent formulas in terms of Y (yaw angle), P (pitch angle), and R (roll angle) for unit vectors $\mathbf{1}_R$, $\mathbf{1}_P$, $\mathbf{1}_Y$ in NED coordinates and $\mathbf{1}_N$, $\mathbf{1}_E$, $\mathbf{1}_D$ in RPY coordinates, corresponding to Eqs. C.86–C.91 of Appendix C.

6.3 Explain why accelerometers cannot sense gravitational accelerations.

6.4 Calculate the numbers of computer multiplies and adds required for

(a) gyroscope scale factor/misalignment compensation (Eq. 6.23),
(b) accelerometer scale factor/misalignment compensation (Eq. 6.28), and
(c) transformation of accelerations to navigation coordinates (Fig. 6.13) using quaternion rotations (Eq. C.243) requiring two quaternion products (Eq. C.234).

If the INS performs these 100 times per second, how many operations per second will be required?

Global Positioning Systems, Inertial Navigation, and Integration,
Mohinder S. Grewal, Lawrence R. Weill, Angus P. Andrews

Print ISBN 0-471-35032-X Electronic ISBN 0-471-20071-9

7

Kalman Filter Basics

7.1 INTRODUCTION

7.1.1 What is a Kalman Filter?

It is an extremely effective and versatile procedure for combining *noisy sensor outputs* to estimate the *state* of a *system* with *uncertain dynamics*.

For our purposes in this book:

- The *noisy sensors* may include *GPS receivers* and *inertial sensors* (accelerometers and gyroscopes, typically) but may also include speed sensors (e.g., wheel speeds of land vehicles, water speed sensors for ships, air speed sensors for aircraft, or Doppler radar), and time sensors (clocks).
- The *system state* in question may include the *position*, *velocity*, *acceleration*, *attitude*, and *attitude rate* of a *vehicle* on land, at sea, in the air, or in space, but the system state may include ancillary "nuisance variables" for modeling *correlated noise sources* (e.g., GPS Selective Availability timing errors) and *time-varying parameters* of the sensors, such as scale factor, output bias, or (for clocks) frequency. Selective Availability has been suspended as of May 1, 2000.
- *Uncertain dynamics* includes unpredictable disturbances of the host vehicle, whether caused by a human operator or by the medium (e.g., winds, surface currents, turns in the road, or terrain changes), but it may also include unpredictable changes in the sensor parameters.

More abstract treatments of the Kalman filter are presented in [18, 19, 40, 46, 67, 69, 71, 72], and a more basic introduction can be found in [31].

7.1.2 How it Works

The Kalman filter maintains two types of variables:

1. *Estimated State Vector.* The components of the estimated state vector include the following:

   (a) The variables of interest (i.e., what we want or need to know, such as position and velocity).
   (b) "Nuisance variables" that are of no intrinsic interest but may be necessary to the estimation process. These nuisance variables may include, for example, the selective availability errors of the GPS satellites. We generally do not wish to know their values but may be obliged to calculate them to improve the receiver estimate of position.
   (c) The Kalman filter state variables for a specific application must include all those system dynamic variables that are measurable by the sensors used in that application. For example, a Kalman filter for a system containing accelerometers and rate gyroscopes must contain acceleration and rotation rate components to which these instruments respond. The acceleration and angular rate components do not have to be those along the sensor input axes, however. The Kalman filter state variables could be the components along locally level earth-fixed coordinates, even though the sensors measure components in vehicle-body-fixed coordinates.

   In similar fashion, the Kalman filter state variables for GPS-only navigation must contain the position coordinates of the receiver antenna, but these could be geodetic latitude, longitude, and altitude with respect to a reference ellipsoid or geocentric latitude, longitude, and altitude with respect to a reference sphere, or ECEF Cartesian coordinates, or ECI coordinates, or any equivalent coordinates.
2. *A Covariance Matrix: a Measure of Estimation Uncertainty.* The equations used to propagate the covariance matrix (collectively called the *Riccati equation*) model and manage *uncertainty*, taking into account how sensor noise and dynamic uncertainty contribute to uncertainty about the estimated system state.

By maintaining an estimate of its own estimation uncertainty and the relative uncertainty in the various sensor outputs, the Kalman filter is able to combine all sensor information "optimally," in the sense that the resulting estimate minimizes any quadratic loss function of estimation error, including the mean-squared value of any linear combination of state estimation errors. The *Kalman gain* is the optimal

weighting matrix for combining new sensor data with a prior estimate to obtain a new estimate.

7.2 STATE AND COVARIANCE CORRECTION

The Kalman filter is a two-step process, the steps of which we call "prediction" and "correction." The filter can start with either step, but we will begin by describing the correction step first.

The correction step makes corrections to an estimate, based on new information obtained from sensor measurements.

The Kalman gain matrix $\overline{\mathbf{K}}$ is the crown jewel of Kalman filtering. All the effort of solving the matrix Riccati equation is for the sole purpose of computing the "optimal" value of the gain matrix $\overline{\mathbf{K}}$ used for correcting an estimate $\hat{\mathbf{x}}$,

$$\underbrace{\hat{\mathbf{x}}(+)}_{\text{corrected}} = \underbrace{\hat{\mathbf{x}}(-)}_{\text{predicted}} + \underbrace{\overline{\mathbf{K}}_{\text{gain}}}\left[\underbrace{\mathbf{z}}_{\text{meas.}} - \underbrace{\mathbf{H}\hat{\mathbf{x}}(-)}_{\text{pred. meas.}}\right], \tag{7.1}$$

based on a measurement

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \text{noise} \tag{7.2}$$

that is a linear function of the vector variable $\mathbf{x}$ to be estimated plus additive noise with known statistical properties.

We will derive a formula for the Kalman gain based on an analogous filter called the *Gaussian maximum-likelihood estimator*. It uses the analogies shown in Table 7.1 between concepts in Kalman filtering, Gaussian probability distributions, and likelihood functions.

The derivation begins with background on properties of Gaussian probability distributions and Gaussian likelihood functions, then development of models for noisy sensor outputs and a derivation of the associated *maximum-likelihood estimate* (MLE) for combining prior estimates with noisy sensor measurements.

7.2.1 Gaussian Probability Density Functions

Probability density functions (PDFs) are nonnegative integrable functions whose integral equals *unity* (i.e., 1). The density functions of *Gaussian* probability distributions all have the form

$$p(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^n \det \mathbf{P}}} \exp(-\tfrac{1}{2}[\mathbf{x} - \boldsymbol{\mu}]^T \mathbf{P}^{-1}[\mathbf{x} - \boldsymbol{\mu}]), \tag{7.3}$$

TABLE 7.1 Analogous Concepts in Three Different Contexts

Context	Kalman filtering	↔	Gaussian probability distributions	↔	Maximum-likelihood estimation
Concepts			Probability distribution	↔	Likelihood function $\mathcal{L}$
	Estimate	↔	Mean	↔	argmax($\mathcal{L}$)[a]
	Covariance	↔	Covariance	↔	Information

[a] Argmax(f) returns the argument x of the function f where $f(x)$ achieves its maximum value. For example, argmax(sin) $= \pi/2$ and argmax(cos)$= 0$.

where n is the dimension of $\mathbf{P}$ (i.e., $\mathbf{P}$ is an $n \times n$ matrix) and the parameters

$$\boldsymbol{\mu} \stackrel{\text{def}}{=} E_{\mathbf{x}\in\mathcal{N}(\boldsymbol{\mu},\mathbf{P})}\langle\mathbf{x}\rangle \tag{7.4}$$

$$\stackrel{\text{def}}{=} \int_{x_1} dx_1 \cdots \int_{x_1} dx_n \, p(\mathbf{x})\mathbf{x}, \tag{7.5}$$

$$\mathbf{P} \stackrel{\text{def}}{=} E_{\mathbf{x}\in\mathcal{N}(\boldsymbol{\mu},\mathbf{P})}\langle(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\rangle \tag{7.6}$$

$$\stackrel{\text{def}}{=} \int_{x_1} dx_1 \cdots \int_{x_1} dx_n \, p(\mathbf{x})(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}. \tag{7.7}$$

The parameter $\boldsymbol{\mu}$ is the *mean* of the distribution. It will be a column vector with the same dimensions as the variate $\mathbf{x}$.

The parameter $\mathbf{P}$ is the *covariance matrix* of the distribution. By its definition, it will always be an $n \times n$ *symmetric* and *nonnegative definite* matrix. However, because its determinant appears in the denominator of the square root and its inverse appears in the exponential function argument, it must be *positive definite* as well. That is, its eigenvalues must be real and positive for the definition to work.

The constant factor $1/\sqrt{(2\pi)^n \det \mathbf{P}}$ in Eq. 7.3 is there to make the integral of the probability density function equal to unity, a necessary condition for all probability density functions.

The operator $E\langle\cdot\rangle$ is the *expectancy operator*, also called the *expected-value operator*.

The notation $\mathbf{x} \in \mathcal{N}(\boldsymbol{\mu}, \mathbf{P})$ denotes that the *variate* (i.e., random variable) $\mathbf{x}$ is drawn from the Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\mathbf{P}$. Gaussian distributions are also called *normal* or *Laplace* distributions.

7.2.2 Likelihood Functions

Likelihood functions are similar to probability density functions, except that their integrals are not constrained to equal unity, or even required to be finite. They are useful for comparing *relative* likelihoods and for finding the value of the unknown independent variable x at which the likelihood function achieves its maximum, as illustrated in Fig. 7.1.

7.2.2.1 Gaussian Likelihood Functions Gaussian likelihood functions have the form

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\mu}, \mathbf{Y}) = \exp(-\tfrac{1}{2}[\mathbf{x} - \boldsymbol{\mu}]^{\mathrm{T}}\mathbf{Y}[\mathbf{x} - \boldsymbol{\mu}]), \tag{7.8}$$

where the parameter $\mathbf{Y}$ is called the *information matrix* of the likelihood function. It replaces $\mathbf{P}^{-1}$ in the Gaussian probability density function. If the information matrix $\mathbf{Y}$ is nonsingular, then its inverse $\mathbf{Y}^{-1} = \mathbf{P}$, a covariance matrix. However, *an information matrix is not required to be nonsingular.* This property of information matrices is important for representing the information from a set of measurements (sensor outputs) with incomplete information for determining the unknown vector $\mathbf{x}$. That is, the measurements may provide *no information* about some linear combinations of the components of $\mathbf{x}$, as illustrated in Fig. 7.2.

7.2.2.2 Scaling of Likelihood Functions Maximum-likelihood estimation is based on the argmax of the likelihood function, but for any positive scalar $c > 0$,

$$\operatorname{argmax}(c\mathcal{L}) = \operatorname{argmax}(\mathcal{L}). \tag{7.9}$$

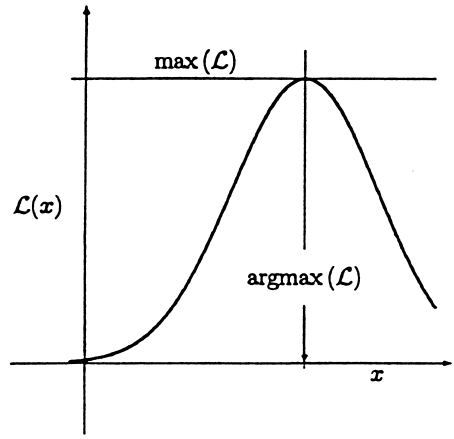


Fig. 7.1 *Maximum-likelihood estimate.*

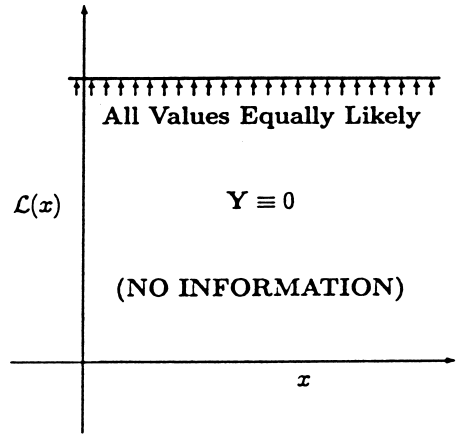


Fig. 7.2 *Likelihood without maximum.*

That is, positive scaling of likelihood functions will have no effect on the maximum-likelihood estimate. As a consequence, likelihood functions can have arbitrary positive scaling.

7.2.2.3 Independent Likelihood Functions The joint probability $P(A\&B)$ of independent events A and B is the product $P(A\&B) = P(A) \times P(B)$. The analogous effect on independent likelihood functions $\mathcal{L}_A$ and $\mathcal{L}_B$ is the pointwise product. That is, at each "point" x,

$$\mathcal{L}_{A\&B}(x) = \mathcal{L}_A(x) \times \mathcal{L}_B(x). \tag{7.10}$$

7.2.2.4 Pointwise Products One of the remarkable attributes of Gaussian likelihood functions is that their pointwise products are also Gaussian likelihood functions, as illustrated in Fig. 7.3.

Given two Gaussian likelihood functions with parameter sets $\{\boldsymbol{\mu}_A, \mathbf{Y}_A\}$ and $\{\boldsymbol{\mu}_B, \mathbf{Y}_B\}$, their pointwise product is a scaled Gaussian likelihood function with parameters $\{\boldsymbol{\mu}_{A\&B}, \mathbf{Y}_{A\&B}\}$ such that, for all $\mathbf{x}$,

$$\begin{aligned}
&\exp(-\tfrac{1}{2}[\mathbf{x} - \boldsymbol{\mu}_{A\&B}]^{\mathrm{T}}\mathbf{Y}_{A\&B}[\mathbf{x} - \boldsymbol{\mu}_{A\&B}]) \\
&\quad = c\ \exp(-\tfrac{1}{2}[\mathbf{x} - \boldsymbol{\mu}_A]^{\mathrm{T}}\mathbf{Y}_A[\mathbf{x} - \boldsymbol{\mu}_A]) \\
&\qquad \times \exp\left(-\frac{1}{2}[\mathbf{x} - \boldsymbol{\mu}_B]^{\mathrm{T}}\mathbf{Y}_B[\mathbf{x} - \boldsymbol{\mu}_B]\right)
\end{aligned} \tag{7.11}$$

for some constant $c > 0$.

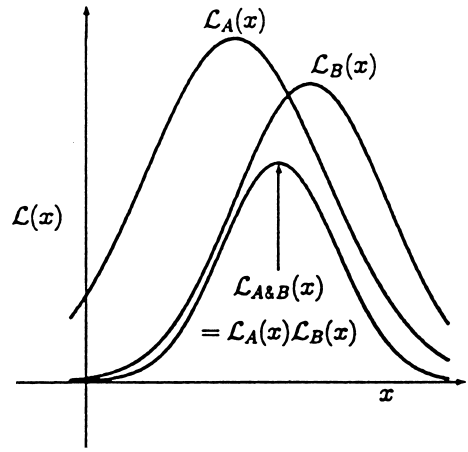


Fig. 7.3 *Pointwise products of Gaussian likelihood functions.*

One can solve Eq. 7.11 for the parameters $\{\boldsymbol{\mu}_{A\&B}, \mathbf{Y}_{A\&B}\}$ as functions of the parameters $\{\boldsymbol{\mu}_A, \mathbf{Y}_A, \boldsymbol{\mu}_B, \mathbf{Y}_B\}$. Taking logarithms of both sides of Eq. 7.11 will produce the equation

$$\begin{aligned} -\tfrac{1}{2}[\mathbf{x} - \boldsymbol{\mu}_{A\&B}]^{\mathrm{T}}\mathbf{Y}_{A\&B}[\mathbf{x} - \boldsymbol{\mu}_{A\&B}] \\ = \log(c) - \tfrac{1}{2}[\mathbf{x} - \boldsymbol{\mu}_A]^{\mathrm{T}}\mathbf{Y}_A[\mathbf{x} - \boldsymbol{\mu}_A] \\ - \tfrac{1}{2}[\mathbf{x} - \boldsymbol{\mu}_B]^{\mathrm{T}}\mathbf{Y}_B[\mathbf{x} - \boldsymbol{\mu}_B]. \end{aligned} \tag{7.12}$$

Next, taking the first and second derivatives with respect to the independent variable $\mathbf{x}$ will produce the equations

$$\mathbf{Y}_{A\&B}(\mathbf{x} - \boldsymbol{\mu}_{A\&B}) = \mathbf{Y}_A(\mathbf{x} - \boldsymbol{\mu}_A) + \mathbf{Y}_B(\mathbf{x} - \boldsymbol{\mu}_B), \tag{7.13}$$

$$\mathbf{Y}_{A\&B} = \mathbf{Y}_A + \mathbf{Y}_B, \tag{7.14}$$

respectively.

Information is Additive The information matrix of the combined likelihood function ($\mathbf{Y}_{A\&B}$ in Eq. 7.14) equals the sum of the individual information matrices of the component likelihood functions ($\mathbf{Y}_A$ and $\mathbf{Y}_B$ in Eq. 7.14).

Combined Maximum-Likelihood Estimate is a Weighted Average Equation 7.13 evaluated at $\mathbf{x} = 0$ is

$$\mathbf{Y}_{A\&B}\boldsymbol{\mu}_{A\&B} = \mathbf{Y}_A\boldsymbol{\mu}_A + \mathbf{Y}_B\boldsymbol{\mu}_B, \tag{7.15}$$

which can be solved for

$$\boldsymbol{\mu}_{A\&B} = \mathbf{Y}^{\dagger}{}_{A\&B}(\mathbf{Y}_A\boldsymbol{\mu}_A + \mathbf{Y}_B\boldsymbol{\mu}_B) \tag{7.16}$$

$$= (\mathbf{Y}_A + \mathbf{Y}_B)^{\dagger}(\mathbf{Y}_A\boldsymbol{\mu}_A + \mathbf{Y}_B\boldsymbol{\mu}_B), \tag{7.17}$$

where † denotes the Moore–Penrose inverse of a matrix (defined in Section B.1.4.7).

Equations 7.14 and 7.17 are key results for deriving the formula for Kalman gain. All that remains is to define likelihood function parameters for noisy sensors.

7.2.3 Noisy Measurement Likelihoods

The term *measurements* refers to outputs of sensors that are to be used in estimating the argument vector $\mathbf{x}$ of a likelihood function. Measurement models represent how these measurements are related to $\mathbf{x}$, including those errors called *measurement errors* or *sensor noise*. These models can be expressed in terms of likelihood functions with $\mathbf{x}$ as the argument.

7.2.3.1 Measurement Vector The collective output values from a multitude ℓ of sensors are the components of a vector

$$\mathbf{z} \stackrel{\text{def}}{=} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_\ell \end{bmatrix}, \tag{7.18}$$

called the *measurement vector*, a column vector with ℓ rows.

7.2.3.2 Measurement Sensitivity Matrix We suppose that the measured values z_i are linearly[1] related to the unknown vector $\mathbf{x}$ we wish to estimate,

$$\mathbf{z} = \mathbf{H}\mathbf{x} \tag{7.19}$$

where $\mathbf{H}$ is the measurement sensitivity matrix.

7.2.3.3 Measurement Noise Measurement noise is the electronic noise at the output of the sensors. It is assumed to be additive,

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{v}, \tag{7.20}$$

[1] The Kalman filter is defined in terms of the *measurement sensitivity matrix* $\mathbf{H}$, but the *extended Kalman filter* (described in Section 7.6.4) can be defined in terms of a suitably differentiable vector-valued function $\mathbf{h}(\mathbf{x})$.

where the measurement noise vector $\mathbf{v}$ is assumed to be zero-mean Gaussian noise with known covariance $\mathbf{R}$,

$$E\langle \mathbf{v} \rangle \stackrel{\text{def}}{=} 0, \tag{7.21}$$

$$\mathbf{R} \stackrel{\text{def}}{=} E\langle \mathbf{v}\mathbf{v}^{\mathrm{T}} \rangle. \tag{7.22}$$

7.2.3.4 Measurement Likelihood A measurement vector $\mathbf{z}$ and its associated covariance matrix of measurement noise $\mathbf{R}$ define a likelihood function for the "true" value of the measurement (i.e., without noise). This likelihood function will have mean

$$\boldsymbol{\mu}_{\mathbf{z}} = \mathbf{z}$$

and information matrix

$$\mathbf{Y}_{\mathbf{z}} = \mathbf{R}^{-1},$$

assuming $\mathbf{R}$ is nonsingular.

7.2.3.5 Unknown Vector Likelihoods The same parameters defining measurement likelihoods also define an inferred likelihood function for the true value of the unknown vector, with mean

$$\boldsymbol{\mu}_{\mathbf{x}} = \mathbf{H}^{\dagger}\boldsymbol{\mu}_{\mathbf{z}} \tag{7.23}$$

$$= \mathbf{H}^{\dagger}\mathbf{z} \tag{7.24}$$

and information matrix

$$\mathbf{Y}_{\mathbf{x}} = \mathbf{H}^{T}\mathbf{Y}_{\mathbf{z}}\mathbf{H} \tag{7.25}$$

$$= \mathbf{H}^{T}\mathbf{R}^{-1}\mathbf{H}, \tag{7.26}$$

where the $n \times \ell$ matrix $\mathbf{H}^{\dagger}$ is defined as the Moore–Penrose generalized inverse (defined in Appendix B) of the $\ell \times n$ matrix $\mathbf{H}$. This information matrix will be singular if $\ell < n$ (i.e., there are fewer sensor outputs than unknown variables), which is not unusual for GPS/INS integration.

7.2.4 Gaussian MLE

7.2.4.1 Variables Gaussian MLE uses the following variables:

- $\hat{\mathbf{x}}$, the maximum likelihood estimate of $\mathbf{x}$. It will always equal the argmax (mean $\boldsymbol{\mu}$) of an associated Gaussian likelihood function, but it can have different values:

$\hat{\mathbf{x}}(-)$, representing the likelihood function prior to using the measurements.

$\hat{\mathbf{x}}(+)$, representing the likelihood function after using the measurements.

P, the covariance matrix of estimation uncertainty. It will always equal the inverse of the information matrix **Y** of the associated likelihood function. It also can have two values:

$\mathbf{P}(-)$, representing the likelihood function prior to using the measurements.

$\mathbf{P}(+)$, representing the likelihood function after using the measurements.

z, the vector of measurements.

H, the measurement sensitivity matrix.

R, the covariance matrix of sensor noise.

7.2.4.2 Update Equations

The MLE formula for updating the variables $\hat{\mathbf{x}}$ and **P** to reflect the effect of measurements can be derived from Eqs. 7.14 and 7.17 with initial likelihood parameters

$$\boldsymbol{\mu}_A = \hat{\mathbf{x}}(-), \tag{7.27}$$

the MLE before measurements, and

$$\mathbf{Y}_A = \mathbf{P}(-)^{-1}, \tag{7.28}$$

the inverse of the covariance matrix of MLE uncertainty before measurements. The likelihood function of **x** inferred from the measurements alone (i.e., without taking into account the prior estimate) is represented by the likelihood function parameters

$$\mathbf{Y}_B = \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}, \tag{7.29}$$

the information matrix of the measurements, and

$$\boldsymbol{\mu}_B = \mathbf{H}^{\dagger}\mathbf{z}, \tag{7.30}$$

where **z** is the measurement vector.

7.2.4.3 Covariance Update

The Gaussian likelihood function with parameters $\{\boldsymbol{\mu}_{A\&B}, \mathbf{Y}_{A\&B}\}$ of Eqs. 7.14 and 7.17 then represents the state of knowledge about the unknown vector **x** combining both sources (i.e., the prior likelihood and the effect of the measurements). That is, the covariance of MLE uncertainty after using the measurements will be

$$\mathbf{P}(+) = \mathbf{Y}_{A\&B}^{-1}, \tag{7.31}$$

and the MLE of **x** after using the measurements will then be

$$\hat{\mathbf{x}}(+) = \boldsymbol{\mu}_{A\&B}. \tag{7.32}$$

Equation 7.14 can be simplified by applying the following general matrix formula[2]:

$$(\mathbf{A}^{-1} + \mathbf{B}\mathbf{C}^{-1}\mathbf{D})^{-1} = \mathbf{A} - \mathbf{A}\mathbf{B}(\mathbf{C} + \mathbf{D}\mathbf{A}\mathbf{B})^{-1}\mathbf{D}\mathbf{A}, \tag{7.33}$$

[2] A formula with many discoverers. Henderson and Searle [53] list some earlier ones.

where

$\mathbf{A}^{-1} = \mathbf{Y}_A$, the prior information matrix for $\hat{\mathbf{x}}$
$\mathbf{A} = \mathbf{P}(-)$, the prior covariance matrix for $\hat{\mathbf{x}}$
$\mathbf{B} = \mathbf{H}^{\mathrm{T}}$, the transpose of the measurement sensitivity matrix
$\mathbf{C} = \mathbf{R}$
$\mathbf{D} = \mathbf{H}$, the measurement sensitivity matrix,

so that Eq. 7.31 becomes

$$\mathbf{P}(+) = \mathbf{Y}_{A\&B}^{-1} \tag{7.34}$$
$$= (\mathbf{Y}_A + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H})^{-1} \quad \text{(Eq.7.14)} \tag{7.35}$$
$$= \mathbf{Y}_A^{-1} - \mathbf{Y}_A^{-1}\mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{Y}_A^{-1}\mathbf{H}^{\mathrm{T}} + \mathbf{R})^{-1}\mathbf{H}\mathbf{Y}_A^{-1} \quad \text{(Eq.7.33)} \tag{7.36}$$
$$= \mathbf{P}(-) - \mathbf{P}(-)\mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{P}(-)\mathbf{H}^{\mathrm{T}} + \mathbf{R})^{-1}\mathbf{H}\mathbf{P}(-), \tag{7.37}$$

a form better conditioned for computation.

7.2.4.4 Estimate Update Equation 7.17 with substitutions from Eqs. 7.27–7.30 will have the form shown in Eq. 7.37

$$\hat{\mathbf{x}}(+) = \boldsymbol{\mu}_{A\&B} \quad \text{(Eq. 7.32)} \tag{7.38}$$
$$= (\mathbf{Y}_A + \mathbf{Y}_B)^{\dagger}(\mathbf{Y}_A\boldsymbol{\mu}_A + \mathbf{Y}_B\boldsymbol{\mu}_B) \quad \text{(Eq. 17.7)} \tag{7.39}$$
$$= \underbrace{\mathbf{P}(+)}_{\text{Eq. 7.31}}\left[\underbrace{\mathbf{P}(-)^{-1}}_{\text{Eq. 7.28}}\underbrace{\hat{\mathbf{x}}(-)}_{\text{Eq. 7.27}} + \underbrace{\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}}_{\text{Eq. 7.29}}\underbrace{\mathbf{H}^{\dagger}\mathbf{z}}_{\text{Eq. 7.30}}\right] \tag{7.40}$$
$$= [\mathbf{P}(-) - \mathbf{P}(-)\mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{P}(-)\mathbf{H}^{\mathrm{T}} + \mathbf{R})^{-1}\mathbf{H}\mathbf{P}(-)] \quad \text{(Eq.7.37)}$$
$$\times [\mathbf{P}(-)^{-1}\hat{\mathbf{x}}(-) + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}\mathbf{H}^{\dagger}\mathbf{z}] \tag{7.41}$$
$$= [\mathbf{I}(-) - \mathbf{P}(-)\mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{P}(-)\mathbf{H}^{\mathrm{T}} + \mathbf{R})^{-1}\mathbf{H}]$$
$$\times [\hat{\mathbf{x}}(-) + \mathbf{P}(-)\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}\mathbf{H}^{\dagger}\mathbf{z}] \tag{7.42}$$
$$= \hat{\mathbf{x}}(-) + \mathbf{P}(-)\mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{P}(-)\mathbf{H}^{\mathrm{T}} + \mathbf{R})^{-1}$$
$$\times \left\{[(\mathbf{H}\mathbf{P}(-)\mathbf{H}^{\mathrm{T}} + \mathbf{R})\mathbf{R}^{-1} - \mathbf{H}\mathbf{P}(-)\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}]\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}(-)\right\} \tag{7.43}$$
$$= \hat{\mathbf{x}}(-) + \mathbf{P}(-)\mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{P}(-)\mathbf{H}^{\mathrm{T}} + \mathbf{R})^{-1}$$
$$\times \left\{[\mathbf{H}\mathbf{P}(-)\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1} + \mathbf{I} - \mathbf{H}\mathbf{P}(-)\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}]\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}(-)\right\} \tag{7.44}$$
$$= \hat{\mathbf{x}}(-) + \underbrace{\mathbf{P}(-)\mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{P}(-)\mathbf{H}^{\mathrm{T}} + \mathbf{R})^{-1}}_{\overline{\mathbf{K}}} \times \left\{\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}(-)\right\}, \tag{7.45}$$

where the matrix $\overline{\mathbf{K}}$ has a special meaning in Kalman filtering.

7.2.5 Kalman Gain Matrix

Equation 7.45 has the form of Eq. 7.1 with Kalman gain matrix

$$\overline{\mathbf{K}} = \mathbf{P}(-)\mathbf{H}^{\mathrm{T}}[\mathbf{H}\mathbf{P}(-)\mathbf{H}^{\mathrm{T}} + \mathbf{R}]^{-1}, \tag{7.46}$$

which can be substituted into Eq. 7.37 to yield a simplified update equation for the covariance matrix update for the effects of using measurements:

$$\mathbf{P}(+) = \mathbf{P}(-) - \overline{\mathbf{K}}\mathbf{H}\mathbf{P}(-). \tag{7.47}$$

This completes the derivation of the Kalman gain matrix based on Gaussian MLE.

7.3 STATE AND COVARIANCE PREDICTION

The rest of the Kalman filter is the prediction step, in which the estimate $\hat{\mathbf{x}}$ and its associated covariance matrix of estimation uncertainty $\mathbf{P}$ are propagated from one time epoch to another. This is the part where the dynamics of the underlying physical processes come into play. The "state" of a dynamic process is a vector of variables that completely specify enough of the initial boundary value conditions for propagating the trajectory of the dynamic process forward in time, and the procedure for propagating that solution forward in time is called "state prediction." The model for propagating the covariance matrix of estimation uncertainty is derived from the model used for propagating the state vector.

7.3.1 State Prediction Models

7.3.1.1 Differential Equation Models Ever since the differential calculus was introduced (more or less simultaneously) by Sir Isaac Newton (1643–1727) and Gottfried Wilhelm Leibnitz (1646–1716), we have been using ordinary differential equations as models for the dynamical behavior of systems of all sorts.

Example 7.1 Differential Equation Model for Harmonic Resonator. Dynamical behavior of the one-dimensional damped mass–spring shown schematically in Fig. 7.4 is modeled by the equations

$$m\frac{d^2\xi}{dt^2} = ma = F = -\underbrace{C_{\text{damping}}\frac{d\xi}{dt}}_{\text{damping force}} - \underbrace{C_{\text{spring}}\xi}_{\text{spring force}} + \underbrace{w(t)}_{\text{disturbance}}$$

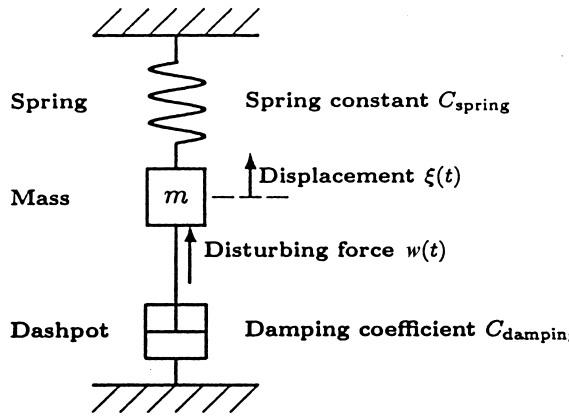


Fig. 7.4 *Schematic model for dynamic system of Example 7.1.*

or

$$\frac{d^2\xi}{dt^2} + \frac{C_{\text{damping}}}{m}\frac{d\xi}{dt} + \frac{C_{\text{spring}}}{m}\xi = \frac{w(t)}{m}, \tag{7.48}$$

where m = mass attached to spring and damper
ξ = upward displacement of mass from its rest position
C_{spring} = spring constant of spring
C_{damping} = damping coefficient of dashpot
$w(t)$ = disturbing force acting on mass

Systems of First-Order Linear Differential Equations The so-called *state space models* for dynamic systems replace higher order differential equations with systems of first-order differential equations. This can be done by defining the first $n-1$ derivatives of an nth-order differential equation as state variables.

Example 7.2 State Space Model for Harmonic Oscillator. Equation 7.48 is a linear second-order ($n = 2$) differential equation. It can be transformed into a system of two linear first-order differential equations with state variables

$$x_1 \stackrel{\text{def}}{=} \xi \quad \text{(mass displacement)}, \qquad x_2 \stackrel{\text{def}}{=} \frac{d\xi}{dt} \quad \text{(mass velocity)},$$

for which

$$\frac{dx_1}{dt} = x_2 \tag{7.49}$$

$$\frac{dx_2}{dt} = \frac{-C_{\text{spring}}}{m}x_1 + \frac{-C_{\text{damping}}}{m}x_2 + \frac{w(t)}{m}. \tag{7.50}$$

Representation in Terms of Vectors and Matrices State space models using systems of linear first-order differential equations can be represented more compactly in terms of a *state vector*, *dynamic coefficient matrix*, and *dynamic disturbance vector*.

Systems of linear first-order differential equations represented in "long-hand" form as

$$\begin{aligned}
\frac{dx_1}{dt} &= f_{11}x_1 + f_{12}x_2 + f_{13}x_3 + \cdots + f_{1n}x_n + w_1,\\
\frac{dx_2}{dt} &= f_{21}x_1 + f_{22}x_2 + f_{23}x_3 + \cdots + f_{2n}x_n + w_2,\\
\frac{dx_3}{dt} &= f_{31}x_1 + f_{32}x_2 + f_{33}x_3 + \cdots + f_{3n}x_n + w_3,\\
&\vdots\\
\frac{dx_n}{dt} &= f_{n1}x_1 + f_{n2}x_2 + f_{n3}x_3 + \cdots + f_{nn}x_n + w_n
\end{aligned}$$

can be represented more compactly in matrix form as

$$\frac{d}{dt}\mathbf{x} = \mathbf{F}\mathbf{x} + \mathbf{w}, \tag{7.51}$$

where the *state vector* $\mathbf{x}$, *dynamic coefficient matrix* $\mathbf{F}$, and *dynamic disturbance vector* $\mathbf{w}$ are given as

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}, \qquad \mathbf{F} = \begin{bmatrix} f_{11} & f_{12} & f_{13} & \cdots & f_{1n} \\ f_{21} & f_{22} & f_{23} & \cdots & f_{2n} \\ f_{31} & f_{32} & f_{33} & \cdots & f_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f_{n1} & f_{n2} & f_{n3} & \cdots & f_{nn} \end{bmatrix}, \qquad \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{bmatrix},$$

respectively.

Example 7.3 Harmonic Resonator Model in Matrix Form. For the system of linear differential equations 7.49 and 7.50,

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \qquad \mathbf{F} = \begin{bmatrix} 0 & 1 \\ \dfrac{-C_{\text{spring}}}{m} & \dfrac{-C_{\text{damping}}}{m} \end{bmatrix}, \qquad \mathbf{w} = \begin{bmatrix} 0 \\ \dfrac{w}{m} \end{bmatrix}.$$

Eigenvalues of Dynamic Coefficient Matrices The coefficient matrix $\mathbf{F}$ of a system of linear differential equations $\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{w}$ has effective units of reciprocal time, or frequency (in units of radians per second). It is perhaps then not surprising that the characteristic values (eigenvalues) of $\mathbf{F}$ are the characteristic frequencies of the dynamic system represented by the differential equations.

The eigenvalues of an $n \times n$ matrix $\mathbf{F}$ are the roots of its *characteristic polynomial*

$$\det(\lambda\mathbf{I} - \mathbf{F}) = \sum_{k=0}^{n} a_n \lambda^n. \tag{7.52}$$

The eigenvalues of $\mathbf{F}$ have the same interpretation as the poles of the related system transfer function, in that the dynamic system $\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{w}$ is stable if and only if the solutions λ of $\det(\lambda\mathbf{I} - \mathbf{F}) = 0$ lie in the left half-plane.

Example 7.4 Damping and Resonant Frequency for Underdamped Harmonic Resonator. For the dynamic coefficient matrix

$$\mathbf{F} = \begin{bmatrix} 0 & 1 \\ \dfrac{-C_{\text{spring}}}{m} & \dfrac{-C_{\text{damping}}}{m} \end{bmatrix}$$

in Example 7.3, the eigenvalues of $\mathbf{F}$ are the roots of its characteristic polynomial

$$\det(\lambda\mathbf{I} - \mathbf{F}) = \det\begin{bmatrix} \lambda & -1 \\ \dfrac{C_{\text{spring}}}{m} & \lambda + \dfrac{C_{\text{damping}}}{m} \end{bmatrix} = \lambda^2 + \frac{C_{\text{damping}}}{m}\lambda + \frac{C_{\text{spring}}}{m},$$

which are

$$\lambda = -\frac{C_{\text{damping}}}{2m} \pm \frac{1}{2m}\sqrt{C_{\text{damping}}^2 - 4mC_{\text{spring}}}.$$

If the discriminant

$$C_{\text{damping}}^2 - 4mC_{\text{spring}} < 0,$$

then the mass–spring system is called underdamped, and its eigenvalues are a complex conjugate pair

$$\lambda = -\frac{1}{\tau_{\text{damping}}} \pm \omega_{\text{resonant}}\mathbf{i}$$

with real part

$$-\frac{1}{\tau_{\text{damping}}} = -\frac{C_{\text{damping}}}{2m}$$

and imaginary part

$$\omega_{\text{resonant}} = \frac{1}{2m}\sqrt{4mC_{\text{spring}} - C_{\text{damping}}^2}.$$

The alternative parameter

$$\tau_{\text{damping}} = \frac{2m}{C_{\text{damping}}}$$

is called the damping time constant of the system, and the other parameter ω_{resonant} is the resonant frequency in units of radians per second.

The dynamic coefficient matrix for the damped harmonic resonator model can also be expressed in terms of the resonant frequency and damping time constant as

$$\mathbf{F}_{\text{harmonic resonator}} = \begin{bmatrix} 0 & 1 \\ -\omega^2 - \dfrac{1}{\tau^2} & -\dfrac{2}{\tau} \end{bmatrix}. \tag{7.53}$$

So long as the damping coefficient $C_{\text{damping}} > 0$, the eigenvalues of this system will lie in the left half-plane. In that case, the damped mass–spring system is guaranteed to be stable.

Matrix Exponential Function The matrix exponential function is defined (in Section B.6.4) as

$$\exp(\mathbf{M}) \stackrel{\text{def}}{=} \sum_{k=0}^{\infty} \frac{1}{k!}\mathbf{M}^k \tag{7.54}$$

for square matrices $\mathbf{M}$. The result is a square matrix of the same dimension as $\mathbf{M}$.

This function has some useful properties:

1. The matrix $\mathbf{N} = \exp(\mathbf{M})$ is always invertible and $\mathbf{N}^{-1} = \exp(-\mathbf{M})$.
2. If $\mathbf{M}$ is *antisymmetric* (i.e., its matrix transpose $\mathbf{M}^{\mathrm{T}} = -\mathbf{M}$), then $\mathbf{N} = \exp(\mathbf{M})$ is an *orthogonal* matrix (i.e., its matrix transpose $\mathbf{N}^T = \mathbf{N}^{-1}$).

3. The eigenvalues of $\mathbf{N} = \exp(\mathbf{M})$ are the (scalar) exponential functions of the eigenvalues of $\mathbf{M}$.
4. If $\mathbf{M}(s)$ is an integrable function of a scalar s, then the derivative

$$\frac{d}{dt}\left(\int^t \mathbf{M}(s)\, ds\right) = \mathbf{M}(t)\left(\int^t \mathbf{M}(s)\, ds\right). \tag{7.55}$$

Forward Solution in Terms of Matrix Exponential Function The property of the matrix exponential function shown in Eq. 7.55 can be used to define the forward solution of Eq. 7.51 as

$$\mathbf{x}(t) = \exp\left(\int_{t_0}^t \mathbf{F}(s)\, ds\right)\left[\mathbf{x}(t_0) + \int_{t_0}^t \exp\left(-\int_{t_0}^s \mathbf{F}(r)\, dr\right)\mathbf{w}(s)\, ds\right], \tag{7.56}$$

where $\mathbf{x}(t_0)$ is the *initial value* of the state vector $\mathbf{x}$ for $t \geq t_0$.

Time Invariant Systems If the dynamic coefficient matrix $\mathbf{F}$ of Eq. 7.51 does not depend on t (time), then the problem is called *time invariant*. In that case,

$$\int_{t_0}^t \mathbf{F}\, ds = (t - t_0)\mathbf{F} \tag{7.57}$$

and the forward solution

$$\mathbf{x}(t) = \exp\big[(t - t_0)\mathbf{F}\big]\left\{\mathbf{x}(t_0) + \int_{t_0}^t \exp\big[-(s - t_0)\mathbf{F}\big]\mathbf{w}(s)\, ds\right\}. \tag{7.58}$$

7.3.1.2 State Models for Discrete Time

Measurements are the outputs of sensors sampled at *discrete times* $\cdots < t_{k-1} < t_k < t_{k+1} < \cdots$. The Kalman filter uses these values to estimate the state of the associated dynamic systems at those discrete times.

If we let $\ldots$, $\mathbf{x}_{k-1}$, $\mathbf{x}_k$, $\mathbf{x}_{k+1}$, $\ldots$ be the corresponding state vector values of a linear dynamic system at those discrete times, then each of these values can be determined from the previous value by using Eq. 7.56 in the form

$$\mathbf{x}_k = \mathbf{\Phi}_{k-1}\mathbf{x}_{k-1} + \mathbf{w}_{k-1}, \tag{7.59}$$

$$\mathbf{\Phi}_{k-1} \stackrel{\text{def}}{=} \exp\left(\int_{t_{k-1}}^{t_k} \mathbf{F}(s)\, ds\right), \tag{7.60}$$

$$\mathbf{w}_{k-1} \stackrel{\text{def}}{=} \mathbf{\Phi}_k \int_{t_{k-1}}^{t_k} \exp\left(-\int_{t_{k-1}}^t \mathbf{F}(s)\, ds\right)\mathbf{w}(t)\, dt. \tag{7.61}$$

Equation 7.59 is the discrete-time dynamic system model corresponding to the continuous-time dynamic system model of Eq. 7.51.

The matrix $\mathbf{\Phi}_{k-1}$ (defined in Eq. 7.60) in the discrete-time model (Eq. 7.59) is called a *state transition matrix* for the dynamic system defined by **F**. Note that $\mathbf{\Phi}$ depends only on **F**, and not on the dynamic disturbance function $\mathbf{w}(t)$.

The noise vectors $\mathbf{w}_k$ are the discrete-time analog of the dynamic disturbance function $\mathbf{w}(t)$. They depend upon **F** and **w**.

Example 7.5 State Transition Matrix for Harmonic Resonator Model. The underdamped harmonic resonator model of Example 7.4 has no time-dependent terms in its coefficient matrix (Eq. 7.53), making it a time-invariant model with state transition matrix

$$\begin{aligned}\mathbf{\Phi} &= \exp(\Delta t\ \mathbf{F}) \\ &= e^{-\Delta t/\tau}\begin{bmatrix} \cos(\omega\ \Delta t) + \dfrac{\sin(\omega\ \Delta t)}{\omega\tau} & \dfrac{\sin(\omega\ \Delta t)}{\omega} \\ -\dfrac{\sin(\omega\ \Delta t)}{\omega\ \tau^2}(1+\omega^2\tau^2) & \cos(\omega\ \Delta t) - \dfrac{\sin(\omega\ \Delta t)}{\omega\ \tau} \end{bmatrix},\end{aligned} \tag{7.62}$$

where $\omega = \omega_{\text{resonant}}$, the resonant frequency
$\tau = \tau_{\text{damping}}$, the damping time constant
$\Delta t =$ discrete-time step.

The eigenvalues of **F** were shown to be $-1/\tau_{\text{damping}} \pm \mathbf{i}\omega_{\text{resonant}}$, so the eigenvalues of $\mathbf{F}\ \Delta t$ will be $-\Delta t/\tau_{\text{damping}} \pm \mathbf{i}\ \Delta t\ \omega_{\text{resonant}}$ and the eigenvalues of $\mathbf{\Phi}$ will be

$$\exp\left(-\frac{\Delta t}{\tau_{\text{damping}}} \pm \mathbf{i}\omega_{\text{resonant}}\ \Delta t\right) = e^{-\Delta t/\tau}[\cos(\omega\ \Delta t) \pm \mathbf{i}\sin(\omega\ \Delta t)].$$

A discrete-time dynamic system will be stable only if the eigenvalues of $\mathbf{\Phi}$ lie inside the unit circle (i.e., $|\lambda_\ell| < 1$).

7.3.2 Covariance Prediction Models

The word *stochastic* derives from the Greek expression for *aiming at a target*, indicating some degree of uncertainty in the dynamics of the projectile between launch and impact. That idea has been formalized mathematically as *stochastic systems theory*, used here for modeling dynamic processes that are not fully predictable.

A *stochastic process* is a model for the evolution over time of a *probability distribution*. For Kalman filtering, this can be viewed as the probability distribution of the true state of a dynamic process. When the underlying probability distribution is Gaussian, the distribution is completely specified by its *mean* and its *covariance*. The mean will be the *estimated value* of the state vector, and the covariance matrix represents the *mean-squared uncertainty* in the estimate. The *time update* equations

of the Kalman filter propagate the estimated value and its associated mean-squared uncertainty forward in time.

7.3.2.1 Zero-Mean White Gaussian Noise Processes

A *zero-mean white Gaussian* noise process in discrete time is a sequence of *independent* samples $\ldots, \mathbf{w}_{k-1}, \mathbf{w}_k, \mathbf{w}_{k+1}, \ldots$ from a normal probability distribution $\mathcal{N}(0, \mathbf{Q}_k)$ with zero mean and known finite covariances $\mathbf{Q}_k$. In Kalman filtering, it is not necessary (but not unusual) that the covariance of all samples be the same.

Sampling is called independent if the expected values of outer products

$$E\langle \mathbf{w}_i \mathbf{w}_j^{\mathrm{T}} \rangle = \begin{cases} 0, & i \neq j, \\ \mathbf{Q}_i, & i = j, \end{cases} \tag{7.63}$$

for all integer indices i and j of the random process.

Zero-mean white Gaussian noise processes are the fundamental random processes used in Kalman filtering. However, it is *not* necessary that all noise sources in the modeled sensors and dynamic systems be zero-mean white Gaussian noise processes. It is only necessary that they can be modeled in terms of such processes (to be addressed in Section 7.5).

7.3.2.2 Gaussian Linear Stochastic Processes in Discrete Time

A linear stochastic processes model in discrete time has the form

$$\mathbf{x}_k = \mathbf{\Phi}_{k-1} \mathbf{x}_{k-1} + \mathbf{w}_{k-1}, \tag{7.64}$$

where $\mathbf{w}_k$ is a zero-mean white Gaussian noise process with known covariances $\mathbf{Q}_k$ and the vector $\mathbf{x}$ represents the state of a dynamic system.

This model for "marginally random" dynamics is quite useful for representing physical systems (e.g., land vehicles, seacraft, aircraft) with zero-mean random disturbances (e.g., wind gusts or sea surface currents). The state transition matrix $\mathbf{\Phi}_k$ represents the known dynamic behavior of the system, and the covariance matrices $\mathbf{Q}_k$ represent the unknown random disturbances. Together, they model the propagation of the necessary statistical properties of the state variable $\mathbf{x}$.

Example 7.6 Harmonic Resonator with White Acceleration Disturbance Noise. If the disturbance acting on the harmonic resonator of Examples 7.1–7.5 were zero-mean white acceleration noise with variance $\sigma^2_{\text{disturbance}}$, then its disturbance noise covariance matrix would have the form

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 \\ 0 & \sigma^2_{\text{disturbance}} \end{bmatrix}. \tag{7.65}$$

7.3.2.3 Noise Distribution Matrix

A common noise source can disturb more than one independent component of the state vector representing a dynamic system.

Forces applied to a rigid body, for example, can affect rotational dynamics as well as translational dynamics. This sort of coupling of common disturbance noise sources into different components of the state dynamics can be represented by using a *noise distribution matrix* $\mathbf{G}$ in the form

$$\frac{d}{dt}\mathbf{x} = \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{w}(t), \tag{7.66}$$

where the components of $\mathbf{w}(t)$ are the common disturbance noise sources and the matrix $\mathbf{G}$ represents how these disturbances are distributed among the state vector components.

The covariance of state vector disturbance noise will then have the form $\mathbf{G}\mathbf{Q}_w\mathbf{G}^{\mathrm{T}}$, where $\mathbf{Q}_w$ is the covariance matrix for the white-noise process $\mathbf{w}(t)$.

The analogous model in discrete time has the form

$$\mathbf{x}_k = \mathbf{\Phi}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{w}_{k-1}, \tag{7.67}$$

where $\{\mathbf{w}_k\}$ is a zero-mean white-noise process in discrete time.

In either case (i.e., continuous or discrete time), it is possible to use the noise distribution matrix for noise scaling, as well, so that the components of $\mathbf{w}_k$ can be independent, uncorrelated unit normal variates and the noise covariance matrix $\mathbf{Q}_w = \mathbf{I}$, the identity matrix.

7.3.2.4 Predictor Equations The linear stochastic process model parameters $\mathbf{\Phi}$ and $\mathbf{Q}$ can be used to calculate how the discrete-time process variables $\boldsymbol{\mu}$ and $\mathbf{P}$ evolve over time.

Using Eq. 7.64 and taking expected values,

$$\begin{aligned}
\boldsymbol{\mu}_k &\stackrel{\text{def}}{=} E\langle \mathbf{x}_k \rangle \\
&= \mathbf{\Phi}_{k-1} E\langle \mathbf{x}_{k-1} \rangle + E\langle \mathbf{w}_{k-1} \rangle \\
&= \mathbf{\Phi}_{k-1}\boldsymbol{\mu}_{k-1} + 0 \\
&= \mathbf{\Phi}_{k-1}\boldsymbol{\mu}_{k-1},
\end{aligned} \tag{7.68}$$

$$\begin{aligned}
\mathbf{P}_k &\stackrel{\text{def}}{=} E\langle (\mathbf{x}_k - \boldsymbol{\mu})(\mathbf{x}_k - \boldsymbol{\mu}_k)^{\mathrm{T}} \rangle \\
&= \mathbf{\Phi}_{k-1} E\langle (\mathbf{x}_{k-1} - \boldsymbol{\mu}_{k-1})(\mathbf{x}_{k-1} - \boldsymbol{\mu}_{k-1})^{\mathrm{T}} \rangle \mathbf{\Phi}_{k-1}^{\mathrm{T}} \\
&\quad + E\langle \mathbf{w}_{k-1}\mathbf{w}_{k-1}^{\mathrm{T}} \rangle + \text{terms with expected value } = 0 \\
&= \mathbf{\Phi}_{k-1}\mathbf{P}_{k-1}\mathbf{\Phi}_{k-1}^{\mathrm{T}} + \mathbf{Q}_{k-1}.
\end{aligned} \tag{7.69}$$

7.4 SUMMARY OF KALMAN FILTER EQUATIONS

7.4.1 Essential Equations

The complete equations for the Kalman filter are summarized in Table 7.2.

TABLE 7.2 Essential Kalman Filter Equations

Predictor (Time Updates)	
Predicted state vector:	
$\hat{\mathbf{x}}_k(-) = \mathbf{\Phi}_k \hat{\mathbf{x}}_{k-1}(+)$	Eq. 7.68
Predicted covariance matrix:	
$\mathbf{P}_k(-) = \mathbf{\Phi}_k \mathbf{P}_{k-1}(+)\mathbf{\Phi}_k^{\mathrm{T}} + \mathbf{Q}_{k-1}$	Eq. 7.69
Corrector (Measurement Updates)	
Kalman gain:	
$\overline{\mathbf{K}}_k = \mathbf{P}_k(-)\mathbf{H}_k^{\mathrm{T}}(\mathbf{H}_k\mathbf{P}_k(-)\mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k)^{-1}$	Eq. 7.46
Corrected state estimate:	
$\hat{\mathbf{x}}_k(+) = \hat{\mathbf{x}}_k(-) + \overline{\mathbf{K}}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k(-))$	Eq. 7.1
Corrected covariance matrix:	
$\mathbf{P}_k(+) = \mathbf{P}_k(-) - \overline{\mathbf{K}}_k\mathbf{H}_k\mathbf{P}_k(-)$	Eq. 7.47

7.4.2 Common Terminology

The symbols used in Table 7.2 for the variables and parameters of the Kalman filter are essentially those used in the original paper by Kalman [71], and this notation is fairly common in the literature.

The following are some names commonly used for the symbols in Table 7.2:

$\mathbf{H}$ is the *measurement sensitivity matrix* or *observation matrix*.

$\mathbf{H}\hat{\mathbf{x}}_k(-)$ is the *predicted measurement*.

$\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}_k(-)$, the difference between the measurement vector and the predicted measurement, is the *innovations vector.*

$\overline{\mathbf{K}}$ is the *Kalman gain*.

$\mathbf{P}_k(-)$ is the *predicted* or *a priori* value of estimation covariance.

$\mathbf{P}_k(+)$ is the *corrected* or *a posteriori* value of estimation covariance.

$\mathbf{Q}_k$ is the covariance of dynamic disturbance noise.

$\mathbf{R}$ is the covariance of *sensor noise* or *measurement uncertainty.*

$\hat{\mathbf{x}}_k(-)$ is the *predicted* or *a priori* value of the estimated state vector.

$\hat{\mathbf{x}}_k(+)$ is the *corrected* or *a posteriori* value of the estimated state vector.

$\mathbf{z}$ is the *measurement vector* or *observation vector.*

7.4.3 Data Flow Diagrams

The matrix-level data flow of the Kalman filter implementation for a time-varying problem is diagrammed in Fig. 7.5, with the inputs shown on the left, the outputs (corrected estimates) on the right, and the symbol z^{-1} representing the unit delay operator.

The dashed lines in the figure enclose two computation loops. The top loop is the estimation loop, with the feedback gain (Kalman gain) coming from the bottom

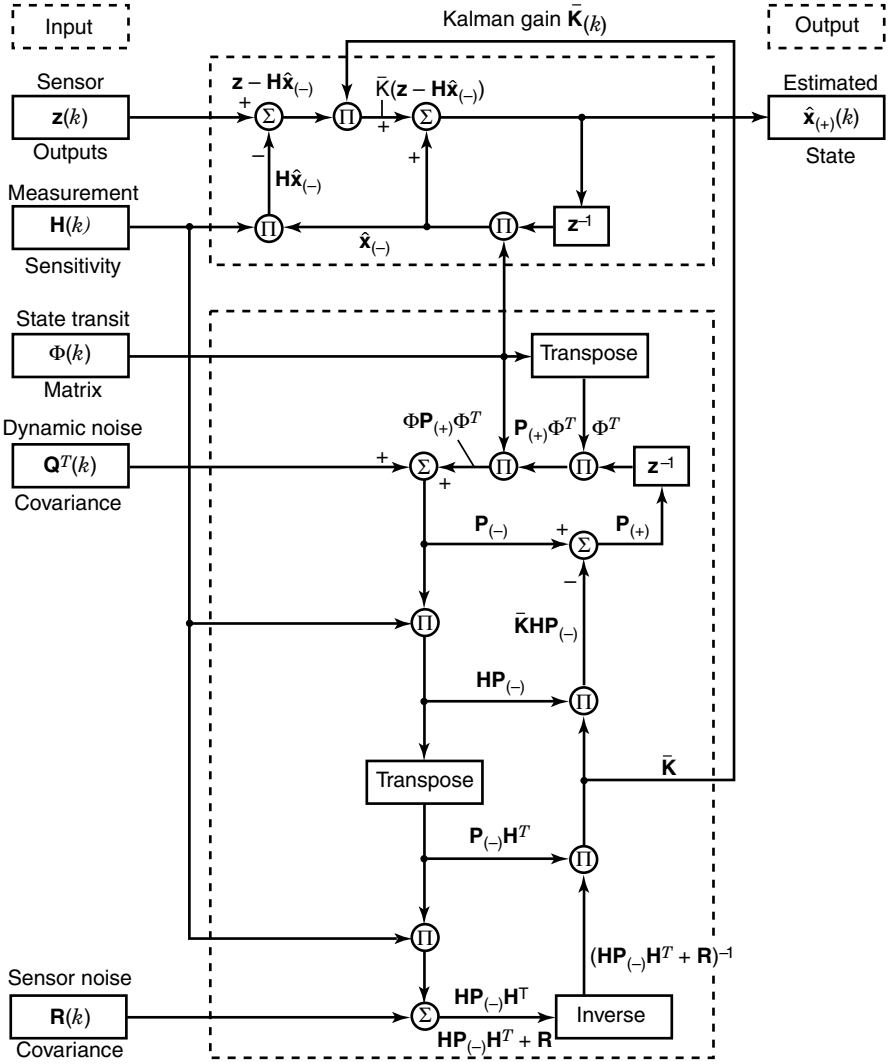


Fig. 7.5 *Kalman filter data array flows for time-varying system.*

loop. The bottom loop implements the Riccati equation solution used to calculate the Kalman gain. This bottom loop runs "open loop," in that there is no feedback mechanism to stabilize it in the presence of roundoff errors. Numerical instability problems with the Riccati equation propagation loop were discovered soon after the introduction of the Kalman filter.

7.5 ACCOMMODATING CORRELATED NOISE

The fundamental noise processes in the basic Kalman filter model are zero-mean white Gaussian noise processes:

$\{\mathbf{w}_k\}$, called *dynamic disturbance*, *plant noise*, or *process noise* and
$\{\mathbf{v}_k\}$, called *sensor noise*, *measurement noise*, or *observation noise*.

However, it is not necessary that the physical noise processes of the real-world application—either the dynamic disturbance or the sensor noise—be uncorrelated in order to apply Kalman filtering.

For applications with uncorrelated noise sources, it is only necessary to model the correlated noise process $\boldsymbol{\xi}_k = \mathbf{v}_k, \mathbf{w}_k$ using a linear stochastic system model of the sort

$$\boldsymbol{\xi}_k = \boldsymbol{\Phi}_{k-1}\boldsymbol{\xi}_{k-1} + \mathbf{y}_{k-1},$$

where $\{\mathbf{y}_k\}$ is a zero-mean white Gaussian noise process, and then augment the state vector by appending the new variable $\boldsymbol{\xi}$,

$$\mathbf{x}_{\text{augmented}} = \begin{bmatrix} \mathbf{x}_{\text{original}} \\ \boldsymbol{\xi} \end{bmatrix} \tag{7.70}$$

and modify the parameter matrices $\boldsymbol{\Phi}$, $\mathbf{Q}$, and $\mathbf{H}$ accordingly.

7.5.1 Correlated Noise Models

7.5.1.1 Autocovariance Functions Correlation of a random sequence $\{\boldsymbol{\xi}_k\}$ is characterized by its discrete-time *autocovariance function* $\mathbf{P}_{\{\boldsymbol{\xi}_k\}}[\Delta k]$, a function of the delay index Δk defined as

$$\mathbf{P}_{\{\boldsymbol{\xi}_k\}}[\Delta k] \stackrel{\text{def}}{=} E_k\langle(\boldsymbol{\xi}_k - \boldsymbol{\mu}_\xi)(\boldsymbol{\xi}_{k+\Delta k} - \boldsymbol{\mu}_\xi)^{\mathrm{T}}\rangle, \tag{7.71}$$

where $\boldsymbol{\mu}_\xi$ is the mean value of the random sequence $\{\boldsymbol{\xi}_k\}$.

For *white*-noise processes,

$$\mathbf{P}[\Delta k] = \begin{cases} 0, & \Delta k \neq 0, \\ \mathbf{C}, & \Delta k = 0, \end{cases} \tag{7.72}$$

where $\mathbf{C}$ is the covariance of the process.

7.5.1.2 Random Walks Random walks, also called *Wiener processes*, are cumulative sums of white-noise processes $\{\mathbf{w}_k\}$:

$$\boldsymbol{\xi}_k = \boldsymbol{\xi}_{k-1} + \mathbf{w}_{k-1}, \tag{7.73}$$

a stochastic process model with state transition matrix $\boldsymbol{\Phi} = \mathbf{I}$, an identity matrix.

Random walks are notoriously unstable, in the sense that the covariance of the variate $\boldsymbol{\xi}_k$ grows linearly with k and without bound as $k \to \infty$. In general, if any of the eigenvalues of a state transition matrix fall on or outside the unit circle in the complex plane (as they all do for identity matrices), the variate of the stochastic process can fail to have a finite steady-state covariance matrix. However, the covariance of *uncertainty* in the *estimated* system state vector can still converge to a finite steady-state value, even if the process itself is unstable. Methods for determining whether estimation uncertainties will diverge are presented in Chapter 8.

7.5.1.3 Exponentially Correlated Noise Exponentially correlated random processes have finite, constant steady-state covariances. A scalar exponentially random process $\{\xi_k\}$ has a model of the sort

$$\xi_k = \varepsilon\xi_{k-1} + w_{k-1}, \tag{7.74}$$

$$0 < \varepsilon < 1,$$

$$\varepsilon = e^{-\Delta t/\tau}, \tag{7.75}$$

where Δt is the time period between samples and τ is the exponential decay time constant of the process. The steady-state variance σ^2 of such a process is the solution to its steady-state variance equation,

$$\sigma^2 = \varepsilon^2\sigma^2 + Q \tag{7.76}$$

$$= \frac{Q}{1-\varepsilon^2} \tag{7.77}$$

$$= \frac{Q}{1-e^{-2\Delta t/\tau}}, \tag{7.78}$$

where Q is the variance of the scalar zero-mean white-noise process $\{w_k\}$.

The autocovariance sequence of an exponentially correlated random process in discrete time has the general form

$$P[\Delta k] = \sigma^2 \exp(-|\Delta k|/N_c), \tag{7.79}$$

which falls off exponentially on either side of its peak value σ^2 (the process variance) at $\Delta k = 0$. The parameter N_c is called the *correlation number* of the process, where $N_c = \tau/\Delta t$ for correlation time τ and sample interval Δt.

7.5.1.4 Harmonic Noise Harmonic noise includes identifiable frequency components, such as those from AC power or from mechanical or electrical resonances. A stochastic process model for such sources has already been developed in the examples of this chapter.

7.5.1.5 SA Noise Autocorrelation A clock dithering algorithm is described in U.S. Patent 4,646,032 [134], including a parametric model of the autocorrelation function (autocovariance function divided by variance) of the resulting timing errors. SA was suspended on May 1, 2000, but can be turned on again. Knowledge of the dithering algorithm does not necessarily give the user any advantage, but there is at least a suspicion that this may be the algorithm used for SA dithering of the individual GPS satellite time references. Its theoretical autocorrelation function is plotted in Fig. 7.6 along with an exponential correlation curve. The two are scaled to coincide at the autocorrelation coefficient value of $1/e \approx 0.36787944\ldots$, the argument at which correlation time is defined. Unlike exponentially correlated noise, this source has greater short-term correlation and less long-term correlation.

The correlation time of SA errors determined from GPS signal analysis is on the order of 10^2–10^3 s. It is possible that the actual correlation time is variable, which might explain the range of values reported in the literature.

Although this is not an exponential autocorrelation function, it could perhaps be modeled as such.

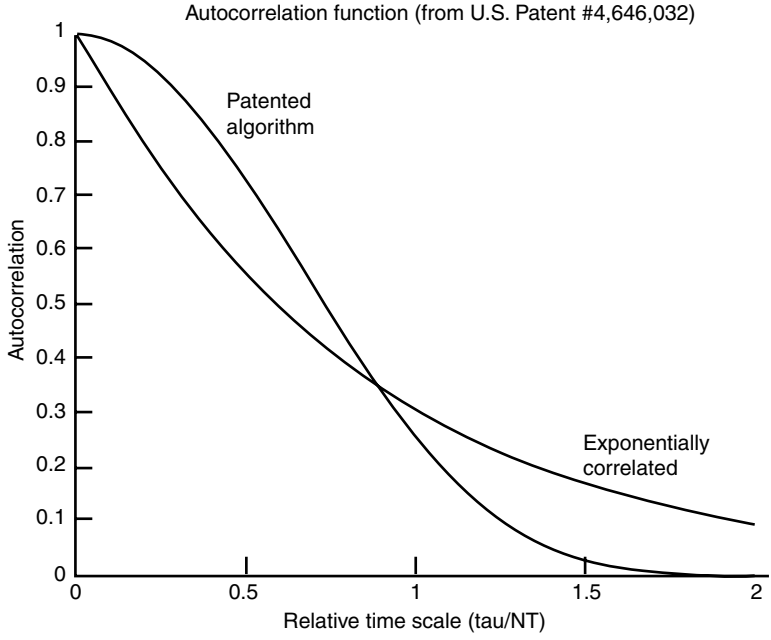


Fig. 7.6 *Autocorrelation function for pseudonoise algorithm.*

7.5.1.6 Slow Variables SA timing errors (if present) are but one of a number of slowly varying error sources in GPS/INS integration. Slow variables may also include many of the calibration parameters of the inertial sensors, which can be responding to temperature variations or other unknown but slowly changing influences. Like SA errors, these other slow variations of these variables can often be tracked and compensated by combining the INS navigation estimates with the GPS-derived estimates. What is different about the calibration parameters is that they are involved nonlinearly in the INS system model.

7.5.2 Empirical Noise Modeling

Noise models used in Kalman filtering should be reasonably faithful representations of the true noise sources. Sensor noise can often be measured directly and used in the design of an appropriate noise model. Dynamic process noise is not always so accessible, and its models must often be inferred from indirect measurements.

7.5.2.1 Spectral Characterization Spectrum analyzers and spectrum analysis software make it relatively easy to calculate the power spectral density of sampled noise data, and the results are useful for characterizing the type of noise and identifying likely noise models.

The resulting noise models can then be simulated using pseudorandom sequences, and the power spectral densities of the simulated noise can be compared to that of the sampled noise to verify the model.

The power spectral density of white noise is constant across the spectrum, and each successive integral changes its slope by $-20\,\mathrm{dB/decade}$ of frequency, as illustrated in Fig. 7.7.

7.5.2.2 Shaping Filters The spectrum of white noise is flat, and the amplitude spectrum of the output of a filter with white-noise input will have the shape of the amplitude transfer function of the filter, as illustrated in Fig. 7.8. Therefore, any noise spectrum can be approximated by white noise passed through a *shaping filter* to yield the desired shape. All correlated noise models for Kalman filters can be implemented by shaping filters.

7.5.3 State Vector Augmentation

7.5.3.1 Correlated Dynamic Disturbance Noise A model for a linear stochastic process model in discrete time with uncorrelated and correlated disturbance noise has the form

$$\mathbf{x}_k = \mathbf{\Phi}_{\mathrm{x},k-1}\mathbf{x}_{k-1} + \mathbf{G}_{wx,k-1}\mathbf{w}_{x,k-1} + \mathbf{D}_{\xi x,k-1}\boldsymbol{\xi}_{k-1}, \tag{7.80}$$

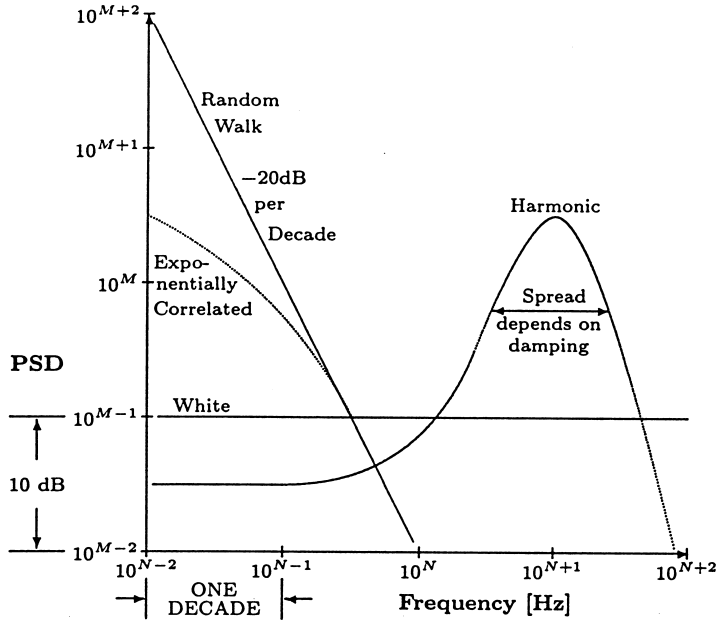


Fig. 7.7 *Spectral properties of some common noise types.*

where $\mathbf{w}_{k-1}$ = zero-mean white (i.e., uncorrelated) disturbance noise
$\mathbf{G}_{wx,k-1}$ = white-noise distribution matrix
$\boldsymbol{\xi}_{k-1}$ = zero-mean correlated disturbance noise
$\mathbf{D}_{\xi x,k-1}$ = correlated noise distribution matrix

If the correlated dynamic disturbance noise can be modeled as yet another linear stochastic process

$$\boldsymbol{\xi}_k = \boldsymbol{\Phi}_{\xi,k-1}\boldsymbol{\xi}_{k-1} + \mathbf{G}_{w\xi,k-1}\mathbf{w}_{\xi,k-1} \tag{7.81}$$

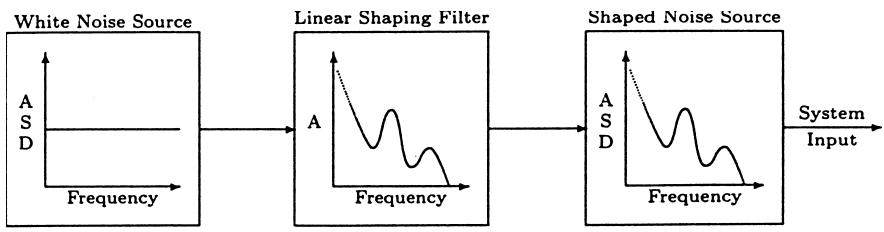


Fig. 7.8 *Putting white noise through shaping filters.*

with only zero-mean white-noise inputs $\{\mathbf{w}_{\mathrm{u},k}\}$, then the augmented state vector

$$\mathbf{x}_{\mathrm{aug},k} \stackrel{\mathrm{def}}{=} \begin{bmatrix} \mathbf{x}_k \\ \boldsymbol{\xi}_k \end{bmatrix} \tag{7.82}$$

has a stochastic process model

$$\begin{aligned} \mathbf{x}_{\mathrm{aug},k} = & \begin{bmatrix} \boldsymbol{\Phi}_{\mathrm{x},k-1} & \mathbf{D}_{\xi x,k-1} \\ 0 & \boldsymbol{\Phi}_{\xi,k-1} \end{bmatrix} \mathbf{x}_{\mathrm{aug},k-1} \\ & + \begin{bmatrix} \mathbf{G}_{wx,k-1} & 0 \\ 0 & \mathbf{G}_{w\xi,k-1} \end{bmatrix} \begin{bmatrix} \mathbf{w}_{\mathrm{x},k-1} \mathbf{w}_{\xi,k-1} \end{bmatrix} \end{aligned} \tag{7.83}$$

having only uncorrelated disturbance noise with covariance

$$\mathbf{Q}_{\mathrm{aug},k-1} = \begin{bmatrix} \mathbf{Q}_{wx,k-1} & 0 \\ 0 & \mathbf{Q}_{w\xi,k-1} \end{bmatrix}. \tag{7.84}$$

The new measurement sensitivity matrix for this augmented state vector will have the block form

$$\mathbf{H}_{\mathrm{aug},k} = [\, \mathbf{H}_k \quad 0 \,]. \tag{7.85}$$

The augmenting block is zero in this case because the uncorrelated noise source is dynamic disturbance noise, not sensor noise.

7.5.3.2 Correlated Noise in Continuous Time There is an analogous procedure for state augmentation using continuous-time models. If $\boldsymbol{\xi}(t)$ is a correlated noise source defined by a model of the sort

$$\frac{d}{dt}\boldsymbol{\xi} = \mathbf{F}_{\xi}\boldsymbol{\xi} + \mathbf{w}_{\xi} \tag{7.86}$$

for $\mathbf{w}_{\xi}(t)$ a white-noise source, then any stochastic process model of the sort

$$\frac{d}{dt}\mathbf{x} = \mathbf{F}_x\mathbf{x} + \mathbf{w}_x(t) + \boldsymbol{\xi}(t) \tag{7.87}$$

with this correlated noise source can also be modeled by the augmented state vector

$$\mathbf{x}_{\mathrm{aug}} \stackrel{\mathrm{def}}{=} \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\xi} \end{bmatrix} \tag{7.88}$$

as

$$\frac{d}{dt}\mathbf{x}_{\text{aug}} = \begin{bmatrix} \mathbf{F}_x & \mathbf{I} \\ 0 & \mathbf{F}_\xi \end{bmatrix}\mathbf{x}_{\text{aug}} + \begin{bmatrix} \mathbf{w}_x \\ \mathbf{w}_\xi \end{bmatrix} \tag{7.89}$$

with only uncorrelated disturbance noise.

7.5.3.3 Correlated Sensor Noise The same sort of state augmentation can be done for correlated sensor noise $\{\boldsymbol{\xi}_k\}$,

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{A}_k\mathbf{v}_k + \mathbf{B}_k\boldsymbol{\xi}_k, \tag{7.90}$$

with the same type of model for the correlated noise (Eq. 7.81) and using the same augmented state vector (Eq. 7.82), but now with a different augmented state transition matrix

$$\boldsymbol{\Phi}_{\text{aug},k-1} = \begin{bmatrix} \boldsymbol{\Phi}_{x,k-1} & 0 \\ 0 & \boldsymbol{\Phi}_{\xi,k-1} \end{bmatrix} \tag{7.91}$$

and augmented measurement sensitivity matrix

$$\mathbf{H}_{\text{aug},k} = [\,\mathbf{H}_k \quad \mathbf{B}_k\,]. \tag{7.92}$$

7.6 NONLINEAR AND ADAPTIVE IMPLEMENTATIONS

Although the Kalman filter is defined for linear dynamic systems with linear sensors, it has been applied more often than not to real-world applications without truly linear dynamics or sensors—and usually with remarkably great success.

7.6.1 Nonlinear Dynamics

State dynamics for nonlinear systems are assumed to be definable in the functional form

$$\frac{d}{dt}\mathbf{x} = \mathbf{f}(\mathbf{x}, t) + \mathbf{w}(t), \tag{7.93}$$

where the function $\mathbf{f}$ is assumed to be differentiable with Jacobian matrix

$$\mathbf{F}(\mathbf{x}, t) \stackrel{\text{def}}{=} \underbrace{\left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}(t)}}_{\text{extended}} \quad \text{or} \quad \underbrace{\left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}^{\text{nom}}(t)}}_{\text{linearized}}, \tag{7.94}$$

where the extended Kalman filter uses the estimated trajectory for evaluating the Jacobian, and linearized Kalman filtering uses a nominal trajectory.

7.6.1.1 Nonlinear Dynamics with Control

In applications with control variables $\mathbf{u}(t)$, Eq. 7.93 can also be expressed in the form

$$\frac{d}{dt}\mathbf{x} = \mathbf{f}(\mathbf{x}, \mathbf{u}(t), t) + \mathbf{w}(t), \tag{7.95}$$

in which case the control vector $\mathbf{u}$ may also appear in the Jacobian matrix $\mathbf{F}$.

7.6.1.2 Propagating Estimates

The estimate $\hat{\mathbf{x}}$ is propagated by solving the differential equation

$$\frac{d}{dt}\hat{\mathbf{x}} = \mathbf{f}(\hat{\mathbf{x}}, t), \tag{7.96}$$

using whatever means necessary (e.g., Runge–Kutta integration). The solution is called the *trajectory* of the estimate.

7.6.1.3 Propagating Covariances

The covariance matrix for nonlinear systems is also propagated over time as the solution to the matrix differential equation

$$\frac{d}{dt}\mathbf{P}(t) = \mathbf{F}(\mathbf{x}(t), t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}^{\mathrm{T}}(\mathbf{x}(t), t) + \mathbf{Q}(t), \tag{7.97}$$

where the values of $\mathbf{F}(t)$ from Eq. 7.94 must be calculated along a trajectory $\mathbf{x}(t)$. This trajectory can be the solution for the estimated value $\hat{\mathbf{x}}$ calculated using the Kalman filter and Eq. 7.96 (for the extended Kalman filter) or along any "nominal" trajectory ("linearized" Kalman filtering).

7.6.2 Nonlinear Sensors

Nonlinear Kalman filtering can accommodate sensors that are not truly linear but can at least be represented in the functional form

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k, \tag{7.98}$$

where $\mathbf{h}$ is a smoothly differentiable function of $\mathbf{x}$. For example, even linear sensors with nonzero biases (offsets) $\mathbf{b}_{\text{sensor}}$ will have sensor models of the sort

$$\mathbf{h}(\mathbf{x}) = \mathbf{H}\mathbf{x} + \mathbf{b}_{\text{sensor}}, \tag{7.99}$$

in which case the Jacobian matrix

$$\frac{\partial \mathbf{h}}{\partial \mathbf{x}} = \mathbf{H}. \tag{7.100}$$

7.6.2.1 Predicted Sensor Outputs The predicted value of nonlinear sensor outputs uses the full nonlinear function applied to the estimated state vector:

$$\hat{\mathbf{z}}_k = \mathbf{h}_k(\hat{\mathbf{x}}_k). \tag{7.101}$$

7.6.2.2 Calculating Kalman Gains The value of the measurement sensitivity matrix $\mathbf{H}$ used in calculating Kalman gains is evaluated as a Jacobian matrix

$$\mathbf{H}_k = \underbrace{\left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}}}_{\text{extended}} \quad \text{or} \quad \underbrace{\left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^{\text{nom}}}}_{\text{linearized}}, \tag{7.102}$$

where the first value (used for extended Kalman filtering) uses the estimated trajectory for evaluation of partial derivatives, and the second value uses a nominal trajectory (for linearized Kalman filtering).

7.6.3 Linearized Kalman Filter

Perhaps the simplest approach to Kalman filtering for nonlinear systems uses linearization of the system model about a nominal trajectory. This approach is necessary for preliminary analysis of systems during the system design phase, when there may be several potential trajectories defined by different mission scenarios. The essential implementation equations for this case are summarized in Table 7.3.

7.6.4 Extended Kalman Filtering

Extended Kalman filtering is nonlinear Kalman filtering with all Jacobian matrices (i.e., $\mathbf{H}$ and/or $\mathbf{F}$) evaluated at $\hat{\mathbf{x}}$, the estimated state. The essential extended Kalman filter equations are summarized in Table 7.4, the major differences from the conventional Kalman filter equations of Table 7.2 being

1. integration of the nonlinear integrand $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ to predict $\hat{\mathbf{x}}_k(-)$,
2. use of the nonlinear function $\mathbf{h}_k(\hat{\mathbf{x}}_k(-))$ in measurement prediction,
3. use of the Jacobian matrix of the dynamic model function $\mathbf{f}$ as the dynamic coefficient matrix $\mathbf{F}$ in the propagation of the covariance matrix, and
4. use of the Jacobian matrix of the measurement function $\mathbf{h}$ as the measurement sensitivity matrix $\mathbf{H}$ in the covariance correction and Kalman gain equations.

TABLE 7.3 Linearized Kalman Filter Equations

Predictor (Time Updates)	
Predicted state vector:	
$\hat{\mathbf{x}}_k(-) = \hat{\mathbf{x}}_{k-1}(+) + \int_{t_{k-1}}^{t_k} \mathbf{f}(\hat{x}, t)\, dt$	Eq. 7.96
Predicted covariance matrix:	
$\dot{\mathbf{P}} = \mathbf{FP} + \mathbf{PF}^{\mathrm{T}} + \mathbf{Q}(t)$	Eq. 7.97
$\mathbf{F} = \left.\dfrac{\partial f}{\partial x}\right\|_{\mathbf{x}^{\mathrm{nom}}(t)}$	Eq. 7.94
or	
$\mathbf{P}_k(-) = \mathbf{\Phi}_k \mathbf{P}_{k-1}(+) \mathbf{\Phi}_k^{\mathrm{T}} + \mathbf{Q}_{k-1}$	Eq. 7.69
Corrector (Measurement Updates)	
Kalman gain:	
$\overline{\mathbf{K}}_k = \mathbf{P}_k(-)\mathbf{H}_k^{\mathrm{T}}(\mathbf{H}_k \mathbf{P}_k(-) \mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k)^{-1}$	Eq. 7.46
$\mathbf{H}_k = \left.\dfrac{\partial h}{\partial x}\right\|_{x = x^{\mathrm{nom}}}$	Eq. 7.102
Corrected state estimate:	
$\hat{\mathbf{x}}_k(+) = \hat{\mathbf{x}}_k(-) + \overline{\mathbf{K}}_k[\mathbf{z}_k - \mathbf{h}_k(\hat{\mathbf{x}}_k(-))]$	Eqs. 7.1, 7.101
Corrected covariance matrix:	
$\mathbf{P}_k(+) = \mathbf{P}_k(-) - \overline{\mathbf{K}}_k \mathbf{H}_k \mathbf{P}_k(-)$	Eqs. 7.47, 7.102

This approach is due to Stanley F. Schmidt, and it has been used successfully in an enormous number of nonlinear applications.

7.6.5 Adaptive Kalman Filtering

In adaptive Kalman filtering, nonlinearities in the model arise from making parameters of the model into functions of state variables. For example, the time constant τ of an exponentially correlated process

$$x_k = \exp\left(-\frac{\Delta t}{\tau}\right) x_{k-1} + w_k \tag{7.103}$$

may be unknown or slowly time varying, in which case it can be made part of the augmented state vector

$$\mathbf{x}_{\mathrm{aug}} = \begin{bmatrix} x \\ \tau \end{bmatrix}$$

TABLE 7.4 Extended Kalman Filter Equations

Predictor (Time Updates)	
Predicted state vector:	
$\hat{\mathbf{x}}_k(-) = \hat{\mathbf{x}}_{k-1}(+) + \int_{t_{k-1}}^{t_k} \mathbf{f}(\hat{\mathbf{x}}, t)\, dt$	Eq. 7.96
Predicted covariance matrix:	
$\dot{\mathbf{P}} = \mathbf{FP} + \mathbf{PF}^{\mathsf{T}} + \mathbf{Q}(t)$	Eq. 7.97
$\mathbf{F} = \left.\dfrac{\partial f}{\partial x}\right\|_{\hat{\mathbf{x}}(t)}$	Eq. 7.94
or	
$\mathbf{P}_k(-) = \mathbf{\Phi}_k \mathbf{P}_{k-1}(+) \mathbf{\Phi}_k^{\mathsf{T}} + \mathbf{Q}_{k-1}$	Eq. 7.69
$\dot{\mathbf{\Phi}} = \mathbf{F\Phi}$	
$\mathbf{\Phi}(t_{k-1}) = \mathbf{I}$	
Corrector (Measurement Updates)	
Kalman gain:	
$\overline{\mathbf{K}}_k = \mathbf{P}_k(-)\mathbf{H}_k^{\mathsf{T}}(\mathbf{H}_k \mathbf{P}_k(-) \mathbf{H}_k^{\mathsf{T}} + \mathbf{R}_k)^{-1}$	Eqs. 7.46, 7.102
Corrected state estimate:	
$\hat{\mathbf{x}}_k(+) = \hat{\mathbf{x}}_k(-) + \overline{\mathbf{K}}_k[\mathbf{z}_k - \mathbf{h}_k(\hat{\mathbf{x}}_k(-))]$	Eqs. 7.1, 7.101
Corrected covariance matrix:	
$\mathbf{P}_k(+) = \mathbf{P}_k(-) - \overline{\mathbf{K}}_k \mathbf{H}_k \mathbf{P}_k(-)$	Eqs. 7.47, 7.102

with dynamic model Jacobian

$$\mathbf{F} = \begin{bmatrix} \exp\left(-\dfrac{\Delta t}{\hat{\tau}}\right) & \dfrac{\Delta t\ \exp(-\Delta t/\tau)\hat{x}}{\hat{\tau}^2} \\ 0 & \exp(-\Delta t/\tau^*) \end{bmatrix},$$

where $\tau^* \gg \tau$ is the exponential time constant of the variations in τ.

Example 7.7 Consider the problem of tracking the phase components of a damped harmonic oscillator with slowly time-varying resonant frequency and damping time constant. The state variables for this nonlinear dynamic system are

- x_1, the in-phase component of the oscillator output signal (i.e, the only observable component);
- x_2, the quadrature-phase component of the signal;
- x_3, the damping time constant of the oscillator (nominally 5 s); and
- x_4, the frequency of oscillator (nominally 2 π rad/s, or 1 Hz).

The dynamic Jacobian matrix will be

$$\mathbf{F} = \begin{bmatrix} -1/x_3 & x_4 & x_1/x_3^2 & x_2 \\ -x_4 & -1/x_3 & x_2/x_3^2 & -x_1 \\ 0 & 0 & -1/\tau_\tau & 0 \\ 0 & 0 & 0 & -1/\tau_\omega \end{bmatrix},$$

where τ_τ is the correlation time for the time-varying oscillator damping time constant, and τ_ω is the correlation time for the time-varying resonant frequency of the oscillator.

If only the in-phase component or the oscillator output can be sensed, then the measurement sensitivity matrix will have the form

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}.$$

Figure 7.9 is a sample output of the MATLAB m-file *osc_ekf.m* on the accompanying diskette, which implements this extended Kalman filter. Note that it tracks the phase, amplitude, frequency, and damping of the oscillator.

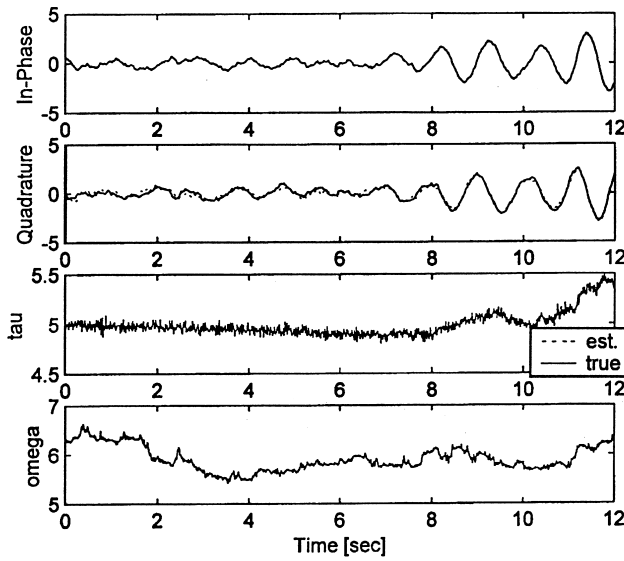


Fig. 7.9 *Extended Kalman filter tracking simulated time-varying oscillator.*

The unknown or time-varying parameters can also be in the measurement model. For example, a sensor output with time-varying scale factor S and bias b can be modeled using an augmented state vector of the sort

$$z = Sx + b,$$
$$\mathbf{x}_{\text{aug}} = \begin{bmatrix} x \\ S \\ b \end{bmatrix}$$

and measurement sensitivity matrix

$$\mathbf{H} = \begin{bmatrix} \hat{S} & \hat{x} & 1 \end{bmatrix}.$$

7.7 KALMAN–BUCY FILTER

7.7.1 Basic Equations

The analog of the Kalman filter in continuous time is the Kalman–Bucy filter, developed jointly by Rudolf Kalman and Richard Bucy [72]. Stochastic process models in continuous time are defined by *stochastic differential equations* of the sort

$$\frac{d}{dt}\mathbf{x} = \mathbf{F}\mathbf{x} + \mathbf{w}(t), \tag{7.104}$$

where $\mathbf{w}(t)$ is a zero-mean white noise-process in continuous time. The mathematical foundations for stochastic differential equations require a different type of calculus (called the *stochastic calculus*[3] or *Itô calculus*), because white-noise processes are not integrable functions in the ordinary (Riemann) calculus. However, the resulting matrix Riccati differential equation for propagation of the covariance matrix

$$\frac{d}{dt}\mathbf{P} = \mathbf{F}\mathbf{P} + \mathbf{P}\mathbf{F}^{\mathrm{T}} + \mathbf{Q} \tag{7.105}$$

is integrable in the ordinary calculus, although the units of this $\mathbf{Q}$ matrix will be different from those of $\mathbf{Q}$ in the Kalman filter.

The analogous differential equation for propagation of the estimate has the form

$$\frac{d}{dt}\hat{\mathbf{x}} = \mathbf{F}\hat{\mathbf{x}} + \underbrace{\mathbf{P}\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}}_{\overline{\mathbf{K}}_{\mathrm{KB}}}(\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}), \tag{7.106}$$

[3] Jazwinski [67] uses the stochastic calculus to develop the Kalman–Bucy filter.

with *Kalman–Bucy gain* $\overline{\mathbf{K}}_{\mathrm{KB}} = \mathbf{P}\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}$, which is quite different from the Kalman gain. The units of this matrix $\mathbf{R}$ (covariance of sensor errors) are different from those of $\mathbf{R}$ in the Kalman filter, also.

7.7.2 Advantages of Kalman-Bucy Filtering

People already familiar with differential equations may find the Kalman–Bucy filter more intuitive and easier to work with than the Kalman filter—despite complications of the stochastic calculus. To its credit, the Kalman–Bucy filter requires only one equation each for propagation of the estimate and its covariance, whereas the Kalman filter requires two (for prediction and correction).

However, if the result must eventually be implemented in a digital processor, then it will have to be put into discrete-time form. Formulas for this transformation are given below. Those who prefer to "think in continuous time" can develop the problem solution first in continuous time as a Kalman–Bucy filter, then transform the result to Kalman filter form for implementation.

7.7.3 Model Parameters

Formulas for the Kalman filter parameters $\mathbf{Q}_k$ and $\mathbf{R}_k$ as functions of the Kalman–Bucy filter parameters $\mathbf{Q}(t)$ and $\mathbf{R}(t)$ can be derived from the process models.

7.7.3.1 $\mathbf{Q}(t)$ and $\mathbf{Q}_k$ The relationship between these two distinct matrix parameters depends on the coefficient matrix $\mathbf{F}(t)$ in the stochastic system model:

$$\mathbf{Q}_k = \int_{t_{k-1}}^{t_k} \exp\left(\int_t^{t_k} \mathbf{F}(s)\, ds\right) \mathbf{Q}(t) \exp\left(\int_t^{t_k} \mathbf{F}(s)\, ds\right)^{\mathrm{T}} dt. \tag{7.107}$$

7.7.3.2 $\mathbf{R}(t)$ and $\mathbf{R}_k$ This relationship will depend on how the sensor outputs in continuous time are filtered before sampling for the Kalman filter. If the sensor outputs were simply sampled without filtering, then

$$\mathbf{R}_k = \mathbf{R}(t_k). \tag{7.108}$$

However, it is common practice to use anti-alias filtering of the sensor outputs before sampling for Kalman filtering. Filtering of this sort can also alter the parameter $\mathbf{H}$ between the two implementations. For an integrate-and-hold filter (an effective anti-aliasing filter), this relationship has the form

$$\mathbf{R}_k = \int_{t_{k-1}}^{t_k} \mathbf{R}(t)\, dt, \tag{7.109}$$

in which case the measurement sensitivity matrix for the Kalman filter will be $\mathbf{H}_K = \Delta t\, \mathbf{H}_{KB}$, where $\mathbf{H}_{KB}$ is the measurement sensitivity matrix for the Kalman–Bucy filter.

7.8 GPS RECEIVER EXAMPLES

The following is a simplified example of the expected performance of a GPS receiver using

1. DOP calculations and
2. covariance analysis using the Riccati equations of a Kalman filter

for given sets of GPS satellites. These examples are implemented in the MATLAB m-file `GPS_perf.m` on the accompanying diskette.

7.8.1 Satellite Models

This example demonstrates how the Kalman filter converges to its minimum error bound and how well the GPS system performs as a function of the different phasings of the four available satellites. In the simulations, the available satellites and their respective initial phasings include the following:

Satellite No.	Ω_0 (deg)	θ_0 (deg)
1	326	68
2	26	340
3	146	198
4	86	271
5	206	90

The simulation runs two cases to demonstrate the criticality of picking the correctly phased satellites. Case 1 chooses satellites 1, 2, 3, and 4 as an example of an optimum set of satellites. Case 2 utilizes satellites 1, 2, 3, and 5 as an example of a nonoptimal set of satellites that will result in the dreaded "GDOP chimney" measure of performance.

Here, the GPS satellites are assumed to be in a circular orbital trajectory at a 55° inclination angle. The angle Ω_0 is the right ascension of the satellite and θ_0 is the angular location of the satellite in its circular orbit. It is assumed that the satellites orbit the earth at a constant rate θ with a period of approximately 43,082 s or slightly less than one half a day. The equations of motion that describe the angular phasing of the satellites are given, as in the simulation:

$$\Omega(t) = \Omega_0 - \Omega t, \qquad \theta(t) = \theta_0 + \theta t,$$

where the angular rates are given as

$$\Omega = 2\frac{\pi}{86{,}164}, \qquad \theta = 2\frac{\pi}{43{,}082},$$

where t is in seconds. The projects simulate the GPS system from $t = 0$ s to 3600 s as an example of the available satellite visibility window.

7.8.2 Measurement Model

In both the GDOP and Kalman filter models, the common observation matrix equations for discrete points is

$$z_k = H_k x_k + v_k,$$

where z, H, and v are the vectors and matrices for the kth observation point in time k. This equation is usually linearized when calculating the pseudorange by defining $z = \rho - \rho_0 = H^{[1]}x + v$.

Measurement noise v is usually assumed to be $\mathcal{N}(0, R)$ (normally distributed with zero mean and variance R). The covariance of receiver error R is usually assumed to be the same error for all measurements as long as all the same conditions exist for all time intervals (0–3600 s) of interest. By defining the measurement Z as the difference in the delta in position, the measurement sensitivity matrix H can be linearized and approximated as $H^{[1]}$ (i.e., first-order linear approximation) by defining

$$H^{[1]} = \frac{\partial \rho_r^i}{\partial x_i},$$

where i refers to the n different states of the Kalman filter and ρ_r is the reference pseudorange.

7.8.3 Coordinates

The orbital frame coordinates used in this simulation simplify the mathematics by using a linear transformation between the ECEF coordinate system to a locally level coordinate frame as the observer's local reference frame. Then, the satellite positions become

$$\acute{x} = y, \qquad \acute{y} = z, \qquad \acute{y} = x - R,$$

where $(\acute{x}, \acute{y}, \acute{z})$ are the locally level coordinates of the satellites x, y, z original ECEF coordinates. Here, R is the earth's radius. This assumes a user position at $(0, 0, 0)$ and makes the math simpler because now the pseudorange can be written as

$$\rho_1(t) = \sqrt{(x_1(t) - 0)^2 + (y_1(t) - 0)^2 + (z_1(t) - 0)^2},$$
$$h_x^{[1]}(t) = \frac{-(x_1(t) - 0)}{\rho_1(t)},$$

where $h^{[1]}$ represents the partial of the pseudorange with respect to x (component of the $H^{[1]}$ matrix). Therefore, the default earth and orbit constants are defined as

$$R = 26560000.0, \qquad E \text{ Rad} = 6380000.0, \qquad \alpha = (55^\circ).$$

7.8.4 Measurement Sensitivity Matrix

The definition of the different elements of the $H^{[1]}$ matrix are

$$x_1(t) = R\{\cos[\theta(t)]\sin[\Omega(t)] + \sin[\theta(t)]\cos[\Omega(t)]\cos\alpha\},$$
$$y_1(t) = R\{\sin[\theta(t)]\}\sin\alpha,$$
$$z_1(t) = R\{\cos[\theta(t)]\cos[\Omega(t)] - \sin[\theta(t)]\sin[\Omega(t)]\cos\alpha\} - E \text{ Rad},$$
$$\rho_1(t) = \sqrt{[x_1(t)]^2 + [y_1(t)]^2 + [z_1(t)]^2},$$
$$h_x(t) = \frac{-x_1(t)}{\rho_1(t)},$$
$$h_y(t) = \frac{-y_1(t)}{\rho_1(t)},$$
$$h_z(t) = \frac{-z_1(t)}{\rho_1(t)},$$

and likewise for each of the other four satellites.

The complete $H^{[1]}$ matrix can then be defined as

$$H(t) = \begin{bmatrix} h1_x(t) & h1_y(t) & h1_z(t) & 1 & 0 \\ h2_x(t) & h2_y(t) & h2_z(t) & 1 & 0 \\ h3_x(t) & h3_y(t) & h3_z(t) & 1 & 0 \\ h4_x(t) & h4_y(t) & h4_z(t) & 1 & 0 \end{bmatrix},$$

where the last two columns refer to the clock bias and clock drift.

In calculating GDOP only, the clock bias is used in the equations, so the H matrix becomes

$$H(t) = \begin{bmatrix} h1_x(t) & h1_y(t) & h1_z(t) & 1 \\ h2_x(t) & h2_y(t) & h2_z(t) & 1 \\ h3_x(t) & h3_y(t) & h3_z(t) & 1 \\ h4_x(t) & h4_y(t) & h4_z(t) & 1 \end{bmatrix}.$$

The calculation of the GDOP and various other DOP are then defined in terms of this $H(t)$ matrix as a function of time t:

$$\begin{aligned} A(t) &= [H(t)^{\mathrm{T}} H(t)]^{-1}, \\ \mathrm{GDOP}(t) &= \sqrt{\mathrm{tr}[A(t)]}, \\ \mathrm{PDOP}(t) &= \sqrt{A(t)_{1,1} + A(t)_{2,2} + A(t)_{3,3}}, \\ \mathrm{HDOP}(t) &= \sqrt{A(t)_{1,1} + A(t)_{2,2}}, \\ \mathrm{VDOP}(t) &= \sqrt{A(t)_{3,3}}, \\ \mathrm{TDOP}(t) &= \sqrt{A(t)_{4,4}}. \end{aligned}$$

7.8.5 Implementation Results

7.8.5.1 DOP Calculations In the MATLAB implementation, the GDOP, PDOP, HDOP, VDOP, and TDOP, are defined and plotted for the two different cases of satellite phasings.

Case 1: Good Geometry The results from Case 1 (satellites 1, 2, 3 and 4) show an excellent GDOP ranging to less 3.2 as a function of time. Figure 7.10 shows the variation of GDOP in meters as a function of time. This is a reasonable GDOP. Figure 7.11 shows all of the DOPs in meters as a function of time.

Case 2: Bad Geometry Case 2 satellite phasing results in the infamous GDOP "chimney peak" during that time when satellite geometry fails to provide observability of user position. Figure 7.12 shows the resulting GDOP plots. It shows that two satellites out of four are close to each other and thereby do not provide linearly independent equations. This combination of satellites cannot be used to find the user position, clock drift, and biases. Figure 7.13 is a multiplot of all the DOPs.

7.8.5.2 Kalman Filter Implementation For the second part of the example, a covariance analysis of the GPS/Kalman filter system is used to evaluate the performance of the system, given initial position estimates and estimates of receiver R and system dynamic Q noise. This type of analysis is done if actual measurement

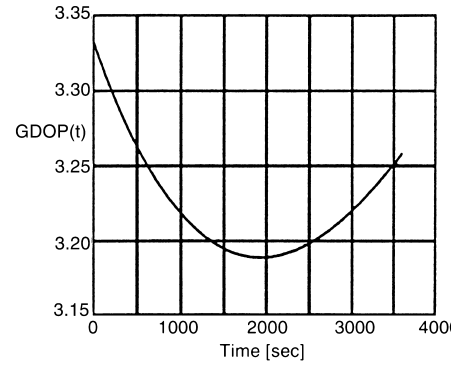


Fig. 7.10 *Case 1 GDOP.*

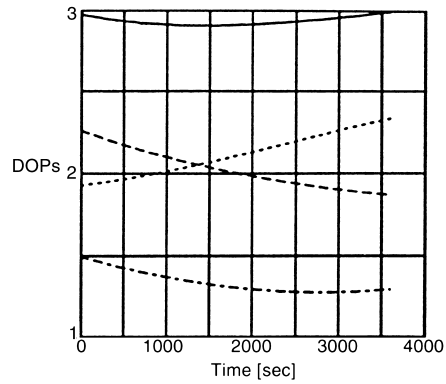


Fig. 7.11 *Case 1 DOPs.*

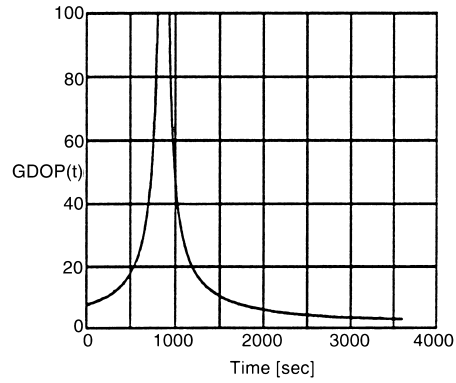


Fig. 7.12 *Case 2 GDOP.*

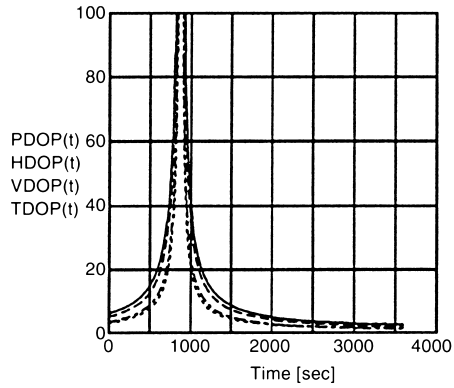


Fig. 7.13 *Case 2 DOPs.*

data is not available and can serve as a predictor of how well the system will converge to a residual error estimate in the position and time. The masking error in the Q matrix is

$$Q = \begin{bmatrix} 0.333 & 0 & 0 & 0 & 0 \\ 0 & 0.333 & 0 & 0 & 0 \\ 0 & 0 & 0.333 & 0 & 0 \\ 0 & 0 & 0 & 0.0833 & 0 \\ 0 & 0 & 0 & 0 & 0.142 \end{bmatrix},$$

where dimensions are in meters squared and meters squared per seconds squared.

The receiver noise R matrix in meters squared is

$$R = \begin{bmatrix} 225 & 0 & 0 & 0 \\ 0 & 225 & 0 & 0 \\ 0 & 0 & 225 & 0 \\ 0 & 0 & 0 & 225 \end{bmatrix}.$$

The initial transformation matrix between the first and next measurement is the matrix

$$\Phi = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

The assumed initial error estimate was 100 m and is presented by the $P_0(+)$ matrix and is an estimate of how far off the initial measurements are from the actual points:

$$P_0(+) = \begin{bmatrix} 10{,}000 & 0 & 0 & 0 & 0 \\ 0 & 10{,}000 & 0 & 0 & 0 \\ 0 & 0 & 10{,}000 & 0 & 0 \\ 0 & 0 & 0 & 90{,}000 & 0 \\ 0 & 0 & 0 & 0 & 900 \end{bmatrix}.$$

These assumptions assume a clock bias error of 300 m and a drift of 30 m/s. The discrete extended Kalman filtering equations, as listed in Table 7.2, are the a priori *covariance matrix*

$$P_k(-) = \Phi P_{k-1}(+)\Phi^{\mathrm{T}} + Q_{k-1},$$

the *Kalman gain equation*

$$\overline{\mathbf{K}}_k = P_k(-)H_k^{[1]\mathrm{T}}[H_k^{[1]} + P_k(-)H_k^{[1]\mathrm{T}} + R_k]^{-1},$$

and the a posteriori *covariance matrix*

$$P_k(+) = \{I - \overline{\mathbf{K}}_k H_k^{[1]}\}P_k(-).$$

The diagonal elements of the covariance matrices $P_k(-)$ (predicted) and $P_k(+)$ (corrected) are plotted as an estimate of how well the individual x, y, z and clock drift errors converge as a function of time for $t = 1$ to $t = 150$ s.

In a real system, the Q, R, and Φ matrices and Kalman gain estimates are under control of the designer and need to be varied individually to obtain an acceptable residual covariance error. This example only analyzes the covariance estimates for the given Q, R, and Φ matrices, which turned out to be a satisfactory set of inputs.

Simulation Procedure Start Simulation $t = 0, \ldots, 3600$.

Case 1: Satellites 1, 2, 3, and 4:

$$\Omega_0 1 = 326\frac{\pi}{180}, \quad \Omega_0 2 = 26\frac{\pi}{180}, \quad \Omega_0 3 = 146\frac{\pi}{180}, \quad \Omega_0 4 = 86\frac{\pi}{180},$$

$$\theta_0 1 = 68\frac{\pi}{180}, \quad \theta_0 2 = 340\frac{\pi}{180}, \quad \theta_0 3 = 198\frac{\pi}{180}, \quad \theta_0 4 = 271\frac{\pi}{180}.$$

Define variables:

$$\Omega_r = 2\frac{\pi}{86{,}164}, \qquad \theta_r = 2\frac{\pi}{43{,}082}.$$

The angular rate equations are

$$\begin{aligned}
\Omega 1(t) &= \Omega_0 1 - \Omega_r t, \qquad & \theta 1(t) &= \theta_0 1 + \theta_r t,\\
\Omega 2(t) &= \Omega_0 2 - \Omega_r t, \qquad & \theta 2(t) &= \theta_0 2 + \theta_r t,\\
\Omega 3(t) &= \Omega_0 3 - \Omega_r t, \qquad & \theta 3(t) &= \theta_0 3 + \theta_r t,\\
\Omega 4(t) &= \Omega_0 4 - \Omega_r t, \qquad & \theta 4(t) &= \theta_0 4 + \theta_r t
\end{aligned}$$

The default earth constants are

$$R = 26560000.0, \qquad E \text{ Rad} = 6380000.0, \qquad \cos\alpha = \cos 55^\circ, \qquad \sin\alpha = \sin 55^\circ,$$

For satellite 1:

$$\begin{aligned}
x_1(t) &= R\big\{\cos[\theta_1(t)]\ \sin[\Omega_1(t)] + \sin[\theta_1(t)]\cos\Omega_1(t)\cos\alpha\big\},\\
y_1(t) &= R\big\{\sin[\theta_1(t)]\sin\alpha\\
z_1(t) &= R\big\{\cos[\theta_1(t)]\cos[\Omega_1(t)] - \sin[\theta(t)]\sin[\Omega_1(t)]\cos\alpha\big\} - E \text{ Rad},\\
\rho_1(t) &= \sqrt{[(x_1(t)]^2 + [y_1(t)]^2 + [z_1(t)]^2}
\end{aligned}$$

and the H matrix elements are

$$h1_x(t) = \frac{-x_1(t)}{\rho_1(t)}, \qquad h1_y(t) = \frac{-y_1(t)}{\rho_1(t)}, \qquad h1_z(t) = \frac{-z_1(t)}{\rho_1(t)}.$$

For satellite 2

$$\begin{aligned}
x_2(t) &= R\{\cos[\theta_2(t)]\sin[\Omega_2(t)] + \sin[\theta_2(t)]\cos\Omega_2(t)\cos\alpha\},\\
y_2(t) &= R\sin[\theta_2(t)]\sin\alpha, .\\
z_2(t) &= R\{\cos[\theta_2(t)]\cos[\Omega_2(t)] - \sin[\theta_2(t)]\sin[\Omega_2(t)]\cos\alpha\} - E \text{ Rad},\\
\rho_2(t) &= \sqrt{[x_2(t)]^2 + [y_2(t)]^2 + [z_2(t)]^2},
\end{aligned}$$

and the H matrix elements are

$$h2_x(t) = \frac{-x_2(t)}{\rho_2(t)}, \qquad h2_y(t) = \frac{-y_2(t)}{\rho_2(t)}, \qquad h2_z(t) = \frac{-z_2(t)}{\rho_2(t)}.$$

For satellite 3

$$
\begin{aligned}
x_3(t) &= R\{\cos[\theta_3(t)]\sin[\Omega_3(t)] + \sin[\theta_3(t)]\cos\Omega_3(t)\cos\alpha\},\\
y_3(t) &= R\sin[\theta_3(t)]\sin\alpha,\\
z_3(t) &= R\{\cos[\theta_3(t)]\cos[\Omega_3(t)] - \sin[\theta_3(t)]\sin[\Omega_3(t)]\cos\alpha\} - E \text{ Rad},\\
\rho_3(t) &= \sqrt{[x_3(t)]^2 + [y_3(t)]^2 + z_3(t)]^2},
\end{aligned}
$$

and the H matrix elements are

$$
h3_x(t) = \frac{-x_3(t)}{\rho_3(t)}, \qquad h3_y(t) = \frac{-y_3(t)}{\rho_3(t)}, \qquad h3_z(t) = \frac{-z_3(t)}{\rho_3(t)}.
$$

For satellite 4

$$
\begin{aligned}
x_4(t) &= R\{\cos[\theta_4(t)]\sin[\Omega_4(t)] + \sin[\theta_4(t)]\cos\Omega_4(t)\cos\alpha\},\\
y_4(t) &= R\sin[\theta_4(t)]\sin\alpha,\\
z_4(t) &= R\{\cos[\theta_4(t)]\cos[\Omega_4(t)] - \sin[\theta_4(t)]\sin[\Omega_4(t)]\cos\alpha\} - E \text{ Rad},\\
\rho_4(t) &= \sqrt{[x_4(t)]^2 + [y_4(t)]^2 + [z_4(t)]^2},
\end{aligned}
$$

and the H matrix elements are

$$
h4_x(t) = \frac{-x_4(t)}{\rho_4(t)}, \qquad h4_y(t) = \frac{-y_4(t)}{\rho_4(t)}, \qquad h4_z(t) = \frac{-z_4(t)}{\rho_4(t)}.
$$

Complete $H^{[1]}$ matrix:

$$
H^{[1]}(t) = \begin{bmatrix}
h1_x(t) & h1_y(t) & h1_z(t) & 1 & 0\\
h2_x(t) & h2_y(t) & h2_z(t) & 1 & 0\\
h3_x(t) & h3_y(t) & h3_z(t) & 1 & 0\\
h4_x(t) & h4_y(t) & h4_z(t) & 1 & 0
\end{bmatrix}.
$$

The H matrix used in the GDOP calculation is

$$
H^{[1]}(t) = \begin{bmatrix}
h1_x(t) & h1_y(t) & h1_z(t) & 1\\
h2_x(t) & h2_y(t) & h2_z(t) & 1\\
h3_x(t) & h3_y(t) & h3_z(t) & 1\\
h4_x(t) & h4_y(t) & h4_z(t) & 1
\end{bmatrix}.
$$

The noise matrix is

$$Q = \begin{bmatrix} 0.333 & 0 & 0 & 0 & 0 \\ 0 & 0.333 & 0 & 0 & 0 \\ 0 & 0 & 0.333 & 0 & 0 \\ 0 & 0 & 0 & 0.0833 & 0 \\ 0 & 0 & 0 & 0 & 0.142 \end{bmatrix}.$$

The initial guess of the $P_0(+)$ matrix is

$$P_0(+) = \begin{bmatrix} 10{,}000 & 0 & 0 & 0 & 0 \\ 0 & 10{,}000 & 0 & 0 & 0 \\ 0 & 0 & 10{,}000 & 0 & 0 \\ 0 & 0 & 0 & 90{,}000 & 0 \\ 0 & 0 & 0 & 0 & 900 \end{bmatrix}$$

and the R matrix is

$$R = \begin{bmatrix} 225 & 0 & 0 & 0 \\ 0 & 225 & 0 & 0 \\ 0 & 0 & 225 & 0 \\ 0 & 0 & 0 & 225 \end{bmatrix},$$

$$A(t) = [H^{[1]\mathrm{T}}(t)H^{[1]}(t)]^{-1},$$

$$\mathrm{GDOP}(t) = \sqrt{\mathrm{tr}[A(t)]}.$$

Kalman Filter Simulation Results Figure 7.14 shows the square roots of the covariance terms P_{11} (RMS east position uncertainty), both predicted (dashed line) and corrected (solid line). After a few iterations, the RMS error in the x position is less than 5 m. Figure 7.15 shows the corresponding RMS north position uncertainty in meters, and Figure 7.16 shows the corresponding RMS vertical position uncertainty in meters.

Figures 7.17 and 7.18 show the square roots of the error covariances in clock bias and clock drift rate in meters.

Problems

7.1 Demonstrate the property of the matrix exponentials in Eq. 7.55 by showing that

$$\frac{d}{d\,\Delta t}\exp(\Delta t\,\mathbf{F}) = \mathbf{F}\Phi,$$

with $\mathbf{F}$ defined by Eq. 7.53 and $\mathbf{\Phi} = \exp(\Delta t\,\mathbf{F})$ defined by Eq. 7.62.

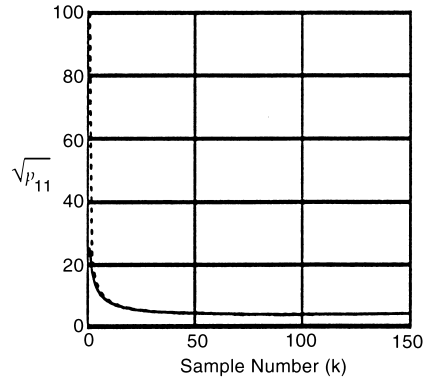


Fig. 7.14 *RMS east position error.*

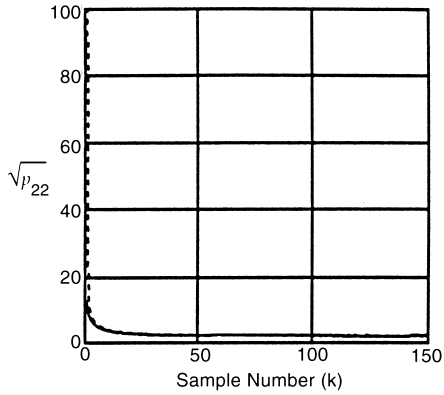


Fig. 7.15 *RMS north position error.*

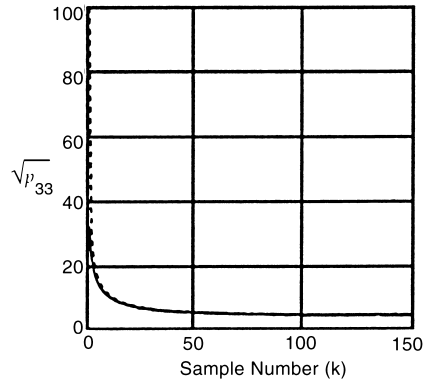


Fig. 7.16 *RMS vertical position error.*

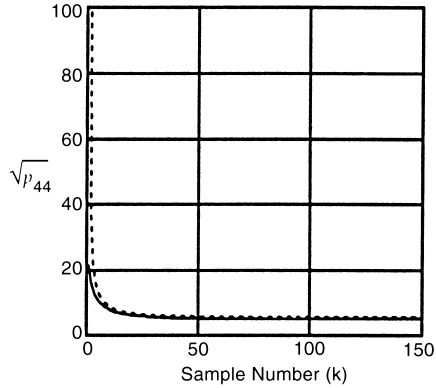


Fig. 7.17 *RMS clock error.*

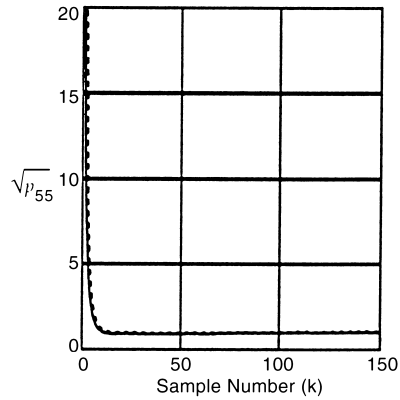


Fig. 7.18 *RMS drift error.*

7.2 Given the scalar plant and observation equations

$$x_k = x_{k-1}, \qquad z_k = x_k + v_k \sim N(0,\ \sigma_v^2)$$

and white noise

$$Ex_0 = 1, \qquad Ex_0^2 = P_0,$$

find the estimate of x_k and the steady-state covariance.

7.3 Given the vector plant and scalar observation equations,

$$x_k = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_{k-1} + w_{k-1} \quad \text{(normal and white)},$$

$$z_k = [\,1 \quad 0\,]x_k + v_k, \quad \text{(normal and white)},$$

$$Ew_K = 0, \qquad Q_k = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix},$$

$$Ev_k = 0, \qquad R_k = 1 + (-1)^k,$$

find the covariances and Kalman gains for $k = 10$, $P_0 = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$,

7.4 Given

$$x_k = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_{k-1} + \begin{bmatrix} 1 \\ \frac{1}{2} \end{bmatrix}(-g),$$

$$z_k = [1 \quad 0]x_k + v_k \sim \text{normal and white},$$

where g is gravity, find $\hat{x}_k$, $P_k(+)$ for $k = 6$:

$$\hat{x}_0 = \begin{bmatrix} 90 \\ 1 \end{bmatrix}, \qquad P_0 = \begin{bmatrix} 10 & 0 \\ 0 & 2 \end{bmatrix},$$

$$Ev_k = 0, \qquad Ev_k^2 = 2$$

7.5 Given

$$x_k = -2x_{k-1} + w_{k-1},$$

$$z_k = x_k + v_k \sim \text{normal and white},$$

$$Ev_k = 0, \qquad Ev_k^2 = 1,$$

$$Ew_k = 0, \qquad E(w_k w_j) = e^{-|k-j|},$$

find the covariances and Kalman gains for $k = 3$, $P_0 = 10$.

7.6 Given

$$E[(w_k - 1)(w_j - 1)] = e^{-|k-j|},$$

find the discrete equation model.

7.7 Given

$$E\langle [w(t_1) - 1][w(t_2) - 1] \rangle = e^{-|t_1 - t_2|},$$

find the differential equation model.

7.8 Based on the 24-satellite GPS constellation, five satellite trajectories are selected, and their parameters tabulated accordingly:

$$\alpha = 55^\circ$$

Satellite ID	$\Omega_0(^\circ)$	$\Theta_0(^\circ)$
6	272.847	268.126
7	332.847	80.956
8	32.847	111.876
9	92.847	135.226
10	152.847	197.046

(a) Choose correctly phased satellites of four.

(b) Calculate DOPs to show their selection by plots.

(c) Use Kalman filter equations for $P_k(-)$, $\bar{K}_k$, and $P_k(+)$ to show the errors. Draw the plots. This should be done with good GDOP.

Choose user positions at $(0, 0, 0)$ for simplicity.

Global Positioning Systems, Inertial Navigation, and Integration,
Mohinder S. Grewal, Lawrence R. Weill, Angus P. Andrews

Print ISBN 0-471-35032-X Electronic ISBN 0-471-20071-9

8

Kalman Filter Engineering

We now consider the following, practical aspects of Kalman filtering applications:

1. how performance of the Kalman filter can degrade due to computer roundoff errors and alternative implementation methods with better robustness against roundoff;
2. how to determine computer memory, word length, and throughput requirements for implementing Kalman filters in computers;
3. ways to implement real-time monitoring and analysis of filter performance;
4. the Schmidt–Kalman suboptimal filter, designed for reducing computer requirements;
5. covariance analysis, which uses the Riccati equations for performance-based predictive design of sensor systems; and
6. Kalman filter architectures for GPS/INS integration.

8.1 MORE STABLE IMPLEMENTATION METHODS

8.1.1 Effects of Computer Roundoff

Computer roundoff limits the precision of numerical representation in the implementation of Kalman filters. It has been shown to cause severe degradation of filter performance in many applications, and alternative implementations of the Kalman filter equations (the Riccati equations, in particular) have been shown to improve robustness against roundoff errors.

Computer roundoff for floating-point arithmetic is often characterized by a single parameter $\varepsilon_{\text{roundoff}}$, which is the largest number such that

$$1 + \varepsilon_{\text{roundoff}} \equiv 1 \text{ in machine precision.} \tag{8.1}$$

The following example, due to Dyer and McReynolds [32], shows how a problem that is well conditioned, as posed, can be made ill-conditioned by the filter implementation.

Example 8.1 Let $\mathbf{I}_n$ denote the $n \times n$ identity matrix. Consider the filtering problem with measurement sensitivity matrix

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1+\delta \end{bmatrix}$$

and covariance matrices

$$\mathbf{P}_0 = \mathbf{I}_3, \text{ and } \mathbf{R} = \delta^2 \mathbf{I}_2,$$

where $\delta^2 < \varepsilon_{\text{roundoff}}$ but $\delta > \varepsilon_{\text{roundoff}}$. In this case, although $\mathbf{H}$ clearly has rank 2 in machine precision, the product $\mathbf{H}\mathbf{P}_0\mathbf{H}^{\mathrm{T}}$ *with roundoff* will equal

$$\begin{bmatrix} 3 & 3+\delta \\ 3+\delta & 3+2\delta \end{bmatrix},$$

which is singular. The result is unchanged when $\mathbf{R}$ is added to $\mathbf{H}\mathbf{P}_0\mathbf{H}^{\mathrm{T}}$. In this case, then, the filter observational update fails because the matrix $\mathbf{H}\mathbf{P}_0\mathbf{H}^{\mathrm{T}} + \mathbf{R}$ is not invertible.

8.1.2 Alternative Implementations

The covariance correction process (observational update) in the solution of the Riccati equation was found to be the dominant source of numerical instability in the Kalman filter implementation, with the more common symptoms of failure being asymmetry of the covariance matrix (easily fixed) or, worse by far, negative terms on its diagonal. These implementation problems could be avoided for some problems by using more precision, but they were eventually solved for most applications by using alternatives to the covariance matrix $\mathbf{P}$ as the dependent variable in the covariance correction equation. However, each of these methods required a compatible method for covariance prediction. Table 8.1 lists several of these compatible implementation methods for improving the numerical stability of Kalman filters.

Figure 8.1 illustrates how these methods perform on the ill-conditioned problem of Example 8.1 as the conditioning parameter $\delta \to 0$. For this particular test case, using 64-bit floating-point precision (52-bit mantissa), the accuracy of the Carlson

TABLE 8.1 Compatible Methods for Implementing the Riccati Equation

Covariance Matrix Format	Implementation Methods: Corrector Method	Implementation Methods: Predictor Method
Symmetric nonnegative definite	Kalman [71], Joseph [19]	Kalman [71]
Square Cholesky factor **C**	Potter [100, 8]	$\mathbf{C}_{k+1}(-) = \mathbf{\Phi}_k \mathbf{C}_k(+)$
Triangular Cholesky factor **C**	Carlson [20]	Kailath–Schmidt[a]
Triangular Cholesky factor **C**	Morf–Kailath combined [93]	
Modified Cholesky factors **U**, **D**	Bierman [10]	Thornton [116]

[a] From unpublished sources.

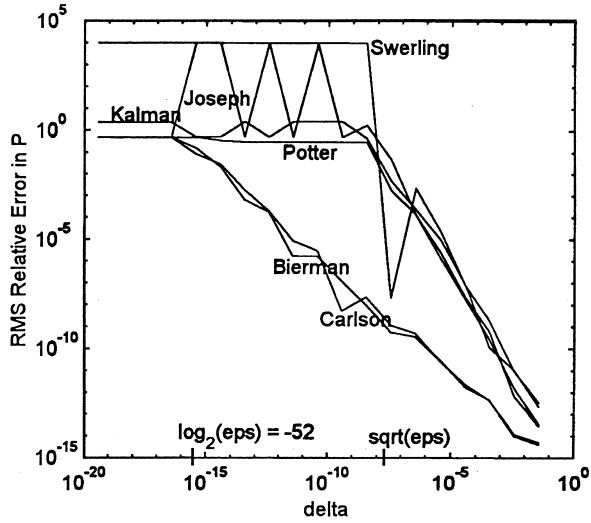


Fig. 8.1 *Degradation of numerical solutions with problem conditioning.*

[20] and Bierman [10] implementations degrade more gracefully than the others as $\delta \to \varepsilon$, the machine precision limit. The Carlson and Bierman solutions still maintain about nine digits ($\approx$30 bits) of accuracy at $\delta \approx \sqrt{\varepsilon}$, when the other methods have essentially no bits of accuracy in the computed solution.

These results, by themselves, do not prove the general superiority of the Carlson and Bierman solutions for the Riccati equation. Relative performance of alternative implementation methods may depend upon details of the specific application, and for many applications, the standard Kalman filter implementation will suffice. For many other applications, it has been found sufficient to constrain the covariance matrix to remain symmetric.

8.1.3 Serial Measurement Processing

It is shown in [73] that it is more efficient to process the components of a measurement vector serially, one component at a time, than to process them as a vector. This may seem counterintuitive, but it is true even if its implementation requires a transformation of measurement variables to make the associated measurement noise covariance $\mathbf{R}$ a diagonal matrix (i.e., with noise uncorrelated from one component to another).

8.1.3.1 Measurement Decorrelation If the covariance matrix $\mathbf{R}$ of measurement noise is *not* a diagonal matrix, then it can be made so by $\mathbf{UDU}^{\mathrm{T}}$ decomposition (Eq. B.22) and changing the measurement variables,

$$\mathbf{R}_{\text{corr}} = \mathbf{U}_R \mathbf{D}_R \mathbf{U}_R^{\mathrm{T}}, \tag{8.2}$$

$$\mathbf{R}_{\text{decorr}} \stackrel{\text{def}}{=} \mathbf{D}_R \text{ (a diagonal matrix)}, \tag{8.3}$$

$$\mathbf{z}_{\text{decorr}} \stackrel{\text{def}}{=} \mathbf{U}_R \backslash \mathbf{z}_{\text{corr}}, \tag{8.4}$$

$$\mathbf{H}_{\text{decorr}} \stackrel{\text{def}}{=} \mathbf{U}_R \backslash \mathbf{H}_{\text{corr}}, \tag{8.5}$$

where $\mathbf{R}_{\text{corr}}$ is the nondiagonal (i.e., correlated component to component) measurement noise covariance matrix, and the new *decorrelated* measurement vector $\mathbf{z}_{\text{decorr}}$ has a diagonal measurement noise covariance matrix $\mathbf{R}_{\text{decorr}}$ and measurement sensitivity matrix $\mathbf{H}_{\text{decorr}}$.

8.1.3.2 Serial Processing of Decorrelated Measurements The components of $\mathbf{z}_{\text{decorr}}$ can now be processed one component at a time using the corresponding row of $\mathbf{H}_{\text{decorr}}$ as its measurement sensitivity matrix and the corresponding diagonal element of $\mathbf{R}_{\text{decorr}}$ as its measurement noise variance.

A MATLAB implementation for this procedure is listed in Table 8.2, where the final line is a "symmetrizing" procedure designed to improve robustness.

TABLE 8.2 Matlab Implementation of Serial Measurement Update

```
x = x̂k[-]
P = Pk[-]
for j=1: ℓ,
     z = zk(j),
     H = Hk(j,:);
     R = Rdecorr (j,j);
     K̄= PH'/(HPH'+R)
     x = K̄(z-Hx);
     P = P-K̄HP;
end;
x̂k[+] = x
Pk[+] = (P+P')/2;
```

8.1.4 Joseph Stabilized Implementation

This implementation of the Kalman filter is in [19], where it is demonstrated that numerical stability of the solution to the Riccati equation can be improved by rearranging the standard formulas for the measurement update into the following formats (given here for scalar measurements):

$$\acute{z} = \frac{z}{\sqrt{R}}, \tag{8.6}$$

$$\acute{\mathbf{H}} = \acute{z}\mathbf{H}, \tag{8.7}$$

$$\overline{\mathbf{K}} = (\acute{\mathbf{H}}\mathbf{P}\acute{\mathbf{H}}^{\mathrm{T}} + 1)^{-1}\mathbf{P}\acute{\mathbf{H}}^{\mathrm{T}}, \tag{8.8}$$

$$\mathbf{P} \leftarrow (I - \overline{\mathbf{K}}\acute{\mathbf{H}})\mathbf{P}(I - \overline{\mathbf{K}}\acute{\mathbf{H}})^{\mathrm{T}} + \overline{\mathbf{K}}\,\overline{\mathbf{K}}^{\mathrm{T}}. \tag{8.9}$$

These equations would replace those for $\overline{\mathbf{K}}$ and $\mathbf{P}$ within the loop in Table 8.2.

The Joseph stabilized implementation and refinements (mostly taking advantage of partial results and the redundancy due to symmetry) in [10], [46] and are implemented in the MATLAB files `Joseph.m`, `Josephb.m`, and `Josephdv.m`, respectively, on the accompanying diskette.

8.1.5 Factorization Methods

8.1.5.1 Historical Background Robust implementation methods were introduced first for the covariance correction (measurement updates), observed to be the principal source of numerical instability. In [100, 8], the idea of using a *Cholesky factor* (defined in Section B.8.1) of the covariance matrix $\mathbf{P}$, as the dependent variable in the Riccati equation is introduced.

Carlson [20] discovered a more robust method using *triangular* Cholesky factors, which have zeros either above or below their main diagonals. Bierman [10] extended

this to *modified Cholesky factors* (defined in Section B.1.8.1), which are *diagonal* and *unit triangular* matrices **D** and **U**, respectively, such that

$$\mathbf{U}\mathbf{D}\mathbf{U}^{\mathrm{T}} = \mathbf{P} \tag{8.10}$$

and **U** is triangular with 1's along its main diagonal.

Compatible covariance prediction methods were discovered by Thomas Kailath and Stanley F. Schmidt (for Carlson's method) and Catherine Thornton [116] (for Bierman's method).

8.1.6 Square-Root Filtering Methods

8.1.6.1 Problems with the Riccati Equation Many early applications of Kalman filtering ran into serious numerical instability problems in solving the ancillary Riccati equation for the Kalman gain. The problem was eventually solved (over the next decade or so) by reformulating the Riccati equation so that its solution was more robust against computer roundoff errors. Some of the more successful of these approaches are collectively called "square-root filtering."

8.1.6.2 Square-Root Filtering The concept for square-root filtering came from James H. Potter when he was at the MIT Instrumentation Laboratory (later the Charles Stark Draper Laboratory) in the early 1960s, and his concept was implemented successfully in the Kalman filter used for onboard navigation in all the Apollo moon missions. Potter's algorithm is implemented on the Matlab m-file `potter.m` on the accompanying diskette. It was originally called square-root filtering because it is based, in part, on an algorithm for taking a symmmetric square root of a special form of a symmetric matrix.

The improved robustness of Potter's approach comes from replacing the covariance matrix **P** with its Cholesky factor[1] as the dependent parameter of the Riccati equation. Some of the observed improvement in numerical stability is attributed to improvement in the *condition number* cond(**C**) (ratio of largest to smallest characteristic value) over cond(**P**), because

$$\mathrm{cond}(\mathbf{C}) = \sqrt{\mathrm{cond}(\mathbf{P})}. \tag{8.11}$$

A matrix is considered ill-conditioned for inversion in a particular computer ("machine") if its condition number is close to $1/\varepsilon_{\mathrm{machine}}$, where $\varepsilon_{\mathrm{machine}}$ is the largest positive number[2] for which

$$1 + \varepsilon_{\mathrm{machine}} \overset{\mathrm{machine}}{\equiv} 1 \tag{8.12}$$

[1]See Section B.1.8.1 for the definition and properties of Cholesky factors.

[2]$\varepsilon_{\mathrm{machine}}$ has the reserved name eps in MATLAB. Its value is returned when you type "eps".

in machine precision. That is, the result of adding $\varepsilon_{\text{machine}}$ to 1 has no effect in machine precision.

8.1.6.3 Triangularization Methods

The so-called "**QR**" theorem of linear algebra states that every real $m \times n$ matrix $\mathbf{S}$ can be factored in the form $\mathbf{S} = \mathbf{QR}$, where $\mathbf{Q}$ is an $m \times m$ orthogonal matrix and $\mathbf{R}$ is an $m \times n$ upper triangular matrix. Depending on the relative magnitudes of m and n, the resulting triangular matrix $\mathbf{R}$ may have any of the forms

$$m < n \qquad m = n \qquad m < n$$

$$\mathbf{R} = \boxed{0 \diagdown} \quad \boxed{\diagdown 0} \quad \boxed{\diagdown \atop 0}$$

with the nonzero part of the upper triangular submatrix in the upper right corner.

There are several algorithms for computing the triangular and orthogonal factors, including some with the order of the factors reversed (effectively "**RQ**" algorithms). These are also called *triangularization methods*. They are key to square-root filtering, because they can transform a nontriangular Cholesky factor $\mathbf{M}$ into a triangular one $\acute{\mathbf{T}}$, because

$$\mathbf{M}\mathbf{M}^{\mathrm{T}} = \acute{\mathbf{T}}\acute{\mathbf{O}}\acute{\mathbf{O}}^{\mathrm{T}}\acute{\mathbf{T}}^{\mathrm{T}} \tag{8.13}$$

$$= \acute{\mathbf{T}}\acute{\mathbf{T}}^{\mathrm{T}}. \tag{8.14}$$

Algorithms that implement **QR** decompositions need not compute the orthogonal factor explicitly, if it is not needed.

Because the matrix symbols $\mathbf{Q}$ (dynamic disturbance noise covariance) and $\mathbf{R}$ (measurement noise covariance) are already used for specific parts of the Kalman filter, we will use alternative symbols here.

8.1.6.4 QR Decomposition by Householder Transformations

Householder transformation matrices[3] are orthogonal matrices of the form

$$\mathcal{H}(\mathbf{v}) = \mathbf{I} - \frac{2\mathbf{v}\mathbf{v}^{\mathrm{T}}}{\mathbf{v}^{\mathrm{T}}\mathbf{v}}, \tag{8.15}$$

where $\mathbf{v}$ is a column vector and I is the compatibly dimensioned identity matrix.

The condition number of an othogonal matrix is perfect (i.e., 1), making it well suited for robust operations in numerical linear algebra. The **QR** decomposition of a matrix $\mathbf{M}$ is effectively accomplished by a series of products by Householder transformation matrices, in the partitioned form

$$\boxed{\begin{matrix} x \\ 0 \end{matrix}} \begin{bmatrix} \mathcal{H}(\mathbf{v}) & 0 \\ 0 & \mathbf{I} \end{bmatrix} = \boxed{\begin{matrix} \\ 0 \end{matrix}\!\!\begin{matrix} x \\ \diagdown \end{matrix}}$$

[3] Named for Alston S. Householder (1904–1993), who developed many of the more robust methods used in numerical linear algebra.

with the vector $\mathbf{v}$ chosen to annihilate all but the end element of the remaining subrow $\boldsymbol{\chi}$ of $\mathbf{M}$ until only the upper triangular part remains. It suffices to let

$$\mathbf{v} = \boldsymbol{\chi}^{\mathrm{T}} - |\boldsymbol{\chi}| \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{8.16}$$

However, operations with Householder matrices are typically *not* implemented by calculating the appropriate Householder matrix and taking a matrix product. They can be implemented quite efficiently as an algorithm operating "in place" on the matrix $\mathbf{M}$, destroying $\mathbf{M}$ and leaving only the matrix $\mathbf{T}$ in its place when completed. The MATLAB function `housetri.m` on the accompanying diskette does just this.

8.1.6.5 Triangularization of Cholesky Factors If $\mathbf{A}$ is any Cholesky factor of $\mathbf{P}$ and $\mathbf{A} = \mathbf{CM}$ is a $\mathbf{QR}$ decomposition of $\mathbf{A}$ such that $\mathbf{M}$ is the orthogonal factor and $\mathbf{C}$ is the triangular factor, then $\mathbf{C}$ is a triangular Cholesky factor of $\mathbf{P}$. That is,

$$\begin{aligned} \mathbf{P} &= \mathbf{A}\mathbf{A}^{\mathrm{T}} \\ &= \mathbf{CM}(\mathbf{CM})^{\mathrm{T}} \\ &= \mathbf{CMM}^{\mathrm{T}}\mathbf{C}^{\mathrm{T}} \\ &= \mathbf{C}\mathbf{C}^{\mathrm{T}} \end{aligned} \tag{8.17}$$

and $\mathbf{C}$ is triangular. This is the basis for the following two types of square-root filtering.

8.1.6.6 Morf–Kailath Square-Root Filter In Morf–Kailath square-root filtering, the entire Riccati equation, including prediction and correction steps, is implemented in a single triangularization procedure. It effectively computes the Cholesky factors of successive covariance matrices of prediction error (required for computing Kalman gain) without ever explicitly computing the intermediate values for corrected estimation errors. Assume the following:

- $\mathbf{G}_k$ is the dynamic disturbance distribution matrix of the system model,
- $\mathbf{C}_{Q_k}$ is a Cholesky factor of $\mathbf{Q}_k$;
- $\boldsymbol{\Phi}_k$ is the state transition matrix from the previous epoch;
- $\mathbf{C}_{P_k}$ is a Cholesky factor of $\mathbf{P}_k(-)$, the covariance matrix of prediction error from the previous epoch;
- $\mathbf{H}_k$ is the measurement sensitivity matrix of the previous epoch;
- $\mathbf{C}_{R_k}$ is the measurement noise covariance matrix of the previous epoch; and

$\mathcal{H}$ is a triangularizing orthogonal matrix for the partitioned matrix such that

$$\begin{bmatrix} \mathbf{G}_k\mathbf{C}_{Q_k} & \mathbf{\Phi}_k\mathbf{C}_{P_k} & 0 \\ 0 & \mathbf{H}_k\mathbf{C}_{P_k} & \mathbf{C}_{R_k} \end{bmatrix}\mathcal{H} = \begin{bmatrix} 0 & C_{P_{k+1}} & \mathbf{\Psi}_{k+1} \\ 0 & 0 & \mathbf{C}_{E_{k+1}} \end{bmatrix}, \tag{8.18}$$

a partitioned upper triangular matrix.

Then $\mathbf{C}_{P_{k+1}}$ is the square triangular Cholesky factor of $\mathbf{P}_{k+1}$, the covariance matrix of prediction error, and the Kalman gain

$$\overline{\mathbf{K}}_{k+1} = \mathbf{\Psi}_{k+1}/\mathbf{C}_{E_{k+1}}. \tag{8.19}$$

8.1.6.7 Carlson–Schmidt Square–Root Filtering In Carlson–Schmidt square-root filtering, only the temporal update (predictor) of the Riccati equation is implemented using triangularization. The observational update is implemented by an algorithm due to Carlson [20]. The Carlson algorithm is implemented in the Matlab m-file `carlson.m` on the accompanying diskette. It calculates the Cholesky factor $\mathbf{C}_{P,k-1}(+)$ of the covariance matrix $\mathbf{P}_{k-1}(+)$ corrected for the effect of taking the measurement.

The temporal update is implemented as

$$[\,0 \quad \mathbf{C}_{P,k}(-)\,] = [\,\mathbf{\Phi}_k\mathbf{C}_{P,k-1}(+) \quad \mathbf{G}_k\mathbf{C}_{Q,k}\,]\mathcal{H}_1\mathcal{H}_2\mathcal{H}_3\cdots\mathcal{H}_n, \tag{8.20}$$

where $\mathbf{C}_{P,k}(-)$ = sought-for triangular Cholesky factor of $\mathbf{P}_k(-)$
$\mathbf{C}_{Q,k}$ = a Cholesky factor of $\mathbf{Q}_k$
$[\,\mathbf{\Phi}_k\mathbf{C}_{P,k-1}(+) \quad \mathbf{G}_k\mathbf{C}_{Q,k}\,]$ = is a Cholesky factor of $\mathbf{P}_k(-)$, and the sequence of Householder transformation matrices $\mathcal{H}_1\mathcal{H}_2\mathcal{H}_3\cdots\mathcal{H}_n$ transforms it into the appropriate triangular form

It can be shown that the matrix $[\,\mathbf{\Phi}_k\mathbf{C}_{P,k-1}(+) \quad \mathbf{G}_k\mathbf{C}_{Q,k}]$ is, indeed, a Cholesky factor of $\mathbf{P}_k(-)$ by multiplying it out:

$$\begin{aligned} &[\,\mathbf{\Phi}_k\mathbf{C}_{P,k-1}(+) \quad \mathbf{G}_k\mathbf{C}_{Q,k}\,][\,\mathbf{\Phi}_k\mathbf{C}_{P,k-1}(+) \quad \mathbf{G}_k\mathbf{C}_{Q,k}]^{\mathrm{T}} \\ &\quad = \mathbf{\Phi}_k\mathbf{C}_{P,k-1}(+)(\mathbf{\Phi}_k\mathbf{C}_{P,k-1}(+))^{\mathrm{T}} + \mathbf{G}_k\mathbf{C}_{Q,k}(\mathbf{G}_k\mathbf{C}_{Q,k})^{\mathrm{T}} \\ &\quad = \mathbf{\Phi}_k\mathbf{C}_{P,k-1}\mathbf{C}_{P,k-1}^{\mathrm{T}}\mathbf{\Phi}_k^{\mathrm{T}} + \mathbf{G}_k\mathbf{C}_{Q,k}\mathbf{C}_{Q,k}^{\mathrm{T}}\mathbf{G}_k^{\mathrm{T}} \\ &\quad = \mathbf{\Phi}_k\mathbf{P}_{k-1}(+)\mathbf{\Phi}_k^{\mathrm{T}} + \mathbf{G}_k\mathbf{Q}_k\mathbf{G}_k^{\mathrm{T}} \\ &\quad = \mathbf{P}_k(-). \end{aligned}$$

The triangularization in Eq. 8.20 is implemented in the Matlab m-file `schmidt.m` on the accompanying diskette.

8.1.7 Bierman–Thornton UD Filter

The Bierman–Thornton square-root filter is analogous to the Carlson–Schmidt square root filter, but with modified Cholesky factors of **P** in place of ordinary Cholesky factors. It is also called "UD filtering," in reference to the modified Cholesky factors **U** and **D**.

The principal differences between Bierman–Thornton UD filtering and Carlson–Schmidt square-root filtering are as follows:

1. The Bierman–Thornton square-root filter uses **U** and **D** in place of **C**.
2. The observational update (due to Bierman [10]) requires no square roots.
3. The temporal update (due to Thornton [116]) uses *modified weighted Gram–Schmidt orthogonalization* in place of Householder triangularization.

The methods of Carlson and Bierman are "rank 1 modification" algorithms for Cholesky factors and modified Cholesky factors, respectively. A rank 1 modification algorithm for a triangular Cholesky factor, for example, calculates the triangular Cholesky factor $\mathbf{C}(+)$ such that

$$\mathbf{C}(+)\mathbf{C}(+)^{\mathrm{T}} = \mathbf{C}(-)\mathbf{C}(-)^{\mathrm{T}} - \mathbf{v}\mathbf{v}^{\mathrm{T}},$$

given the prior Cholesky factor $\mathbf{C}(-)$ and the vector **v**. Rank 1 modification in this case refers to the matrix rank of the modification $\mathbf{v}\mathbf{v}^{\mathrm{T}}$ of $\mathbf{C}(-)\mathbf{C}(-)^{\mathrm{T}}$. In this particular application of rank 1 modification, the matrix and vector are

$$\mathbf{P}(-) = \mathbf{C}(-)\mathbf{C}(-)^{\mathrm{T}} \quad \text{(predicted covariance)},$$
$$\mathbf{v} = \frac{\mathbf{P}(-)\mathbf{H}^{\mathrm{T}}}{\sqrt{\mathbf{H}\mathbf{P}(-)\mathbf{H}^{\mathrm{T}} + R}},$$

respectively. This only works if the dimension of the measurement equals 1 (i.e., the rank of **H** is 1), which is the reason that square-root filtering must process measurement vector components one at a time.

The corresponding *UD* predictor algorithm was discovered by Catherine Thornton, and was the subject of her Ph.D. dissertation [116]. It is based on a relatively robust orthogonalization method developed by Åke Björck [11] and called "modified Gram–Schmidt." Bierman [10] refers to it as "modified weighted Gram–Schmidt" (MWGS), which is much longer than its appropriate name, "Thornton's method." A listing of its implementation in Matlab (from `thornton.m` on the accompanying diskette) is presented in Table 8.3.

The corresponding Matlab listing of the Bierman corrector algorithm (from `bierman.m` on the accompanying diskette) is given in Table 8.4.

TABLE 8.3 UD Filter Part 1: Thornton Predictor

```
function [x,U,D] = thornton(xin,Phi,Uin,Din,Gin,Q)
x    = Phi*xin;   % state prediction
[n,r]= size(Gin); % get model dimensions
G    = Gin;       % move to internal array
U    = eye(n);    % initialize U
PhiU = Phi*Uin;
for i=n:-1:1,
   sigma = 0;
   for j=1:n,
      sigma = sigma + PhiU(i,j)^2 *Din(j,j);
      if (j <= r)
         sigma = sigma + G(i,j)^2 *Q(j,j);
      end;
   end;
   D(i,i) = sigma;
   for j=1:i-1,
      sigma = 0;
      for k=1:n,
         sigma = sigma + PhiU(i,k)*Din(k,k)*PhiU(j,k);
      end;
      for k=1:r,
         sigma = sigma + G(i,k)*Q(k,k)*G(j,k);
      end;
      U(j,i) = sigma/D(i,i);
      for k=1:n,
         PhiU(j,k) = PhiU(j,k) - U(j,i)*PhiU(i,k);
      end;
      for k=1:r,
         G(j,k) = G(j,k) - U(j,i)*G(i,k);
      end;
    end;
end;
```

8.1.7.1 Potter Implementation

The original square-root filter is due to James H. Potter, who first introduced the idea of recasting the Riccati equation in terms of Cholesky factors of the covariance matrix **P**. The Matlab m-file `potter.m` on the accompanying diskette is a Matlab implementation of the Potter square-root filter. The Potter approach handles only the observational update. It has been generalized in [5] for vector-valued observations, with a corresponding differential equation for the temporal propagation of the covariance equation.

8.2 IMPLEMENTATION REQUIREMENTS

Computer requirements for implementing Kalman filters tend to be dominated by the need to solve the matrix Riccati equation, and many of these requirements can be expressed as functions of the dimensions of the matrices in the Riccati equation.

TABLE 8.4 UD Filter Part 2: Bierman Corrector

```
a     = U'*H';% a is not modified, but
b     = D*a; % b is modified to become unscaled Kalman gain.
dz    = z - H*xin;
alpha = R;
gamma = 1/alpha;
for j=1:length(xin),
   beta   = alpha;
   alpha  = alpha + a(j)*b(j);
   lambda = -a(j)*gamma;
   gamma  = 1/alpha;
   D(j,j) = beta*gamma*D(j,j);
   for i=1:j-1,
      beta   = U(i,j);
      U(i,j) = beta + b(i)*lambda;
      b(i)   = b(i) + b(j)*beta;
   end;
end;
dzs = gamma*dz; % apply scaling to innovations
x = x + dzs*b; % multiply by unscaled Kalman gain
```

8.2.1 Throughput

Computer throughput is measured in arithmetic operations per second (ops). Minimum throughput required for implementing the Kalman filter will be the product

$$\text{Throughput (ops)} \approx \text{operations per cycle} \times \text{cycles per second},$$

where the operations per cycle depends on the number of state variables (n) and measurement variables (ℓ) and cycles per second depends on attributes of the sensors and the dynamic system model.

8.2.1.1 Cycles per Second The eigenvalues of the dynamic coefficient matrix $\mathbf{F}$ determine the natural frequencies of the dynamic system model, with the real parts representing inverse decay times and the imaginary parts representing natural resonant frequencies. Sampling rates much faster than the largest eigenvalue of $\mathbf{F}$ are likely to be sufficient for Kalman filter implementation, but they may not be necessary. This sort of analysis is used for calculating the size of the time steps required for reliably integrating the differential equations for the system state estimate $\hat{\mathbf{x}}$ and its associated covariance matrix $\mathbf{P}$, but determining workable update rates for a particular application usually relies on simulation studies. Only in simulation can we calculate differences beween the true solution and an approximated solution.

8.2.1.2 Operations per Cycle Factors that influence the numbers of arithmetic operations per cycle of a Kalman filter implementation on a specific application include the following:

1. The dimensions of the state vector $\mathbf{x}$ (n), measurement vector $\mathbf{z}$ (ℓ), and process noise vector $\mathbf{w}$ (p). These and the implementation methods (next item) are the only factors considered in Fig. 8.2, so the results should be considered upper bounds on just the estimation loop and Riccati equation. Computations required to compute any of the parameter matrices are not included in the calculations.
2. The implementation methods used, such as
   (a) the original Kalman implementation,
   (b) the Carlson–Schmidt square-root implementation,
   (c) the Bierman–Thornton *UD* implementation, and
   (d) the Morf–Kailath combined square-root implementation.

   However, choices among these methods are more likely to be driven by cost and numerical stability issues than by computational requirements. The more stable implementation methods may perform as well with shorter wordlengths as the less stable methods with longer wordlengths, which could make a big difference in processor speed and cost.
3. Processor hardware architecture issues, including the following:
   (a) Processor type, including
      (i) reduced instruction set computers (RISCs), in which all arithmetic operations take place between registers and can be completed in one machine cycle and all data transfers between registers and memory also require one machine cycle;
      (ii) complex instruction set computers (CISCs), which can perform many RISC-type operations per instruction but with each instruction executing in several machine cycles, and
      (iii) DSP processors designed for pipelined dot products and analog interfaces.
   (b) The types of arithmetic operations available as machine instructions versus those that must be implemented in software, such as square roots.
   (c) Hardware interrupt structure, which may determine how well the processor supports real-time programming constraints.
   (d) Availability of real-time debugging hardware and software (compilers, editors, source-on-line code debuggers).
   (e) Data wordlength options.
   (f) Arithmetic processing speed with representative instruction mix for Kalman filtering.

4. Whether the implementation includes a dynamic disturbance noise distribution matrix **G**.
5. Whether any or all of the following matrices must be computed on each interation:
   (a) **Φ** ($n \times n$ state transition matrix),
   (b) **H** ($\ell \times n$ measurement sensitivity matrix),
   (c) **R** ($\ell \times \ell$ measurement noise covariance matrix),
   (d) **Q** ($p \times p$ dynamic disturbance noise covariance matrix), and
   (e) **G** ($n \times p$ dynamic disturbance noise distribution matrix).
6. Whether the estimate and covariance propagation is done using a state transition matrix or (for nonlinear filtering) by numerical integration.
7. Whether the predicted measurement is computed using a measurement sensitivity matrix or a nonlinear measurement function.
8. The sparse structure (if any) of the matrix parameters in the Kalman filter equations. For example:
   (a) For Kalman filters integrating independent systems with no dynamic interactions (e.g., GPS and INS), dynamic coefficient matrices and state transition matrices will have blocks of zeros representing the dynamic uncoupling.
   (b) Because most sensors measure only a limited number of variables, the number of nonzero elements in the measurement senstivity matrix **H** usually tends to grow as the number of sensors (ℓ) and not as the total number of possible elements ($n\ell$).
   (c) It is uncommon that the dynamic disturbance noise covariance matrix **Q** and/or sensor noise covariance matrix **R** are dense (i.e., no zeros), and it is not uncommon that they are diagonal matrices.
9. Details of the programming implementation, such as
   (a) whether the programming takes advantage of any offered matrix sparseness by skipping multiplications by zero,
   (b) whether the programming takes advantage of symmetry in the Riccati equation solution, and
   (c) multiplication order in evaluating matrix expressions, which can make a significant difference in the computation required.

   In Carlson–Schmidt square-root filtering (Section 8.1.6.7), for example, it is not necessary to place the blocks of the matrix

$$[\,\mathbf{\Phi}_k\mathbf{C}_{P,k-1}(+)|\mathbf{G}_k\mathbf{C}_{Q,k}\,]$$

   into a separate common array for triangularization. The additional array space can be saved by doing an implicit triangularization of the array (by modified indexing) without physically relocating the blocks.

Not Quite Upper Bounds The contour plots in Fig. 8.2 are "not-quite-worst-case computational cost" plotted in terms of equivalent multiply-and-accumulate operations per update cycle as functions of the parameter matrix dimensions and the compatible implementation methods used. These are effectively "not-quite upper bounds" because they assume the worst-case parameter matrix conditions (i.e., all parameter matrices full and time varying) but do not include the added computational cost of calculating the time-varying parameter matrices. We cannot present a comparable upper bound (other than Eq. 8.21) for the computational costs for calculating the parameter matrices ($\mathbf{\Phi}$, $\mathbf{H}$, $\mathbf{G}$, $\mathbf{Q}$ and $\mathbf{R}$), because these computational costs will be very much application-dependent.

The results also assume that advantage is taken of symmetry, where possible.

Decorrelation Costs These computational costs for all implementations of the corrector step (observational update) include those for diagonalizing $\mathbf{R}$, which is on the order of $\ell(\ell^2 - 1)/6$ multiply-and-accumulate operations per update cycle, where $\mathbf{R}$ is $\ell \times \ell$. They also include the measurement error decorrelation operations for calculating the matrix ratios $\mathbf{U_R}\backslash\mathbf{H}$ and $\mathbf{U_R}\backslash\mathbf{z}$, which is in the order of $\frac{1}{2}(n+1)\ell(\ell-1)$ multiply-and-accumulate operations per update cycle, where n is the number of state variables.

More detailed computational requirements are tabulated in [10], assuming diagonal $\mathbf{Q}$ and $\mathbf{R}$ matrices (i.e., without the added worst-case computational requirements for diagonalization and/or decorrelation included).

Kalman–Kalman The computational cost for the Kalman filter corrector step (observational update, including decorrelation) will be on the order of $\frac{1}{6}\ell(9n^2 + \ell^2 + 3\ell n + 18n + 3\ell + 2)$ multiply-and-accumulate operations per update cycle [10, p. 104, plus decorrelation], and that of the predictor step (temporal update) will be on the order of $n(2n^2 + p^2 + n)$ multiply-and-accumulate operations per update cycle, where p is the number of components in the dynamic disturbance noise vector (i.e., $\mathbf{Q}$ is $p \times p$).

Carlson–Schmidt Computational complexity of the Carlson algorithm is on the order of $\frac{1}{6}\ell(12n^2 + 3\ell n + \ell^2 + 75n + 3\ell - 4)$ multiply-and-accumulate operations per update cycle [10, p. 108, plus decorrelation]. (This algorithm requires taking square roots, the computational costs for which were approximated as being equivalent to six multiply-and-accumulate operations.)

The corresponding Schmidt–Kailath temporal update (using Householder triangularization) has computational cost on the order of $\frac{1}{3}n(2n^2 + 3np + 6n + 12p + 25)$ multiply-and-accumulate operations per update cycle in the worst-case scenario ($\mathbf{G}$ and $\mathbf{Q}$ time varying and $\mathbf{Q}$ nondiagonal). This includes the cost of forming the products $\mathbf{\Phi}C_{\mathbf{P}}$ and $\mathbf{G}\mathbf{C}_{\mathbf{Q}}$, where $\mathbf{C_P}$ is a Cholesky factor of $\mathbf{P}$ and $\mathbf{C_Q}$ is a Cholesky factor of $\mathbf{Q}$.

Bierman–Thornton The computational cost of the Bierman algorithm is on the order of $\frac{1}{6}\ell(9n^2 + 3\ell n + \ell^2 + 3\ell + 12n - 4)$ multiply-and-accumulate operations

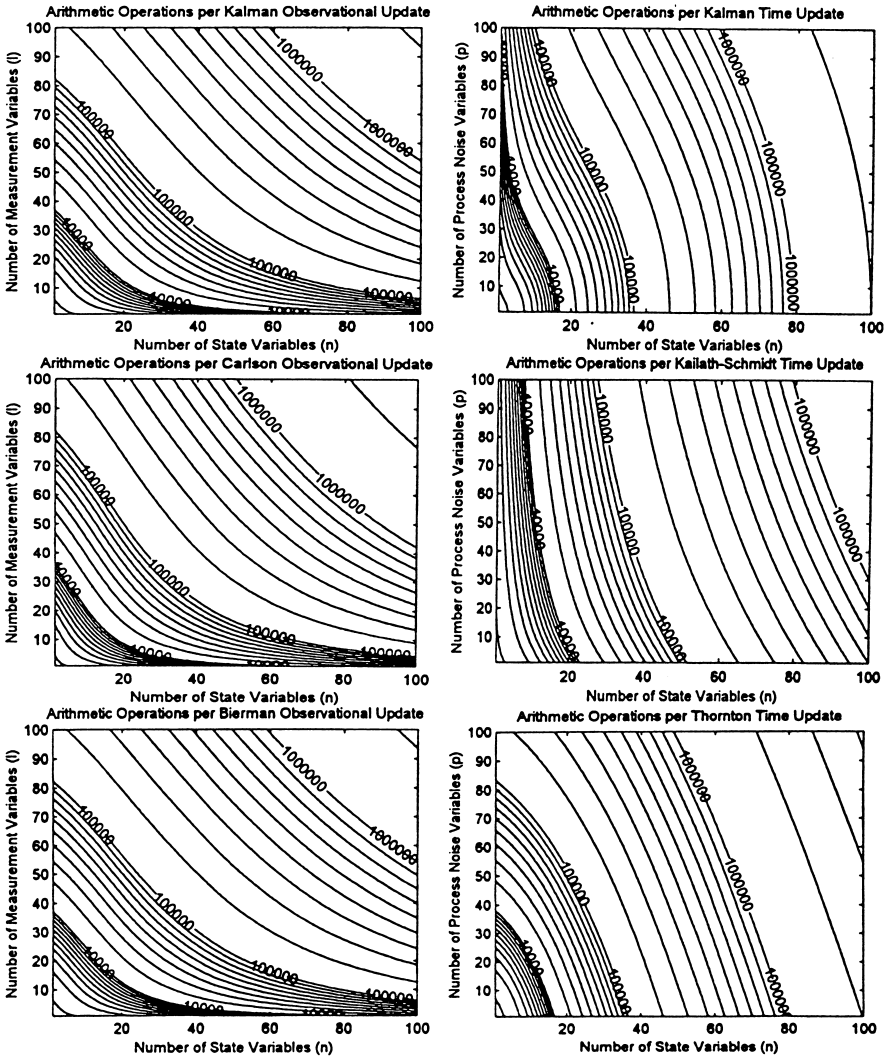


Fig. 8.2 *Arithmetic operations (multiply, accumulate) per update for Kalman filter implementations.*

per update cycle, which is not significantly different from that of the conventional Kalman filter [10, p. 107, plus decorrelation].

The corresponding not-quite-worst-case computational cost for the Thornton update algorithm is on the order of $\frac{1}{6}n(4n^2 + 3np + p^2 + n - 1)/2 + p(p^2 - 1)$ multiply-and-accumulate operations per update cycle. These computational costs include those for diagonalizing $\mathbf{Q}$ (by UD factorization), which is on the order of $\frac{1}{6}p(p^2 - 1)$ multiply-and-accumulate operations per update cycle, where $\mathbf{Q}$ is $p \times p$. These formulas also include the additional $\frac{1}{2}np(p - 1)$ multiply-and-accumulate

operations per update cycle required for taking the product $\mathbf{GU_Q}$, where $\mathbf{U_Q}$ is the U-factor from UD factorization of $\mathbf{Q}$, before beginning the Thornton algorithm listed in [10].

Not shown in the plots is the "Joseph stabilized" implementation, which generally has better numerical stability than the conventional Kalman implementation but requires approximately three times the number of arithmetic operations per update cycle [10, pp. 104 105].

The additional computational cost required for computing the parameter matrices $\mathbf{\Phi}$, $\mathbf{H}$, $\mathbf{G}$, $\mathbf{Q}$, and $\mathbf{R}$ tends to grow as

$$a\left(\underbrace{n^2}_{\mathbf{\Phi}}+\underbrace{n\ell}_{\mathbf{H}}+\underbrace{np}_{\mathbf{G}}+\underbrace{\tfrac{1}{2}p(p+1)}_{\mathbf{Q}}+\underbrace{\tfrac{1}{2}\ell(\ell+1)}_{\mathbf{R}}\right) \tag{8.21}$$

multiply-and-accumulate operations per update cycle, where a is the average number of multiply-and-accumulate operations per matrix element.

8.2.2 Memory

The minimum number of data words required for conventional Kalman filtering is approximately $4n^2 + 3n\ell + 2\ell^2$ [46], where n is the number of state variables and ℓ is the number of measurement variables. If necessary, these memory requirements can be reduced somewhat by exploiting symmetry of the covariance matrix $\mathbf{P}$ to store only the unique triangular part. Memory requirements for square-root filtering are not substantially different.

8.2.2.1 Wordlength Bierman [10] has conjectured that square-root filtering (Section 8.1.6.2) gives comparable accuracy with half as many bits of precision as conventional Kalman filtering. No proof of this conjecture was offered, but Example 8.1 and Fig. 8.1 would seem to support it.

8.3 KALMAN FILTER MONITORING

8.3.1 Rejecting Anomalous Sensor Data

8.3.1.1 Effects of Anomalous Sensor Data Anomalous sensor data can result from sensor failures or from corruption of the signals from sensors, and it is important to detect these events before the anomalous data corrupts the estimate. The filter is not designed to accept errors due to sensor failures or signal corruption, and they can seriously degrade the accuracy of estimates. The Kalman filter has infinite impulse response, so errors of this sort can persist for some time.

8.3.1.2 Detecting Anomalous Sensor Data Fortunately, the Kalman filter implementation includes parameters that can be used to detect anomalous data. The Kalman gain matrix

$$\overline{\mathbf{K}}_k = \mathbf{P}_k(-)\mathbf{H}_k^{\mathrm{T}} \underbrace{(\mathbf{H}_k\mathbf{P}_k(-)\mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k)^{-1}}_{\mathbf{Y}_{vk}} \tag{8.22}$$

includes the factor

$$\mathbf{Y}_{vk} = \left(\mathbf{H}_k\mathbf{P}_k(-)\mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k\right)^{-1}, \tag{8.23}$$

the information matrix of innovations. The innovations are the measurement residuals

$$v_k \overset{\text{def}}{=} \mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k(-), \tag{8.24}$$

the differences between the apparent sensor outputs and the predicted sensor outputs. The associated likelihood function for innovations is

$$\mathcal{L}(\boldsymbol{\nu}_k) = \exp(-\tfrac{1}{2}\boldsymbol{\nu}_k^{\mathrm{T}}\mathbf{Y}_{vk}\boldsymbol{\nu}_k), \tag{8.25}$$

and the log-likelihood is

$$\log[\mathcal{L}(\boldsymbol{\nu}_k)] = -\boldsymbol{\nu}_k^{\mathrm{T}}\mathbf{Y}_{vk}\boldsymbol{\nu}_k, \tag{8.26}$$

which can easily be calculated. The equivalent statistic

$$\chi^2 = \frac{\boldsymbol{\nu}_k^{\mathrm{T}}\mathbf{Y}_{vk}\boldsymbol{\nu}_k}{\ell} \tag{8.27}$$

(i.e., without the sign change and division by 2, but divided by the dimension of $\boldsymbol{\nu}_k$) is nonnegative with a minimum value of zero. If the Kalman filter were perfectly modeled and all white-noise sources were Gaussian, this would be a chi-squared statistic with distribution as plotted in Fig. 8.3. An upper limit threshold value on χ^2 can be used to detect anomalous sensor data, but a practical value of that threshold should be determined by the operational values of χ^2, not the theoretical values. That is, first its range of values should be determined by monitoring the system in operation, then a threshold value $\chi^2_{\max}$ chosen such that the fraction of good data rejected when $\chi^2 > \chi^2_{\max}$ will be acceptable.

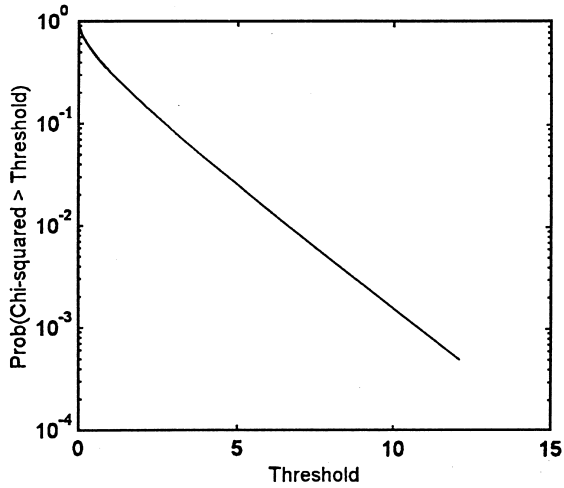


Fig. 8.3 *Chi-squared distribution.*

8.3.1.3 Exception Handling for Anomalous Sensor Data The log-likelihood test can be used to detect and reject anomalous data, but it can also be important to use the measurement innovations in other ways:

1. as a minimum, to raise an alarm whenever something anomalous has been detected;
2. to talley the relative frequency of sensor data anomalies, so that trending or incipient failure may be detectable; and
3. to aid in identifying the source, such as which sensor or system may have failed.

8.3.2 Monitoring Filter Health

Filter health monitoring methods are useful for detecting disparities between the physical system and the model of the system used in Kalman filtering (useful in filter development), for detecting numerical instabilities in the solution of the Riccati equation, for detecting the onset of poor observability conditions, for detecting when sensors fail, or for detecting gradual degradation of sensors.

8.3.2.1 Empirical Chi-Squared Distributions Calculating the empirical distribution of the statistic χ^2 of Eq. 8.27 and comparing it to the theoretical distribution in Fig. 8.3 is another way of checking that the filter model is in reasonable agreement with the physical system. It does require many thousands of samples to obtain a reasonable assessment of the distribution, however. These distributions are also handy for setting the thresholds for data rejection, because they characterize the frequency of type 2 errors (i.e., rejecting legitimate measurements).

8.3.2.2 Covariance Analysis Covariance analysis in this context means monitoring selected diagonal terms of the covariance matrix $\mathbf{P}$ of estimation uncertainty. These are the variances of state estimation uncertainty; system requirements are often specified in terms of the variance or RMS uncertainties of key state variables, and this is a way of checking that these requirements are being met. It is not always possible to cover all operational trajectories in the design of the sensor system, it is possible that situations can occur when these requirements are not being met in operation, and it can be useful to know that.

8.3.2.3 Covariance Matrix Condition Number The condition number of a matrix is the ratio of its largest to smallest characteristic values. A rough rule of thumb for a healthy covariance matrix is that its condition number $N_{\text{cond}}(\mathbf{P}) \ll 1/\varepsilon$, where the machine precision limit $\varepsilon = 2^{-N_{\text{bits}}}$ and N_{bits} is the number of bits in the mantissa of data words.

8.3.2.4 Checking Covariance Symmetry Square-root filtering (Section 8.1.6.2) is designed to ensure that the covariance matrix of estimation uncertainty (the dependent variable of the matrix Riccati equation) remains symmetric and positive definite. Otherwise, the fidelity of the solution of the Riccati equation can degrade to the point that it corrupts the Kalman gain, and that in turn corrupts the estimate. If you should choose not to use square-root filtering, then you may need some assurance that the decision was justified.

Verhaegen and Van Dooren [124] have shown that asymmetry of $\mathbf{P}$ is one of the factors contributing to numerical instability of the Riccati equation. If square-root filtering is not used, then the covariance matrix can be "symmetrized" occasionally by adding it to its transpose and rescaling:

$$\mathbf{P} := \tfrac{1}{2}(\mathbf{P} + \mathbf{P}^{\mathrm{T}}). \tag{8.28}$$

This trick has been used for many years to head off numerical instabilities.

8.3.2.5 Test for Positive Definiteness Cholesky decomposition [9] is one way to test whether $\mathbf{P}$ is positive definite, although the method may not be very robust. Each diagonal term of the Cholesky factor of $\mathbf{P}$ requires taking a square root, and the radicand will fail to be positive if $\mathbf{P}$ is indefinite. This can happen due to roundoff in Cholesky decomposition if $\mathbf{P}$ is close to being indefinite. In either case, it is probably a reasonable indication that the problem should be implemented using square-root filtering.

8.3.2.6 Checking Innovations Means and Autocorrelations Innovations are the differences between what comes out of the sensors and what was expected, based on the estimated system state. If the system were perfectly modeled in the Kalman filter, the innovations would be a zero-mean white-noise process and its autocorrelation function would be zero except at zero delay. The departure of the

empirical autocorrelation of innovations from this model is a useful tool for analysis of mismodeling in real-world applications.

Calculation of Autocovariance and Autocorrelation Functions The mean of the innovations should be zero. If not, the mean must be subtracted from the innovations before calculating the autocovariance and autocorrelation functions of the innovations.

For vector-valued innovations, the autocovariance function is a matrix-valued function, defined as

$$\mathcal{A}_{\text{covar},k} \stackrel{\text{def}}{=} E_i\langle \mathbf{z}_{v,i}\mathbf{z}_{v,i+k}^{\mathrm{T}}\rangle, \tag{8.29}$$

$$\mathbf{z}_{v,i} \stackrel{\text{def}}{=} \mathbf{z}_i - \mathbf{H}_i\hat{\mathbf{x}}_i(-) \text{ (innovations)}, \tag{8.30}$$

and the autocorrelation function is defined by

$$\mathcal{A}_{\text{correl},k} \stackrel{\text{def}}{=} \mathbf{D}_\sigma^{-1}\mathcal{A}_{\text{covar},k}\mathbf{D}_\sigma^{-1}, \tag{8.31}$$

$$\mathbf{D}_\sigma \stackrel{\text{def}}{=} \begin{bmatrix} \sigma_1 & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & 0 & \cdots & 0 \\ 0 & 0 & \sigma_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_\ell \end{bmatrix}, \tag{8.32}$$

$$\sigma_j \stackrel{\text{def}}{=} \sqrt{\{\mathcal{A}_{\text{covar},0}\}_{jj}}, \tag{8.33}$$

where the jth diagonal element $\{\mathcal{A}_{\text{covar},0}\}_{jj}$ of $\mathcal{A}_{\text{covar},0}$ is the variance of the jth component of the innovations vector.

Calculation of Spectra and Cross-Spectra The Fourier transforms of the diagonal elements of the autocovariance function $\mathcal{A}_{\text{covar},k}$ (i.e., as functions of k) are the power spectral densities (spectra) of the corresponding components of the innovations, and the Fourier transforms of the off-diagonal elements are the cross-spectra between the respective components.

Interpretation of Results Simple patterns to look for include the following:

1. Nonzero means of innovations may indicate the presence of uncompensated sensor output biases, or mismodeled output biases. The modeled variance of the bias may be seriously underestimated, for example.
2. Short-term means increasing or varying with time may indicate output noise that is a random walk or an exponentially correlated process.

3. Exponential decay of the autocorrelation functions is a reasonable indication of unmodeled (or mismodeled) random walk or exponentially correlated noise.
4. Spectral peaks may indicate unmodeled harmonic noise, but it could also indicate that there is an unmodeled harmonic term in the state dynamic model.
5. The autocovariance function $\mathcal{A}_{\text{covar},0}$ should equal $\mathbf{HPH}^{\mathrm{T}} + \mathbf{R}$ for time-invariant or very slowly time-varying systems. If $\mathcal{A}_{\text{covar},0}$ is much bigger than $\mathbf{HPH}^{\mathrm{T}} + \mathbf{R}$, it could indicate that $\mathbf{R}$ is too small or that the process noise $\mathbf{Q}$ is too small, either of which may cause $\mathbf{P}$ to be too small. If $\mathcal{A}_{\text{covar},0}$ is much smaller than $\mathbf{HPH}^{\mathrm{T}} + \mathbf{R}$, $\mathbf{R}$ and/or $\mathbf{Q}$ may be too large.
6. If the off-diagonal elements of $\mathcal{A}_{\text{correl},0}$ are much bigger than those of $\mathbf{D}_{\sigma}^{-1}(\mathbf{HPH}^{\mathrm{T}} + \mathbf{R})\mathbf{D}_{\sigma}^{-1}$, then there may be unmodeled correlations between sensor outputs. These correlations could be caused by mechanical vibration or power supply noise, for example.

8.4 SCHMIDT–KALMAN SUBOPTIMAL FILTERING

This is a methodology proposed by Schmidt [108] for reducing the processing and memory requirements for Kalman filtering, with predictable performance degradation. It has been used in GPS navigation as a means of eliminating additional variables (one per GPS satellite) required for Kalman filtering with correlated satellite clock phase errors due to selective availability (principally) and ionospheric delay errors. However, prospective users should always quantify the computational savings before adopting this approach.

8.4.1 State Vector Partitioning

Schmidt–Kalman filtering partitions the state vector into "essential" variables and "nuisance" variables,

$$\mathbf{x} = \left[\frac{\mathbf{x}_{\varepsilon}}{\mathbf{x}_{v}}\right], \tag{8.34}$$

where $\mathbf{x}_{\varepsilon}$ is the $n_{\varepsilon} \times 1$ subvector of essential variables to be estimated, $\mathbf{x}_{v}$ is the $n_{v} \times 1$ subvector that will not be estimated, and

$$n_{\varepsilon} + n_{v} = n, \text{ the total number of state variables.} \tag{8.35}$$

Even though the subvector $\mathbf{x}_{v}$ of nuisance variables is not estimated, the effects of not doing so must be reflected in the covariance matrix $\mathbf{P}_{\varepsilon\varepsilon}$ of uncertainty in the estimated variables. For that purpose, the Schmidt–Kalman filter calculates the covariance matrix $\mathbf{P}_{vv}$ of uncertainty in the unestimated state variables and the cross-

TABLE 8.5 Summary Implementation of Schmidt–Kalman Filter

Corrector (Observational Update)

$$
\begin{aligned}
\mathcal{C} &= [H_{\varepsilon k}(P_{\varepsilon\varepsilon k}(-)H_{\varepsilon k}^{\mathrm{T}} + P_{\varepsilon v k}(-)H_{vk}^{\mathrm{T}}) \\
&\quad + H_{vk}(P_{v\varepsilon k}(-)H^{\mathrm{T}} + P_{vvk}(-)H_{vk}^{\mathrm{T}}) \\
\overline{\mathbf{K}}_{\mathrm{SK},k} &= (P_{\varepsilon\varepsilon k}(-)H_{\varepsilon k}^{\mathrm{T}} + P_{\varepsilon v k}(-)H_{vk}^{\mathrm{T}})\mathcal{C} \\
\mathbf{x}_{\varepsilon,k}+ &= \mathbf{x}_{\varepsilon,k}(-) + \overline{\mathbf{K}}_{\mathrm{SK},k}(\mathbf{z}_k - H_{\varepsilon k}\mathbf{x}_{\varepsilon,k}(-) \\
\mathcal{A} &= I_{n_\varepsilon} - \overline{\mathbf{K}}_{\mathrm{SK},k}H_{\varepsilon,k} \\
\mathcal{B} &= \mathcal{A}P_{\varepsilon v,k}(-)H_{v,k}^{\mathrm{T}}\overline{\mathbf{K}}_{\mathrm{SK},k}^{\mathrm{T}} \\
P_{\varepsilon\varepsilon,k}+ &= \mathcal{A}P_{\varepsilon\varepsilon,k} - \mathcal{A}^{\mathrm{T}} - \mathcal{B} - \mathcal{B}^{\mathrm{T}} \\
&\quad + \overline{\mathbf{K}}_{\mathrm{SK},k}(H_{v,k}P_{vv,k} - H_{v,k}^{\mathrm{T}} + R_k)\overline{\mathbf{K}}_{\mathrm{SK},k}^{\mathrm{T}} \\
P_{\varepsilon v,k}+ &= \mathcal{A}P_{\varepsilon v,k}(-) - \overline{\mathbf{K}}_{\mathrm{SK},k}H_{v,k}P_{vv,k}- \\
P_{v\varepsilon,k}+ &= P_{\varepsilon v,k}+^{\mathrm{T}} \\
P_{vv,k}+ &= P_{vv,k}-
\end{aligned}
$$

Predictor (Time Update)

$$
\begin{aligned}
\hat{\mathbf{x}}_{\varepsilon,k+1-} &= \Phi_{\varepsilon k}\hat{\mathbf{x}}_{\varepsilon,k+} \\
P_{\varepsilon\varepsilon k+1-} &= \Phi_{\varepsilon k}P_{\varepsilon\varepsilon k+}\Phi_{\varepsilon k}^{\mathrm{T}} + Q_{\varepsilon\varepsilon} \\
P_{\varepsilon v k+1-} &= \Phi_{\varepsilon k}P_{\varepsilon v k+}\Phi_{vk}^{\mathrm{T}} \\
P_{v\varepsilon k+1-} &= P_{\varepsilon v k+1-}^{\mathrm{T}} \\
P_{vvk+1-} &= \Phi_{vk}P_{vvk+}\Phi_{vk}^{\mathrm{T}} + Q_{vv}
\end{aligned}
$$

covariance matrix $\mathbf{P}_{v\varepsilon}$ between the two types. These other covariance matrices are used in the calculation of the Schmidt–Kalman gain.

8.4.2 Implementation Equations

The essential implementation equations for the Schmidt–Kalman (SK) filter are listed in Table 8.5. These equations have been arranged for reusing intermediate results to reduce computational requirements.

8.5 COVARIANCE ANALYSIS

The dependent variable of the Riccati equation is the covariance matrix of estimation uncertainty, and using the Riccati equation to predict performance of a Kalman filter is called *covariance analysis*. It is highly recommended, if not essential, in the development of any integrated sensor system. It is useful for the following purposes:

1. quantifying expected system performance under the operating conditions for which it is designed;

2. determining how different operating conditions (e.g., momentary loss of GPS signals) will influence system performance;
3. evaluating performance of alternative designs, with different sensors and/or different sensor noise;
4. identifying the dominating error sources limiting system performance and determining the payoffs for improving performance of critical subsystems;
5. finding where relaxation of sensor requirements will not significantly degrade overall system performance; and
6. determining whether computer roundoff is likely to compromise the accuracy of the solution.

Covariance analysis is generally much easier than full Kalman filter implementation, especially if the model is nonlinear. One may not have to integrate nonlinear differential equations for covariance analysis, and many details that are critical for implementing the Kalman filter are not that important for covariance analysis.

8.6 GPS/INS INTEGRATION ARCHITECTURES

GPS architecture is likely to change with the addition of more channels and aiding signals, and INSs already have a wide variety of architectures. However, even if there were only one GPS and one possible INS, there would still be many ways to integrate the two. For these reasons, there is no all-encompassing Kalman filter architecture for GPS/INS integration. As an alternative, we present here some representative examples of integration architectures, depending on the type of INS and the level of coupling required between the GPS, INS, and Kalman filter. These examples cover a range from the simpler "loosely coupled" or "aiding" architectures to the more complex "tightly coupled" integration architectures.

The term *tightly coupled* is usually applied to systems using a single Kalman filter to integrate all sensor data, whereas *loosely coupled* systems may contain more than one Kalman filter, but there are many possible levels of coupling between the extremes.

8.6.1 GPS/Land Vehicle Integration

This is not an example of GPS/INS integration, but a simpler integration using different sensor types. It would be considered loosely coupled in that the internal workings of the GPS receiver are not altered, and there is no feedback from the Kalman filter to the receiver.

Figure 8.4 shows an architecture for integrating GPS with a wheel speed sensor (odometer) and magnetic compass for improving the navigational accuracy of highway vehicle navigation systems. The schematic shows differential GPS and map matching as part of the system, although they have little impact on the Kalman filter design beyond reducing some covariance parameters.

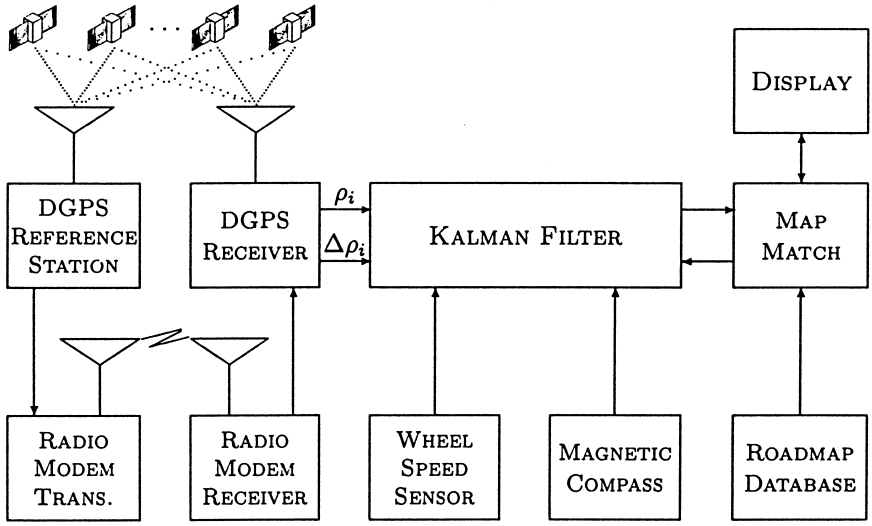


Fig. 8.4 *GPS/Automobile integration architecture.*

8.6.1.1 State Vector Wheel speed sensors tend to have slowly time-varying scale factors (due to slow variations in tire pressure and temperature), and magnetic compasses tend to have time-varying biases (due to localized magnetic anomalies). These two parameters (wheel speed scale factor and compass bias) can be added to the system state vector. A suitable system state vector for this application could include the following 10 state variables:

- three position components (e.g., easting, northing, and altitude with respect to a map reference point);
- three velocity components, each modeled as a random walk, with greater rate of variation in the horizontal components;
- receiver clock bias and drift (two variables, model shown in Fig. 5.9);
- wheel speed scale factor (modeled as a slow random walk); and
- magnetic compass heading bias (modeled as another slow random walk).

The 10×10 state transition matrix for this model would have the block form

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{I}_3 & \Delta t\mathbf{I}_3 & 0 & 0 & 0 & 0 & \text{(position)} \\ 0 & \mathbf{I}_3 & 0 & 0 & 0 & 0 & \text{(velocity)} \\ 0 & 0 & 1 & \Delta t & 0 & 0 & \text{(clock bias)} \\ 0 & 0 & 0 & 1 & 0 & 0 & \text{(clock drift)} \\ 0 & 0 & 0 & 0 & 1 & 0 & \text{(wheel speed scale factor)} \\ 0 & 0 & 0 & 0 & 0 & 1 & \text{(compass offset)} \end{bmatrix} \tag{8.36}$$

with nonzero process noise covariances on all but the position variables.

8.6.1.2 Measurement Vector

Inputs to the Kalman filter include the following:

- From the GPS receiver:
  (a) pseudoranges ρ_i to each satellite being tracked by the receiver and
  (b) integrated Doppler components $\Delta\rho_i$ for each of the satellites.
- From the wheel speed sensor: indicated vehicle speed.
- From the magnetic compass: magnetic heading angle.

The measurement sensitivity matrix for the GPS receiver inputs will be the same as that used in the GPS receiver Kalman filter implementation (Section 7.8.5.2). If the vehicle heading angle from the magnetic compass is measured clockwise from north, then the nonlinear function

$$\begin{bmatrix} v_E \\ v_N \end{bmatrix} = \mathbf{h}\left(\alpha, \delta\alpha, s_{\text{wheel}}, \delta S_{\text{wheelspeed}}\right) + \mathbf{v} \tag{8.37}$$

$$= \begin{bmatrix} (1 + \delta S_{\text{wheelspeed}}) s_{\text{wheel}} & \sin(\alpha - \delta\alpha) \\ \left(1 + \delta S_{\text{wheelspeed}}\right) s_{\text{wheel}} & \cos(\alpha - \delta\alpha) \end{bmatrix} + \mathbf{v}, \tag{8.38}$$

where v_E = east velocity
v_N = north velocity
$\mathbf{v}$ = sensor noise
α = magnetic compass output angle
$\delta\alpha$ = magnetic compass output bias
s_{wheel} = wheel speed sensor output
$\delta S_{\text{wheelspeed}}$ = wheel speed sensor scale factor offset

The measurement sensitivity submatrix for these variables will be the partial derivatives of this $\mathbf{h}$ evaluated at the estimated values of the state variables.

8.6.1.3 Potential Improvements

The schematic shows no feedback from the Kalman filter to the GPS receiver, but such feedback could be used to enhance

- reacquisition of signals lost momentarily by occlusions from buildings and other structures and
- the receiver tracking loops by using velocity changes.

8.6.2 GPS/INS Loosely Coupled Integration

Figure 8.5 is a schematic of a loosely coupled GPS/INS integration scheme with two Kalman filters. The GPS Kalman filter will be similar in function to that described in

Section 7.8.5.2, and the INS Kalman filter uses the GPS Kalman filter outputs for estimating sensor errors that the INS would be incapable of estimating by itself. An implementation of this type is called "GPS-aided INS" because the INS Kalman filter treats the outputs of the GPS Kalman filter as sensor outputs and does not include the GPS state variables (clock bias and drift).

The models used in the INS Kalman filter for estimating the INS parameters will be the same or similar in form to those used in tightly coupled implementations (Section 8.6.3). The sensor noise on the position and velocity data from the GPS receiver are modeled as exponentially correlated random processes to account for the fact that their variances remain bounded. The time constants for these exponentially correlated processes will usually be on the order of 10^2 s.

The advantages of this type of implementation include the fact that the GPS receiver can be treated as a separate subsystem requiring no alteration for the system-level integration. It is a relatively simple way to make an inexpensive IMU perform as well as an expensive one.

8.6.3 GPS/INS Tightly Coupled Integration

The common system wide state vector in this implementation architecture includes receiver state variables (clock bias and drift), GPS satellite state variables (propagation delay and selective availability timing errors), the INS navigation solution (position, velocity, acceleration, attitude, and attitude rate), and INS sensor compensation parameters.

The resulting Kalman filter implementation is doubly nonlinear, in that it has nonlinear dynamics and nonlinear measurements. The primary source of nonlinearities is in the attitude model, which is inherently nonlinear. For Kalman filter implementation, one would have to use extended Kalman filtering. For covariance analysis, one must at least use a set of simulated or nominal system trajectories for performance analysis.

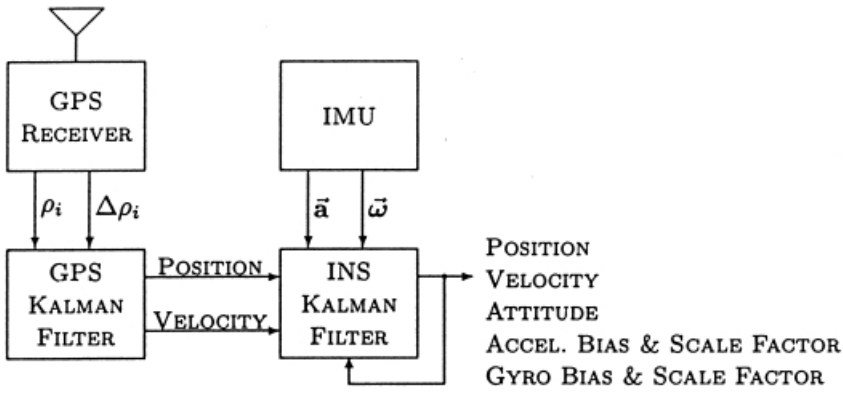


Fig. 8.5 *GPS-aided strapdown INS.*

A model of this type can be used for preliminary assessment of expected system performance and assessing how performance depends on details of the system design and application, such as

1. operating with or without the influence of Selective Availability,
2. operating at higher latitudes where satellite coverage is more sparse,
3. using Schmidt–Kalman filtering (Section 8.4) in place of conventional Kalman filtering,
4. performance specifications of the inertial sensors,
5. receiver performance characteristics,
6. vehicle trajectories and dynamics, and
7. momentary loss of GPS signals due to surrounding features (e.g., trees, buildings, mountains) or jamming.

8.6.3.1 Combined System State Vector The system state vector of 65+ state variables includes the following:

Fifteen Vehicle Dynamic Variables

$\theta, \phi_{\text{geodetic}}, h$—the longitude, geodetic latitude, and orthometric altitude of the vehicle.

v_E, v_N, v_U—the east, north, and vertical components of vehicle velocity, in meters per second.

a_E, a_N, a_U—the east, north, and vertical components of vehicle acceleration, in meters per second squared. It is necessary to include acceleration in the state vector because accelerometers measure the nongravitational component.

ρ_E, ρ_N, ρ_U—the east, north, and vertical components of the rotation vector rotating locally level ENU coordinates into vehicle-fixed RPY coordinates.

$\dot{\rho}_E, \dot{\rho}_N, \dot{\rho}_U$—time derivatives of the east, north, and vertical components of the rotation vector rotating locally level ENU coordinates into vehicle-fixed RPY coordinates.

Twelve Accelerometer Parameters

$b_{a_R}, b_{a_P}, b_{a_Y}$—roll-, pitch-, and yaw-axis components of accelerometer bias.

$\mathbf{M}_a$—nine elements of the 3×3 accelerometer scale factor and misalignment matrix.

Twelve Gyroscope Parameters

$b_{a_R}, b_{a_P}, b_{a_Y}$—roll-, pitch-, and yaw-axis components of gyroscope bias.

$\mathbf{M}_a$—nine elements of the 3×3 gyroscope scale factor and misalignment matrix.

Two Receiver Clock Parameters

Δp_{Clock}—normalized clock pseudorange error (equivalent clock bias error in seconds times c).

Δv_{Clock}—normalized clock pseudorange rate error (equivalent clock drift rate error in seconds per second times c).

Twenty-Four (or More) GPS Timing Errors

$\Delta \rho_{SAi}$—pseudorange error for ith GPS satellite, in meters.

In the minimum configuration (24 satellites active), 26 of the 65 state variables are for GPS, and the remaining 39 are for the INS. The pseudorange and velocity units for the 24 GPS system state variables are chosen to avoid factors of $c \approx 3 \times 10^8$ m/s that could cause scaling problems in numerical processing of the covariance matrix **P**.

Error Modeling with Temporary State Variables It is possible (and even desirable) to reduce the number of state variables to include just those satellites currently being used by the receiver. This approach does not apply to momentary loss of a satellite for a few seconds (or less) due to signal blockage or multipath interference but to the long-term loss of a satellite availability for minutes or hours when it passes behind the earth. When the latter occurs, the corresponding state variable for satellite pseudorange offset can be expunged from the state vector, along with the corresponding element of the measurement vector and the corresponding rows and columns of **P**, **Q**, **H**, and **R**. Similarly, when a new satellite first becomes available (after "satelliterise"), its pseudorange offset can be added as a new state variable with corresponding changes to the measurement vector. This approach changes the dimensions of the measurement vector **z**, state vector $\hat{\mathbf{x}}$, state transition matrix $\mathbf{\Phi}$, measurement sensitivity matrix **H**, measurement noise covariance matrix **R**, and dynamic disturbance noise covariance **Q**—all of which will have pronounced effects on the programming of the Kalman filter but which we do not consider here. Setting the associated measurement sensitivity submatrix to zero has exactly the same effect, with less programming agony but potentially greater impact on throughput and memory requirements.

Momentary loss of a satellite is usually modeled by zeroing the corresponding row(s) of the measurement sensitivity matrix, even in implementations using temporary state variables for the satellites currently being tracked by the receiver.

Rotation Vectors as Attitude Variables The coordinate transformation matrix from RPY to ENU coordinates is represented in terms of the equivalent rotation vector

$$\boldsymbol{\rho}_{\text{ENU}} \stackrel{\text{def}}{=} \begin{bmatrix} \rho_E \\ \rho_N \\ \rho_U \end{bmatrix} \tag{8.39}$$

in ENU coordinates. We use the rotation vector to represent attitude because attitude is three dimensional and the higher dimensioned representations (quaternions, coordinate transformation matrices) use redundant parameters that would cause the system state vector to be underdetermined and Euler angles to have effective "gimbal lock" problems. This complicates the model derivations a bit, but it avoids observability problems in the analysis.

Sensor Parameter Model Cluster-level sensor calibration is used to compensate for the biases, scale factor errors, and input axis misalignments of three-axis sensor suites (e.g., accelerometers or rate gyroscopes), often with a model in the form

$$\mathbf{v}_{\text{output}} = \mathbf{b} + \mathbf{M}_{\text{cal}}\mathbf{v}_{\text{input}}, \tag{8.40}$$

where $\mathbf{b}$ is the vector of biases and $\mathbf{M}$ is the calibration matrix of scale factors and misalignments. The 3 components of the bias vector $\mathbf{b}$ and 9 elements of the calibration matrix $\mathbf{M}_{\text{cal}}$ make up the 15 sensor parameters for each type of sensor (i.e., accelerometer or gyroscope), making a total of 30 sensor parameters in the system state vector used for analysis.

For instrument compensation, it is usually preferred to have the model in compensation form, as

$$\mathbf{v}_{\text{input}} = \mathbf{M}_{\text{comp}}\left(\mathbf{v}_{\text{out}} - \text{b}\right),$$

where the misalignment and scale factor compensation matrix would be

$$\mathbf{M}_{\text{comp}} = \mathbf{M}_{\text{cal}}^{-1}.$$

However, for the purposes of covariance analysis, it will be more convenient to use the calibration form of the scale factor and misalignment matrix shown in Eq. 8.40.

8.6.3.2 Measurement Model

The 30+ equivalent sensors include 3 accelerometers, 3 gyroscopes, and 24 or more potential GPS receiver channels, the outputs of which are pseudoranges and Doppler rates to the satellites in view. The GPS receiver can only track the satellites in view, and we model this by setting to zero the measurement sensitivities of receiver channels to GPS satellites occluded by the earth.

Measurement Variables The elements of the measurement vector for this model will be

a_R, a_P, a_Y—roll-, pitch-, and yaw-axis accelerometer outputs;
$\omega_R, \omega_P, \omega_Y$—roll-, pitch- and yaw-axis gyroscope outputs;
ρ_i—GPS receiver pseudorange output for ith GPS satellite, if available; and
$\dot{\rho}_i$—GPS receiver Doppler output for ith GPS satellite, if available.

Acceleration Sensitivities The sensitivities of the outputs of vehicle-fixed accelerometers to the state variables are governed by the equations

$$\mathbf{a}_{\text{ENU}} = \mathbf{C}_{\text{RPY}}^{\text{ENU}}(\boldsymbol{\rho}_{\text{ENU}})\mathbf{a}_{\text{RPY}}, \tag{8.41}$$

$$\begin{aligned}\mathbf{C}_{\text{RPY}}^{\text{ENU}} &= [\mathbf{C}_{\text{ENU}}^{\text{RPY}}(\boldsymbol{\rho}_{\text{ENU}})]^{\text{T}} \\ &= \cos(|\boldsymbol{\rho}_{\text{ENU}}|)\mathbf{I}_3 + \frac{1-\cos(|\boldsymbol{\rho}_{\text{ENU}}|)}{|\boldsymbol{\rho}_{\text{ENU}}|^2}\boldsymbol{\rho}_{\text{ENU}}\boldsymbol{\rho}_{\text{ENU}}^{\text{T}} \\ &\quad - \frac{\sin(|\boldsymbol{\rho}_{\text{ENU}}|)}{|\boldsymbol{\rho}_{\text{ENU}}|}\begin{bmatrix} 0 & -\rho_U & \rho_N \\ \rho_U & 0 & -\rho_E \\ -\rho_N & \rho_E & 0 \end{bmatrix} \quad \text{(Eq. C.112)},\end{aligned} \tag{8.42, 8.43}$$

$$\mathbf{a}_{\text{RPY}} = \mathbf{M}_a(\mathbf{a}_{\text{measured}} - \mathbf{b}_a), \tag{8.44}$$

so that the dependence of the accelerometer output vector on the state variables can be expressed as

$$\begin{aligned}\mathbf{a}_{\text{measured}} &= \mathbf{b}_a + \mathbf{M}_a\mathbf{a}_{RPY} \\ &= \mathbf{b}_a + \mathbf{M}_a^{-1}\left(\cos(|\boldsymbol{\rho}_{\text{ENU}}|)\mathbf{I}_3 + \frac{1-\cos(|\boldsymbol{\rho}_{\text{ENU}}|)}{|\boldsymbol{\rho}_{\text{ENU}}|^2}\boldsymbol{\rho}_{\text{ENU}}\boldsymbol{\rho}_{\text{ENU}}^{\text{T}}\right. \\ &\quad \left. - \frac{\sin(|\boldsymbol{\rho}_{\text{ENU}}|)}{|\boldsymbol{\rho}_{\text{ENU}}|}[\boldsymbol{\rho}_{\text{ENU}}\otimes]\right)\mathbf{a}_{\text{ENU}} \qquad (8.46) \\ &= \mathbf{h}_a(\boldsymbol{\rho}_{\text{ENU}}, \mathbf{a}_{\text{ENU}}, \mathbf{b}_a, \mathbf{M}_a), \qquad (8.47)\end{aligned}$$

a function linear in the state variables $\mathbf{a}_{\text{ENU}}$, $\mathbf{b}_a$, and $\mathbf{M}_a$ and nonlinear in $\boldsymbol{\rho}_{\text{ENU}}$.

GPS Pseudorange Sensitivities Pseudorange output from the GPS receiver will be with respect to the receiver antenna, whereas the position estimated by the INS system is usually a physical reference point within the sensor cluster. It is necessary to take the relative antenna offsets into account in the Kalman filter implementation. In addition, for high-performance vehicles that use more than one GPS antenna to maintain tracking during inverted maneuvers, it is necessary to switch offsets whenever the antenna is switched. The measurement sensitivities for pseudorange were derived in Section 7.2.3.2.

Attitude Rate Sensitivities The inputs of the gyroscopes are related to the state vector components through Eq. C.148, which we can put into the form

$$\dot{\boldsymbol{\rho}}_{\text{ENU}} = \frac{\partial\dot{\boldsymbol{\rho}}}{\partial\boldsymbol{\omega}_{\text{RPY}}}\boldsymbol{\omega}_{\text{RPY}} + \frac{\partial\dot{\boldsymbol{\rho}}}{\partial\boldsymbol{\omega}_{\text{ENU}}}\boldsymbol{\omega}_{\text{ENU}}, \tag{8.48}$$

where $\boldsymbol{\omega}_{\text{RPY}}$ is the vector of inputs to the gyroscopes and the inertial rotation rate for ENU coordinates is

$$\boldsymbol{\omega}_{\text{ENU}} = \begin{bmatrix} -\dfrac{v_N}{r_M + h} \\ \dfrac{v_E}{r_T + h} + \omega_{\oplus} \cos(\phi_{\text{geodetic}}) \\ \omega_{\oplus} \sin(\phi_{\text{geodetic}}) \end{bmatrix}, \tag{8.49}$$

where $\omega_{\oplus}$ = earth rotation rate
ϕ_{geodetic} = geodetic latitude
v_E = east component of velocity with respect to the surface of the earth
r_T = transverse radius of curvature of the ellipsoid (Eq. 6.41)
v_N = north component of velocity with respect to the surface of the earth.
r_M = meridional radius of curvature of the ellispoid (Eq. 6.38)
h = altitude above (+) or below (−) the reference ellispoid surface (≈mean sea level)

The deritative of $\boldsymbol{\omega}_{\text{ENU}}$ (Eq. 8.48) with respect to time will be needed in the implementation and can be derived by differentiating Eq. 8.48 as

$$\dot{\boldsymbol{\omega}}_{\text{ENU}} = \begin{bmatrix} -\dfrac{a_N}{r_M + h} + \dfrac{v_N(\dot{r}_M + v_U)}{(r_M + h)^2} \\ \dfrac{a_E}{r_T + h} - \dfrac{v_E(\dot{r}_T + v_U)}{(r_T + h)^2} - \dfrac{\omega_{\oplus} \sin(\phi_{\text{geodetic}}) v_N}{r_M + h} \\ \dfrac{\omega_{\oplus} \cos(\phi_{\text{geodetic}}) v_N}{r_M + h} \end{bmatrix}, \tag{8.50}$$

where a_E = east component of acceleration with respect to the surface of the earth
a_N = north component of acceleration with respect to the surface of the earth
v_U = vertical component of velocity with respect to the surface of the earth
$\dot{r}_M$ = time derivative of meridional radius of curvature (this effect is usually ignored but can be calculated by taking the time derivative of Eq. 6.38)
$\dot{r}_T$ = time derivative of transverse radius of curvature (this effect is also ignored, as a rule, but can be calculated by taking the time derivative of Eq. 6.41)

Equation 8.48 can be put into the form

$$\boldsymbol{\omega}_{\mathrm{RPY}} = \left(\frac{\partial\dot{\boldsymbol{\rho}}}{\partial\boldsymbol{\omega}_{\mathrm{RPY}}}\right)^{-1}\left\{\dot{\boldsymbol{\rho}} - \frac{\partial\dot{\boldsymbol{\rho}}}{\partial\boldsymbol{\omega}_{\mathrm{ENU}}}\begin{bmatrix} -\dfrac{v_N}{r_M + h} \\ \dfrac{v_E}{r_T + h} + \omega_{\oplus}\cos(\phi_{\mathrm{geodetic}}) \\ \omega_{\oplus}\sin(\phi_{\mathrm{geodetic}}) \end{bmatrix}\right\}, \tag{8.51}$$

where the matrices of partial derivatives are given in Eqs. C.151 and C.154, the matrix inverse

$$\left(\frac{\partial\dot{\boldsymbol{\rho}}}{\partial\boldsymbol{\omega}_{\mathrm{RPY}}}\right)^{-1} = \frac{\sin(|\boldsymbol{\rho}_{\mathrm{ENU}}|)}{|\boldsymbol{\rho}_{\mathrm{ENU}}|}\mathbf{I} - \left(\frac{|\boldsymbol{\rho}_{\mathrm{ENU}}|^2 - \sin(|\boldsymbol{\rho}_{\mathrm{ENU}}|)}{|\boldsymbol{\rho}_{\mathrm{ENU}}|^3}\right)\boldsymbol{\rho}_{\mathrm{ENU}}\boldsymbol{\rho}_{\mathrm{ENU}}^{\mathrm{T}}$$
$$- \frac{1 - \cos(|\boldsymbol{\rho}_{\mathrm{ENU}}|)}{|\boldsymbol{\rho}_{\mathrm{ENU}}|^2}[\boldsymbol{\rho}_{\mathrm{ENU}}\otimes] \tag{8.52}$$

$$= \theta\mathbf{1}_{\rho}\mathbf{1}_{\rho}^{\mathrm{T}} + \frac{\sin(\theta)}{\theta}[\mathbf{I} - \mathbf{1}_{\rho}\mathbf{1}_{\rho}^{\mathrm{T}}] - \frac{1 - \cos(\theta)}{\theta}[\mathbf{1}_{\rho}\otimes] \tag{8.53}$$

$$\theta \stackrel{\mathrm{def}}{=} |\boldsymbol{\rho}_{\mathrm{ENU}}| \tag{8.54}$$

$$\mathbf{1}_{\rho} \stackrel{\mathrm{def}}{=} \frac{\boldsymbol{\rho}_{\mathrm{ENU}}}{|\boldsymbol{\rho}_{\mathrm{ENU}}|}, \tag{8.55}$$

and the matrix fraction

$$\frac{\partial\dot{\boldsymbol{\rho}}/\partial\boldsymbol{\omega}_{\mathrm{RPY}}}{\partial\dot{\boldsymbol{\rho}}/\partial\boldsymbol{\omega}_{\mathrm{ENU}}} = -\cos(|\boldsymbol{\rho}_{\mathrm{ENU}}|)\mathbf{I} - \frac{1 - \cos(|\boldsymbol{\rho}_{\mathrm{ENU}}|)}{|\boldsymbol{\rho}_{\mathrm{ENU}}|^2}\boldsymbol{\rho}_{\mathrm{ENU}}\boldsymbol{\rho}_{\mathrm{ENU}}^{\mathrm{T}}$$
$$+ \frac{\sin(|\boldsymbol{\rho}_{\mathrm{ENU}}|)}{|\boldsymbol{\rho}_{\mathrm{ENU}}|}[\boldsymbol{\rho}_{\mathrm{ENU}}\otimes] \tag{8.56}$$

$$= -\cos(\theta)\mathbf{I} - [1 - \cos(\theta)]\mathbf{1}_{\rho}\mathbf{1}_{\rho}^{\mathrm{T}} + \sin(\theta)[\mathbf{1}_{\rho}\otimes]. \tag{8.57}$$

The full nonlinear functional dependence of the gyroscope ouputs on the state variables can then be expressed in the form[4]

$$\boldsymbol{\omega}_{\text{output}} = \mathbf{h}_{\omega}(\dot{\boldsymbol{\rho}}, \boldsymbol{\rho}, v_E, v_N, h, \phi_{\text{geodetic}}, \mathbf{M}_{\text{cal}}, \mathbf{b}_{\text{gyro}}) \tag{8.58}$$

$$= \mathbf{b}_{\text{gyro}} + \mathbf{M}_{\text{cal}}\{\boldsymbol{\omega}_{\text{RPY}}\} \tag{8.59}$$

$$\begin{aligned} &= \mathbf{b}_{\text{gyro}} + \mathbf{M}_{\text{cal}} \\ &\quad \times \left\{ \left[\frac{\sin(|\boldsymbol{\rho}_{\text{ENU}}|)}{|\boldsymbol{\rho}_{\text{ENU}}|}\mathbf{I} - \left(\frac{|\boldsymbol{\rho}_{\text{ENU}}|^2 - \sin(|\boldsymbol{\rho}_{\text{ENU}}|)}{|\boldsymbol{\rho}_{\text{ENU}}|^3} \right) \boldsymbol{\rho}_{\text{ENU}}\boldsymbol{\rho}_{\text{ENU}}^{\text{T}} \right. \right. \\ &\quad \left. - \frac{1 - \cos(|\boldsymbol{\rho}_{\text{ENU}}|)}{|\boldsymbol{\rho}_{\text{ENU}}|^2}[\boldsymbol{\rho}_{\text{ENU}}\otimes] \right] \dot{\boldsymbol{\rho}} \\ &\quad - \left[\frac{-\cos(|\boldsymbol{\rho}_{\text{ENU}}|)}{|\boldsymbol{\rho}_{\text{ENU}}|^2}\boldsymbol{\rho}_{\text{ENU}}\boldsymbol{\rho}_{\text{ENU}}^{\text{T}} + \frac{\sin(|\boldsymbol{\rho}_{\text{ENU}}|)}{|\boldsymbol{\rho}_{\text{ENU}}|}[\boldsymbol{\rho}_{\text{ENU}}\otimes] \right] \\ &\quad \left. \times \begin{bmatrix} \dfrac{-v_N}{r_M + h} \\ \dfrac{v_E}{r_T + h} + \omega_{\oplus}\cos(\phi_{\text{geodetic}}) \\ \omega_{\oplus}\sin(\phi_{\text{geodetic}}) \end{bmatrix} \right\}. \end{aligned} \tag{8.60}$$

Equation 8.60 is the nonlinear model for the dependence of the gyroscope inputs on the other state variables. Its partial derivatives with respect to the state variables are those used in the extended Kalman filter implementation.

8.6.3.3 State Dynamics

The dynamics of rotation vector models are somewhat nonstandard, but the rest of the dynamic model is relatively straightforward.

Vehicle Dynamic Model This part of the dynamic model is fairly standard, with

$$\begin{aligned} \frac{d}{dt}\theta &= \text{longitude rate, given by Eq. 6.42,} \\ \frac{d}{dt}\phi &= \text{latitude rate, given by Eq. 6.39,} \\ \frac{d}{dt}h &= v_U \quad \text{(altitude rate),} \\ \dot{\mathbf{v}}_{\text{ENU}} &= \mathbf{a}_{\text{ENU,}} \\ \frac{d}{dt}\mathbf{a}_{\text{ENU}} &= \mathbf{w}_{\dot{a}} \quad \text{(white noise),} \\ \frac{d}{dt}\boldsymbol{\rho}_{\text{ENU}} &= \dot{\boldsymbol{\rho}}_{\text{ENU.}} \end{aligned}$$

[4] The meridional and transverse curvatures are also functions of geodetic latitude, but the sensitivities are usually considered to be too weak to be taken seriously.

The second derivative of the rotation vector can be derived by taking the time derivative of Eq. 8.48:

$$\ddot{\boldsymbol{\rho}}_{\text{ENU}} = \left[\frac{d}{dt}\frac{\partial\dot{\boldsymbol{\rho}}_{\text{ENU}}}{\partial\boldsymbol{\omega}_{\text{RPY}}}\right]\boldsymbol{\omega}_{\text{RPY}} + \left[\frac{d}{dt}\frac{\partial\dot{\boldsymbol{\rho}}_{\text{ENU}}}{\partial\boldsymbol{\omega}_{\text{ENU}}}\right]\boldsymbol{\omega}_{\text{ENU}} + \left[\frac{\partial\dot{\boldsymbol{\rho}}_{\text{ENU}}}{\partial\boldsymbol{\omega}_{\text{ENU}}}\right]\left[\frac{d}{dt}\boldsymbol{\omega}_{\text{ENU}}\right] + \underbrace{\frac{\partial\dot{\boldsymbol{\rho}}_{\text{ENU}}}{\partial\boldsymbol{\omega}_{\text{RPY}}}}_{\mathbf{G}_{\ddot{\rho}}}\underbrace{\dot{\boldsymbol{\omega}}_{\text{RPY}}}_{\mathbf{w}_{\ddot{\omega}}} \tag{8.61}$$

$$= \mathbf{f}(\boldsymbol{\rho}_{\text{ENU}}, \dot{\boldsymbol{\rho}}_{\text{ENU}}, \mathbf{v}_{ENU}, \mathbf{a}_{ENU}, \phi) + \mathbf{G}_{\ddot{\rho}}\mathbf{w}_{\ddot{\omega}}, \tag{8.62}$$

where

$\boldsymbol{\omega}_{\text{RPY}}$ is the input to the gyroscopes, which can be computed from Eq. 8.58 for real-time implementation,

$\dfrac{\partial\dot{\boldsymbol{\rho}}_{\text{ENU}}}{\partial\boldsymbol{\omega}_{\text{ENU}}}$ is given by Eq. C.154;

$\dfrac{\partial\dot{\boldsymbol{\rho}}_{\text{ENU}}}{\partial\boldsymbol{\omega}_{\text{RPY}}}$ is given by Eq. C.151;

$\dfrac{d}{dt}\dfrac{\partial\dot{\boldsymbol{\rho}}_{\text{ENU}}}{\partial\boldsymbol{\omega}_{\text{ENU}}}$ is given by Eq. C.181;

$\dfrac{d}{dt}\dfrac{\partial\dot{\boldsymbol{\rho}}_{\text{ENU}}}{\partial\boldsymbol{\omega}_{\text{RPY}}}$ is given by Eq. C.173;

$\boldsymbol{\omega}_{\text{ENU}}$ is given by Eq. 8.49;

$\dot{\boldsymbol{\omega}}_{\text{ENU}}$ is given by Eq. 8.50; and

$\dot{\boldsymbol{\omega}}_{\text{RPY}}$ is essentially the attitude dynamic disturbance noise in RPY coordinates, a random process.

Equation 8.62 is in the standard form for nonlinear dynamics with additive process noise $\mathbf{w}_{\ddot{\boldsymbol{\omega}}}(t)$ in RPY coordinates and a process noise distribution matrix $\mathbf{G}_{\ddot{\rho}}$. The angular acceleration process noise covariance matrix

$$\mathbf{Q}_{\ddot{\boldsymbol{\omega}}} = E\left\langle \mathbf{w}_{\ddot{\boldsymbol{\omega}}}\mathbf{w}_{\ddot{\boldsymbol{\omega}}}^{\text{T}} \right\rangle \tag{8.63}$$

generally depends on the vehicle type, with higher covariances for more agile vehicles. For some vehicles, the covariances will be ordered as $q_{\ddot{\omega}11} \geq q_{\ddot{\omega}22} \geq q_{\ddot{\omega}33}$, with random roll maneuvers dominating pitch maneuvers and pitch maneuvers dominating yaw maneuvers.

Equation C.194 in Appendix C is a general formula for taking the partial derivatives of the nonlinear dynamic function in Eq. 8.62 with respect to $\boldsymbol{\rho}_{\text{ENU}}$. Some of these derivatives will be submatrices of the extended Kalman filter dynamic coefficient matrix $\mathbf{F}$.

Parameter Dynamics All the sensor parameters can be modeled as random walks or, if a reasonable correlation time is known, as exponentially correlated processes.

Pseudorange Offset Dynamics These tend to be dominated by SA, which can be modeled as an exponentially correlated process with time constant on the order of a minute. An independent state vector and process model is required for each GPS satellite used.

Problems

8.1 Show that the matrix $\mathbf{C}_{P_{k+1}}$ defined by Eq. 8.18 is the square triangular Cholesky factor of $\mathbf{P}_{k+1}$, the covariance matrix of prediction error.

8.2 Show that the matrix $\overline{\mathbf{K}}_k$ defined by Eq. 8.19 is the Kalman gain matrix.

8.3 If $\mathbf{C}$ is a Cholesky factor of $\mathbf{P}$, is the block matrix $[\mathbf{C}|0]$ (i.e., padded with any number of zero columns on the right) also a Cholesky factor? How about $[0|\mathbf{C}]$?

8.4 Calculate the partial derivatives of the function $\mathbf{h}$ (two components) in Eq. 8.38 with respect to magnetic compass bias ($\delta\alpha$) and wheel speed sensor scale factor offset ($\delta S_{\text{wheelspeed}}$). (There should be four partial derivatives in the resulting 2×2 submatrix.)

Global Positioning Systems, Inertial Navigation, and Integration,
Mohinder S. Grewal, Lawrence R. Weill, Angus P. Andrews

Print ISBN 0-471-35032-X Electronic ISBN 0-471-20071-9

9
Differential GPS

9.1 INTRODUCTION

Differential GPS (DGPS) is a technique for reducing the error in GPS-derived positions by using additional data from a reference GPS receiver at a known position. The most common form of DGPS involves determining the combined effects of navigation message ephemeris and satellite clock errors [including the effects of selective availability (SA), if active] at a reference station and transmitting pseudorange corrections, in real time, to a user's receiver. The receiver applies the corrections in the process of determining its position [63]. This results in the following:

- Some error sources are canceled completely:
  (a) selective availability and
  (b) satellite ephemeris and clock errors.
- With other error sources, cancelation degrades with distance:
  (a) ionospheric delay error and
  (b) tropospheric delay error.
- Still other error sources are not canceled at all:
  (a) multipath errors and
  (b) receiver errors.

9.2 LADGPS, WADGPS, AND WAAS

9.2.1 Description of Local-Area DGPS (LADGPS)

LADGPS is a form of DGPS in which the user's GPS receiver receives real-time pseudorange and, possibly, carrier phase corrections from a reference receiver generally located within the line of sight. The corrections account for the combined effects of navigation message ephemeris and satellite clock errors (including the effects of SA) and, usually, atmospheric propagation delay errors at the reference station. With the assumption that these errors are also common to the measurements made by the user's receiver, the application of the corrections will result in more accurate coordinates [81].

9.2.2 Description of Wide-Area DGPS (WADGPS)

WADGPS is a form of DGPS in which the user's GPS receiver receives corrections determined from a network of reference stations distributed over a wide geographical area. Separate corrections are usually determined for specific error sources, such as satellite clock, ionospheric propagation delay, and ephemeris. The corrections are applied in the user's receiver or attached computer in computing the receiver's coordinates. The corrections are typically supplied in real time by way of a geostationary communications satellite or through a network of ground-based transmitters. Corrections may also be provided at a later date for post-processing collected data [81].

9.2.3 Description of Wide Area Augmentation System (WAAS)

WAAS enhances the GPS SPS and is available over a wide geographical area. The WAAS being developed by the Federal Aviation Administration, together with other agencies, will provide WADGPS corrections, additional ranging signals from geostationary (GEO) satellites, and integrity data on the GPS and GEO satellites [81].

The GEO Uplink Subsytem includes a closed-loop control algorithm and special signal generator hardware. These ensure that the downlink signal to the users is controlled adequately to be used as a ranging source to supplement the GPS satellites in view.

The primary mission of WAAS is to provide a means for air navigation for all phases of flight in the National Airspace System (NAS) from departure, en route, arrival, and through approach. GPS augmented by WAAS offers the capability for both nonprecision approach (NPA) and precision approach (PA) within a specific service volume. A secondary mission of the WAAS is to provide a WAAS network time (WNT) offset between the WNT and Coordinated Universal Time (UTC) for nonnavigation users.

WAAS provides improved en route navigation and PA capability to WAAS certified avionics. The safety critical WAAS system consists of the equipment and

software necessary to augment the Department of Defense (DoD) provided GPS SPS. WAAS provides a signal in space (SIS) to WAAS certified aircraft avionics using the WAAS for any FAA-approved phase of flight. The SIS provides two services: (1) data on GPS and GEO satellites and (2) a ranging capability.

The GPS satellite data is received and processed at widely dispersed wide-area reference Stations (WRSs), which are strategically located to provide coverage over the required WAAS service volume. Data is forwarded to wide-area master stations (WMSs), which process the data from multiple WRSs to determine the integrity, differential corrections, and residual errors for each monitored satellite and for each predetermined ionospheric grid point (IGP). Multiple WMSs are provided to eliminate single-point failures within the WAAS network. Information from all WMSs is sent to each GEO uplink subsystem (GUS) and uplinked along with the GEO navigation message to GEO satellites. The GEO satellites downlink this data to the users via the GPS SPS L-band ranging signal (L_1) frequency with GPS-type modulation. Each ground-based station/subsystem communicates via a terrestrial communications subsystem (TCS). See Fig. 9.1.

In addition to providing augmented GPS data to the users, WAAS verifies its own integrity and takes any necessary action to ensure that the system meets the WAAS performance requirements. WAAS also has a system operation and maintenance function that provides status and related maintenance information to FAA airway facilities (AFs) NAS personnel.

WAAS has a functional verification system (FVS) that is used for early development test and evaluation (DT&E), refinement of contractor site installation procedures, system-level testing, WAAS operational testing, and long-term support for WAAS.

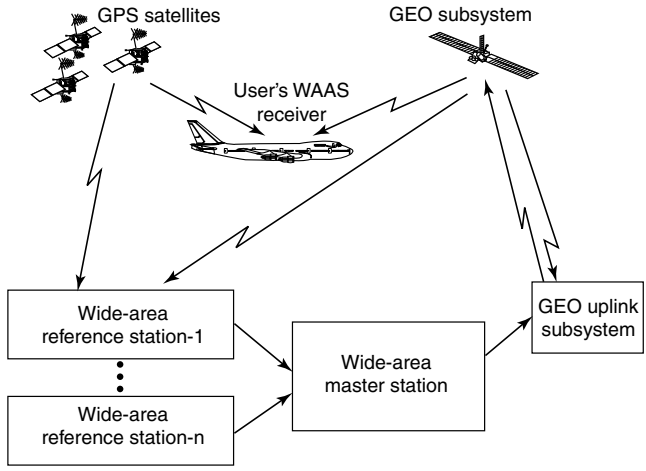


Fig. 9.1 *WAAS Top Level View*

Correction and Verification (C&V) processes data from all WRSs to determine integrity, differential corrections, satellite orbits, and residual error bounds for each monitored satellite. It also determines ionospheric vertical delays and their residual error bounds at each of the IGPs. C&V schedules and formats WAAS messages and forwards them to the GUSs for broadcast to the GEO satellites.

C&V's capabilities are as follows:

1. Control C&V Operations and Maintenance (COM) supports the transfer of files, performs remotely initiated software configuration checks, and accepts requests to start and stop execution of the C&V application software.
2. Control C&V Modes (CMD) manage mode transitions in the C&V subsystem while the application software is running.
3. Monitor C&V (MCV) reports line replaceable unit (LRU) faults and configuration status. In addition, it monitors software processes and provides performance data for the local C&V subsystems.
4. Process Input Data (PID) selects and monitors data from the wide-area reference equipment (WREs). Data that passes PID screening is repackaged for other C&V capabilities. PID performs clock and L_1 GPS Precision Positioning Service L-band ranging signal (L_2) receiver bias calculations, cycle slip detection, outlier detection, data smoothing, and data monitoring. In addition, PID calculates and applies the windup correction to the carrier phase, accumulates data to estimate the pseudorange to carrier phase bias, and computes the ionosphere corrected carrier phase and measured slant delay.
5. Satellite Orbit Determination (SOD) determines the GPS and GEO satellite orbits and clock offsets, WRE receiver clock offsets, and troposphere delay.
6. Ionosphere Correction Computation (ICC) determines the L_1 IGP vertical delays, grid ionosphere vertical error (GIVE) for all defined IGPs, and L_1–L_2 interfrequency bias for each satellite transmitter and each WRS receiver.
7. Satellite Correction Processing (SCP) determines the fast and long-term satellite corrections, including the user differential range error (UDRE). It determines the WNT and the GEO and WNT clock steering commands [99].
8. Independent Data Verification (IDV) compares satellite corrections, GEO navigation data, and ionospheric corrections from two independent computational sources, and if the comparisons are within limits, one source is selected from which to build the WAAS messages. If the comparisons are not within limits, various responses may occur, depending on the data being compared, all the way from alarms being generated to the C&V being faulted.
9. Message Output Processing (MOP) transmits messages containing independently verified results of C&V calculations to the GUS processing (GP) for broadcast.
10. C&V Playback (PLB) processes the playback data that has been recorded by the other C&V capabilities.

11. Integrity Data Monitoring (IDM) checks both the broadcast and the to-be-broadcast UDREs and GIVEs to ensure that they are properly bounding their errors. In addition, it monitors and validates that the broadcast messages are sent correctly. It also performs the WAAS time-to-alarm validation [1, 99].

9.2.3.1 WRS Algorithms Each WRS collects raw pseudorange (PR) and accumulated delta range (ADR) measurements from GPS and GEO satellites selected for tracking. Each WRS performs smoothing on the measurements and corrects for atmospheric effects, that is, ionospheric and tropospheric delays. These smoothed and atmospherically corrected measurements are provided to the WMS.

9.2.3.2 WMS Foreground (Fast) Algorithms The WMS foreground algorithms are applicable to real-time processing functions, specifically the computation of fast correction, determination of satellite integrity status and WAAS message formatting. This processing is done at a 1-HZ rate.

9.2.3.3 WMS Background (Slow) Algorithms The WMS background processing consists of algorithms that estimate slowly varying parameters. These algorithms consist of WRS clock error estimation, grid ionospecific delay computation, broadcast ephemeris computation, satellite orbit determination, satellite ephemeris error computation, and satellite visibility computation.

9.2.3.4 Independent Data Verification and Validation Algorithms This includes a set of WRS and at least one WMS, which enable monitoring the integrity status of GPS and the determination of wide-area DGPS correction data. Each WRS has three dual frequency GPS receivers to provide parallel sets of measurement data. The presence of parallel data streams enables Independent Data Verification and Validation (IDV&V) to be employed to ensure the integrity of GPS data and their corrections in the WAAS messages broadcast via one or more GEOs. With IDV&V active, the WMS applies the corrections computed from one stream to the data from the other stream to provide verification of the corrections prior to transmission. The primary data stream is also used for the validation phase to check the active (already broadcast) correction and to monitor their SIS performance. These algorithms are continually being improved. The latest versions can be found in references [48, 96, 97, 137, 99] and [98, pp. 397–425].

9.3 GEO UPLINK SUBSYSTEM (GUS)

Corrections from the WMS are sent to the ground uplink subsystem (GUS) for uplink to the GEO. The GUS receives integrity and correction data and WAAS specific messages from the WMS, adds forward error correction (FEC) encoding, and transmits the messages via a C-band uplink to the GEO satellites for broadcast to the WAAS user. The GUS signal uses the GPS standard positioning service

waveform (C/A-code, BPSK modulation); however, the data rate is higher (250 bps). The 250 bps of data are encoded with a one-half rate convolutional code, resulting in a 500-symbols/s transmission rate.

Each symbol is modulated by the C/A-code, a 1.023×10^6-chips/s pseudo random sequence to provide a spread-spectrum signal. This signal is then BPSK modulated by the GUS onto an IF carrier, upconverted to a C-band frequency, and uplinked to the GEO. It is the C/A-code modulation that provides the ranging capability if its phase is properly controlled.

Control of the carrier frequency and phase is also required to eliminate uplink Doppler and to maintain coherence between code and carrier. The GUS monitors the C-band and L_1 downlinks from the GEO to provide closed-loop control of the PRN code and L_1 carrier coherency. WAAS short- and long-term code carrier coherence requirements are met.

9.3.1 Description of the GUS Algorithm

The GUS control loop algorithm "precorrects" the code phase, carrier phase, and carrier frequency of the GEO uplink signal to maintain GEO broadcast code–carrier coherence. The uplink effects such as ionospheric code–carrier divergence, uplink Doppler, equipment delays, and frequency offsets must be corrected in the GUS control loop algorithm.

Figure 9.2 provides an overview of the functional elements of the GUS control loop. The control loop contains algorithm elements (shaded boxes) and hardware elements that either provide inputs to the algorithm or are controlled or affected by outputs from the algorithm. The hardware elements include a WAAS GPS receiver, GEO satellite, and GUS signal generator.

Downlink ionospheric delay is estimated in the ionospheric delay and rate estimator using pseudorange measurements from the WAAS GPS receiver on L_1 and L_2 (downconverted from the GEO C-band downlink at the GUS). This is a two-state Kalman filter that estimates the ionospheric delay and delay rate.

At each measurement interval, a range measurement is taken and fed into the range, rate, and acceleration estimator. This measurement is the average between the reference pseudorange from the GUS signal generator PR_{sign} and the received pseudorange from the L_1 downlink as measured by the WAAS GPS Receiver (PR_{geo}) and adjusted for estimated ionospheric delay (PR_{iono}). The equation for the range measurement is then

$$z = \tfrac{1}{2}[(PR_{\text{geo}} - PR_{\text{iono}}) + PR_{\text{sign}}] - T_{C\text{up}} - T_{L1\text{dwn}S},$$

where $T_{C\text{up}} =$ C-band uplink delay (m)
$T_{L1\text{dwn}S} = L_1$ receiver delay of the GUS (m)

The GUS signal generator is initialized with a pseudorange value from satellite ephemeris data. This is the initial reference from which corrections are made.

The range, rate and acceleration estimator is a three-state Kalman filter that drives the frequency and code control loops.

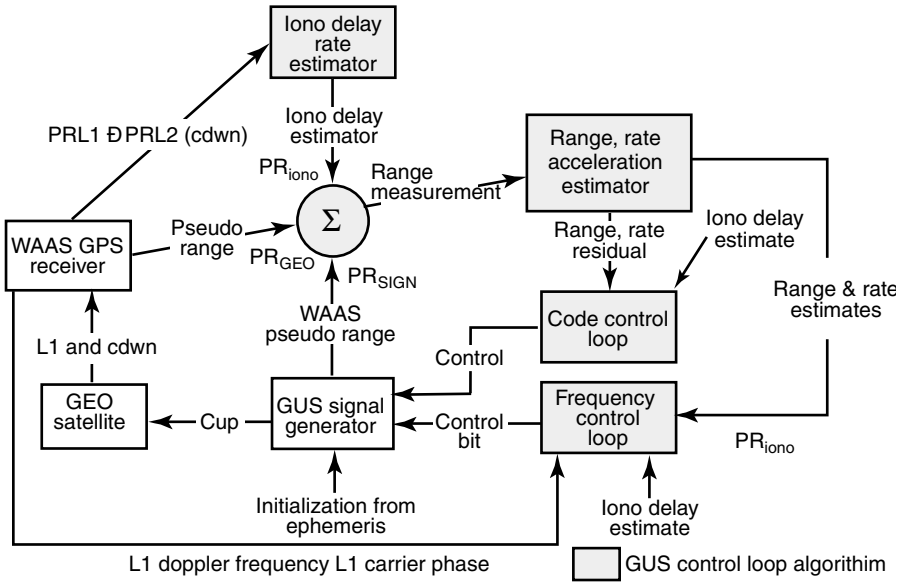


Fig. 9.2 *GUS control loop block diagram.*

The code control loop is a second-order control system. The error signal for this control system is the difference between the WAAS pseudorange (P_{rsign}) and the estimated pseudorange from the Kalman filter. The loop output is the code rate adjustments to the GUS signal generator.

The frequency control loop has two modes. First, it adjusts the signal generator frequency to compensate for uplink Doppler effects. This is accomplished using a first-order control system. The error signal input is the difference between the L_1 Doppler frequency from the WAAS GPS receiver and the estimated range rate (converted to a Doppler frequency) from the Kalman filter.

Once the frequency error is below a threshold value, the carrier phase is controlled. This is accomplished using a second-order control system. The error signal input to this system is the difference between the L_1 carrier phase and a carrier phase estimate based on the Kalman filter output. This estimated range is converted to carrier cycles using the range estimate at the time carrier phase control starts as a reference. Fine adjustments are made to the signal generator carrier frequency to maintain phase coherence [35, 47–49, 94].

9.3.2 In-Orbit Tests

Two separate series of in-orbit tests (IOTs) were conducted, one at the COMSAT GPS Earth Station (GES) in Santa Paula, California with Pacific Ocean Region (POR) and Atlantic Ocean Region-West (AOR-W) I-3 satellites and the other at the COMSAT GES in Clarksburg, Maryland, using AOR-W. The IOTs were conducted

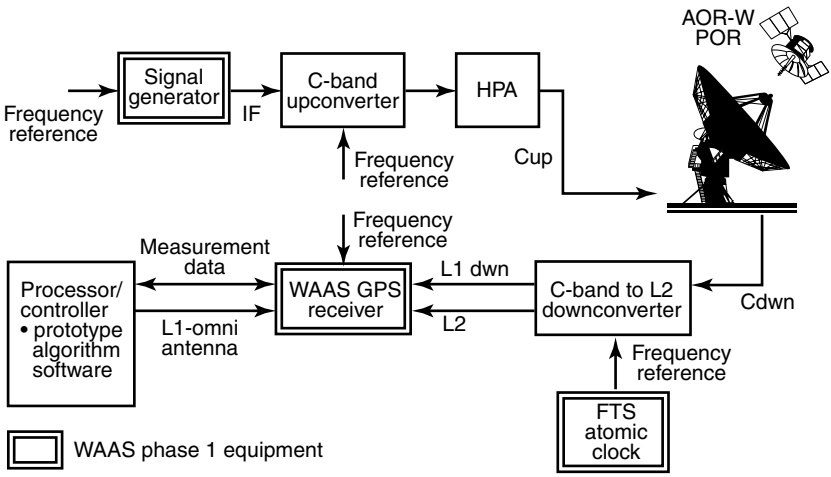


Fig. 9.3 *IOT test GUS setup.*

to validate a prototype version of the GUS control loop algorithm. Data was collected to verify the ionospheric estimation and code–carrier coherence performance capability of the control loop and the short–term carrier frequency stability of the I-3 satellites with a prototype ground station. The test results were also used to validate the GUS control loop simulation.

Figure 9.3 illustrates the IOT setup at a high level. Prototype ground station hardware and software were used to assess algorithm performance at two different ground stations with two different Inmarsat-3 satellites.

9.3.3 Ionospheric Delay Estimation

The GUS control loop estimates the ionospheric delay contribution of the GEO C-band uplink to maintain code–carrier coherence of the broadcast SIS. Figures 9.4–9.6 provide the delay estimates for POR using the Santa Paula GES and AOR-W using both the Santa Paula and Clarksburg GES. Each plot shows the estimated ionospheric delay (output of the two-state Kalman filter) versus the calculated delay using the L_1 and C pseudorange data from a WAAS GPS receiver. Calculated delay is noisier and varying about 1 m/s, whereas the estimated delay by the Kalman filter is right in middle of the measured delay, as shown in Figures 9.4–9.6. Delay measurements were calculated using the equation

$$\text{Ionospheric delay} = \frac{P_{RL1} - P_{RC} - tau\ L_1 + tau\ C}{1 - [L_1\ freq]^2/[C\ freq]^2}$$

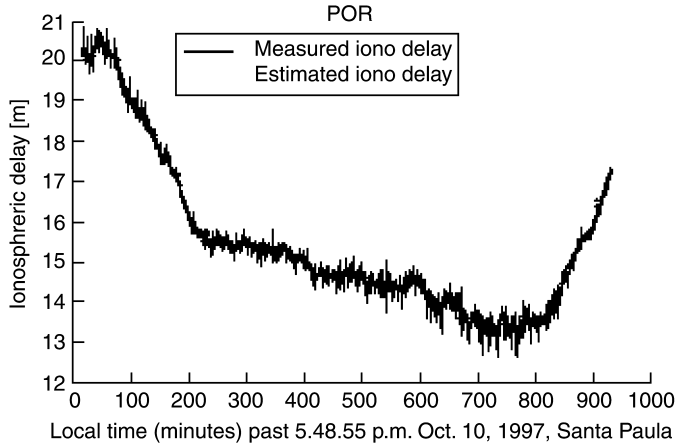


Fig. 9.4 *Measured and estimated ionospheric delay, POR, Santa Paula.*

where $P_{RL1} = L_1$ pseudorange (m)
$P_{RC} = C$ pseudorange (m)
$tau\ L_1 = L_1$ downlink delay (m)
$tau\ C = C$ downlink delay (m)
$L_1 freq = L_1$ frequency, $= 1575.42$ MHz
$Cfreq = C$ frequency, $= 3630.42$ MHz

The ionosphere during the IOTs was fairly benign with no high levels of solar activity observed. Table 9.1 provides the ionospheric delay statistics (in meters)

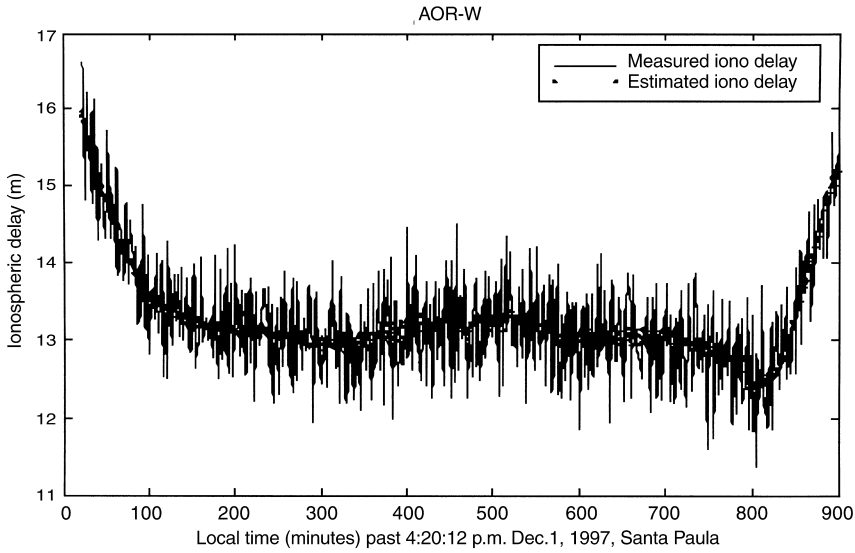


Fig. 9.5 *Measured and estimated ionospheric delay, AOR-W, Santa Paula.*

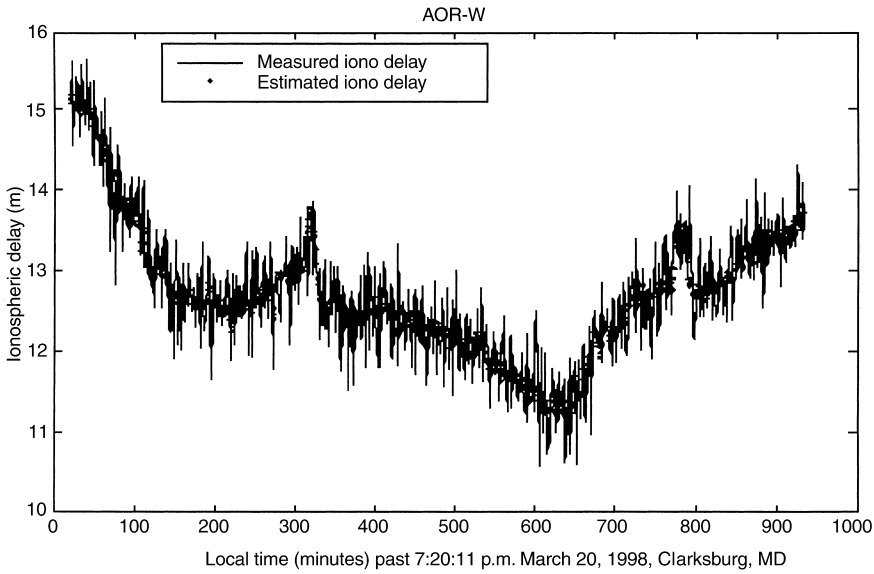


Fig. 9.6 *Measured and estimated ionospheric delay, Clarksburg.*

between the output of the ionospheric Kalman filter in the control loop, and the calculated delay from the WAAS GPS receiver's L_1 and L_2 pseudoranges. The statistics show that the loop's ionospheric delay estimation is very close (low RMS) to the ionospheric delay calculated using the measured pseudorange from the WAAS GPS receiver.

9.3.4 Code–Carrier Frequency Coherence

The GEO's broadcast code–carrier frequency coherence requirement is specified in the WAAS System Specification and Appendix A of reference [106]. It states:

> The lack of coherence between the broadcast carrier phase and the code phase shall be limited. The short term fractional frequency difference between the code phase rate and the carrier frequency will be less than 5×10^{-11}. That is,

$$\left| \frac{f_{code}}{1.023\ MHz} - \frac{f_{carrier}}{1575.42\ MHz} \right| < 5 \times 10^{-11}$$

TABLE 9.1 Observed RMS WAAS Ionospheric Correction Errors

In-Orbit Test	RMS Error (m)
Santa Paula GES, Oct. 10, 1997, POR	0.20
Santa Paula GES, Dec. 1, 1997, AOR-W	0.45
Clarksburg GES, Mar. 20, 1998, AOR-W	0.34

Over the long term, the difference between the broadcast code phase (1/1540) and the broadcast carrier phase will be within one carrier cycle, one sigma. This does not include code-carrier divergence due to inospheric refraction in the downlink propagation path.

For the WAAS program, short term is defined as less than 10 s and long term less than 100 s.

Pseudorange minus the ionospheric estimates averaged over τ seconds is expressed as

$$F_{\mathrm{PR}} = \frac{P_{RL1}(t) - \text{Ionoestimate } (t)}{\tau} \qquad \text{meters/s.}$$

Carrier phase minus the ionospheric estimate average over τ seconds is expressed as

$$F_{\mathrm{PH}} = \frac{-\phi_{L1}(t) + (\text{Ionoestimate } (t)/\lambda_{L1})}{\tau} \qquad \text{cycles/s.}$$

For long-term code–carrier coherence calculations, a τ of 60 s was chosen to mitigate receiver bias errors in the pseudorange and carrier phase measurements of the WAAS GPS receiver. For short-term code–carrier coherence a shorter 30-s averaging time was selected. The code–carrier coherence requirement is specified at the output of the GEO and not the receiver, so data averaging has to be employed to back out receiver effects such as multipath and noise. Each averaging time was based upon analyzing GPS satellite code–carrier coherence data and selecting the minimum averaging time required for GPS to meet the WAAS code–carrier coherence requirements.

For long term code–carrier coherence calculations, the difference between the pseudorange and the phase measurements is given by

$$\Delta_{\mathrm{PR-PH}} = [F_{\mathrm{PR}}/\lambda_{L1}] - F_{\mathrm{PH}} \qquad \text{cycles/s,}$$

where λ_{L1} is the wavelength of the L_1 carrier frequency and "long term coherence" equals $|\Delta_{\mathrm{PR-PH}}(t + 100) - \Delta_{\mathrm{PR-PH}}(t)|$ cycles.

For short-term code–carrier coherence calculations, the difference between the pseudorange and the phase measurements is given by

$$\delta_{\mathrm{PR-PH}} = \frac{F_{\mathrm{PR}} - F_{\mathrm{PH}}}{10 \times c \text{ (speed of light)}}$$

and "short term coherence" is $|\delta_{\mathrm{PR-PH}}(t + 10) - \delta_{\mathrm{PR-PH}}(t)|$.

The IOT long- and short-term code–carrier results from Santa Paula and Clarksburg are shown in Table 2. The results indicate that the control loop algorithm performance meets the long- and short-term code–carrier requirements of WAAS with the I-3 satellites.

TABLE 9.2 Code Carrier Coherence

		Short Term[a] (10 s)	Long Term[b] (100 s)
Requirement		$<5 \times 10^{-11}$	<1 cycle
Santa Paula prototyping	POR, Oct. 10, 1997	1.89×10^{-11}	0.326
	AOR-W, Dec. 1, 1997	1.78×10^{-11}	0.392
Clarksburg prototyping	AOR-W, Mar. 20, 1998	1.92×10^{-11}	0.434

[a]Data averaging 30 s for short term.
[b]Data averaging 60 s for long term.

9.3.5 Carrier Frequency Stability

Carrier frequency stability is a function of both the uplink frequency standard, GUS signal generator, and I-3 transponder. The GEO's short-term carrier frequency stability requirement is specified in the WAAS System Specification and Appendix A of reference [106]. It states:

> The short term stability of the carrier frequency (square root of the Allan variance) at the input of the user's receiver antenna shall be better than 5×10^{-11} over 1 to 10 seconds, excluding the effects of the ionosphere and Doppler.

The Allan variance [2] is calculated on the second difference of L_1 phase data divided by the center frequency over 1–10 s. Effects of smoothed ionosphere and Doppler are compensated for in the data prior to this calculation. Test results in Table 9.3 show that the POR and AOR-W I-3 GEOs, in conjunction with WAAS ground station equipment, meet the short-term carrier frequency stability requirement of WAAS.

9.4 GEO UPLINK SUBSYSTEM (GUS) CLOCK STEERING ALGORITHMS

The local oscillator (cesium frequency standard) at the GUS is not perfectly stable with respect to WAAS network time (WNT). Even though the cesium frequency standard is very stable, it has inherent drift. Over a long period of operation, as in the

TABLE 9.3 Carrier Frequency Stability Requirements Satisfied

Requirement for L_1		1 s, $<5 \times 10^{-11}$	10 s, $<5 \times 10^{-11}$
Santa Paula prototyping	Oct. 10, 1997, POR	4.52×10^{-11}	5.32×10^{-12}
	Dec. 1, 1997, AOR-W	3.93×10^{-11}	4.5×10^{-12}
Clarksburg prototyping	Mar. 20, 1998	4.92×10^{-11}	4.73×10^{-12}

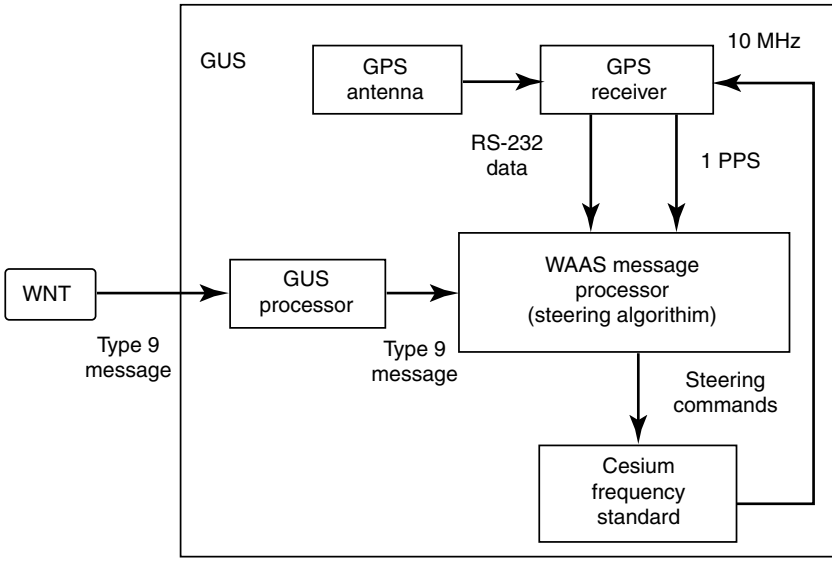


Fig. 9.7 *WMS to GUS clock steering.*

WAAS scenario, this slow drift will accumulate and result in an offset so large that the value will not fit in the associated data fields in the WAAS Type 9 message. This is why a clock steering algorithm is necessary at the GUS. This drifting effect will cause GUS time and WNT to slowly diverge. The GUS can compensate for this drift by periodically re-synchronizing the receiver time with the WNT using the estimated receiver clock offset $[a_0(t_k)]$. This clock offset is provided by the WMS in WAAS Type 9 messages. (See Fig. 9.7.)

GUS steering algorithms for the primary and backup GEO uplink subsystems [103, 46] are discussed in the next section.

The primary GUS clock steering is closed loop via the signal generator, GEO, WRS, WMS, to the GUS processor. The backup GUS clock steering is an open-loop system, because the backup does not uplink to the GEO. The clock offset is calculated using the estimated range and the range calculated from the C&V provided GEO positions.

The GUS also contains the WAAS clock steering algorithm. This algorithm uses the WAAS Type 9 messages from the WMS to align the GEO's epoch with the GPS epoch. The WAAS Type 9 message contains a term referred to as a_0 , or clock offset. This offset represents a correction, or time difference, between the GEOs epoch and WNT. WNT is the internal time reference scale of WAAS and is required to track the GPS time scale, while at the same time providing the users with the translation to UTC. Since GPS master time is not directly obtainable, the WAAS architecture requires that WNT be computed at multiple WMSs using potentially differing sets of measurements from potentially differing sets of receivers and clocks (WAAS

reference stations). WNT is required to agree with GPS to within 50 ns. At the same time, the WNT-to-UTC offset must be provided to the user, with the offset being accurate to 20 ns. The GUS calculates local clock adjustments. Based upon these clock adjustments, the frequency standard can be made to speed up or slow the GUS clock. This will keep the total GEO clock offset within the range allowed by the WAAS Type 9 message so that users can make the proper clock corrections in their algorithms.

9.4.1 Primary GUS Clock Steering Algorithm

The GUS clock steering algorithm calculates the fractional frequency control adjustment required to slowly steer the GUS's cesium frequency standard to align the GEO's epoch. These frequency control signals are very small so normal operation of the code and frequency control loops of any user receiver is not disturbed. Figure 9.8 shows the primary GUS's closed-loop control system block diagram. The primary GUS is the active uplink dedicated to either the AOR-W or POR GEO satellite. If this primary GUS fails, then the hot "backup GUS" is switched to primary.

The clock steering algorithm is designed using a proportional and integral (PI) controller. This algorithm allows one to optimize by adjusting the parameters a, b, and T. Values of a and b are optimized to 0.707 damping ratio.

The value $\bar{a}_0(t_k)$ is the range residual for the primary GUS:

$$\bar{a}_0(t_k) = \frac{1}{N}\sum_{n=1}^{N} a_0(t_{k-n}).$$

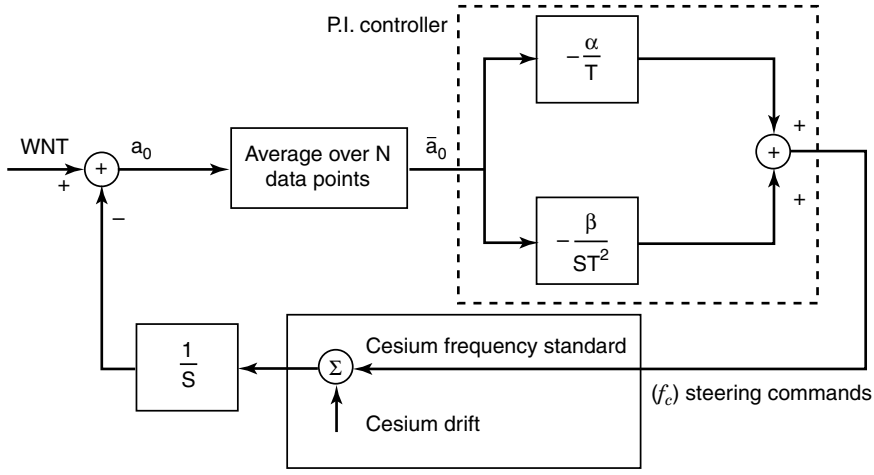


Fig. 9.8 *Clock steering block diagram.*

The value $f_c(t_k)$ is the frequency control signal to be applied at time t_k to the GUS cesium frequency standard:

$$f_c(t_k) = -\left[\frac{\alpha}{T}\,\bar{a}_0(t_k) + \frac{\beta}{T^2}\int_0^{t_k} \bar{a}_0(t)\,dt\right],$$

where T = large time constant
α, β = control parameters
N = number of data points within period t
t = time of averaging period
t_k = time when the frequency control signal is applied to the cesium frequency standard
$a_0(t_k)$ = time offset for GEO at time t_k provided by WMS for primary GUS
S = Laplace transform variable (see Fig. 9.8)

9.4.2 Backup GUS Clock Steering Algorithm

The backup GUS must employ a different algorithm for calculating the range residual. Since the backup GUS is not transmitting to the satellite, the WMS cannot model the clock drift at the backup GUS, and therefore an a_0 term is not present in the WAAS Type 9 message. In lieu of the a_0 term provided by the WMS, the backup GUS calculates an equivalent a_0 parameter.

The range residual $a_0(t_k)$ for the backup GUS is calculated as follows [44]:

$$a_0(t_k) = \frac{B_{\mathrm{RE}} - R_{\mathrm{WMS}}}{c - S(t_k)}$$

where B_{RE} = range estimate in the backup range estimator
R_{WMS} = range estimate calculated from the GEO position supplied by WMs Type 9 message
c = speed of light
$S(t_k)$ = Sagnac effect correction in an inertial frame

The backup GUS uses the same algorithm $f_c(t_k)$ as the primary GUS.

9.4.3 Clock Steering Test Results Description

AOR-W Primary (Clarksburg, MD) Figure 9.9 shows the test results for the first nine days. The first two to three days had cold-start transients and WMS switch overs (LA to DC and DC to LA). From the third to the sixth day, the clock stayed within ±250 ns. At the end of the sixth day, a maneuver took place and caused a small transient and the clock offset went to −750 ns. On the eighth day, the primary GUS was switched to Santa Paula, and another transient was observed. Clock

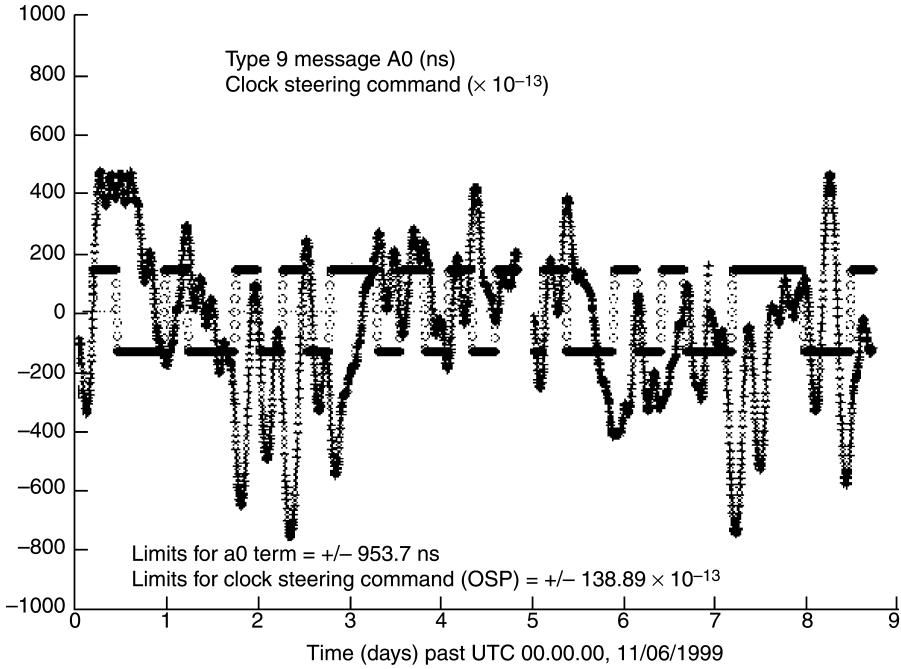


Fig. 9.9 *Primary GP clock steering parameters, AOR-W, Clarksburg.*

steering command limits are $\pm 138.89 \times 10^{-13}$. Limits on the clock offset from the WAAS Type 9 messages are ± 953.7 ns.

AOR-W Backup (Santa Paula, CA) Figure 9.10 shows that the backup GUS stayed within ± 550 ns for the first six days after initial transients. At the end of the sixth day, a GEO maneuver caused a transient.

POR Primary (Brewster, WA) Figure 9.11 shows a cold-start transients and WMS switchovers (LA to DC and DC to LA); the primary GUS stayed within ± 450 ns after initial transients. There was a GUS switchover after the seventh day, which caused transients.

POR Backup (Santa Paula, CA) Figure 9.12 shows cold-start transients. After initial transients, the backup GUS stayed within ± 550 ns for nine days.

The clock offsets in all four cases are less than ± 953.7 ns (limit on WAAS Type 9 Message) for nine days.

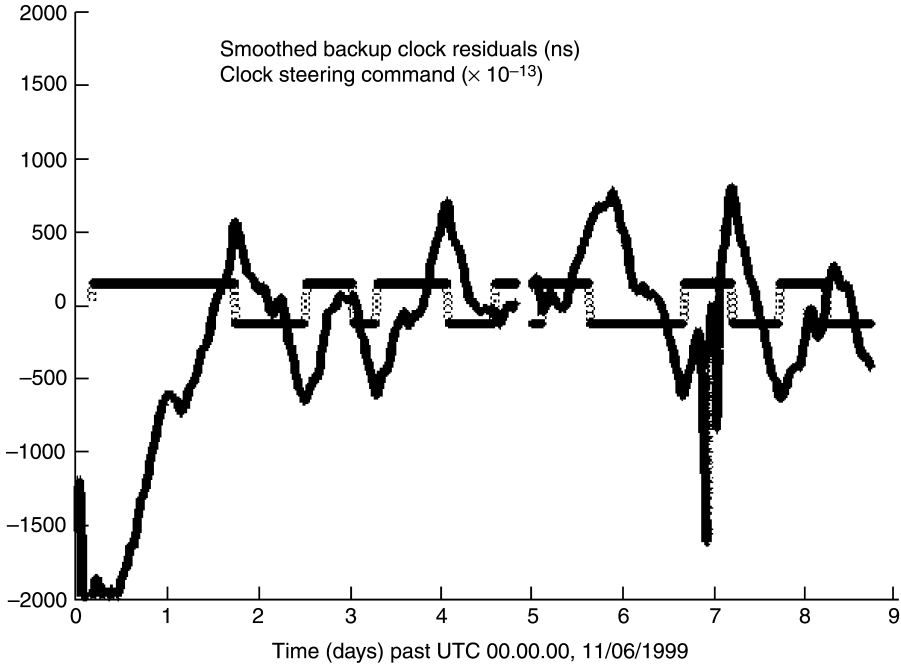


Fig. 9.10 *Backup GP clock steering parameters, POR, Santa Paula.*

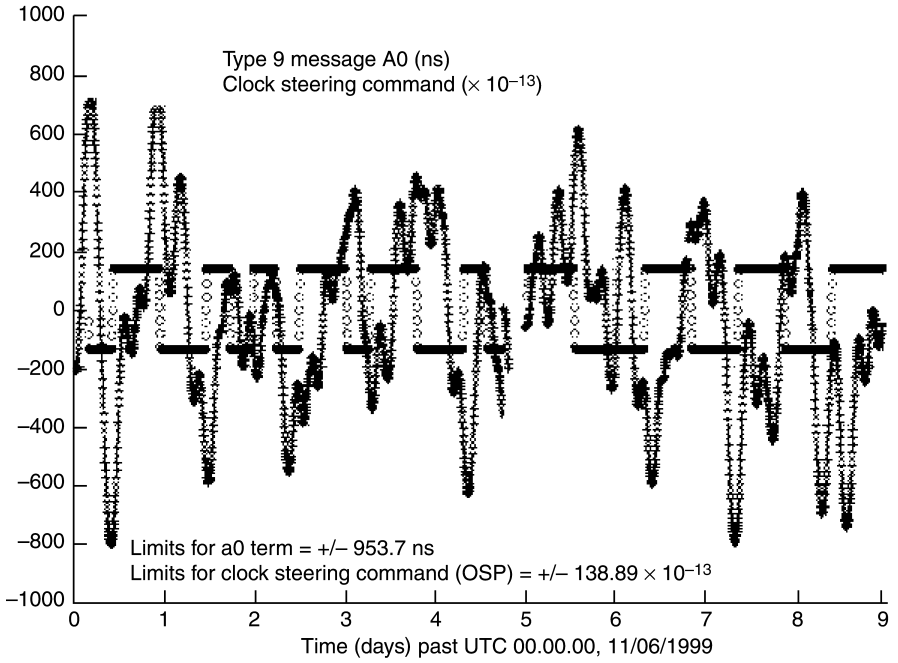


Fig. 9.11 *Primary GUS Processor clock steering parameters for POR (Brewster, WA).*

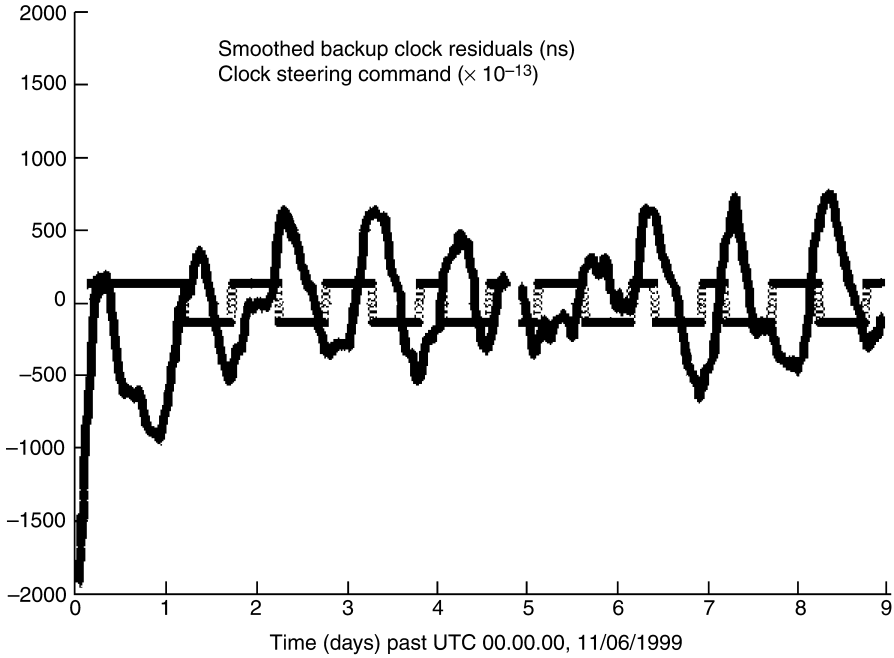


Fig. 9.12 *Backup GPs clock steering parameters for POR (Santa Paula, CA).*

9.5 GEO ORBIT DETERMINATION

The purpose of WAAS is to provide pseudorange and ionospheric corrections for GPS satellites to improve the accuracy for the GPS navigation user and to protect the user with "integrity." Integrity is the ability to provide timely warnings to the user whenever any navigation parameters estimated using the system are outside tolerance limits. WAAS may also augment the GPS constellation by providing additional ranging sources using GEO satellites that are being used to broadcast the WAAS signal.

The two parameters having the most influence on the integrity bounds for the broadcast data are user differential ranging error (UDRE) for the pseudorange corrections and grid ionospheric vertical error (GIVE) for the ionospheric corrections. With these, the on-board navigation system estimates the horizontal protection limit (HPL) and the vertical protection limit (VPL), which are then compared to the Horizontal Alert Limit (HAL) and the Vertical Alert Limit (VAL) requirements for the particular phase of flight involved, that is, oceanic/remote, en-route, terminal, nonprecision approach, and precision approach. If the estimated protection limits are greater than the alert limits, the navigation system is declared unavailable. Therefore, the UDRE and GIVE values obtained by the WAAS (in concert with the GPS and

GEO constellation geometry and reliability) essentially determine the degree of availability of the WADGPS navigation service to the user.

The WAAS algorithms calculate the broadcast corrections and the corresponding UDREs and GIVEs by processing the satellite signals received by the network of ground stations. Therefore, the expected values for UDREs and GIVEs are dependent on satellite and station geometries, satellite signal and clock performance, receiver performance, environmental conditions (such as multipath, ionospheric storms, etc.) and algorithm design [50, 95].

9.5.1 Geometric Analysis

In this section, GDOP will be defined in a reverse direction, as compared to Chapter 2, Section 2.5.2, and Chapter 7, Section 7.8, respectively. There is only one satellite and multiple ground stations. GDOP will be calculated to determine the optimal locations of the ground stations with respect to fixed satellites, such as GEOs [AOR-W, POR, Atlantic Ocean Region East (AOR-E), Indian Ocean East (IOP), Multi-functional Transport Satellite (MTSAT)].

The static geometry of the relation between the GEO and the ground stations is characterized by treating the GEO as a ranging signal source with respect to a network of synchronized ground stations. The states of the GEO are position x, y, z and clock offset ct, where c is the speed of light. A least-squares estimate from the linearized pseudorange differential is developed to obtain the GDOP (geometric dilution of precision). Also introduced are the PDOP (position dilution of precision), TDOP (time dilution of precision), VDOP (vertical dilution of precision), and HDOP (horizontal dilution of precision). Some results relating the behavior of these DOPs with respect to various ground geometries are then presented.

The equation for the pseudorange for each station is

$$\rho_i = \sqrt{(x_i - X)^2 + (y_i - Y)^2 + (z_i - Z)^2} + C_b.$$

Let the state of the GEO be identified by the vector $\mathbf{x} = [x \;\; y \;\; z \;\; C_b]^{\mathrm{T}}$ and the measurements defined by the vector $\rho = [\rho_1 \;\; \rho_2 \;\; \rho_3 \;\; \cdots \;\; \rho_N]^{\mathrm{T}}$, where N is the number of stations that can measure the pseudorange. One can then obtain the H matrix, defined as

$$H \equiv \frac{\partial \rho}{\partial x}.$$

The ith row of H is given by

$$H_i = \left[\frac{x_i - X}{r_i} \quad \frac{y_i - Y}{r_i} \quad \frac{z_i - Z}{r_i} \quad 1 \right]$$

with

$$r_i = \sqrt{(x_i - X)^2 + (y_i - Y)^2 + (z_i - Z)^2}.$$

The first three columns of H are the direction cosines of the line-of-sight directions from the receiver antenna to GPS satellite antennas. Then, taking the linear approximation, the pseudorange differential is given by

$$d\rho = H\ dx + d_\varepsilon,$$

where d_ε is the residual.

If the residuals are random variables with zero mean and no correlations, the least mean squares estimate for dx, called $d\hat{x}$, is given by

$$d\hat{x} = (H^{\mathrm{T}}H)^{-1}H^{\mathrm{T}}\ d\rho.$$

This is obtained by minimizing the square of the residual with respect to $d\hat{x}$.

Let the mean of the residuals be zero and the covariance be given by the matrix R, that is, $E[d_\varepsilon] = 0$ and $E[d_\varepsilon d_\varepsilon^{\mathrm{T}}] = R$. In this case, the least mean squares estimate is

$$d\hat{x} = (H^{\mathrm{T}}R^{-1}H)^{-1}H^{\mathrm{T}}R^{-1}\ d\rho.$$

Then the error covariance matrix for $d\hat{x}$ is given by

$$\begin{aligned}\Sigma &\equiv E[(d\hat{x} - dx)(d\hat{x} - dx)^{\mathrm{T}}] \\ &= E[(H^{\mathrm{T}}R^{-1}H)^{-1}H^{\mathrm{T}}R^{-1}d_\varepsilon d_\varepsilon^{\mathrm{T}}R^{-1}H(H^{T}R^{-1}H)^{-1} \\ &= (H^{\mathrm{T}}R^{-1}H)^{-1}.\end{aligned}$$

If the covariance matrix for the residuals is diagonal and given by $R = \sigma^2 I$, then this reduces to

$$\Sigma = \sigma^2(H^{\mathrm{T}}H)^{-1}.$$

If we define the GEO frame such that x points in the cross-track direction (i.e., horizontal and orthogonal to the orbital track), y points along track, and z points toward the earth, the DOPs can then be defined as follows:

$$\text{GDOP} \equiv \frac{\sqrt{\text{tr}(\Sigma)}}{\sigma^2} = \frac{\sqrt{\Sigma_{xx} + \Sigma_{yy} + \Sigma_{zz} + \Sigma_{tt}}}{\sigma^2},$$

$$\text{PDOP} \equiv \frac{\sqrt{\Sigma_{xx} + \Sigma_{yy} + \Sigma_{zz}}}{\sigma^2},$$

$$\text{VDOP} \equiv \frac{\sqrt{\Sigma_{zz}}}{\sigma^2},$$

$$\text{TDOP} \equiv \frac{\sqrt{\Sigma_{tt}}}{\sigma^2},$$

$$\text{HDOP} \equiv \frac{\sqrt{\Sigma_{xx} + \Sigma_{yy}}}{\sigma^2}.$$

Due to the fact that the DOPs are inversely related to the square root of the number of stations, one can introduce a normalized DOP index, $\text{nDOP} \equiv \text{GDOP}\sqrt{N_{\text{stat}}}/100$, that roughly characterizes the "geometry per station" obtained from a particular station architecture. (N_{stat} is the number of reference stations.)

9.5.2 GEO Synchronous Satellite Orbit Determination via Covariance Analysis

A full WAAS algorithm contains three Kalman filters—an orbit determination filter, an ionospheric corrections filter, and a fast corrections filter. The fast corrections filter is a Kalman filter that estimates the GEO, GPS, and ground station clock states every second. In this section, we derive an estimated lower bound of the GEO UDRE for a WAAS algorithm that contains only the orbit determination Kalman filter, called the UDRE(OD), where OD refers to orbit determination.

A method is proposed to approximate the UDRE obtained for a WAAS including both the orbit determination filter and the fast corrections filter from UDRE(OD). From case studies of the geometries studied in the previous section, we obtain the essential dependence of UDRE on ground station geometry.

A covariance analysis on the orbit determination is performed using a simplified version of the orbit determination algorithms. The performance of the ionospheric corrections filter is treated as perfect, and therefore, the ionospheric filter model is ignored. The station clocks are treated as if perfectly synchronized using the GPS satellite measurements. Therefore, the station clock states are ignored. This allows the decoupling of the orbit determinations for all the statellites from each other, simplifying the orbit determination problem to that for one satellite with its corresponding ground station geometry and synchronized station clocks. Both of these assumptions are liberal, and therefore, the UDRE(OD) obtained here is a lower bound for the actual UDRE(OD). Finally, we consider only users within the service

volume covered by the stations and, therefore, ignore any degradation factors depending on user location.

To simulate the Kalman filter for the covariance matrix P, the following four matrices are necessary:

Φ = state transition matrix;
H = measurement sensitivity matrix;
Q = process noise covariance matrix; and
R = measurement noise covariance matrix.

The methods used to determine these matrices are described below.

The state vector for the satellite is

$$\mathbf{x} = \begin{bmatrix} r \\ \dot{r} \\ C_b \end{bmatrix},$$

where

$$r \equiv [x \;\; y \;\; z]^{\mathrm{T}}$$

is the satellite position in the ECI frame,

$$\dot{r} \equiv [\dot{x} \;\; \dot{y} \;\; \dot{z}]^{\mathrm{T}}$$

is the satellite velocity in the ECI frame, and C_b is the satellite clock offset relative to the synchronized station clocks. Newton's second and third (gravitational) laws provide the equations of motion for the satellite:

$$\ddot{r} \equiv \frac{d^2 r}{dt^2} = -\frac{\mu_E r}{|r|^3} + M,$$

where $\ddot{r}$ is the acceleration in the ECI frame, μ_E is the gravitational constant for the earth, and M is the total perturbation vector in the ECI frame containing all the perturbing accelerations. For this analysis, only the perturbation due to the oblateness of the earth is included. The effect of this perturbation on the behavior of the covariance is negligible, and therefore higher order perturbations are ignored. (Note that although the theoretical model is simplified, the process noise covariance matrix Q is chosen to be consistent with a far more sophisticated orbital model.)

Therefore,

$$M = -\frac{3}{2} J_2 \frac{\mu_E}{|r|^3} \frac{a_E^2}{|r|^2} [I_{3\times 3} + 2\hat{z}\hat{z}^{\mathrm{T}}] r,$$

where a_E is the semimajor axis of the earth shape model, J_2 is the second zonal harmonic coefficient of the earth shape model, and $\hat{z} \equiv [0 \;\; 0 \;\; 1]^{\mathrm{T}}$ [7].

The second-order differential equation of motion can be rewritten as a pair of first-order differential equations

$$\dot{r}_1 = r_2, \qquad \dot{r}_2 = \frac{\mu_E r_1}{|r|^3} + M, \tag{9.1}$$

where r_1 and r_2 are vectors ($r_1 \equiv r$), which therefore gives a system of six first order equations.

The variational equations are differential equations describing the rates of change of the satellite position and velocity vectors as functions of variations in the components of the estimation state vector. These lead to the state transition matrix Φ used in the Kalman filter. The variational equations are

$$\ddot{Y}(t) = A(t)Y(t) + B(t)\dot{Y}(t), \tag{9.2}$$

where

$$Y(t_k)_{3\times 6} \equiv \left[\left(\frac{\partial r(t_k)}{\partial r(t_{k-1})}\right)_{3\times 3} \left(\frac{\partial r(t_k)}{\partial \dot{r}(t_{k-1})}\right)_{3\times 3}\right],$$

$$\dot{Y}(t_k)_{3\times 6} \equiv \left[\left(\frac{\partial \dot{r}(t_k)}{\partial r(t_{k-1})}\right)_{3\times 3} \left(\frac{\partial r(t_k)}{\partial \dot{r}(t_{K-1})}\right)_{3\times 3}\right], \tag{9.3}$$

$$A(t)_{3\times 3} \equiv \frac{\partial \ddot{r}}{\partial r} \tag{9.4}$$

$$= \frac{-\mu_E}{|r|^3}[I_{3\times 3} - 3\hat{r}\hat{r}^{\mathrm{T}}] - \frac{3}{2}J_2\frac{\mu E}{|r|^3}\frac{a_E^2}{|r|^2}$$

$$\times \left[I_{3\times 3} + 2\hat{z}\hat{z}^{\mathrm{T}} - 10(\hat{r}^{\mathrm{T}}\hat{z}^{\mathrm{T}})(\hat{z}\hat{r}^{\mathrm{T}} + \hat{r}\hat{z}^{\mathrm{T}}) + (10(\hat{r}_{\mathrm{T}}\hat{z})^2 - 5)(\hat{\mathrm{r}}\hat{r}^{\mathrm{T}})\right], \tag{9.4}$$

$$B(t)_{3\times 3} \equiv \frac{\partial \ddot{r}}{\partial \dot{r}} = 0_{3\times 3}, \tag{9.5}$$

where $\hat{r} = r/|r|$.

Equations 9.3–9.5 are substituted into Eq. 9.2 with Eq. 9.1, and the differential equations are solved using the fourth-order Runge–Kutta method. The time step used is a 5-min interval. The initial conditions for the GEO are specified for the particular case given and propagated forward for each time step, whereas the initial conditions for the Y's are

$$Y(t_{k-1})_{3\times 6} = [I_{3\times 3} \quad 0_{3\times 3}], \qquad \dot{Y}(t_k)_{3\times 6} = [0_{3\times 3} \quad I_{3\times 3}]$$

and reset for each time step. This is due to the divergence of the solution of the differential equation used in this method to calculate the state transition matrix for the Kepler problem.

This gives the state $\mathbf{x}^{\mathrm{T}} = \begin{bmatrix}\mathbf{r}_1^{\mathrm{T}} & \mathbf{r}_2^{\mathrm{T}}\end{bmatrix}$ and the state transition matrix

$$\Phi_{k,k-1_{7\times7}} = \begin{bmatrix} Y(t_k)_{3\times6} & 0_{3\times1} \\ \dot{Y}(t_k)_{3\times6} & 0_{3\times1} \\ 0_{1\times6} & 1_{1\times1} \end{bmatrix} \tag{9.7}$$

for the Kalman filter.

The measurement sensitivity matrix is given by

$$H_{N\times7} \equiv \frac{\partial\rho}{\partial x} = \left[\left(\frac{\partial\rho}{\partial r}\right)_{N\times3}\left(\frac{\partial\rho}{\partial\dot{r}}\right)_{N\times3}\left(\frac{\partial\rho}{\partial(ct)}\right)_{N\times1}\right],$$

where ρ is the pseudorange for a station and N is the number of stations in view of the satellite. Note that this is essentially the same H as in the previous section. Ignoring relativistic corrections and denoting the station position by the vector $r_S \equiv [x_S \;\; y_S \;\; z_S]^{\mathrm{T}}$, the matrices above are given by

$$\frac{\partial\rho}{\partial r} = \frac{[r - r_S]^{\mathrm{T}}}{|r - r_S|}\frac{\partial r(t_k)}{\partial r(t_{k-1})},$$

$$\frac{\partial\rho}{\partial\dot{r}} = \frac{[r - r_S]^{\mathrm{T}}}{|r - r_S|}\frac{\partial r(t_k)}{\partial\dot{r}(t_{k-1})},$$

and

$$\frac{\partial\rho}{\partial(ct)} = 1.$$

The station position is calculated with the WGS-84 model for the earth and converted to the ECI frame using the J2000 epoch. (See Appendix C.)

These are then combined with the measurement noise covariance matrix, R and the process noise covariance matrix Q to obtain the Kalman filter equations for the covariance matrix P as follows:

$$\begin{aligned} P_k(-) &= \Phi_{k,k-1}P_{k-1}(+)\Phi_{k,k-1}^{\mathrm{T}} + Q, \\ \overline{K}_k &= P_k(-)H_k^{\mathrm{T}}[H_kP_k(-)H_k^{\mathrm{T}} + R]^{-1}, \\ P_k(+) &= [I - \overline{K}_kH_k]P_k(-)[I - \overline{K}_kH_k]^{\mathrm{T}} + \overline{K}_kR\overline{K}_k^{\mathrm{T}}. \end{aligned}$$

The initial condition, $P_0(+)$, and Q are chosen to be consistent with the WAAS algorithms. The value of R is chosen by matching the output of the GEO covariance

for AOR-W with $R = \sigma^2 I$ and is used as the input R for all other satellites and station geometries (note that this therefore gives approximate results). This corresponds to carrier phase ranging for the stations. The results corresponding to the value of R for code ranging are also presented.

From this covariance, the lower bound on the UDRE is obtained by

$$\text{UDRE} \geq \text{EMRBE} + K_{SS}\sqrt{\text{tr}(P)},$$

where EMRBE is the estimated maximum range and bias error. To obtain the .999 level of bounding for the UDRE with EMRBE $= 0$, $K_{SS} = 3.29$. Finally, since the message is broadcast every second, $\Delta t = 1$, so the trace can be used for the velocity components as well.

Figure 9.13 shows the relationship between UDRE and GDOP for various GEO satellites and WRS locations. Table 9.4 describes the various cases considered in this analysis.

The numerical values used for the filter are as follows [all units are Système International (SI)]:

- Earth parameters:

$$\mu_E = 3.98600441 \times 10^{14}, \quad J_2 = 1082.63 \times 10^{-6},$$
$$a_E = 6{,}378{,}137.0, \quad b_E = 6{,}356{,}752.3142.$$

- Filter parameters:

$$Q_{\text{pos}} = 0, \quad Q_{\text{vel}} = 0.75 \times 10^6, \quad Q_{ct} = 60,$$
$$\sigma_R = 0.013, \quad P_{0,\text{pos}} = 144.9, \quad P_{0,\text{vel}} = 1 \times 10^{-4} \quad P_{0,\text{ct}} = 100.9.$$

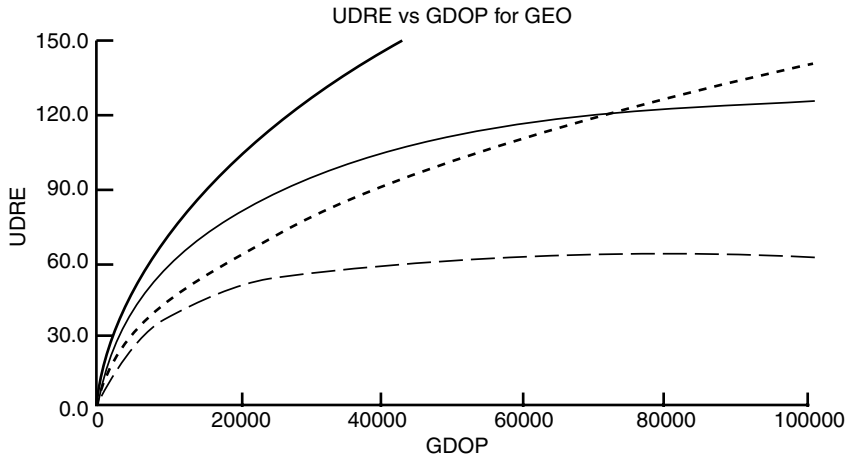


Fig. 9.13 *Relationship between UDRE and GDOP.*

TABLE 9.4 Cases Used in Geometry-per-Station Analysis

Case	UDRE	GDOP	Satellite	Geometry
1	17.9	905	AOR-W	WAAS stations (25), 21 in view
2	45.8	2516	AOR-W	4 WAAS stations (CONUS)
3	135	56536	AOR-W	4 WAAS stations (NE)
4	4.5	254	AOR-W	WAAS stations + Santiago
5	5.8	212	AOR-W	WAAS stations + London
6	4.0	154	AOR-W	WAAS stations+ Santiago + London
7	7.5	439	AOR-W	4 WAAS stations (CONUS) + Santiago
8	8.6	337	AOR-W	4 WAAS stations (CONUS) + London
9	6.6	271	AOR-W	4 WAAS stations (CONUS) + Santiago + London
10	47.7	2799	AOR-W	4 WAAS stations (NE) + Santiago
11	21.5	1405	AOR-W	4 WAAS stations (NE) + London
12	16.4	1334	AOR-W	4 WAAS stations (NE) + Santiago + London
13	28.5	1686	POR	WAAS stations (25), 8 in view
14	45.4	3196	POR	WAAS stations, Hawaii
15	31.1	1898	POR	WAAS stations, Cold Bay
16	55.0	4204	POR	WAAS stations, Hawaii, Cold Bay
17	6.7	257	POR	WAAS stations + Sydney
18	8.3	338	POR	WAAS stations + Tokyo
19	6.7	257	POR	WAAS stations + Sydney + Tokyo
20	21.0	1124	MTSAT	MSAS stations, 8 in view
21	22.0	1191	MTSAT	MSAS stations − Hawaii
22	24.9	1407	MTSAT	MSAS stations − Australia
23	54.6	4149	MTSAT	MSAS stations − Hawaii, Australia
24	22.0	1198	MTSAT	MSAS stations − Ibaraki
25	29.0	1731	MTSAT	MSAS stations − Ibaraki, Australia
26	54.8	4164	MTSAT	MSAS stations − Ibaraki, Australia, Hawaii
27	13.2	609	MTSAT	MSAS stations + Cold Bay
A		139	TEST	Theta = 75°
B		422	TEST	Theta = 30°
C		3343	TEST	Theta = 10°
D		13211	TEST	Theta = 5°
E		67	TEST	41 stations
F		64	TEST	41 + 4 stations

4 WAAS stations (CONUS) are Boston, Miami, Seattle, and Los Angeles.
4 WAAS stations (NE) are Boston, New York, Washington D.C., and Cleveland.

- Curve fit parameters:

$$\sigma_{Q,\text{fit}} = 6.12, \qquad \varepsilon_{\phi,\text{fit}} = .0107.$$

Problems

9.1 Determine the code–carrier coherency at the GUS location using L_1 code and carrier.

9.2 Determine the frequency stability of the AOR and POR transponder using Allan variance for the L_1 using 1–10-s intervals.

Global Positioning Systems, Inertial Navigation, and Integration,
Mohinder S. Grewal, Lawrence R. Weill, Angus P. Andrews

Print ISBN 0-471-35032-X Electronic ISBN 0-471-20071-9

Appendix A

Software

MATLAB m-files on the accompanying diskette are divided into folders for each chapter of the book. The following sections describe what the m-files demonstrate.

A.1 CHAPTER 3 SOFTWARE

A.1.1 Satellite Position Determination

The MATLAB script `ephemeris.m` calculates a GPS satellite position in ECEF coordinates from its ephemeris parameters. The ephemeris parameters comprise a set of Keplerian orbital parameters and describe the satellite orbit during a particular time interval. From these parameters, ECEF coordinates are calculated using the equations from the text. Note that time t is the GPS time at transmission and t_k (`tk` in the script) is the total time difference between time t and the epoch time t_{oe} (`toe`). Kepler's equation for eccentric anomaly is nonlinear in E_k (`Ek`) and is solved numerically using the Newton–Raphson method.

A.2 CHAPTER 5 SOFTWARE

The MATLAB script `Klobuchar.m` calculates the ionospheric delay by using the Klobuchar model.

A.3 CHAPTER 6 SOFTWARE

The m-file `latitude.m` computes and plots the differences between geodetic and geocentric latitude and parametric latitude for the WGS 84 ellipsoid model.

A.3.1 Quaternion Utilities

The subdirectory "quaternions" contains the following m-files:

`vrot2qrot.m`	Transforms a rotation vector to the equivalent quaternion matrix and applies it to an initial quaternion matrix
`rotvec2qvec.m`	Converts a rotation vector to the equivalent quaternion vector
`quattrim.m`	Re-scales and adjusts the symmetric and antisymmetric parts of a 4×4 matrix to make it a legitimate quaternion matrix
`qvec2mat.m`	Converts the quaternion 4-vector to the equivalent quaternion 4×4 matrix
`rotdemo.m`	Computes and plots direction cosines of a coordinate transformation matrix during a rotation (implemented using quaternions for attitude rate integration)
`rotdemo1.m`	Plots the locations of body coordinates on the unit sphere during a rotation about an axis with azimuth 45° and elevation about 35.26°
`rotdemo2.m`	Plots the locations of body coordinates on the unit sphere during a rotation about the east axis
`rotdemo3.m`	Plots the locations of body coordinates on the unit sphere during two successive rotations about different axes
`quatmats.m`	Creates the four quaternion basis matrices $\mathbf{Q}_1$, $\mathbf{Q}_2$, $\mathbf{Q}_3$, $\mathbf{Q}_4$
`trimdemo.m`	Demonstrates "quaternion trimming" (using `quattrim.m`) by comparing trimmed and untrimmed quaternions computed along a random-walk attitude trajectory
`qmat2vec.m`	Converts a 4×4 quaternion matrix to the equivalent quaternion 4-vector.
`viewfrom.m`	A coordinate transformation utility used in the quaternion demonstrations to compute the view of trajectories on the unit sphere from any perspective

In addition, the MATLAB m-file `quatdemo.m`, which is bundled with some versions of MATLAB, is an excellent demonstration of three-dimensional rotation, written by Loren Dean of The MathWorks.

A.4 CHAPTER 7 SOFTWARE

A.4.1 Pointwise Products of Likelihood Functions

The MATLAB m-file `pwprod.m` demonstrates the "Gaussian-ness" of pointwise products of Gaussian likelihood functions and that the maximum-likelihood estimation formulas do pick the peaks of the pointwise products. Two examples are programmed: one using Gaussian likelihood functions that could be derived from

probability distributions with covariance matrices and one using a degenerate likelihood function that could not be derived from a Gaussian probability distribution with a covariance matrix but could be encountered in maximum-likelihood estimation.

The examples chosen are two dimensional, because we cannot plot and view likelihood functions defined over a higher number of dimensions. In practice, the matrix $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$ represents the information about the system state $\mathbf{x}$ gained from a measurement with measurement sensitivity matrix $\mathbf{H}$ and measurement noise covariance $\mathbf{R}$. This information matrix is often rank deficient and has no corresponding covariance matrix.

A.4.2 GPS Navigation Performance Using Kalman Filtering

A.3.2.1 (`GPS_perf.m`) The m-file `GPS_perf.m` performs covariance analysis of GPS navigation performance by solving the associated Riccati equation. The program allows the user to choose four of five satellites, performs both DOP analysis and covariance analysis, and plots both the GDOPs and the RMS navigation uncertainties from Kalman filtering. (Press the RETURN key to continue from one plot to the next.)

The default selection (satellite numbers 1, 2, 3, 4) gives good GDOPs (<4) and good performance. The RMS uncertainties in position and clock errors (calculated from the Riccati equation solution) settle down to steady-state values within a minute or two.

The alternate selection of satellite numbers 1, 2, 3, and 5, however, provides an example in which observability is lost momentarily when these four satellites get into a bad configuration for GPS navigation. In this situation, although the GDOP goes to infinity, the RMS navigation uncertainties with Kalman filtering suffer only slight degradation.

A.4.2.2 (`init_var.m`) The m-file `init_var.m` initializes parameters and variables used in the GPS navigation Kalman filter analysis.

A.4.2.3 (`choose_sat.m`) The m-file `choose_sat.m` allows the user to choose the satellite set (or use a default set). The default set (1, 2, 3, 4) gives good performance.

A.4.2.4 (`gps_init.m`) The m-file `gps_init.m` initializes the selected constellation of GPS satellites.

A.4.2.5 (`calcH.m`) The m-file `calcH.m` solves for satellite motion and calculates the resulting measurement sensitivity matrix for the Kalman filter. This example uses a simplified model for the GPS satellite dynamics.

A.4.2.6 (`gdop.m`) The m-file `gdop.m` calculates the GDOP as a function of time for the chosen satellite constellation.

A.4.2.7 (`covar.m`) The m-file `covar.m` solves the Riccati equation and calculates the Kalman gain matrix. The square roots of the diagonal elements of the covariance matrix **P** are saved for plotting.

A.4.2.8 (`plot_covar.m`) The m-file `plot_covar.m` plots the results of the GPS navigation performance analysis for a time period of 1 h.

A.4.2.9 (`osc_ekf.m`) Demonstrates extended Kalman filter on harmonic oscillator model.

A.5 CHAPTER 8 SOFTWARE

A.5.1 Square-Root Filter Comparison

A.5.1.1 (`shootout.m`) The Matlab m-file `shootout.m` provides a demonstration of the relative fidelity of nine different ways to perform the covariance correction on Example 8.1.

To test how different solution methods perform as conditioning worsens, the observational update is performed for $10^{-9}\varepsilon^{2/3} \leq \delta \leq 10^{9}\varepsilon^{2/3}$ using nine different implementation methods:

1. the conventional Kalman filter, as published by R. E. Kalman;
2. Swerling inverse implementation, published by P. Swerling before the Kalman filter;
3. Joseph "stabilized" implementation as given by P. D. Joseph;
4. Joseph "stabilized" implementation as modified by G. J. Bierman;
5. Joseph "stabilized" implementation as modified by T. W. DeVries;
6. the Potter algorithm (due to J. E. Potter);
7. the Carlson "triangular" algorithm (N. A. Carlson);
8. the Bierman "U-D" algorithm (G. J. Bierman); and
9. the closed-form solution for this particular problem.

The first, second, and last of these are implemented in the m-file `shootout.m`. The others are implemented in the m-files listed below.

The results are plotted as the RMS error in the computed value of P relative to the closed-form solution. In order that all results, including failed results, can be plotted, the value "`NaN`" (not a number) is interpreted as an underflow and set to zero, and the value "`Inf`" is interpreted as the result of a divide-by-zero and set to 10^4.

This demonstration should show that, for this particular problem, the accuracies of the Carlson and Bierman implementations degrade more gracefully than the others as $\delta \to \varepsilon$. This might encourage the use of the Carlson and Bierman methods for applications with suspected roundoff problems, although it does not necessarily demonstrate the superiority of these methods for all applications.

A.5.1.2 Bierman* UD *Corrector (`bierman.m`) Performs the Bierman "U-D" implementation of the Kalman filter measurement update.

A.5.1.3 Carlson "Triangular" Corrector (`carlson.m`) Performs the Carlson "fast triangular" implementation of the Kalman filter measurement update.

A.5.1.4 Joseph "Stabilized" Corrector There are several forms of this Riccati equation corrector implementation, which helps to preserve symmetry of **P**, among other things:

`joseph.m` Performs the Joseph "stabilized" implementation of the Kalman filter measurement update, as proposed by Peter Joseph [19].

`josephb.m` Performs the Joseph "stabilized" implementation of the Kalman filter measurement update, as modified by G. J. Bierman.

`josephdv.m` Performs the Joseph "stabilized" implementation of the Kalman filter measurement update, as modified by T. W. DeVries.

A.5.1.5 Potter's Original Square-Root Filter (`potter.m`) Performs the Potter "square root" implementation of the Kalman filter measurement update.

A.5.1.6 Upper Triangular Cholesky Factorization (`utchol.m`) Performs upper triangular Cholesky factorization for initializing the Carlson "fast triangular" implementation of the Kalman filter measurement update.

A.5.2 Rotation Vector Time Derivatives

The Matlab function `rhodtrpy.m` computes the 3×3 Jacobian matrix of partial derivatives $\dfrac{\partial \dot{\rho}+}{\partial \boldsymbol{\omega}_{\text{RPY}}}$ from Eq. C.151 used in the tightly coupled Kalman filter model.

The Matlab function `rhodtenu.m` computes the 3×3 Jacobian matrix of partial derivatives $\dfrac{\partial \dot{\rho}}{\partial \boldsymbol{\omega}_{\text{ENU}}}$ from Eq. C.154 used in the tightly coupled Kalman filter model.

Global Positioning Systems, Inertial Navigation, and Integration,
Mohinder S. Grewal, Lawrence R. Weill, Angus P. Andrews

Print ISBN 0-471-35032-X Electronic ISBN 0-471-20071-9

Appendix B

Vectors and Matrices

The "S" in "GPS" and in "INS" stands for "system," and *"systems science"* for modeling, analysis, design, and integration of such systems is based largely on linear algebra and matrix theory. Matrices model the ways that components of systems interact dynamically and how overall system performance depends on characteristics of components and subsystems and on the ways they are used within the system.

This appendix presents an overview of matrix theory used for GPS/INS integration and the matrix notation used in this book. The level of presentation is intended for readers who are already somewhat familiar with vectors and matrices. A more thorough treatment can be found in most college-level textbooks on linear algebra and matrix theory.

B.1 SCALARS

Vectors and matrices are arrays composed of scalars, which we will assume to be real numbers. Unless constrained by other conventions, we represent scalars by italic lowercase letters.

In computer implementations, these real numbers will be approximated by floating-point numbers, which are but a finite subset of the rational numbers. The default MATLAB representation for real numbers on 32-bit personal computers is in 64-bit ANSI standard floating point, with a 52-bit mantissa.

B.2 VECTORS

B.2.1 Vector Notation

Vectors are arrays of scalars, either column vectors,

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_n \end{bmatrix},$$

or row vectors,

$$\mathbf{y} = [y_1, y_2, y_3, \ldots, y_m].$$

Unless specified otherwise, vectors can be assumed to be column vectors.

The scalars v_k or y_k are called the *components* of $\mathbf{v}$ or $\mathbf{y}$, respectively. The number of components of a vector (rows in a column vector or columns in a row vector) is called its *dimension.* The dimension of $\mathbf{v}$ shown above is the integer n and the dimension of $\mathbf{y}$ is m. An n-dimensional vector is also called an n-vector.

Vectors are represented by boldface lowercase letters, and the corresponding italic lowercase letters with subscripts represent the scalar components of the associated vector.

B.2.2 Unit Vectors

A *unit vector* (i.e., a vector with magnitude equal to 1) is represented by the symbol $\mathbf{1}$.

B.2.3 Subvectors

Vectors can be partitioned and represented in *block form* as a vector of subvectors:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \vdots \\ \mathbf{x}_\ell \end{bmatrix},$$

where each subvector $\mathbf{x}_k$ is also a vector, as indicated by boldfacing.

B.2.4 Transpose of a Vector

Vector *transposition,*, represented by the post-superscript T transforms row vectors to column vectors, and *vice versa*:

$$\mathbf{v}^{\mathrm{T}} = \left[v_1, v_2, v_3, \ldots, v_n\right], \qquad \mathbf{y}^{T} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_m \end{bmatrix}.$$

In MATLAB, the transpose of vector $\mathbf{v}$ is written as $\mathbf{v}'$.

B.2.5 Vector Inner Product

The inner product or dot product of two m-vectors is the sum of the products of their corresponding components:

$$\mathbf{x}^{\mathrm{T}}\mathbf{y} \quad \text{or} \quad \mathbf{x} \cdot \mathbf{y} \stackrel{\text{def}}{=} \sum_{k=1}^{m} x_k y_k.$$

B.2.6 Orthogonal Vectors

Vectors $\mathbf{x}$ and $\mathbf{y}$ are called *orthogonal* or *normal* if their inner product is zero.

B.2.7 Magnitude of a Vector

The *magnitude* of a vector is the root-sum-squared of its components, denoted by $|\cdot|$ and defined as

$$\begin{aligned} |\mathbf{v}| &\stackrel{\text{def}}{=} \sqrt{\mathbf{v}\mathbf{v}^{\mathrm{T}}} \quad \text{(row vector)} \\ &= \sqrt{\sum_{k=1}^{n} v_k^2}, \\ |\mathbf{y}| &\stackrel{\text{def}}{=} \sqrt{\mathbf{y}^{\mathrm{T}}\mathbf{y}} \quad \text{(column vector)} \\ &= \sqrt{\sum_{k=1}^{m} y_k^2}. \end{aligned}$$

B.2.8 Unit Vectors and Orthonormal Vectors

A unit vector has magnitude equal to 1, and a pair or set of mutually orthogonal unit vectors is called *orthonormal*.

B.2.9 Vector Norms

The magnitude of a column n-vector $\mathbf{x}$ is also called its *Euclidean norm*. This is but one of a class of norms called "*Hölder norms*,"[1] "l_p *norms*," or simply "*p-norms*":

$$\|x\|_p \stackrel{\text{def}}{=} \left[\sum_{i=1}^{n} |x_i|^p\right]^{1/p},$$

and in the limit (as $p \to \infty$) as the *sup*[2] norm, or ∞ norm:

$$\|x\|_\infty \stackrel{\text{def}}{=} \max_i |x_i|.$$

These norms satisfy the *Hölder inequality*:

$$|x^{\mathrm{T}} y| \leq \|x\|_p \|y\|_q \quad \text{for } \frac{1}{p} + \frac{1}{q} = 1.$$

They are also related by inequalities such as

$$\|x\|_\infty \leq \|x\|_E \leq \|x\|_1 \leq n\|x\|_\infty.$$

The Euclidean norm (Hölder 2-norm) is the default norm for vectors. When no other norm is specified, the implied norm is the Euclidean norm.

B.2.10 Vector Cross-product

Vector cross-products are only defined for vectors with three components (i.e., 3-vectors). For any two 3-vectors $\mathbf{x}$ and $\mathbf{y}$, their vector cross-products are defined as

$$\mathbf{x} \otimes \mathbf{y} \stackrel{\text{def}}{=} \begin{bmatrix} x_2 y_3 - x_3 y_2 \\ x_3 y_1 - x_1 y_3 \\ x_1 y_2 - x_2 y_1 \end{bmatrix},$$

which has the properties

$$\begin{aligned} \mathbf{x} \otimes \mathbf{y} &= -\mathbf{y} \otimes \mathbf{x}, \\ \mathbf{x} \otimes \mathbf{x} &= 0, \\ |\mathbf{x} \otimes \mathbf{y}| &= \sin(\theta)|\mathbf{x}||\mathbf{y}|, \end{aligned}$$

where θ is the angle between the vectors $\mathbf{x}$ and $\mathbf{y}$.

[1] Named for the German mathematician Otto Ludwig Hölder (1859–1937).

[2] "Sup" (sounds like "soup") stands for *supremum*, a mathematical term for the *least upper bound* of a set of real numbers. The maximum (max) is the supremum over a finite set.

B.2.11 Right-Handed Coordinate Systems

A Cartesian coordinate system in three dimensions is considered "right handed" if its three coordinate axes are numbered consecutively such that the unit vectors $\mathbf{1}_k$ along its respective coordinate axes satisfy the cross-product rules

$$\mathbf{1}_1 \otimes \mathbf{1}_2 = \mathbf{1}_3, \tag{B.2}$$

$$\mathbf{1}_2 \otimes \mathbf{1}_3 = \mathbf{1}_1, \tag{B.3}$$

$$\mathbf{1}_3 \otimes \mathbf{1}_1 = \mathbf{1}_2. \tag{B.4}$$

B.2.12 Vector Outer Product

The vector outer product of two column vectors

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_m \end{bmatrix}$$

is defined as the $n \times m$ array

$$\mathbf{x}\mathbf{y}^{\mathrm{T}} \stackrel{\text{def}}{=} \begin{bmatrix} x_1y_1 & x_1y_2 & x_1y_2 & \dots & x_1y_m \\ x_2y_1 & x_2y_2 & x_2y_2 & \dots & x_2y_m \\ x_3y_1 & x_3y_2 & x_3y_2 & \dots & x_3y_m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_ny_1 & x_ny_2 & x_ny_2 & \dots & x_ny_m \end{bmatrix},$$

a matrix.

B.3 MATRICES

B.3.1 Matrix Notation

For positive integers m and n, an *m-by-n* real *matrix* $\mathbf{A}$ is a two-dimensional rectangular array of scalars, designated by the subscript notation a_{ij}, and usually displayed in the following format:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{bmatrix}.$$

The scalars a_{ij} are called the *elements* of $\mathbf{A}$. Uppercase *bolded* letters are used for matrices, with the corresponding lowercase letter denoting scalar elements of the associated matrices.

Row and Column Subscripts The first subscript (i) on the element a_{ij} refers to the *row* in which the element occurs, and the second subscript (j) refers to the *column* in which a_{ij} occurs in this format. The integers i and j in this notation are also called *indices* of the elements. The first index is called the *row index*, and the second index is called the *column index* of the element. The term "(ij)th *position*" in the matrix **A** refers to the position of a_{ij}, and a_{ij} is called the "(ij)th *element*" of **A**:

←		columns		→		rows
1st	2nd	3rd	...	nth		
↓	↓	↓		↓		
a_{11}	a_{12}	a_{13}	...	a_{1n}	←	1st
a_{21}	a_{22}	a_{23}	...	a_{2n}	←	2nd
a_{31}	a_{32}	a_{33}	...	a_{3n}	←	3rd
⋮	⋮	⋮	⋱	⋮	⋮	⋮
a_{m1}	a_{m2}	a_{m3}	...	a_{mn}	←	mth

If juxtaposition of subscripts leads to confusion, they may be separated by commas. The element in the eleventh row and first column of the matrix **A** would then be denoted by $a_{11,1}$, not a_{111}.

Dimensions The positive integers m and n are called the *dimensions* of a matrix **A**: m is called the *row dimension* of **A** and n is called the *column dimension* of **A**. The dimensions of **A** may also be represented as "$m \times n$," which is to be read as "m by n." The symbol "$\times$" in this notation does not indicate multiplication. (The number of elements in the matrix **A** equals the product mn, however, and this is important for determining memory requirements for data structures to hold **A**.)

B.3.2 Special Matrix Forms

Square Matrices A matrix is called *square* if it has the same row and column dimensions. The *main diagonal* of a square matrix **A** is the set of elements a_{ij} for which $i = j$. The other elements are called *off-diagonal.* If all the off-diagonal elements of a square matrix **A** are zero, **A** is called a *diagonal* matrix. This and other special forms of square matrices are illustrated Fig. B.1.

Sparse and Dense Matrices A matrix with a "significant fraction" (typically, half or more) of zero elements is called *sparse*. Matrices that are decidedly not sparse are called *dense*, although both sparsity and density are matters of degree. All the forms except symmetric shown in Fig. B.1 are sparse, although sparse matrices do not have to be square. Sparsity is an important characteristic for implementation of matrix methods, because it can be exploited to reduce computer memory and computational requirements.

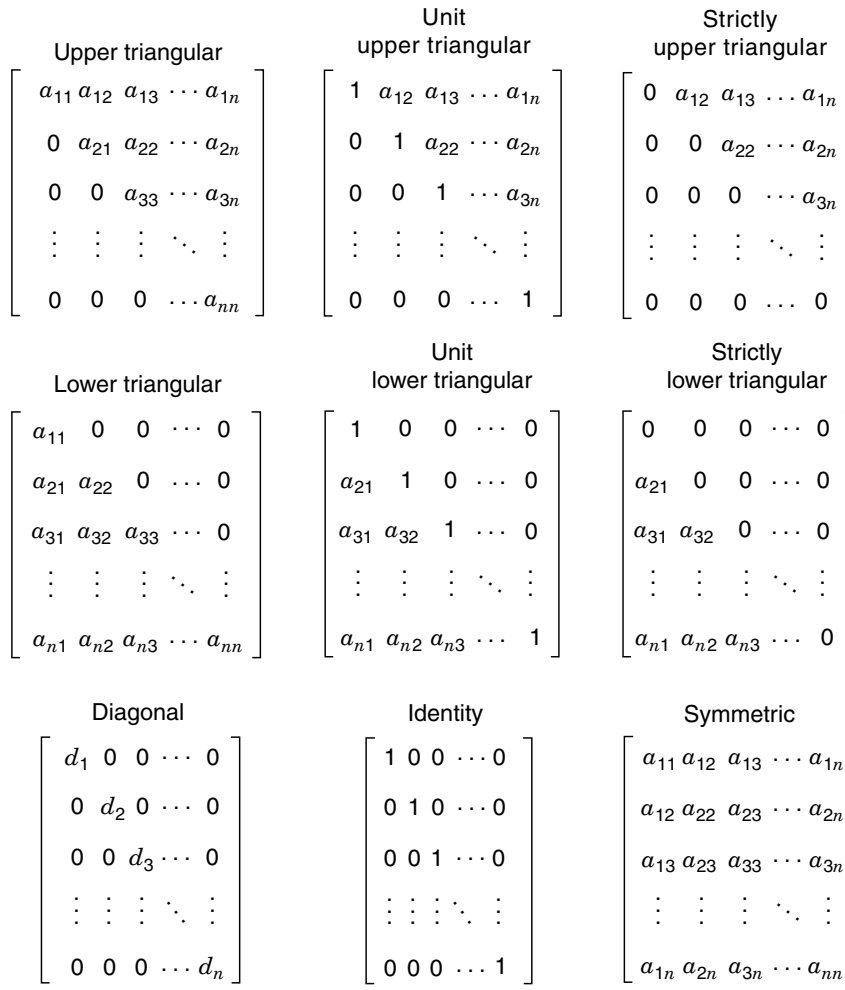


Fig. B.1 *Special forms of square matrices.*

Zero Matrices The ultimate sparse matrix is a matrix in which *all* elements are 0 (zero). It is called a *zero matrix,* and it is represented by the symbol "0" (zero). The equation $A = 0$ indicates that **A** is a zero matrix. Whenever it is necessary to specify the dimensions of a zero matrix, they may be indicated by subscripting: $0_{m\times n}$ will indicate an $m \times n$ zero matrix. If the matrix is square, only one subscript will be used: 0_n will mean an $n \times n$ zero matrix.

Identity Matrices The identity matrix will be represented by the symbol **I**. If it is necessary to denote the dimension of **I** explicitly, it will be indicated by subscripting the symbol: $\mathbf{I}_n$ denotes the $n \times n$ identity matrix.

B.4 MATRIX OPERATIONS

B.4.1 Matrix Transposition

The *transpose* of $\mathbf{A}$ is the matrix $\mathbf{A}^{\mathrm{T}}$ (with the superscript "T" denoting the transpose operation), obtained from $\mathbf{A}$ by interchanging rows and columns:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} a_{11} & a_{21} & a_{31} & \dots & a_{m1} \\ a_{12} & a_{22} & a_{32} & \dots & a_{m2} \\ a_{13} & a_{23} & a_{33} & \dots & a_{m3} \\ \dots & \dots & \dots & \ddots & \vdots \\ a_{1n} & a_{2n} & a_{3n} & \dots & a_{mn} \end{bmatrix}.$$

The transpose of an $m \times n$ matrix is an $n \times m$ matrix.

The transpose of the matrix $\mathbf{M}$ in MATLAB is written as $\mathbf{M}'$.

Symmetric Matrices A matrix $\mathbf{A}$ is called *symmetric* if $\mathbf{A}^{\mathrm{T}} = \mathbf{A}$ and *skew symmetric* (or *anti-symmetric*) if $\mathbf{A}^{T} = -\mathbf{A}$. Only square matrices can be symmetric or skew symmetric. Therefore, whenever a matrix is said to be symmetric or skew-symmetric, it is implied that it is a square matrix. Any square matrix $\mathbf{A}$ can be expressed as a sum of its symmetric and antisymmetric parts:

$$\mathbf{A} = \underbrace{\tfrac{1}{2}\left(\mathbf{A} + \mathbf{A}^{\mathrm{T}}\right)}_{\text{symmetric}} + \underbrace{\tfrac{1}{2}\left(\mathbf{A} - \mathbf{A}^{\mathrm{T}}\right)}_{\text{antisymmetric}}.$$

Cross-Product Matrices The vector cross-product $\boldsymbol{\rho} \otimes \boldsymbol{\alpha}$ can also be expressed in matrix form as

$$\boldsymbol{\rho} \otimes \boldsymbol{\alpha} = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \end{bmatrix} \otimes \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} \tag{B.5}$$

$$= \begin{bmatrix} \rho_2\alpha_3 - \rho_3\alpha_2 \\ \rho_3\alpha_1 - \rho_1\alpha_3 \\ \rho_1\alpha_2 - \rho_2\alpha_1 \end{bmatrix} \tag{B.6}$$

$$= [\boldsymbol{\rho}\otimes]\boldsymbol{\alpha} \tag{B.7}$$

$$= \begin{bmatrix} 0 & -\rho_3 & \rho_2 \\ \rho_3 & 0 & -\rho_1 \\ -\rho_2 & \rho_1 & 0 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}, \tag{B.8}$$

where the "cross-product matrix"

$$[\boldsymbol{\rho}\otimes] \stackrel{\text{def}}{=} \begin{bmatrix} 0 & -\rho_3 & \rho_2 \\ \rho_3 & 0 & -\rho_1 \\ -\rho_2 & \rho_1 & 0 \end{bmatrix} \tag{B.9}$$

is skew-symmetric.

B.4.2. Subscripted Matrix Expressions

Subscripts represent an operation on a matrix that extracts the designated matrix element. Subscripts may also be applied to matrix expressions. The element in the (ij)th position of a matrix expression can be indicated by subscripting the expression, as in

$$\{\mathbf{A}^T\}_{ij} = a_{ji}.$$

Here, we have used braces { } to indicate the scope of the expression to which the subscripting applies. This is a handy device for defining matrix operations.

B.4.3 Multiplication of Matrices by Scalars

Multiplication of a matrix $\mathbf{A}$ by a scalar s is equivalent to multiplying every element of $\mathbf{A}$ by s:

$$\{\mathbf{A}s\}_{ij} = \{s\mathbf{A}\}_{ij} = sa_{ij}.$$

B.4.4 Addition and Multiplication of Matrices

Addition of Matrices Is Associative and Commutative. Matrices can be added together if and only if they share the same dimensions. If $\mathbf{A}$ and $\mathbf{B}$ have the same dimensions, then addition is defined by adding corresponding elements:

$$\{\mathbf{A}+\mathbf{B}\}_{ij} = a_{ij} + b_{ij.}$$

Addition of matrices is *commutative* and *associative.* That is, $\mathbf{A}+\mathbf{B} = \mathbf{B}+\mathbf{A}$ and $\mathbf{A}+(\mathbf{B}+\mathbf{C}) = (\mathbf{A}+\mathbf{B})+\mathbf{C}$.

Additive Inverse of a Matrix The product of a matrix $\mathbf{A}$ by the scalar -1 yields its *additive inverse* $-\mathbf{A}$:

$$(-1)\,\mathbf{A} = -\mathbf{A}, \qquad \mathbf{A}+(-\mathbf{A}) = \mathbf{A}-\mathbf{A} = 0.$$

Here, we have followed the not uncommon practice of using the symbol "−" both as a unary (additive inverse) and binary (subtraction) operator. *Subtraction* of a matrix $\mathbf{A}$ from a matrix $\mathbf{B}$ is equivalent to adding the additive inverse of $\mathbf{A}$ to $\mathbf{B}$:

$$\mathbf{B} - \mathbf{A} = \mathbf{B} + (-\mathbf{A}).$$

Multiplication of Matrices Is Associative but Not Commutative. Multiplication of an $m \times n$ matrix $\mathbf{A}$ by a matrix $\mathbf{B}$ on the right-hand side of $\mathbf{A}$, as in the matrix product $\mathbf{AB}$, is defined only if *the row dimension of* $\mathbf{B}$ *equals the column dimension of* $\mathbf{A}$. That is, we can multiply an $m \times n$ matrix $\mathbf{A}$ by a $p \times q$ matrix $\mathbf{B}$ in this order only if $n = p$. In that case, the matrices $\mathbf{A}$ and $\mathbf{B}$ are said to be *conformable* for multiplication in that order, and the matrix product is defined element by element by

$$\{\mathbf{AB}\}_{ij} \stackrel{\text{def}}{=} \sum_{k=1}^{n} a_{ik} b_{kj},$$

the result of which is an $m \times q$ matrix. Whenever matrices appear as a product in an expression, it is implied that they are conformable for multiplication.

Products with Identity Matrices Multiplication of any $m \times n$ matrix $\mathbf{A}$ by a conformable *identity matrix* yields the original matrix $\mathbf{A}$ as the product:

$$\mathbf{A}\mathbf{I}_n = \mathbf{A}, \qquad \mathbf{I}_m\mathbf{A} = \mathbf{A}.$$

B.4.5 Powers of Square Matrices

Square matrices can always be multiplied by themselves, and the resulting matrix products are again conformable for multiplication. Consequently, one can define the pth power of a square matrix $\mathbf{A}$ as

$$\mathbf{A}^p = \underbrace{\mathbf{A} \times \mathbf{A} \times \mathbf{A} \times \cdots \times \mathbf{A}}_{p \text{ elements}}.$$

B.4.6 Matrix Inversion

If $\mathbf{A}$ and $\mathbf{B}$ are square matrices of the same dimension, and such that their product

$$\mathbf{AB} = \mathbf{I},$$

then $\mathbf{B}$ is the *matrix inverse* of $\mathbf{A}$ and $\mathbf{A}$ is the matrix inverse of $\mathbf{B}$. (It turns out that $\mathbf{BA} = \mathbf{AB} = \mathbf{I}$ in this case.) The inverse of a matrix $\mathbf{A}$ is unique, if it exists, and is denoted by $\mathbf{A}^{-1}$. Not all matrices have inverses. Matrix *inversion* is the process of finding a matrix inverse, if it exists. If the inverse of a matrix $\mathbf{A}$ does not exist, $\mathbf{A}$ is called *singular*. Otherwise, it is called *non-singular*.

B.4.7 Generalized Matrix Inversion

Even nonsquare and/or singular matrices can have *generalized inverses*. The *Moore-Penrose generalized inverse* of an $m \times n$ matrix $\mathbf{A}$ is the $n \times m$ matrix $\mathbf{A}^+$ such that

$$\begin{aligned}\mathbf{A}\mathbf{A}^+\mathbf{A} &= \mathbf{A},\\ \mathbf{A}^+\mathbf{A}\mathbf{A}^+ &= \mathbf{A}^+,\\ (\mathbf{A}\mathbf{A}^+)^{\mathrm{T}} &= \mathbf{A}\mathbf{A}^+,\\ (\mathbf{A}^+\mathbf{A})^{\mathrm{T}} &= \mathbf{A}^+\mathbf{A}.\end{aligned}$$

B.4.8 Orthogonal Matrices

A square matrix $\mathbf{A}$ is called *orthogonal* if $\mathbf{A}^{\mathrm{T}} = \mathbf{A}^{-1}$. Orthogonal matrices have several useful properties:

- Orthogonality of a matrix $\mathbf{A}$ implies that the row vectors of $\mathbf{A}$ are jointly orthonormal vectors, and the column vectors of $\mathbf{A}$ are also jointly orthonormal vectors.
- The dot products of vectors are invariant under multiplication by a conformable orthogonal matrix. That is, if $\mathbf{A}$ is orthogonal, then $\mathbf{x}^T\mathbf{y} = (\mathbf{A}\mathbf{x})^T(\mathbf{A}\mathbf{y})$ for all conformable $\mathbf{x}$ and $\mathbf{y}$.
- Products and inverses of orthogonal matrices are orthogonal.

As a rule, multiplications by orthogonal matrices tend to be numerically well conditioned, compared to general matrix multiplications. (The inversion of orthogonal matrices is obviously extremely well conditioned.)

B.5 BLOCK MATRIX FORMULAS

B.5.1 Submatrices, Partitioned Matrices, and Blocks

For any $m \times n$ matrix $\mathbf{A}$ and any subset $S_{\text{rows}} \subseteq \{1, 2, 3, \ldots, m\}$ of the row indices and subset $S_{\text{cols}} \subseteq \{1, 2, 3, \ldots, n\}$ of the column indices, the subset of elements

$$\mathbf{A}' = \{a_{ij} | i \in S_{\text{rows}}, j \in S_{\text{cols}}\}$$

is called a *submatrix* of $\mathbf{A}$.

A *partitioning* of an integer n is an exhaustive collection of contiguous subsets S_k of the form

$$\overbrace{1, 2, 3, \ldots, \ell_1}^{S_1}, \overbrace{(\ell_1 + 1), \ldots, \ell_2}^{S_2}, \ldots, \overbrace{(\ell_{p-1} + 1), \ldots, n}^{S_p}.$$

The collection of submatrices formed by partitionings of the row and column dimensions of a matrix is called a *partitioning* of the matrix, and the matrix is said to be *partitioned* by that partitioning. Each submatrix of a partitioned matrix $\mathbf{A}$ is called a *partitioned submatrix*, *partition*, submatrix *block*, *subblock*, or *block* of $\mathbf{A}$. Each block of a partitioned matrix $\mathbf{A}$ can be represented by a conformable matrix expression, and $\mathbf{A}$ can be displayed as a *block matrix*:

$$\mathbf{A} = \begin{bmatrix} \mathbf{B} & \mathbf{C} & \mathbf{D} & \dots & \mathbf{F} \\ \mathbf{G} & \mathbf{H} & \mathbf{J} & \dots & \mathbf{L} \\ \mathbf{M} & \mathbf{N} & \mathbf{P} & \dots & \mathbf{R} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{V} & \mathbf{W} & \mathbf{X} & \dots & \mathbf{Z} \end{bmatrix}$$

where $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$, ... stand for matrix expressions. Whenever a matrix is displayed as a block matrix, it is implied that all block submatrices in the same row have the same row dimension and that all block submatrices in the same column have the same column dimension.

A block matrix of the form

$$\begin{bmatrix} \mathbf{A} & 0 & 0 & \dots & 0 \\ 0 & \mathbf{B} & 0 & \dots & 0 \\ 0 & 0 & \mathbf{C} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \mathbf{M} \end{bmatrix},$$

in which the off-diagonal block submatrices are zero matrices, is called a *block diagonal matrix*, and a block matrix in which the block submatrices on one side of the diagonal are zero matrices is called a *block triangular matrix*.

Columns and Rows as Blocks There are two special partitionings of matrices in which the block submatrices are vectors. The *column vectors* of an $m \times n$ matrix $\mathbf{A}$ are the block submatrices of the partitioning of $\mathbf{A}$ for which all column dimensions are 1 and all row dimensions are m. The *row vectors* of $\mathbf{A}$ are the block submatrices of the partitioning for which all row dimensions are 1 and all column dimensions are n. All column vectors of an $m \times n$ matrix are m-vectors, and all row vectors are n-vectors.

B.5.2 Rank and Linear Dependence

A *linear combination* of a finite set of n-vectors $\{\mathbf{v}_i\}$ is a summation of the sort $\sum_i a_i \mathbf{v}_i$ for some set of scalars $\{a_i\}$. If some linear combination $\sum a_i \mathbf{v}_i = 0$ and at least one coefficient $a_i \neq 0$, the set of vectors $\{\mathbf{v}_i\}$ is called *linearly dependent*. Conversely, if the only linear combination for which $\sum a_i \mathbf{v}_i = 0$ is the one for which all the $a_i = 0$, then the set of vectors $\{\mathbf{v}_i\}$ is called *linearly independent*.

The *rank* of a $n \times m$ matrix $\mathbf{A}$ equals the size of the *largest* collection of its column vectors that is linearly independent. Note that any such linear combination can be expressed in the form $\mathbf{Aa}$, where the nonzero elements of the column m-vector $\mathbf{A}$ are the associated scalars of the linear combination, and the number of nonzero components of $\mathbf{A}$ is the size of the collection of column vectors in the linear combination. The same value for the rank of a matrix is obtained if the test is applied to its row vectors, where any linear combination of row vectors can be expressed in the form $\mathbf{a}^{\mathrm{T}}\mathbf{A}$ for some column n-vector $\mathbf{A}$.

An $n \times n$ matrix is nonsingular if and only if its rank equals its dimension n.

B.5.3 Conformable Block Operations

Block matrices with conformable partitionings may be transposed, added, subtracted, and multiplied in block format. For example,

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} \mathbf{A}^{\mathrm{T}} & \mathbf{C}^{\mathrm{T}} \\ \mathbf{B}^{\mathrm{T}} & \mathbf{D}^{\mathrm{T}} \end{bmatrix},$$

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} + \begin{bmatrix} \mathbf{E} & \mathbf{F} \\ \mathbf{G} & \mathbf{H} \end{bmatrix} = \begin{bmatrix} \mathbf{A}+\mathbf{E} & \mathbf{B}+\mathbf{F} \\ \mathbf{C}+\mathbf{G} & \mathbf{D}+\mathbf{H} \end{bmatrix},$$

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \times \begin{bmatrix} \mathbf{E} & \mathbf{F} \\ \mathbf{G} & \mathbf{H} \end{bmatrix} = \begin{bmatrix} \mathbf{AE}+\mathbf{BG} & \mathbf{AF}+\mathbf{BH} \\ \mathbf{CE}+\mathbf{DG} & \mathbf{CF}+\mathbf{DH} \end{bmatrix}.$$

B.5.4 Block Matrix Inversion Formula

The inverse of a partitioned matrix with square diagonal blocks may be represented in block form as [53]

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{E} & \mathbf{F} \\ \mathbf{G} & \mathbf{H} \end{bmatrix},$$

where

$$\begin{aligned} \mathbf{E} &= \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{BHCA}^{-1}, \\ \mathbf{F} &= -\mathbf{A}^{-1}\mathbf{BH}, \\ \mathbf{G} &= -\mathbf{HCA}^{-1}, \\ \mathbf{H} &= [\mathbf{D} - \mathbf{CA}^{-1}\mathbf{B}]^{-1}. \end{aligned}$$

This formula can be proved by multiplying the original matrix times its alleged inverse and verifying that the result is the identity matrix.

B.5.5 Inversion Formulas for Matrix Expressions

Sherman–Morrison Formula A "rank 1" modification of a square matrix $\mathbf{A}$ is a sum of the form $\mathbf{A} + \mathbf{bc}^{\mathrm{T}}$, where $\mathbf{b}$ and $\mathbf{c}$ are conformable column vectors. Its inverse is given by the formula

$$\left[\mathbf{A} + \mathbf{bc}^{\mathrm{T}}\right]^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{bc}^{\mathrm{T}}\mathbf{A}^{-1}}{1 + \mathbf{c}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{b}}.$$

Sherman–Morrison–Woodbury Formula This is the generalization of the above formula for conformable matrices in place of vectors:

$$\left[\mathbf{A} + \mathbf{BC}^{\mathrm{T}}\right]^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}\left[\mathbf{I} + \mathbf{C}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{B}\right]^{-1}\mathbf{C}^{\mathrm{T}}\mathbf{A}^{-1}.$$

Hemes Inversion Formula A further generalization of this formula (used in the derivation of the Kalman filter equations) includes an additional conformable square matrix factor in the modification:

$$\left[\mathbf{A} + \mathbf{BC}^{-1}\mathbf{D}^{\mathrm{T}}\right]^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}\left[\mathbf{C} + \mathbf{D}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{B}\right]^{-1}\mathbf{D}^{\mathrm{T}}\mathbf{A}^{-1}. \tag{B.10}$$

B.6 FUNCTIONS OF SQUARE MATRICES

B.6.1 Determinants and Characteristic Values

Elementary Permutation Matrices An *elementary permutation matrix* is formed by interchanging rows or columns of an identity matrix $\mathbf{I}_n$:

$$\mathbf{P}_{[ij]} = \begin{array}{c} \\ i \\ \\ j \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccccc} & & i & & j & & \end{array} \\ \begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & 0 & \cdots & 1 & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & 1 & \cdots & 0 & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{pmatrix} \end{array}.$$

Multiplication of a vector $\mathbf{x}$ by $\mathbf{P}_{[ij]}$ permutes the ith and jth elements of $\mathbf{x}$. Note that $\mathbf{P}_{[ij]}$ is an *orthogonal* matrix and that $\mathbf{P}_{[ii]} = \mathbf{I}_n$, the identity matrix.

Determinants of Elementary Permutation Matrices The *determinant* of an elementary permutation matrix $\mathbf{P}_{[ij]}$ is defined to be -1, unless $i = j$ (i.e., $\mathbf{P}_{[ij]} = \mathbf{I}_n$):

$$\det(\mathbf{P}_{[ij]}) \stackrel{\text{def}}{=} \begin{cases} -1, & i \neq j, \\ +1, & i = j. \end{cases}$$

Permutation Matrices A *permutation matrix* is any product of elementary permutation matrices. These are also orthogonal matrices. Let $\mathcal{P}_n$ denote the set of all distinct $n \times n$ permutation matrices. There are $n! = 1 \times 2 \times 3 \times \cdots \times n$ of them, corresponding to the $n!$ permutations of n indices.

Determinants of Permutation Matrices The determinant of a permutation matrix can be defined by the rule that the determinant of a product of matrices is the product of the determinants:

$$\det(\mathbf{AB}) = \det(\mathbf{A})\ \det(\mathbf{B}).$$

Therefore, the determinant of a permutation matrix will be either $+1$ or -1. A permutation matrix is called "even" if its determinant is $+1$ and "odd" if its determinant equals -1.

Determinants of Square Matrices The determinant of any $n \times n$ matrix $\mathbf{A}$ can be defined as follows:

$$\det(\mathbf{A}) \stackrel{\text{def}}{=} \sum_{\mathbf{P} \in \mathcal{P}_n} \det(\mathbf{P}) \prod_{i=1}^{n} \{\mathbf{AP}\}_{ii}.$$

This formula has $\mathcal{O}(n \times n!)$ computational complexity (for a sum over $n!$ products of n elements each).

Characteristic Values of Square Matrices For a free variable λ, the polynomial

$$p_A(\lambda) \stackrel{\text{def}}{=} \det\ [\mathbf{A} - \lambda \mathbf{I}] = \sum_{i=0}^{n} a_i \lambda^i$$

is called the *characteristic polynomial* of $\mathbf{A}$. The roots of $p_A(\lambda)$ are called the *characteristic values* (or *eigenvalues*) of $\mathbf{A}$. The determinant of $\mathbf{A}$ equals the product of its characteristic values, with each characteristic value occurring as many times in the product as the multiplicity of the associated root of the characteristic polynomial.

Definiteness of Symmetric Matrices If $\mathbf{A}$ is symmetric, all its characteristic values are real numbers, which implies that they can be ordered. They are usually expressed in descending order:

$$\lambda_1(\mathbf{A}) \geq \lambda_2(\mathbf{A}) \geq \lambda_3(\mathbf{A}) \geq \cdots \geq \lambda_n(\mathbf{A}).$$

A real square symmetric matrix $\mathbf{A}$ is called

$$\begin{array}{rll} \textit{positive definite} & \text{if} & \lambda_n(\mathbf{A}) > 0, \\ \textit{non-negative definite} & \text{if} & \lambda_n(\mathbf{A}) \geq 0, \\ \textit{indefinite} & \text{if} & \lambda_1(\mathbf{A}) > 0 \text{ and } \lambda_n(\mathbf{A}) < 0, \\ \textit{non-positive definite} & \text{if} & \lambda_1(\mathbf{A}) \leq 0, \text{ and} \\ \textit{negative definite} & \text{if} & \lambda_1(\mathbf{A}) < 0. \end{array}$$

Non-negative definite matrices are also called *positive semidefinite*, and non-positive definite matrices are also called *negative semidefinite*.

Characteristic Vectors For each real characteristic value $\lambda_i(\mathbf{A})$ of a real symmetric $\mathbf{A}$, there is a corresponding *characteristic vector* (or *eigenvector*) $\mathbf{e}_i(\mathbf{A})$ such that $\mathbf{e}_i(\mathbf{A}) \neq 0$ and $\mathbf{A}\mathbf{e}_i(\mathbf{A}) = \lambda_i(\mathbf{A})\mathbf{e}_i(\mathbf{A})$. The characteristic vectors corresponding to distinct characteristic values are mutually orthogonal.

B.6.2 The Matrix Trace

The *trace* of a square matrix is the sum of its diagonal elements. It also equals the sum of the characteristic values and has the property that the trace of the product of conformable matrices is independent of the order of multiplication—a very useful attribute:

$$\begin{aligned} \text{trace}(\mathbf{AB}) &= \sum_i \{\mathbf{AB}\}_{ii} && \text{(B.11)} \\ &= \sum_i \sum_j \mathbf{A}_{ij}\mathbf{B}_{ji} && \text{(B.12)} \\ &= \sum_j \sum_i \mathbf{B}_{ji}\mathbf{A}_{ij} && \text{(B.13)} \\ &= \text{trace}(\mathbf{BA}). && \text{(B.14)} \end{aligned}$$

Note the product $\mathbf{AB}$ is conformable for the trace function only if it is a square matrix, which requires that $\mathbf{A}$ and $\mathbf{B}^{\mathrm{T}}$ have the same dimensions. If they are $m \times n$ (or $n \times m$), then the computation of the trace of their product requires mn multiplications, whereas the product itself would require m^2n (or mn^2) multiplications.

B.6.3 Algebraic Functions of Matrices

An algebraic function may be defined by an expression in which the independent variable (a matrix) is a free variable, such as the truncated power series

$$f(\mathbf{A}) = \sum_{k=-n}^{n} \mathbf{B}_k \mathbf{A}^k,$$

where the negative power $\mathbf{A}^{-p} = \left\{\mathbf{A}^{-1}\right\}^p = \{\mathbf{A}^p\}^{-1}$. In this representation, the matrix $\mathbf{A}$ is the independent (free) variable and the other matrix parameters ($\mathbf{B}_k$) are assumed to be known and fixed.

B.6.4 Analytic Functions of Matrices

An analytic function is defined in terms of a convergent power series. It is necessary that the power series converge to a limit, and the matrix norms defined in Section B.1.7 must be used to define and prove convergence of a power series. This level of rigor is beyond the scope of this book, but we do need to use one particular analytic function: the exponential function.

Matrix Exponential Function The power series

$$e^{\mathbf{A}} \stackrel{\text{def}}{=} \sum_{k=0}^{\infty} \frac{1}{k!} \mathbf{A}^k, \tag{B.15}$$

$$k! \stackrel{\text{def}}{=} 1 \times 2 \times 3 \cdots \times k, \tag{B.16}$$

does converge[3] for all square matrices $\mathbf{A}$. It defines the exponential function of the matrix $\mathbf{A}$. This definition is sufficient to prove some elementary properties of the exponential function for matrices, such as

- $e^{0_n} = \mathbf{I}_n$ for 0_n, the $n \times n$ zero matrix.
- $e^{\mathbf{I}_n} = e\mathbf{I}_n$ for $\mathbf{I}_n$, the $n \times n$ identity matrix.
- $e^{\mathbf{A}^{\mathrm{T}}} = \left\{e^{\mathbf{A}}\right\}^{\mathrm{T}}$.
- $(d/dt)e^{\mathbf{A}t} = \mathbf{A}e^{\mathbf{A}t} = e^{\mathbf{A}t}\mathbf{A}$.
- The exponential of a skew-symmetric matrix is an orthogonal matrix.
- The characteristic vectors of $\mathbf{A}$ are also the characteristic vectors of $e^{\mathbf{A}}$.
- If λ is a characteristic value of $\mathbf{A}$, then e^{λ} is a characteristic value of $e^{\mathbf{A}}$.

Powers and Exponentials of Cross-product Matrices The fact that exponential functions of skew-symmetric matrices are orthogonal matrices will have important consequences for coordinate transformations (Appendix C), because the matrices transforming vectors from one right-handed coordinate system (defined in Section B.1.2.11) to another can can be represented as the exponentials of cross-

[3]However, convergence is not fast enough to make this a reasonable general-purpose formula for approximating the exponential of $\mathbf{A}$. More reliable and efficient methods can be found, e.g., in [41].

product matrices (defined in Eq. B.9). We show here how to represent the exponential of a cross-product matrix

$$[\boldsymbol{\rho}\otimes] = \begin{bmatrix} 0 & -\rho & \rho_2 \\ \rho_3 & 0 & -\rho_1 \\ -\rho_2 & \rho_1 & 0 \end{bmatrix}$$

in closed form. The first few powers can be calculated by hand, as

$$\begin{aligned}
[\boldsymbol{\rho}\otimes]^0 &= \mathbf{I}_3, \\
[\boldsymbol{\rho}]^1 &= [\boldsymbol{\rho}\otimes], \\
[\boldsymbol{\rho}\otimes]^2 &= \begin{bmatrix} -\rho_3^2-\rho_2^2 & \rho_2\rho_1 & \rho_3\rho_1 \\ \rho_2\rho_1 & -\rho_3^2-\rho_1^2 & \rho_3\rho_2 \\ \rho_3\rho_1 & \rho_3\rho_2 & -\rho_2^2-\rho_1^2 \end{bmatrix} \\
&= \boldsymbol{\rho}\boldsymbol{\rho}^{\mathrm{T}} - |\boldsymbol{\rho}|^2\mathbf{I}_3, \\
[\boldsymbol{\rho}\otimes]^3 &= [\boldsymbol{\rho}\otimes][\boldsymbol{\rho}\otimes]^2 \\
&= [\boldsymbol{\rho}\otimes]\left[\boldsymbol{\rho}\boldsymbol{\rho}^{\mathrm{T}} - |\boldsymbol{\rho}|^2\mathbf{I}_3\right]^2 \\
&= -|\boldsymbol{\rho}|^2[\boldsymbol{\rho}\otimes], \\
[\boldsymbol{\rho}\otimes]^4 &= -|\boldsymbol{\rho}|^2[\boldsymbol{\rho}\otimes]^2, \\
&\vdots \\
[\boldsymbol{\rho}\otimes]^{2k+1} &= (-1)^k|\boldsymbol{\rho}|^{2k}[\boldsymbol{\rho}\otimes], \\
[\boldsymbol{\rho}\otimes]^{2k+2} &= (-1)^k|\boldsymbol{\rho}|^{2k}[\boldsymbol{\rho}\otimes]^2,
\end{aligned}$$

so that the exponential expansion

$$\begin{aligned}
\exp([\boldsymbol{\rho}\otimes]) &= \sum_{l=1}^{+\infty} \frac{1}{\ell!}[\boldsymbol{\rho}\otimes]^\ell \\
&= [\boldsymbol{\rho}\otimes]^0 + \frac{1}{|\boldsymbol{\rho}|}\left\{\sum_{k=0}^{+\infty}\frac{(-1)^k|\boldsymbol{\rho}|^{2k+1}}{(2k+1)!}\right\}[\boldsymbol{\rho}\otimes] + \frac{1}{|\boldsymbol{\rho}|^2}\left\{\sum_{k=0}^{+\infty}\frac{(-1)^k|\boldsymbol{\rho}|^{2k+2}}{(2k+2)!}\right\}[\boldsymbol{\rho}\otimes]^2 \\
&= \cos(|\boldsymbol{\rho}|)\mathbf{I}_2 + \frac{1-\cos(|\boldsymbol{\rho}|)}{|\boldsymbol{\rho}|^2}\boldsymbol{\rho}\boldsymbol{\rho}^{\mathrm{T}} + \frac{\sin(|\boldsymbol{\rho}|)}{|\boldsymbol{\rho}|}\begin{bmatrix} 0 & -\rho_3 & \rho_2 \\ \rho_3 & 0 & -\rho_1 \\ -\rho_2 & \rho_1 & 0 \end{bmatrix},
\end{aligned} \tag{B.17}$$

where ! denotes the factorial function (defined in Eq. B.16).

B.6.5 Similarity Transformations and Analytic Functions

For any $n \times n$ nonsingular matrix $\mathbf{A}$, the transform $\boldsymbol{X} \rightarrow \mathbf{A}^{-1}\boldsymbol{X}\mathbf{A}$ is called a *similarity transformation* of the $n \times n$ matrix $\boldsymbol{X}$. It is a useful transformation for analytic functions of matrices

$$f(\boldsymbol{X}) = \sum_{k=0}^{\infty} a_k \boldsymbol{X}^k,$$

because

$$\begin{aligned} f(\mathbf{A}^{-1}\boldsymbol{X}\mathbf{A}) &= \sum_{k=0}^{\infty} a_k (\mathbf{A}^{-1}\boldsymbol{X}\mathbf{A})^k \\ &= \mathbf{A}^{-1}\left(\sum_{k=0}^{\infty} a_k \boldsymbol{X}^k\right)\mathbf{A} \\ &= \mathbf{A}^{-1} f(\boldsymbol{X})\mathbf{A}. \end{aligned}$$

If the characteristic values of $\boldsymbol{X}$ are distinct, then the similarity transform performed with the characteristic vectors of $\boldsymbol{X}$ as the column vectors of $\mathbf{A}$ will diagonalize $\boldsymbol{X}$ with its characteristic values along the main diagonal:

$$\begin{aligned} \mathbf{A}^{-1}\boldsymbol{X}\mathbf{A} &= \operatorname{diag}_\ell\{\lambda_\ell\}, \\ f(\mathbf{A}^{-1}\boldsymbol{X}\mathbf{A}) &= \operatorname{diag}_\ell\{\mathbf{F}(\lambda_l \ell)\}, \\ f(\boldsymbol{X}) &= \mathbf{A} \operatorname{diag}_\ell\{\mathbf{F}(\lambda_\ell)\}\mathbf{A}^{-1}. \end{aligned}$$

(Although this is a useful analytical approach for demonstrating functional dependencies, it is not considered a robust numerical method.)

B.7 NORMS

B.7.1 Normed Linear Spaces

Vectors and matrices can be considered as elements of *linear spaces*, in that they can be added and multiplied by scalars. A *norm* is *any* nonnegative real-valued function $\|\cdot\|$ defined on a linear space such that, for any scalar s and elements $\mathbf{x}$ and $\mathbf{y}$ of the linear space (vectors *or* matrices),

$$\begin{aligned} \|x\| = 0 \quad &\text{iff} \quad x = 0, \\ \|x\| > 0 \quad &\text{iff} \quad x \neq 0, \\ \|sx\| \quad &= \quad |s|\|x\|, \\ \|x + y\| \quad &\leq \quad \|x\| + \|y\|, \end{aligned}$$

where iff stands for "if and only if." These constraints are rather loose, and many possible norms can be defined for a particular linear space. A linear space with a specified norm is called a *normed linear space*. The norm induces a *topology* on the linear space, which is used to define continuity and convergence. Norms are also used in numerical analysis for establishing error bounds and in sensitivity analysis for bounding sensitivities. The multiplicity of norms is useful in these applications, because the user is free to pick the one that works best for her or his particular problem.

We define here many of the more popular norms, some of which are known by more than one name.

B.7.2 Matrix Norms

Many norms have been defined for matrices. Two general types are presented here. Both are derived from vector norms, but by different means.

Generalized Vector Norms Vector norms can be generalized to matrices by treating the matrix like a doubly-subscripted vector. For example, the Hölder norms for vectors can be generalized to matrices as

$$\|\mathbf{A}\|_{(p)} = \left\{ \sum_{i=1}^{m} \sum_{j=1}^{n} |a_{i,j}|^p \right\}^{1/p}.$$

The matrix (2)-norm defined in this way is also called the *Euclidean norm*, *Schur norm*, or *Frobenius norm*. We will use the notation $\|\cdot\|_F$ in place of $\|\cdot\|_{(2)}$ for the Frobenius norm.

The reason for putting the parentheses around the subscript p in the above definition is that there is another way that the vector p-norms are used to define matrix norms, and it is with this alternative definition that they are usually allowed to wear an unadorned p subscript. These alternative norms also have the following desirable properties.

Desirable Multiplicative Properties of Matrix Norms Because matrices can be multiplied, one could also apply the additional constraint that

$$\|\mathbf{AB}\|_M \leq \|\mathbf{A}\|_M \, \|\mathbf{B}\|_M$$

for conformable matrices $\mathbf{A}$ and $\mathbf{B}$ and a matrix norm $\|\cdot\|_M$. This is a good property to have for some applications. One might also insist on a similar property with respect to multiplication by vector $\mathbf{x}$, for which a norm $\|\cdot\|_{V_1}$ may already be defined:

$$\|\mathbf{A}x\|_{V_2} \leq \|\mathbf{A}\|_M \|x\|_{V_1}.$$

This property is called *compatibility* between the matrix norm $\|\cdot\|_M$ and the vector norms $\|\cdot\|_{V_1}$ and $\|\cdot\|_{V_2}$. (Note that there can be two distinct vector norms associated with a matrix norm: one in the normed linear space containing $\mathbf{x}$ and one in the space containing $\mathbf{A}x$.)

Matrix Norms Subordinate to Vector Hölder Norms There is a family of alternative matrix "p-norms" [but not (p)-norms] defined by the formula

$$\|\mathbf{A}\|_p \stackrel{\text{def}}{=} \sup_{\|x\|\neq 0} \frac{\|\mathbf{A}x\|_p}{\|x\|_p},$$

where the norms on the right-hand side are the vector Hölder norms and the induced matrix norms on the left are called *subordinate* to the corresponding Hölder norms. The 2-norm defined in this way is also called the *spectral norm* of $\mathbf{A}$. It has the properties:

$$\|\text{diag}_i\{\lambda_i\}\|_2 = \max_i |\lambda_i| \quad \text{and} \quad \|\mathbf{A}x\|_2 \leq \|\mathbf{A}\|_2 \|x\|_2.$$

The first of these properties implies that $\|\mathbf{I}\|_2 = 1$. The second property is compatibility between the spectral norm and the vector Euclidean norm. (Subordinate matrix norms are guaranteed to be compatible with the vector norms used to define them.) All matrix norms subordinate to vector norms also have the property that $\|\mathbf{I}\| = 1$.

Computation of Matrix Hölder Norms The following formulas may be used in computing 1-norms and ∞-norms of $m \times n$ matrices $\mathbf{A}$:

$$\|\mathbf{A}\|_1 = \max_{i\leq j\leq n}\left\{\sum_{i=1}^{m} |a_{ij}|\right\},$$

$$\|\mathbf{A}\|_\infty = \max_{1\leq i\leq m}\left\{\sum_{j=1}^{n} |a_{ij}|\right\}.$$

The norm $\|\mathbf{A}\|_2$ can be computed as the square root of the largest characteristic value of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$, which takes considerably more effort.

Default Matrix Norm When the type of norm applied to a matrix is not specified (by an appropriate subscript), the default will be the spectral norm (Hölder matrix

2-norm). It satisfies the following bounds with respect to the Frobenius norm and the other matrix Hölder norms for $m \times n$ matrices $\mathbf{A}$:

$$\begin{aligned}
\|\mathbf{A}\|_2 &\leq \|\mathbf{A}\|_F \leq \sqrt{n}\|\mathbf{A}\|_2, \\
\frac{1}{\sqrt{m}}\|\mathbf{A}\|_1 &\leq \|\mathbf{A}\|_2 \leq \sqrt{n}\|\mathbf{A}\|_1, \\
\frac{1}{\sqrt{n}}\|\mathbf{A}\|_\infty &\leq \|\mathbf{A}\|_2 \leq \sqrt{m}\|\mathbf{A}\|_\infty, \\
\max_{\substack{1\leq i\leq m \\ 1\leq j\leq n}} |a_{ij}| &\leq \|\mathbf{A}\|_F \leq \sqrt{mn} \max_{\substack{1\leq i\leq m \\ 1\leq j\leq n}} |a_{ij}|.
\end{aligned}$$

B.8 FACTORIZATIONS AND DECOMPOSITIONS

Decompositions are also called *factorizations* of matrices. These are generally represented by algorithms or formulas for representing a matrix as a product of matrix factors with useful properties. The two factorization algorithms described here have either triangular or diagonal factors in addition to orthogonal factors.

Decomposition methods are algorithms for computing the factors, given the matrix to be "decomposed."

B.8.1 Cholesky Decomposition

This decomposition is named after André Louis Cholesky [9], who was perhaps not the first discoverer of the method for factoring a symmetric, positive-definite matrix $\mathbf{P}$ as a product of triangular factors.

Cholesky Factors A Cholesky factor of a symmetric positive-definite matrix $\mathbf{P}$ is a matrix $\mathbf{C}$ such that

$$\mathbf{C}\mathbf{C}^{\mathrm{T}} = \mathbf{P}. \tag{B.18}$$

Note that it does not matter whether we write this equation in the alternative form $\mathbf{F}^{\mathrm{T}}\mathbf{F} = \mathbf{P}$, because the two solutions are related by $\mathbf{F} = \mathbf{C}^{\mathrm{T}}$.

Cholesky factors are not unique, however. If $\mathbf{C}$ is a Cholesky factor of $\mathbf{P}$, then for any conformable orthogonal matrix $\mathbf{M}$, the matrix

$$\mathbf{A} \stackrel{\text{def}}{=} \mathbf{C}\mathbf{M}$$

satisfies the equation

$$\begin{aligned}
\mathbf{A}\mathbf{A}^{\mathrm{T}} &= \mathbf{C}\mathbf{M}(\mathbf{C}\mathbf{M})^{\mathrm{T}} \\
&= \mathbf{C}\mathbf{M}\mathbf{M}^{\mathrm{T}}\mathbf{C}^{\mathrm{T}} \\
&= \mathbf{C}\mathbf{C}^{\mathrm{T}} \\
&= \mathbf{P}.
\end{aligned} \tag{B.19}$$

That is, $\mathbf{A}$ is also a legitimate Cholesky factor. The ability to transform one Cholesky factor into another using orthogonal matrices will turn out to be very important in square-root filtering (in Section 8.1.6).

Cholesky Factoring Algorithms There are two possible forms of the Cholesky factorization algorithm, corresponding to two possible forms of the defining equation:

$$\mathbf{P} = \mathbf{L}_1\mathbf{L}_1^{\mathrm{T}} = \mathbf{U}_1^{\mathrm{T}}\mathbf{U}_1 \tag{B.20}$$

$$= \mathbf{U}_2\mathbf{U}_2^{\mathrm{T}} = \mathbf{L}_2^{\mathrm{T}}\mathbf{L}_2, \tag{B.21}$$

where the Cholesky factors $\mathbf{U}_1$, $\mathbf{U}_2$ are upper triangular and their respective transposes $\mathbf{L}_1$, $\mathbf{L}_2$ are lower triangular.

The first of these is implemented by the built-in MATLAB function `chol(P)`, with argument $\mathbf{P}$ a symmetric positive-definite matrix. The call `chol(P)` returns an upper triangular matrix $\mathbf{U}_1$ satisfying Eq. B.20. The MATLAB m-file `chol2.m` on the accompanying diskette implements the solution to Eq. B.21. The call `chol2(P)` returns an upper triangular matrix $\mathbf{U}_2$ satisfying Eq. B.21.

Modified Cholesky Factorization The algorithm for Cholesky factorization of a matrix requires taking square roots, which can be avoided by using a *modified Cholesky factorization* in the form

$$\mathbf{P} = \mathbf{U}\mathbf{D}\mathbf{U}^{\mathrm{T}}, \tag{B.22}$$

where $\mathbf{D}$ is a diagonal matrix with positive diagonal elements and $\mathbf{U}$ is a *unit triangular matrix* (i.e., $\mathbf{U}$ has 1's along its main diagonal). This algorithm is implemented in the m-file `modchol.m` on the accompanying diskette.

B.8.2 QR Decomposition (Triangularization)

The $\mathbf{QR}$ *decomposition* of a matrix $\mathbf{A}$ is a representation in the form

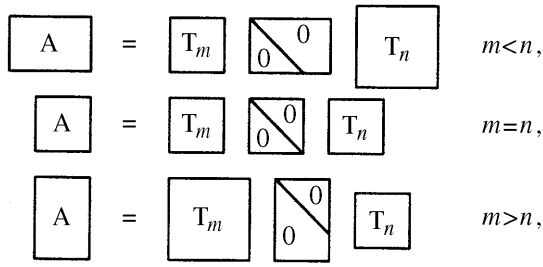
$$\mathbf{A} = \mathbf{Q}\mathbf{R},$$

where $\mathbf{Q}$ is an orthogonal matrix and $\mathbf{R}$ is a triangular matrix. Numerical methods for $\mathbf{QR}$ decomposition are also called "triangularization" methods. Some of these methods are an integral part of square-root Kalman filtering and are presented in Section 8.1.6.3.

B.8.3 Singular-Value Decomposition

The *singular-value decomposition* of an $m \times n$ matrix $\mathbf{A}$ is a representation in the form $\mathbf{A} = \mathbf{T}_m\mathbf{D}\mathbf{T}_n$, where $\mathbf{T}_m$ and $\mathbf{T}_n$ are orthogonal matrices (with square dimen-

sions as specified by their subscripts) and **D** is an $m \times n$ matrix filled with zeros everywhere except along the main diagonal of its maximal upper left square submatrix. This decomposition will have either of three forms:



depending on the relative values of m and n. The middle matrix **D** has the block form

$$\mathbf{D} = \begin{cases} \left[\text{diag}_i\{\sigma_i\} | 0_{m\times(n-m)}\right] & \text{if } m < n, \\ \text{diag}_i\{\sigma_i\} & \text{if } m = n, \\ \begin{bmatrix} \text{diag}_i\{\sigma_i\} \\ 0_{(m-n)\times n} \end{bmatrix} & \text{if } m > n, \end{cases}$$

$$\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \cdots \geq \sigma_p \geq 0,$$

$$p = \min(m, n).$$

That is, the diagonal nonzero elements of **D** are in *descending order*, and nonnegative. These are called the *singular values* of **A**. For a proof that this decomposition exists, and an algorithm for computing it, see the book by Golub and Van Loan [41].

The singular values of a matrix characterize many useful matrix properties, such as

$$\|\mathbf{A}\|_2 = \sigma_1(\mathbf{A}),$$

rank $(\mathbf{A}) = r$ such that $\sigma_r > 0$ and either $\sigma_{r+1} = 0$ or $r = p$ (the rank of a matrix is defined in Section B.1.5.2), and

the *condition number* of **A** equals σ_1/σ_p.

The condition number of the matrix **A** in the linear equation $\mathbf{A}x = b$ bounds the sensitivity of the solution **x** to variations in b and the sensitivity of the solution to roundoff errors in determining it. The singular-value decomposition may also be used to define the "pseudorank" of **A** as the smallest singular value σ_i such that $\sigma_i > \varepsilon\sigma_1$, where ε is a processor- and precision-dependent constant such that $0 < \varepsilon \ll 1$ and $1 + \varepsilon \equiv 1$ in machine precision.

These relationships are useful for the analysis of state transition matrices Φ of Kalman filters, which can be singular or close enough to being singular that numerical roundoff can cause the product $\Phi P \Phi^{\mathrm{T}}$ to be essentially singular.

B.8.4 Eigenvalue–Eigenvector Decompositions of Symmetric Matrices

Symmetric QR Decomposition The so-called "symmetric QR" decomposition of an $n \times n$ symmetric real matrix $\mathbf{A}$ has the special form $\mathbf{A} = \mathbf{TDT}^T$, where the right orthogonal matrix is the transposed left orthogonal matrix and the diagonal matrix

$$\mathbf{D} = \mathrm{diag}_i\{\lambda_i\}.$$

That is, the diagonal elements are the characteristic values of the symmetric matrix. Furthermore, the column vectors of the orthogonal matrix $\mathbf{T}$ are the associated characteristic vectors $\mathbf{e}_i$ of $\mathbf{A}$:

$$\begin{aligned}\mathbf{A} &= \mathbf{TDT}^{\mathrm{T}} \\ &= \sum_{i=1}^{n} \lambda_i \mathbf{e}_i \mathbf{e}_i^{\mathrm{T}}, \\ \mathbf{T} &= [\mathbf{e}_1 \quad \mathbf{e}_2 \quad \mathbf{e}_3 \quad \cdots \quad \mathbf{e}_n].\end{aligned}$$

These relationships are useful for the analysis of covariance matrices, which are constrained to have nonnegative characteristic values, although their numerical values may stray enough in practice (due to computer roundoff errors) to develop negative characteristic values.

B.9 QUADRATIC FORMS

Bilinear and Quadratic Forms For a matrix $\mathbf{A}$ and all conformable column vectors $\mathbf{x}$ and $\mathbf{y}$, the functional mapping $(x, y) \rightarrow x^{\mathrm{T}}\mathbf{A}y$ is called a *bilinear form*. As a function of $\mathbf{x}$ and $\mathbf{y}$, it is linear in both $\mathbf{x}$ and $\mathbf{y}$ and hence *bilinear*. In the case that $x = y$, the functional mapping $x \rightarrow x^{\mathrm{T}}\mathbf{A}x$ is called a *quadratic form*. The matrix $\mathbf{A}$ of a quadratic form is always a square matrix.

B.9.1 Symmetric Decomposition of Quadratic Forms

Any square matrix $\mathbf{A}$ can be represented uniquely as the sum of a symmetric matrix and a skew-symmetric matrix:

$$\mathbf{A} = \tfrac{1}{2}\left(\mathbf{A} + \mathbf{A}^{\mathrm{T}}\right) + \tfrac{1}{2}\left(\mathbf{A} - \mathbf{A}^{\mathrm{T}}\right),$$

where $\frac{1}{2}\left(\mathbf{A} + \mathbf{A}^{\mathrm{T}}\right)$ is called the *symmetric part* of $\mathbf{A}$ and $\frac{1}{2}\left(\mathbf{A} - \mathbf{A}^{\mathrm{T}}\right)$ is called the *skew-symmetric part* of $\mathbf{A}$. The quadratic form $x^{\mathrm{T}}\mathbf{A}x$ depends only on the symmetric part of $\mathbf{A}$:

$$x^{\mathrm{T}}\mathbf{A}x = x^{\mathrm{T}}\left\{\tfrac{1}{2}\left(\mathbf{A} + \mathbf{A}^{\mathrm{T}}\right)\right\}x.$$

Therefore, one can always assume that the matrix of a quadratic form is symmetric, and one can express the quadratic form in summation form as

$$x^{\mathrm{T}}\mathbf{A}x = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}, x_i x_j = \sum_{i=j} a_{ij}x_i x_j + \sum_{i\neq j} a_{ij}x_i x_j = \sum_{i=1}^{n} a_{ii}x_i^2 + 2\sum_{i<j} a_{ij}x_i x_j$$

for symmetric $\mathbf{A}$.

Ranges of Quadratic Forms The domain of a quadratic form for an $n \times n$ matrix is n-dimensional Euclidean space, and the range is in $(-\infty, +\infty)$, the real line. In the case that $x \neq 0$,

$$\begin{aligned}
\text{if } \mathbf{A} \text{ is } \textit{positive definite}, &\quad \text{the range of } x \to x^{\mathrm{T}}\mathbf{A}x \text{ is } &(0, +\infty);\\
\text{if } \mathbf{A} \text{ is } \textit{non-negative definite}, &\quad \text{the range of } x \to x^{\mathrm{T}}\mathbf{A}x \text{ is } &[0, +\infty);\\
\text{if } \mathbf{A} \text{ is } \textit{indefinite}, &\quad \text{the range of } x \to x^{\mathrm{T}}\mathbf{A}x \text{ is } &(-\infty, +\infty);\\
\text{if } \mathbf{A} \text{ is } \textit{non-positive definite}, &\quad \text{the range of } x \to x^{\mathrm{T}}\mathbf{A}x \text{ is } &(-\infty, 0];\\
\text{if } \mathbf{A} \text{ is } \textit{negative definite}, &\quad \text{the range of } x \to x^{\mathrm{T}}\mathbf{A}x \text{ is } &(-\infty, 0).
\end{aligned}$$

If $x^{\mathrm{T}}x = 1$, then $\lambda_n(\mathbf{A}) \leq x^{\mathrm{T}}\mathbf{A}x \leq \lambda_1(\mathbf{A})$. That is, the quadratic form maps the unit n-sphere onto the closed interval $[\lambda_n(\mathbf{A}), \lambda_1(\mathbf{A})]$.

B.10 DERIVATIVES OF MATRICES

B.10.1 Derivatives of Matrix-Valued Functions

The derivative of a matrix with respect to a scalar is the matrix of derivatives of its elements:

$$\mathbf{F}(t) = \begin{bmatrix}
f_{11}(t) & f_{12}(t) & f_{13}(t) & \cdots & f_{1n}(t)\\
f_{21}(t) & f_{22}(t) & f_{23}(t) & \cdots & f_{2n}(t)\\
f_{31}(t) & f_{32}(t) & f_{33}t & \cdots & f_{3n}(t)\\
\vdots & \vdots & \vdots & & \vdots\\
f_{m1}(t) & f_{m2}(t) & f_{m3}(t) & \ldots f_{mn}(t) &
\end{bmatrix},$$

$$\frac{d}{dt}\,\mathbf{F}(t) = \begin{bmatrix}
\frac{d}{dt}f_{11}(t) & \frac{d}{dt}f_{12}(t) & \frac{d}{dt}f_{13}(t) & \cdots & \frac{d}{dt}f_{1n}(t)\\
\frac{d}{dt}f_{21}(t) & \frac{d}{dt}f_{22}(t) & \frac{d}{dt}f_{23}(t) & \cdots & \frac{d}{dt}f_{2n}(t)\\
\frac{d}{dt}f_{31}(t) & \frac{d}{dt}f_{32}(t) & \frac{d}{dt}f_{33}(t) & \cdots & \frac{d}{dt}f_{3n}(t)\\
\vdots & \vdots & \vdots & & \vdots\\
\frac{d}{dt}f_{m1}(t) & \frac{d}{dt}f_{m2}(t) & \frac{d}{dt}f_{m3}(t) & \cdots & \frac{d}{dt}f_{mn}(t)
\end{bmatrix}.$$

The rule for the derivative of a product applies also to matrix products:

$$\frac{d}{dt}[\mathbf{A}(t)\mathbf{B}(t)] = \left[\frac{d}{dt}\,\mathbf{A}(t)\right]\mathbf{B}(t) + \mathbf{A}(t)\left[\frac{d}{dt}\,\mathbf{B}(t)\right],$$

provided that the order of the factors is preserved.

Derivative of Matrix Inverse If $\mathbf{F}(t)$ is square and nonsingular, then $\mathbf{F}(t)\mathbf{F}^{-1}(t) = \mathbf{I}$, a constant. As a consequence, its derivative will be zero. This fact can be used to derive the formula for the derivative of a matrix inverse:

$$\begin{aligned} 0 &= d/dt\mathbf{I} \\ &= d/dt[\mathbf{F}(t)\mathbf{F}^{-1}(t)] \\ &= [d/dt\mathbf{F}(t)]\mathbf{F}^{-1}(t) + \mathbf{F}(t)[d/dt\mathbf{F}^{-1}(t)], \\ d/dt\mathbf{F}^{-1}(t) &= -\mathbf{F}^{-1}[d/dt\mathbf{F}(t)]\mathbf{F}^{-1}. \end{aligned} \tag{B.23}$$

Derivative of Orthogonal Matrix If the $\mathbf{F}(t)$ is orthogonal, its inverse $\mathbf{F}^{-1}(t) = \mathbf{F}^{\mathrm{T}}(t)$, its transpose, and because

$$d/dt\mathbf{F}^{\mathrm{T}}(t) = [d/dt\mathbf{F}(t)]^{\mathrm{T}} = \dot{\mathbf{F}}^{\mathrm{T}},$$

one can show that orthogonal matrices satisfy matrix differential equations with antisymmetric dynamic coefficient matrices:

$$\begin{aligned} 0 &= \frac{d}{dt}\mathbf{I} \\ &= \frac{d}{dt}[\mathbf{F}(t)\mathbf{F}^{\mathrm{T}}(t)] \\ &= \dot{\mathbf{F}}(t)\mathbf{F}^{\mathrm{T}}(t) + \mathbf{F}(t)\dot{\mathbf{F}}^{\mathrm{T}}(t), \\ \dot{\mathbf{F}}(t)\mathbf{F}^{\mathrm{T}}(t) &= -[\mathbf{F}(t)\dot{\mathbf{F}}^{\mathrm{T}}(t)] \\ &= -[\dot{\mathbf{F}}(t)\mathbf{F}^{\mathrm{T}}(t)]^{\mathrm{T}} \\ &= \text{antisymmetric matrix} \\ &= \Omega_{\text{left}}, \\ \dot{\mathbf{F}}(t) &= \Omega_{\mathit{left}}\mathbf{F}(t), \end{aligned} \qquad \begin{aligned} 0 &= \frac{d}{dt}\mathbf{I} \\ &= \frac{d}{dt}[\mathbf{F}^{\mathrm{T}}(t)\mathbf{F}(t)] \\ &= \dot{\mathbf{F}}^{\mathrm{T}}(t)\mathbf{F}(t) + \mathbf{F}^{\mathrm{T}}(t)\dot{\mathbf{F}}(t), \\ \mathbf{F}^{\mathrm{T}}(t)\dot{\mathbf{F}}(t) &= -[\dot{\mathbf{F}}(t)\mathbf{F}(t)\mathbf{f}(t)] \\ &= -[\mathbf{F}^{\mathrm{T}}(t)\dot{\mathbf{F}}(t)]^{\mathrm{T}} \\ &= \text{antisymmetric matrix} \\ &= \Omega_{\text{right}}, \\ \dot{\mathbf{F}}(t) &= \mathbf{F}(t)\Omega_{\text{right}}. \end{aligned}$$

That is, all time-differentiable orthogonal matrices $\mathbf{F}(t)$ satisfy dynamic equations with antisymmetric coefficient matrices, which can be either left- or right-side coefficient matrices.

B.10.2 Gradients of Quadratic Forms

If $f(x)$ is a differentiable scalar-valued function of an n-vector $\mathbf{x}$, then the vector

$$\frac{\partial f}{\partial x} = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \frac{\partial f}{\partial x_3}, \ldots, \frac{\partial f}{\partial x_n}\right]^{\mathrm{T}}$$

is called the *gradient* of f with respect to $\mathbf{x}$. In the case that f is a quadratic form with symmetric matrix $\mathbf{A}$, then the ith component of its gradient will be

$$\begin{aligned}
\left[\frac{\partial}{\partial x}(x^{\mathrm{T}}\mathbf{A}x)\right]_i &= \frac{\partial f}{\partial x_i}\left(\sum_j a_{jj}x_j^2 + 2\sum_{j<k} a_{jk}x_j x_k\right) \\
&= \left(2a_{ii}x_i + 2\sum_{i<k} a_{ik}x_k + 2\sum_{j<i} a_{ji}x_j\right) \\
&= \left(2a_{ii}x_i + 2\sum_{i\neq k} a_{ik}x_k\right) \\
&= 2\sum_{k=1}^{n} a_{ik}x_k \\
&= [2\mathbf{A}x]_i.
\end{aligned}$$

That is, the gradient vector can be expressed as

$$\frac{\partial}{\partial x}\left(x^T\mathbf{A}x\right) = 2\mathbf{A}x.$$

Global Positioning Systems, Inertial Navigation, and Integration,
Mohinder S. Grewal, Lawrence R. Weill, Angus P. Andrews

Print ISBN 0-471-35032-X Electronic ISBN 0-471-20071-9

Appendix C

Coordinate Transformations

C.1 NOTATION

We use the notation $\mathbf{C}_{\text{to}}^{\text{from}}$ to denote a coordinate transformation matrix from one coordinate frame (designated by "from") to another coordinated frame (designated by "to"). For example,

- $\mathbf{C}_{\text{ENU}}^{\text{ECI}}$ denotes the coordinate transformation matrix from earth-centered inertial (ECI) coordinates to earth-fixed east–north–up (ENU) local coordinates and
- $\mathbf{C}_{\text{NED}}^{\text{RPY}}$ denotes the coordinate transformation matrix from vehicle body-fixed roll–pitch–yaw (RPY) coordinates to earth-fixed north–east–down (NED) coordinates.

Coordinate transformation matrices satisfy the composition rule

$$\mathbf{C}_C^B \mathbf{C}_B^A = \mathbf{C}_C^A,$$

where A, B, and C represent different coordinate frames.

What we mean by a coordinate transformation matrix is that if a vector $\mathbf{v}$ has the representation

$$\mathbf{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \tag{C.1}$$

in *XYZ* coordinates and the same vector $\mathbf{v}$ has the alternative representation

$$\mathbf{v} = \begin{bmatrix} v_u \\ v_v \\ v_w \end{bmatrix} \tag{C.2}$$

in *UVW* coordinates, then

$$\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \mathbf{C}_{XYW}^{UVW} \begin{bmatrix} v_u \\ v_v \\ v_w \end{bmatrix}, \tag{C.3}$$

where "*XYZ*" and "*UVW*" stand for any two Cartesian coordinate systems in three-dimensional space.

The components of a vector in either coordinate system can be expressed in terms of the vector components along unit vectors parallel to the respective coordinate axes. For example, if one set of coordinate axes is labeled X, Y and Z, and the other set of coordinate axes are labeled U, V, and W, then the same vector $\mathbf{v}$ can be expressed in either coordinate frame as

$$\mathbf{v} = v_x \mathbf{1}_x + v_y \mathbf{1}_y + v_z \mathbf{1}_z \tag{C.4}$$

$$= v_u \mathbf{1}_u + v_v \mathbf{1}_v + v_w \mathbf{1}_w, \tag{C.5}$$

where

- the unit vectors $\mathbf{1}_x$, $\mathbf{1}_y$, and $\mathbf{1}_z$ are along the *XYZ* axes;
- the scalars v_x, v_y, and v_z are the respective components of $\mathbf{v}$ along the *XYZ* axes;
- the unit vectors $\mathbf{1}_u$, $\mathbf{1}_v$, and $\mathbf{1}_w$ are along the *UVW* axes; and
- the scalars v_u, v_v, and v_w are the respective components of $\mathbf{v}$ along the *UVW* axes.

The respective components can also be represented in terms of dot products of $\mathbf{v}$ with the various unit vectors,

$$v_x = \mathbf{1}_x^{\mathrm{T}} \mathbf{v} = v_u \mathbf{1}_x^{\mathrm{T}} \mathbf{1}_u + v_v \mathbf{1}_x^{\mathrm{T}} \mathbf{1}_v + v_w \mathbf{1}_x^{\mathrm{T}} \mathbf{1}_w, \tag{C.6}$$

$$v_y = \mathbf{1}_y^{\mathrm{T}} \mathbf{v} = v_u \mathbf{1}_y^{\mathrm{T}} \mathbf{1}_u + v_v \mathbf{1}_y^{\mathrm{T}} \mathbf{1}_v + v_w \mathbf{1}_y^{\mathrm{T}} \mathbf{1}_w, \tag{C.7}$$

$$v_z = \mathbf{1}_z^{\mathrm{T}} \mathbf{v} = v_u \mathbf{1}_z^{\mathrm{T}} \mathbf{1}_u + v_v \mathbf{1}_z^{\mathrm{T}} \mathbf{1}_v + v_w \mathbf{1}_z^{\mathrm{T}} \mathbf{1}_w, \tag{C.8}$$

which can be represented in matrix form as

$$\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} \mathbf{1}_x^{\mathrm{T}}\mathbf{1}_u & \mathbf{1}_x^{\mathrm{T}}\mathbf{1}_v & \mathbf{1}_x^{\mathrm{T}}\mathbf{1}_w \\ \mathbf{1}_y^{\mathrm{T}}\mathbf{1}_u & \mathbf{1}_y^{\mathrm{T}}\mathbf{1}_v & \mathbf{1}_y^{\mathrm{T}}\mathbf{1}_w \\ \mathbf{1}_z^{\mathrm{T}}\mathbf{1}_u & \mathbf{1}_z^{\mathrm{T}}\mathbf{1}_v & \mathbf{1}_z^{\mathrm{T}}\mathbf{1}_w \end{bmatrix} \begin{bmatrix} v_u \\ v_v \\ v_w \end{bmatrix} \tag{C.9}$$

$$\stackrel{\text{def}}{=} \mathbf{C}_{XYZ}^{UVW} \begin{bmatrix} v_u \\ v_v \\ v_w \end{bmatrix}, \tag{C.10}$$

which defines the coordinate transformation matrix $\mathbf{C}_{XYZ}^{UVW}$ from UVW to XYZ coordinates in terms of the dot products of unit vectors. However, dot products of unit vectors also satisfy the cosine rule (defined in Section B.1.2.5)

$$\mathbf{1}_a^T\mathbf{1}_b = \cos\left(\theta_{ab}\right), \tag{C.11}$$

where θ_{ab} is the angle between the unit vectors $\mathbf{1}_a$ and $\mathbf{1}_b$. As a consequence, the coordinate transformation matrix can also be written in the form

$$C_{XYZ}^{UVW} = \begin{bmatrix} \cos(\theta_{xu}) & \cos(\theta_{xv}) & \cos(\theta_{xw}) \\ \cos(\theta_{yu} & \cos(\theta_{yv}) & \cos(\theta_{yw}) \\ \cos(\theta_{zu}) & \cos(\theta_{zv}) & \cos(\theta_{zw}) \end{bmatrix}, \tag{C.12}$$

which is why coordinate transformation matrices are also called "direction cosines matrices."

Navigation makes use of coordinates that are natural to the problem at hand: inertial coordinates for inertial navigation, orbital coordinates for GPS navigation, and earth-fixed coordinates for representing locations on the earth.

The principal coordinate systems used in navigation, and the transformations between these different coordinate systems, are summarized in this appendix. These are primarily Cartesian (orthogonal) coordinates, and the transformations between them can be represented by orthogonal matrices. However, the coordinate transformations can also be represented by rotation vectors or quaternions, and all representations are used in the derivations and implementation of GPS/INS integration.

C.2 INERTIAL REFERENCE DIRECTIONS

C.2.1 Vernal Equinox

The equinoxes are those times of year when the length of the day equals the length of the night (the meaning of "equinox"), which only happens when the sun is over the

equator. This happens twice a year: when the sun is passing from the Southern Hemisphere to the Northern Hemisphere (vernal equinox) and again when it is passing from the Northern Hemisphere to the Southern Hemisphere (autumnal equinox). The time of the vernal equinox defines the beginning of spring (the meaning of "vernal") in the Northern Hemisphere, which usually occurs around March 21–23.

The direction from the earth to the sun at the instant of the vernal equinox is used as a "quasi-inertial" direction in some navigation coordinates. This direction is defined by the intersection of the equatorial plane of the earth with the ecliptic (earth–sun plane). These two planes are inclined at about 23.45°, as illustrated in Fig. C.1. The inertial direction of the vernal equinox is changing ever so slowly, on the order of 5 arc seconds per year, but the departure from truly inertial directions is neglible over the time periods of most navigation problems. The vernal equinox was in the constellation Pisces in the year 2000. It was in the constellation Aries at the time of Hipparchus (190–120 BCE) and is sometimes still called "the first point of Aries."

C.2.2 Polar Axis of Earth

The one inertial reference direction that remains invariant in earth-fixed coordinates as the earth rotates is its polar axis, and that direction is used as a reference direction in inertial coordinates. Because the polar axis is (by definition) orthogonal to the earth's equatorial plane and the vernal equinox is (by definition) in the earth's equatorial plane, the earth's polar axis will always be orthogonal to the vernal equinox.

A third orthogonal axis can then be defined (by their cross-product) such that the three axes define a right-handed (defined in Section B.2.11) orthogonal coordinate system.

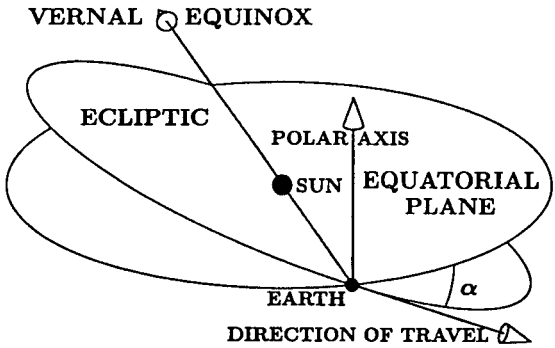


Fig. C.1 *Direction of Vernal Equinox.*

C.3 COORDINATE SYSTEMS

Although we are concerned exclusively with coordinate systems in the three dimensions of the observable world, there are many ways of representing a location in that world by a set of coordinates. The coordinates presented here are those used in navigation with GPS and/or INS.

C.3.1 Cartesian and Polar Coordinates

René Descartes (1596–1650) introduced the idea of representing points in three-dimensional space by a triplet of coordinates, called "Cartesian coordinates" in his honor. They are also called "Euclidean coordinates," but not because Euclid discovered them first. The Cartesian coordinates (x, y, z) and polar coordinates (θ, ϕ, r) of a common reference point, as illustrated in Fig. C.2, are related by the equations

$$x = r\cos(\theta)\cos(\phi), \tag{C.13}$$

$$y = r\sin(\theta)\cos(\phi), \tag{C.14}$$

$$z = r\sin(\phi), \tag{C.15}$$

$$r = \sqrt{x^2 + y^2 + z^2}, \tag{C.16}$$

$$\phi = \arcsin\left(\frac{z}{r}\right) \qquad \left(-\tfrac{1}{2}\pi \le \phi \le +\tfrac{1}{2}\pi\right), \tag{C.17}$$

$$\theta = \arctan\left(\frac{y}{x}\right) \qquad (-\pi < \theta \le +\pi), \tag{C.18}$$

with the angle θ (in radians) undefined if $\phi = \pm\frac{1}{2}\pi$.

C.3.2 Celestial Coordinates

The "celestial sphere" is a system for inertial directions referenced to the polar axis of the earth and the vernal equinox. The polar axis of these celestial coordinates is

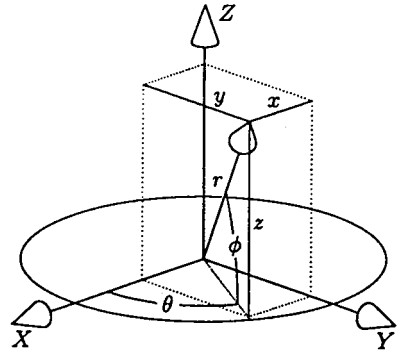


Fig. C.2 *Cartesian and polar coordinates.*

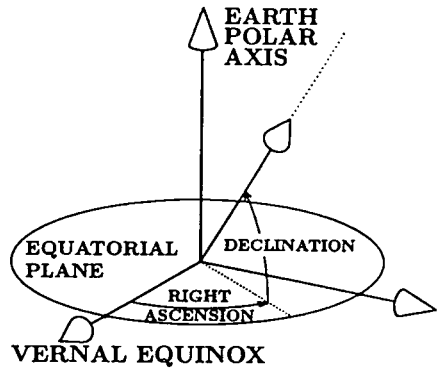


Fig. C.3 *Celestial coordinates.*

parallel to the polar axis of the earth and its prime meridian is fixed to the vernal equinox. Polar celestial coordinates are *right ascension* (the celestial analog of longitude, measured eastward from the vernal equinox) and *declination* (the celestial analog of latitude), as illustrated in Fig. C.3. Because the celestial sphere is used primarily as a reference for direction, no origin need be specified.

Right ascension is zero at the vernal equinox and increases eastward (in the direction the earth turns). The units of right ascension (RA) can be radians, degrees, or hours (with 15 deg/h as the conversion factor).

By convention, declination is zero in the equatorial plane and increases toward the north pole, with the result that celestial objects in the Northern Hemisphere have positive declinations. Its units can be degrees or radians.

C.3.3 Satellite Orbit Coordinates

Johannes Kepler (1571–1630) discovered the geometric shapes of the orbits of planets and the minimum number of parameters necessary to specify an orbit (called "Keplerian" parameters). Keplerian parameters used to specify GPS satellite orbits in terms of their orientations relative to the equatorial plane and the vernal equinox (defined in Section C.2.1 and illustrated in Fig. C.1) include the following:

- Right ascension of the ascending node and orbit inclination, specifying the orientation of the orbital plane with respect to the vernal equinox and equatorial plane, is illustrated in Fig. C.4.
  (a) Right ascension is defined in the previous section and is shown in Fig. C.3.
  (b) The intersection of the orbital plane of a satellite with the equatorial plane is called its "line of nodes," where the "nodes" are the two intersections of the satellite orbit with this line. The two nodes are dubbed "ascending"[1]

[1]The astronomical symbol for the ascending node is ☊, often read as "earphones."

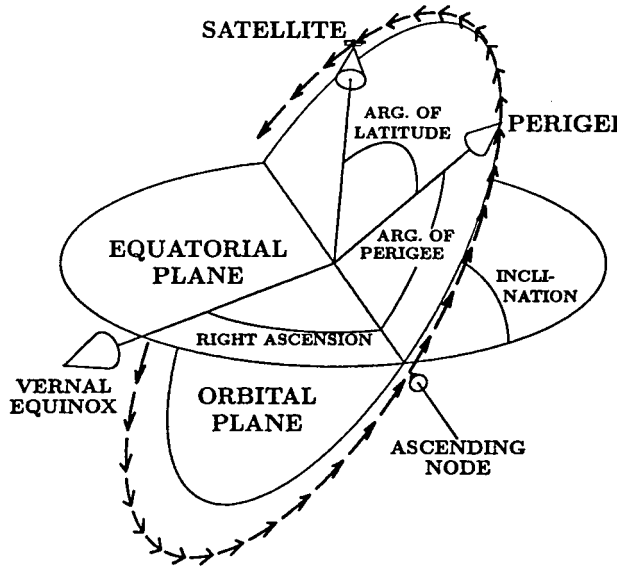


Fig. C.4 *Keplerian parameters for satellite orbit.*

(i.e., ascending from the Southern Hemisphere to the Northern Hemisphere) and "descending". The right ascension of the ascending node (RAAN) is the angle in the equatorial plane from the vernal equinox to the ascending node, measured counterclockwise as seen looking down from the north pole direction.

(c) Orbital inclination is the dihedral angle between the orbital plane and the equatorial plane. It ranges from zero (orbit in equatorial plane) to 180°.

- Semimajor axis a and semiminor axis b (defined in Section C.3.5.2 and illustrated in Fig. C.6) specify the size and shape of the elliptical orbit within the orbital plane.
- Orientation of the ellipse within its orbital plane, specified in terms of the "argument of perigee," the angle between the ascending node and the perigee of the orbit (closest approach to earth), is illustrated in Fig. C.4.
- Position of the satellite relative to perigee of the elliptical orbit, specified in terms of the angle from perigee, called the "argument of latitude" or "true anomaly," is illustrated in Fig. C.4.

For computer simulation demonstrations, GPS satellite orbits can usually be assumed to be circular with radius $a = b = R = 26{,}560$ km and inclined at 55° to the equatorial plane. This eliminates the need to specify the orientation of the elliptical orbit within the orbital plane. (The argument of perigee becomes overly sensitive to orbit perturbations when eccentricity is close to zero.)

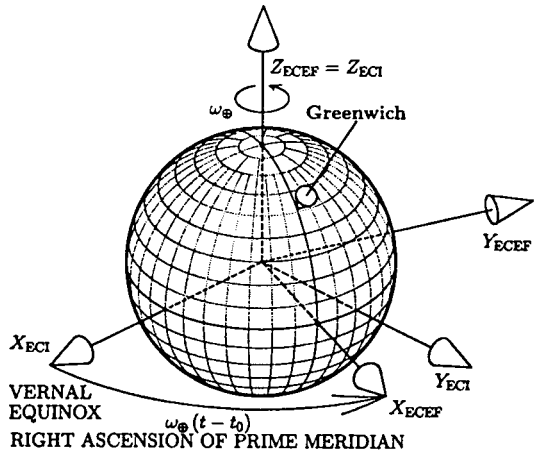


Fig. C.5 *ECI and ECEF Coordinates.*

C.3.4 ECI Coordinates

Earth-centered inertial (ECI) coordinates are the favored inertial coordinates in the near-earth environment. The origin of ECI coordinates is at the center of gravity of the earth, with (Fig. C.5)

1. axis in the direction of the vernal equinox,
3. axis direction parallel to the rotation axis (north polar axis) of the earth, and
2. an additional axis to make this a right-handed orthogonal coordinate system, with the polar axis as the third axis (hence the numbering).

The equatorial plane of the earth is also the equatorial plane of ECI coordinates, but the earth itself is rotating relative to the vernal equinox at its sidereal rotation rate of about $7{,}292{,}115{,}167 \times 10^{-14}$ rad/s, or about 15.04109 deg/h, as illustrated in Fig. C.5.

C.3.5 ECEF Coordinates

Earth-centered, earth-fixed (ECEF) coordinates have the same origin (earth center) and third (polar) axis as ECI coordinates but rotate with the earth, as shown in Fig. C.5. As a consequence, ECI and ECEF longitudes differ only by a linear function of time.

Longitude in ECEF coordinates is measured east (+) and west (−) from the prime meridian passing through the principal transit instrument at the observatory at Greenwich, UK, a convention adopted by 41 representatives of 25 nations at the International Meridian Conference, held in Washington, DC, in October of 1884.

Latitudes are measured with respect to the equatorial plane, but there is more than one kind of "latitude." *Geocentric latitude* would be measured as the angle between the equatorial plane and a line from the reference point to the center of the earth, but this angle could not be determined accurately (before GPS) without running a transit survey over vast distances. The angle between the pole star and the local vertical direction could be measured more readily, and that angle is more closely approximated as *geodetic latitude*. There is yet a third latitude (parametric latitude) that is useful in analysis. The latter two latitudes are defined in the following subsections.

C.3.5.1 Ellipsoidal Earth Models *Geodesy* is the study of the size and shape of the earth and the establishment of physical control points defining the origin and orientation of coordinate systems for mapping the earth. Earth shape models are very important for navigation using either GPS or INS, or both. INS alignment is with respect to the local vertical, which does not generally pass through the center of the earth. That is because the earth is not spherical.

At different times in history, the earth has been regarded as being flat (first-order approximation), spherical (second-order), and ellipsoidal (third-order). The third-order model is an ellipsoid of revolution, with its shorter radius at the poles and its longer radius at the equator.

C.3.5.2 Parametric Latitude For geoids based on ellipsoids of revolution, every meridian is an ellipse with equatorial radius a (also called "semimajor axis") and polar radius b (also called "semiminor axis"). If we let z be the Cartesian coordinate in the polar direction and $x_{\text{meridional}}$ be the equatorial coordinate in the meridional plane, as illustrated in Fig. C.6, then the equation for this ellipse will be

$$\frac{x_{\text{meridional}}^2}{a^2} + \frac{z^2}{b^2} = 1 \tag{C.19}$$

$$= \cos^2(\phi_{\text{parametric}}) + \sin^2(\phi_{\text{parametric}}) \tag{C.20}$$

$$= \frac{a^2 \cos^2(\phi_{\text{parametric}})}{a^2} + \frac{b^2 \sin^2(\phi_{\text{parametric}})}{b^2} \tag{C.21}$$

$$= \frac{[a \cos(\phi_{\text{parametric}})]^2}{a^2} + \frac{[b \sin(\phi_{\text{parametric}})]^2}{b^2}. \tag{C.22}$$

That is, a parametric solution for the ellipse is

$$x_{\text{meridional}} = a \cos(\phi_{\text{parametric}}), \tag{C.23}$$

$$z = b \sin(\phi_{\text{parametric}}), \tag{C.24}$$

as illustrated in Fig. C.6. Although the parametric latitude $\phi_{\text{parametric}}$ has no physical significance, it is quite useful for relating geocentric and geodetic latitude, which do have physical significance.

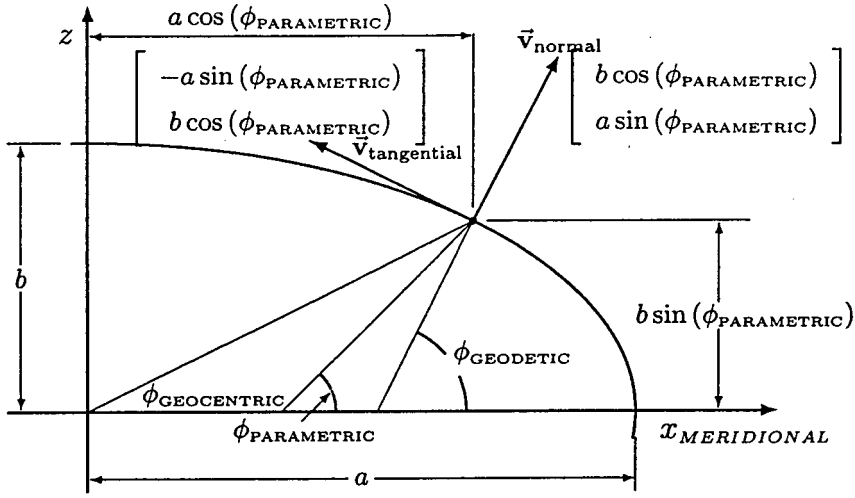


Fig. C.6 *Geocentric, parametric, and geodetic latitudes in meridional plane.*

C.3.5.3 Geodetic Latitude Geodetic latitude is defined as the elevation angle above (+) or below (−) the equatorial plane of the normal to the ellipsoidal surface. This direction can be defined in terms of the parametric latitude, because it is orthogonal to the meridional tangential direction.

The vector tangential to the meridian will be in the direction of the derivative to the elliptical equation solution with respect to parametric latitude:

$$\mathbf{v}_{\text{tangential}} \propto \frac{\partial}{\partial \phi_{\text{parametric}}} \begin{bmatrix} a\cos(\phi_{\text{parametric}}) \\ b\sin(\phi_{\text{parametric}}) \end{bmatrix} \tag{C.25}$$

$$= \begin{bmatrix} -a\sin(\phi_{\text{parametric}}) \\ b\cos(\phi_{\text{parametric}}) \end{bmatrix}, \tag{C.26}$$

and the meridional normal direction will be orthogonal to it, or

$$\mathbf{v}_{\text{normal}} \propto \begin{bmatrix} b\cos(\phi_{\text{parametric}}) \\ a\sin(\phi_{\text{parametric}}) \end{bmatrix}, \tag{C.27}$$

as illustrated in Fig. C.6.

The tangent of geodetic latitude is then the ratio of the z- and x-components of the surface normal vector, or

$$\tan(\phi_{\text{geodetic}}) = \frac{a\sin(\phi_{\text{parametric}})}{b\cos(\phi_{\text{parametric}})} \tag{C.28}$$

$$= \frac{a}{b}\tan(\phi_{\text{parametric}}), \tag{C.29}$$

from which, using some standard trigonometric identities,

$$\sin(\phi_{\text{geodetic}}) = \frac{\tan(\phi_{\text{geodetic}})}{\sqrt{1+\tan^2(\phi_{\text{geodetic}})}} \tag{C.30}$$

$$= \frac{a\sin(\phi_{\text{parametric}})}{\sqrt{a^2\sin^2(\phi_{\text{parametric}}) + b^2\cos^2(\phi_{\text{parametric}})}}, \tag{C.31}$$

$$\cos(\phi_{\text{geodetic}}) = \frac{1}{\sqrt{1+\tan^2(\phi_{\text{geodetic}})}} \tag{C.32}$$

$$= \frac{b\cos(\phi_{\text{parametric}})}{\sqrt{a^2\sin^2(\phi_{\text{parametric}}) + b^2\cos^2(\phi_{\text{parametric}})}}. \tag{C.33}$$

The inverse relationship is

$$\tan(\phi_{\text{parametric}}) = \frac{b}{a}\tan(\phi_{\text{geodetic}}), \tag{C.34}$$

from which, using the same trigonometric identities as before,

$$\sin(\phi_{\text{parametric}}) = \frac{\tan(\phi_{\text{parametric}})}{\sqrt{1+\tan^2(\phi_{\text{parametric}})}} \tag{C.35}$$

$$= \frac{b\sin(\phi_{\text{geodetic}})}{\sqrt{a^2\cos^2(\phi_{\text{geodetic}}) + b^2\sin^2(\phi_{\text{geodetic}})}}, \tag{C.36}$$

$$\cos(\phi_{\text{parametric}}) = \frac{1}{\sqrt{1+\tan^2(\phi_{\text{parametric}})}} \tag{C.37}$$

$$= \frac{a\cos(\phi_{\text{geodetic}})}{\sqrt{a^2\cos^2(\phi_{\text{geodetic}}) + b^2\sin^2(\phi_{\text{geodetic}})}}, \tag{C.38}$$

and the two-dimensional X-Z Cartesian coordinates in the meridional plane of a point on the geoid surface will

$$x_{\text{meridional}} = a\cos(\phi_{\text{parametric}}) \tag{C.39}$$

$$= \frac{a^2\cos(\phi_{\text{geodetic}})}{\sqrt{a^2\cos^2(\phi_{\text{geodetic}}) + b^2\sin^2(\phi_{\text{geodetic}})}}, \tag{C.40}$$

$$z = b\sin(\phi_{\text{parametric}}) \tag{C.41}$$

$$= \frac{b^2\sin(\phi_{\text{geodetic}})}{\sqrt{a^2\cos^2(\phi_{\text{geodetic}}) + b^2\sin^2(\phi_{\text{geodetic}})}} \tag{C.42}$$

in terms of geodetic latitude.

Equations C.40 and C.42 apply only to points on the geoid surface. Orthometric height h above (+) or below (−) the geoid surface is measured along the surface normal, so that the X-Z coordinates for a point with altitude h will be

$$x_{\text{meridional}} = \cos(\phi_{\text{geodetic}}) \times \left(h + \frac{a^2}{\sqrt{a^2 \cos^2(\phi_{\text{geodetic}}) + b^2 \sin^2(\phi_{\text{geodetic}})}} \right), \tag{C.43}$$

$$z = \sin(\phi_{\text{geodetic}}) \times \left(h + \frac{b^2}{\sqrt{a^2 \cos^2(\phi_{\text{geodetic}}) + b^2 \sin^2(\phi_{\text{geodetic}})}} \right). \tag{C.44}$$

In three-dimensional ECEF coordinates, with the X-axis passing through the equator at the prime meridian (at which longitude $\theta = 0$),

$$x_{\text{ECEF}} = \cos(\theta) x_{\text{meridional}} \tag{C.45}$$

$$= \cos(\theta) \cos(\phi_{\text{geodetic}}) \times \left(h + \frac{a^2}{\sqrt{a^2 \cos^2(\phi_{\text{geodetic}}) + b^2 \sin^2(\phi_{\text{geodetic}})}} \right), \tag{C.46}$$

$$y_{\text{ECEF}} = \sin(\theta) x_{\text{meridional}} \tag{C.47}$$

$$= \sin(\theta) \cos(\phi_{\text{geodetic}}) \times \left(h + \frac{a^2}{\sqrt{a^2 \cos^2(\phi_{\text{geodetic}}) + b^2 \sin^2(\phi_{\text{geodetic}})}} \right), \tag{C.48}$$

$$z_{\text{ECEF}} = \sin(\phi_{\text{geodetic}}) \times \left(h + \frac{b^2}{\sqrt{a^2 \cos^2(\phi_{\text{geodetic}}) + b^2 \sin^2(\phi_{\text{geodetic}})}} \right), \tag{C.49}$$

in terms of geodetic latitude ϕ_{geodetic}, longitude θ, and orthometric altitude h with respect to the reference geoid.

The inverse transformation, from ECEF *XYZ* to geodetic longitude–latitude–altitude coordinates, is

$$\theta = \text{atan2}(y_{\text{ECEF}}, x_{\text{ECEF}}), \tag{C.50}$$

$$\phi_{\text{geodetic}} = \text{atan2}\left(z_{\text{ECEF}} + \frac{e^2 a^2 \sin^3(\zeta)}{b}, \xi - e^2 a \cos^3(\zeta)\right), \tag{C.51}$$

$$h = \frac{\xi}{\cos(\phi)} - r_T, \tag{C.52}$$

where atan2 is the four-quadrant arctangent function in MATLAB and

$$\zeta = \text{atan2}(az_{\text{ECEF}}, b\xi), \tag{C.53}$$

$$\xi = \sqrt{x_{\text{ECEF}}^2 + y_{\text{ECEF}}^2}, \tag{C.54}$$

$$r_T = \frac{a}{\sqrt{1 - e^2 \sin(\phi)}}, \tag{C.55}$$

where r_T is the transverse radius of curvature on the ellipsoid, a is the equatorial radius, b is the polar radius, and e is elliptical eccentricity.

C.3.5.4 Geocentric Latitude For points on the geoid surface, the tangent of geocentric latitude is the ratio of distance above (+) or below (−) the equator [$z = b \sin(\phi_{\text{parametric}})$] to the distance from the polar axis [$(x_{\text{meridional}} = a \cos(\phi_{\text{parametric}})$], or

$$\tan(\phi_{\text{GEOCENTRIC}}) = \frac{b \sin(\phi_{\text{parametric}})}{a \cos(\phi_{\text{parametric}})} \tag{C.56}$$

$$= \frac{b}{a} \tan(\phi_{\text{parametric}}) \tag{C.57}$$

$$= \frac{b^2}{a^2} \tan(\phi_{\text{geodetic}}), \tag{C.58}$$

from which, using the same trigonometric identities as were used for geodetic latitude,

$$\sin(\phi_{\text{geocentric}}) = \frac{\tan(\phi_{\text{geocentric}})}{\sqrt{1+\tan^2(\phi_{\text{geocentric}})}} \tag{C.59}$$

$$= \frac{b\sin(\phi_{\text{parametric}})}{\sqrt{a^2\cos^2(\phi_{\text{parametric}}) + b^2\sin^2(\phi_{\text{parametric}})}} \tag{C.60}$$

$$= \frac{b^2\sin(\phi_{\text{geodetic}})}{\sqrt{a^4\cos^2(\phi_{\text{geodetic}}) + b^4\sin^2(\phi_{\text{geodetic}})}}, \tag{C.61}$$

$$\cos(\phi_{\text{geocentric}}) = \frac{1}{\sqrt{1+\tan^2(\phi_{\text{geocentric}})}} \tag{C.62}$$

$$= \frac{a\cos(\phi_{\text{parametric}})}{\sqrt{a^2\cos^2(\phi_{\text{parametric}}) + b^2\sin^2(\phi_{\text{parametric}})}} \tag{C.63}$$

$$= \frac{a^2\cos(\phi_{\text{geodetic}})}{\sqrt{a^4\cos^2(\phi_{\text{geodetic}}) + b^4\sin^2(\phi_{\text{geodetic}})}}. \tag{C.64}$$

The inverse relationships are

$$\tan(\phi_{\text{parametric}}) = \frac{a}{b}\tan(\phi_{\text{geocentric}}), \tag{C.65}$$

$$\tan(\phi_{\text{geodetic}}) = \frac{a^2}{b^2}\tan(\phi_{\text{geocentric}}), \tag{C.66}$$

from which, using the same trigonometric identities again,

$$\sin(\phi_{\text{parametric}}) = \frac{\tan(\phi_{\text{parametric}})}{\sqrt{1+\tan^2(\phi_{\text{parametric}})}} \tag{C.67}$$

$$= \frac{a\sin(\phi_{\text{geocentric}})}{\sqrt{a^2\sin^2(\phi_{\text{geocentric}}) + b^2\cos^2(\phi_{\text{geocentric}})}}, \tag{C.68}$$

$$\sin(\phi_{\text{geodetic}}) = \frac{a^2\sin(\phi_{\text{geocentric}})}{\sqrt{a^4\sin^2(\phi_{\text{geocentric}}) + b^4\cos^2(\phi_{\text{geocentric}})}}, \tag{C.69}$$

$$\cos(\phi_{\text{parametric}}) = \frac{1}{\sqrt{1+\tan^2(\phi_{\text{parametric}})}} \tag{C.70}$$

$$= \frac{b\cos(\phi_{\text{geocentric}})}{\sqrt{a^2\sin^2(\phi_{\text{geocentric}}) + b^2\cos^2(\phi_{\text{geocentric}})}}, \tag{C.71}$$

$$\cos(\phi_{\text{geodetic}}) = \frac{b^2\cos(\phi_{\text{geocentric}})}{\sqrt{a^4\sin^2(\phi_{\text{geocentric}}) + b^4\cos^2(\phi_{\text{geocentric}})}}. \tag{C.72}$$

C.3.6 LTP Coordinates

Local tangent plane (LTP) coordinates, also called "locally level coordinates," are a return to the first-order model of the earth as being flat, where they serve as local reference directions for representing vehicle attitude and velocity for operation on or near the surface of the earth. A common orientation for LTP coordinates has one horizontal axis (the north axis) in the direction of increasing latitude and the other horizontal axis (the east axis) in the direction of increasing longitude, as illustrated in Fig. C.7. Horizontal location components in this local coordinate frame are called "relative northing" and "relative easting."

C.3.6.1 Alpha Wander Coordinates Maintaining east–north orientation was a problem for some INSs at the poles, where north and east directions change by 180°. Early gimbaled inertial systems could not slew the platform axes fast enough for near-polar operation. This problem was solved by letting the platform axes "wander" from north but keeping track of the angle α between north and a reference platfrom axis, as shown in Fig. C.8. This LTP orientation came to be called "alpha wander.""

C.3.6.2 ENU/NED Coordinates East–north–up (ENU) and north–east–down (NED) are two common right-handed LTP coordinate systems. ENU coordinates may be preferred to NED coordinates because altitude increases in the upward direction. But NED coordinates may also be preferred over ENU coordinates because the direction of a right (clockwise) turn is in the positive direction with respect to a downward axis, and NED coordinate axes coincide with vehicle-fixed roll–pitch–yaw (RPY) coordinates (Section C.3.7) when the vehicle is level and headed north.

The coordinate transformation matrix $\mathbf{C}_{\text{NED}}^{\text{ENU}}$ from ENU to NED coordinates and the transformation matrix $\mathbf{C}_{\text{ENU}}^{\text{NED}}$ from NED to ENU coordinates are one and the same:

$$\mathbf{C}_{\text{NED}}^{\text{ENU}} = \mathbf{C}_{\text{ENU}}^{\text{NED}} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}. \tag{C.73}$$

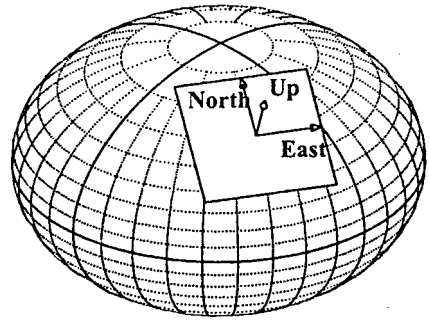


Fig. C.7 *ENU coordinates.*

C.3.6.3 ENU/ECEF Coordinates The unit vectors in local *east*, *north*, and *up* directions, as expressed in ECEF Cartesian coordinates, will be

$$\mathbf{1}_E = \begin{bmatrix} -\sin(\theta) \\ \cos(\theta) \\ 0 \end{bmatrix}, \tag{C.74}$$

$$\mathbf{1}_N = \begin{bmatrix} -\cos(\theta)\ \sin(\phi_{\text{geodetic}}) \\ -\sin(\theta)\ \sin(\phi_{\text{geodetic}}) \\ \cos(\phi_{\text{geodetic}}) \end{bmatrix}, \tag{C.75}$$

$$\mathbf{1}_U = \begin{bmatrix} \cos(\theta)\ \cos(\phi_{\text{geodetic}}) \\ \sin(\theta)\ \cos(\phi_{\text{geodetic}}) \\ \sin(\phi_{\text{geodetic}}) \end{bmatrix}, \tag{C.76}$$

and the unit vectors in the ECEF X, Y, and Z directions, as expressed in ENU coordinates, will be

$$\mathbf{1}_X = \begin{bmatrix} -\sin(\theta) \\ -\cos(\theta)\ \sin(\phi_{\text{geodetic}}) \\ \cos(\theta)\ \cos(\phi_{\text{geodetic}}) \end{bmatrix}, \tag{C.77}$$

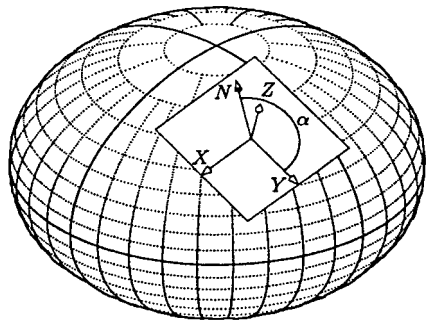


Fig. C.8 *Alpha wander.*

$$\mathbf{1}_Y = \begin{bmatrix} \cos(\theta) \\ -\sin(\theta)\ \sin(\phi_{\text{geodetic}}) \\ \sin(\theta)\ \cos(\phi_{\text{geodetic}}) \end{bmatrix}, \tag{C.78}$$

$$\mathbf{1}_Z = \begin{bmatrix} 0 \\ \cos(\phi_{\text{geodetic}}) \\ \sin(\phi_{\text{geodetic}}) \end{bmatrix}. \tag{C.79}$$

C.3.6.4 NED/ECEF Coordinates It is more natural in some applications to use NED directions for locally level coordinates. This coordinate system coincides with vehicle-body-fixed RPY coordinates (shown in Fig. C.9) when the vehicle is level headed north. The unit vectors in local *north*, *east*, and *down* directions, as expressed in ECEF Cartesian coordinates, will be

$$\mathbf{1}_N = \begin{bmatrix} -\cos(\theta)\ \sin(\phi_{\text{geodetic}}) \\ -\sin(\theta)\ \sin(\phi_{\text{geodetic}}) \\ \cos(\phi_{\text{geodetic}}) \end{bmatrix}, \tag{C.80}$$

$$\mathbf{1}_E = \begin{bmatrix} -\sin(\theta) \\ \cos(\theta) \\ 0 \end{bmatrix}, \tag{C.81}$$

$$\mathbf{1}_D = \begin{bmatrix} -\cos(\theta)\ \cos(\phi_{\text{geodetic}}) \\ -\sin(\theta)\ \cos(\phi_{\text{geodetic}}) \\ -\sin(\phi_{\text{geodetic}}) \end{bmatrix}, \tag{C.82}$$

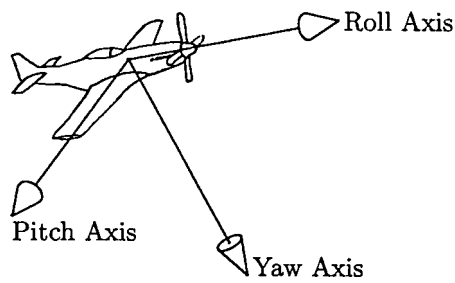


Fig. C.9 *Roll-pitch-yaw axes.*

and the unit vectors in the ECEF X, Y, and Z directions, as expressed in NED coordinates, will be

$$\mathbf{1}_X = \begin{bmatrix} -\cos(\theta)\sin(\phi_{\text{geodetic}}) \\ -\sin(\theta) \\ -\cos(\theta)\cos(\phi_{\text{geodetic}}) \end{bmatrix}, \tag{C.83}$$

$$\mathbf{1}_Y = \begin{bmatrix} -\sin(\theta)\sin(\phi_{\text{geodetic}}) \\ \cos(\theta) \\ -\sin(\theta)\cos(\phi_{\text{geodetic}}) \end{bmatrix}, \tag{C.84}$$

$$\mathbf{1}_Z = \begin{bmatrix} \cos(\phi_{\text{geodetic}}) \\ 0 \\ -\sin(\phi_{\text{geodetic}}) \end{bmatrix}, \tag{C.85}$$

C.3.7 RPY Coordinates

The RPY coordinates are vehicle fixed, with the roll axis in the nominal direction of motion of the vehicle, the pitch axis out the right-hand side, and the yaw axis such that turning to the right is positive, as illustrated in Fig.C.9. The same orientations of vehicle-fixed coordinates are used for surface ships and ground vehicles. They are also called "SAE coordinates," because they are the standard body-fixed coordinates used by the Society of Automotive Engineers.

For rocket boosters with their roll axes vertical at lift-off, the pitch axis is typically defined to be orthogonal to the plane of the boost trajectory (also called the "pitch plane" or "ascent plane").

C.3.8 Vehicle Attitude Euler Angles

The attitude of the vehicle body with respect to local coordinates can be specified in terms of rotations about the vehicle roll, pitch, and yaw axes, starting with these axes aligned with NED coordinates. The angles of rotation about each of these axes are called *Euler angles*, named for the Swiss mathematician Leonard Euler (1707–1783). It is always necessary to specify the order of rotations when specifying Euler (pronounced "oiler") angles.

A fairly common convention for vehicle attitude Euler angles is illustrated in Fig. C.10, where, starting with the vehicle level with roll axis pointed north:

1. *Yaw/Heading*. Rotate through the yaw angle (Y) about the vehicle yaw axis to the intended azimuth (heading) of the vehicle roll axis. Azimuth is measured clockwise (east) from north.
2. *Pitch*. Rotate through the pitch angle (P) about the vehicle pitch axis to bring the vehicle roll axis to its intended elevation. Elevation is measured positive upward from the local horizontal plane.

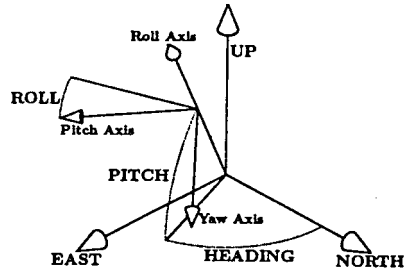


Fig. C.10 *Vehicle Euler angles.*

3. *Roll*. Rotate through the roll angle (R) about the vehicle roll axis to bring the vehicle attitude to the specified orientation.

Euler angles are redundant for vehicle attitudes with 90° pitch, in which case the roll axis is vertical. In that attitude, heading changes also rotate the vehicle about the roll axis. This is the attitude of most rocket boosters at lift-off. Some boosters can be seen making a roll maneuver immediately after lift-off to align their yaw axes with the launch azimuth in the ascent plane. This maneuver may be required to correct for launch delays on missions for which launch azimuth is a function of launch time.

C.3.8.1 RPY/ENU Coordinates With vehicle attitude specified by yaw angle (Y), pitch angle (P), and roll angle (R) as specified above, the resulting unit vectors of the roll, pitch, and yaw axes in ENU coordinates will be

$$\mathbf{1}_R = \begin{bmatrix} \sin(Y)\cos(P) \\ \cos(Y)\cos(P) \\ \sin(P) \end{bmatrix}, \tag{C.86}$$

$$\mathbf{1}_P = \begin{bmatrix} \cos(R)\cos(Y) + \sin(R)\sin(Y)\sin(P) \\ -\cos(R)\sin(Y) + \sin(R)\cos(Y)\sin(P) \\ -\sin(R)\cos(P) \end{bmatrix}, \tag{C.87}$$

$$\mathbf{1}_Y = \begin{bmatrix} -\sin(R)\cos(Y) + \cos(R)\sin(Y)\sin(P) \\ \sin(R)\sin(Y) + \cos(R)\cos(Y)\sin(P) \\ -\cos(R)\cos(P) \end{bmatrix}; \tag{C.88}$$

the unit vectors of the east, north, and up axes in RPY coordinates will be

$$\mathbf{1}_E = \begin{bmatrix} \sin(Y)\cos(P) \\ \cos(R)\cos(Y) + \sin(R)\sin(Y)\sin(P) \\ -\sin(R)\cos(Y) + \cos(R)\sin(Y)\sin(P) \end{bmatrix}, \tag{C.89}$$

$$\mathbf{1}_N = \begin{bmatrix} \cos(Y)\cos(P) \\ -\cos(R)\sin(Y) + \sin(R)\cos(Y)\sin(P) \\ \sin(R)\sin(Y) + \cos(R)\cos(Y)\sin(P) \end{bmatrix}, \tag{C.90}$$

$$\mathbf{1}_U = \begin{bmatrix} \sin(P) \\ -\sin(R)\cos(P) \\ -\cos(R)\cos(P) \end{bmatrix}; \tag{C.91}$$

and the coordinate transformation matrix from RPY coordinates to ENU coordinates will be

$$C_{\text{ENU}}^{\text{RPY}} = [\mathbf{1}_R \quad \mathbf{1}_P \quad \mathbf{1}_Y] = \begin{bmatrix} \mathbf{1}_E^{\text{T}} \\ \mathbf{1}_N^{\text{T}} \\ \mathbf{1}_U^{\text{T}} \end{bmatrix} \tag{C.92}$$

$$= \begin{bmatrix} S_Y C_P & C_R C_Y + S_R S_Y S_P & -S_R C_Y + C_R S_Y S_P \\ C_Y C_P & -C_R S_Y + S_R C_Y S_P & S_R S_Y + C_R C_Y S_P \\ S_P & -S_R C_P & -C_R C_P \end{bmatrix}, \tag{C.93}$$

where

$$S_R = \sin(R), \tag{C.94}$$

$$C_R = \cos(R), \tag{C.95}$$

$$S_P = \sin(P), \tag{C.96}$$

$$C_P \cos(P), \tag{C.97}$$

$$S_Y = \sin(Y), \tag{C.98}$$

$$C_Y = \cos(Y). \tag{C.99}$$

C.3.9 GPS Coordinates

The parameter Ω in Fig. C.12 is the RAAN, which is the ECI longitude where the orbital plane intersects the equatorial plane as the satellite crosses from the Southern Hemisphere to the Northern Hemisphere. The orbital plane is specified by Ω and α, the inclination of the orbit plane with respect to the equatorial plane ($\alpha \approx 55^\circ$ for GPS satellite orbits). The θ parameter represents the location of the satellite within the orbit plane, as the angular phase in the circular orbit with respect to ascending node.

For GPS satellite orbits, the angle θ changes at a nearly constant rate of about 1.4584×10^{-4} rad/s and a period of about 43,082 s (half a day).

The nominal satellite position in ECEF coordinates is then given as

$$x = R[\cos\theta\cos\Omega - \sin\theta\sin\Omega\cos\alpha], \tag{C.100}$$

$$y = R[\cos\theta\cos\Omega + \sin\theta\sin\Omega\cos\alpha], \tag{C.101}$$

$$z = R\sin\theta\sin\alpha, \tag{C.102}$$

$$\theta = \theta_0 + (t - t_0)\frac{360}{43{,}082}\text{deg}, \tag{C.103}$$

$$\Omega = \Omega_0 - (t - t_0)\frac{360}{86{,}164}\text{deg}, \tag{C.104}$$

$$R = 26{,}560{,}000\,\text{m}. \tag{C.105}$$

GPS satellite positions in the transmitted navigation message are specified in the ECEF coordinate system of WGS 84. A locally level x^1, y^1, z^1 reference coordinate system (described in Section C.3.6) is used by an observer location on the earth, where the x^1–y^1 plane is tangential to the surface of the earth, x^1 pointing east, y^1 pointing north, and z^1 normal to the plane. See Fig. C.11. Here,

$$X_{\text{ENU}} = \mathbf{C}_{\text{ENU}}^{\text{ECEF}} X_{\text{ECEF}} + \mathbf{S},$$

$\mathbf{C}_{\text{ENU}}^{\text{ECEF}}$ = coordinate transformation matrix from ECEF to ENU,

$\mathbf{S}$ = coordinate origin shift vector from ECEF to local reference,

$$\mathbf{C}_{\text{ENU}}^{\text{ECEF}} = \begin{bmatrix} -\sin\theta & \cos\theta & 0 \\ -\sin\phi\cos\theta & -\sin\phi\sin\theta & \cos\phi \\ \cos\phi\cos\theta & \cos\phi\sin\theta & \sin\phi \end{bmatrix},$$

$$S = \begin{bmatrix} X_U\sin\theta - Y_U\cos\theta \\ X_U\sin\phi\cos\theta - Y_U\sin\phi\sin\theta - Z_U\cos\phi \\ -X_U\cos\phi\cos\theta - Y_U\cos\phi\sin\theta - Z_U\sin\phi \end{bmatrix},$$

X_U, Y_U, Z_U = user's position,

θ = local reference longitude,

ϕ = local geometric latitude.

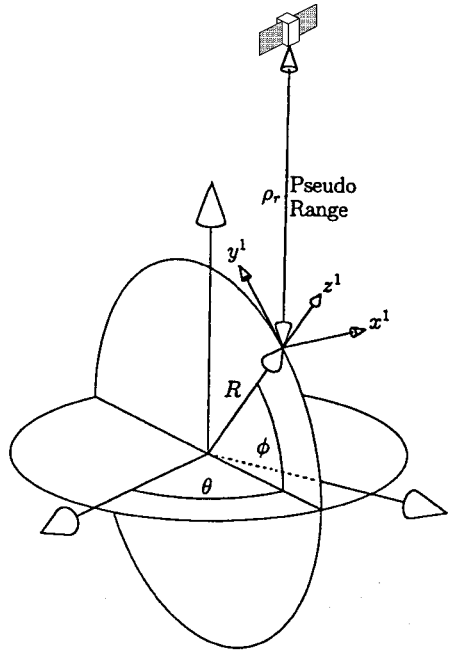


Fig. C.11 *Pseudorange.*

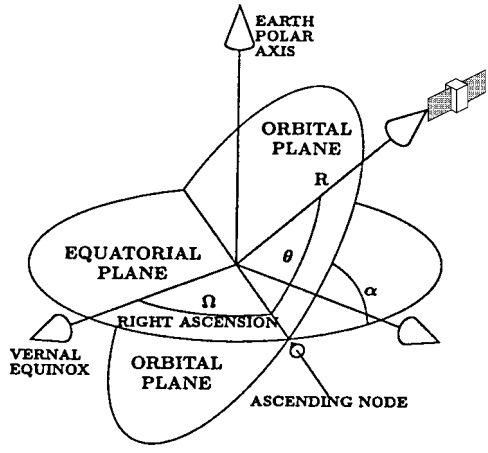


Fig. C.12 *Satellite coordinates.*

C.4 COORDINATE TRANSFORMATION MODELS

Coordinate transformations are methods for transforming a vector represented in one coordinate system into the appropriate representation in another coordinate system. These coordinate transformations can be represented in a number of different ways, each with its advantages and disadvantages.

These transformations generally involve translations (for coordinate systems with different origins) and rotations (for Cartesian coordinate systems with different axis directions) or transcendental transformations (between Cartesian and polar or geodetic coordinates). The transformations between Cartesian and polar coordinates have already been discussed in Section C.3.1 and translations are rather obvious, so we will concentrate on the rotations.

C.4.1 Euler Angles

Euler angles were used for defining vehicle attitude in Section C.3.8, and vehicle attitude representation is a common use of Euler angles in navigation.

Euler angles are used to define a coordinate transformation in terms of a set of three angular rotations, performed in a specified sequence about three specified orthogonal axes, to bring one coordinate frame to coincide with another. The coordinate transformation from RPY coordinates to NED coordinates, for example, can be composed from three Euler rotation matrices:

$$C_{\text{NED}}^{\text{RPY}} = \overbrace{\begin{bmatrix} C_Y & -S_Y & 0 \\ S_Y & C_Y & 0 \\ 0 & 0 & 1 \end{bmatrix}}^{\text{Yaw}} \overbrace{\begin{bmatrix} C_P & 0 & S_P \\ 0 & 1 & 0 \\ -S_P & 0 & C_P \end{bmatrix}}^{\text{Pitch}} \overbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & C_R & -S_R \\ 0 & S_R & C_R \end{bmatrix}}^{\text{Roll}} \tag{C.106}$$

$$= \underbrace{\begin{bmatrix} C_Y P_P & -S_Y C_R + C_Y S_P S_R & S_Y S_R + C_Y S_P C_R \\ S_Y C_P & C_Y C_R + S_Y S_P S_R & -C_Y S_R + S_Y S_P C_R \\ -S_P & C_P S_R & C_P C_R \\ \text{(roll axis)} & \text{(pitch axis)} & \text{(yaw axis)} \end{bmatrix}}_{\text{in NED coordinates}}, \tag{C.107}$$

where the matrix elements are defined in Eqs. C.94–C.99. This matrix also rotates the NED coordinate axes to coincide with RPY coordinate axes. (Compare this with the transformation from RPY to ENU coordinates in Eq. C.93.)

For example, the coordinate transformation for nominal booster rocket launch attitude (roll axis straight up) would be given by Eq. with pitch angle $P = \frac{1}{2}\pi$ ($C_P = 0,\ S_P = 1$), which becomes

$$\mathbf{C}_{\text{NED}}^{\text{RPY}} = \begin{bmatrix} 0 & \sin(R-Y) & \cos(R-Y) \\ 0 & \cos(R-Y) & -\sin(R-Y) \\ 1 & 0 & 0 \end{bmatrix}.$$

That is, the coordinate transformation in this attitude depends only on the difference between roll angle (R) and yaw angle (Y). Euler angles are a concise representation for vehicle attitude. They are handy for driving cockpit displays such as compass cards (using Y) and artificial horizon indicators (using R and P), but they are not particularly handy for representing vehicle attitude dynamics. The reasons for the latter include the following:

- Euler angles have discontinuities analogous to "gimbal lock" (Section 6.4.1.2) when the vehicle roll axis is pointed upward, as it is for launch of many rockets. In that orientation, tiny changes in vehicle pitch or yaw cause $\pm 180^\circ$ changes in heading angle. For aircraft, this creates a slewing rate problem for electromechanical compass card displays.
- The relationships between sensed body rates and Euler angle rates are mathematically complicated.

C.4.2 Rotation Vectors

All right-handed orthogonal coordinate systems with the same origins in three dimensions can be transformed one onto another by single rotations about fixed axes. The corresponding *rotation vectors* relating two coordinate systems are defined by the direction (rotation axis) and magnitude (rotation angle) of that transformation.

For example, the rotation vector for rotating ENU coordinates to NED coordinates (and vice versa) is

$$\boldsymbol{\rho}_{\mathrm{NED}}^{\mathrm{ENU}} = \begin{bmatrix} \pi/\sqrt{2} \\ \pi/\sqrt{2} \\ 0 \end{bmatrix}, \tag{C.108}$$

which has magnitude $|\boldsymbol{\rho}_{\mathrm{NED}}^{\mathrm{ENU}}| = \pi$ (180°) and direction north–east, as illustrated in Fig. C.13. (For illustrative purposes only, NED coordinates are shown as being translated from ENU coordinates in Fig. C.13. In practice, rotation vectors represent pure rotations, without any translation.)

The rotation vector is another minimal representation of a coordinate transformation, along with Euler angles. Like Euler angles, rotation vectors are concise but also have some drawbacks:

1. It is not a unique representation, in that adding multiples of $\pm 2\pi$ to the magnitude of a rotation vector has no effect on the transformation it represents.
2. It is a nonlinear and rather complicated representation, in that the result of one rotation followed by another is a third rotation, the rotation vector for which is a fairly complicated function of the first two rotation vectors.

But, unlike Euler angles, rotation vector models do not exhibit "gimbal lock."

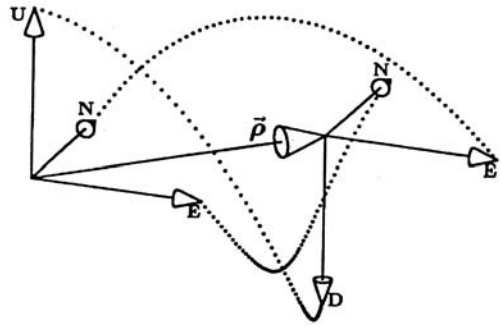


Fig. C.13 *Rotation from ENU to NED coordinates.*

C.4.2.1 Rotation Vector to Matrix The rotation represented by a rotation vector

$$\boldsymbol{\rho} = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \end{bmatrix} \tag{C.109}$$

can be implemented as multiplication by the matrix

$$\mathbf{C}(\boldsymbol{\rho}) \stackrel{\text{def}}{=} \exp(\boldsymbol{\rho}\otimes) \tag{C.110}$$

$$\stackrel{\text{def}}{=} \exp\left(\begin{bmatrix} 0 & -\rho_3 & \rho_2 \\ \rho_3 & 0 & -\rho_1 \\ -\rho_2 & \rho_1 & 0 \end{bmatrix}\right) \tag{C.111}$$

$$= \cos(|\boldsymbol{\rho}|)\mathbf{I}_3 + \frac{1-\cos(|\boldsymbol{\rho}|)}{|\boldsymbol{\rho}|^2}\boldsymbol{\rho}\boldsymbol{\rho}^{\mathrm{T}} + \frac{\sin(|\boldsymbol{\rho}|)}{|\boldsymbol{\rho}|}\begin{bmatrix} 0 & -\rho_3 & \rho_2 \\ \rho_3 & 0 & -\rho_1 \\ -\rho_2 & \rho_1 & 0 \end{bmatrix} \tag{C.112}$$

$$= \cos(\theta)\mathbf{I}_3 + (1-\cos(\theta))\mathbf{1}_\rho\mathbf{1}_\rho^{\mathrm{T}} + \sin(\theta)\begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix}, \tag{C.113}$$

$$\theta \stackrel{\text{def}}{=} |\boldsymbol{\rho}|, \tag{C.114}$$

$$\mathbf{1}_\rho \stackrel{\text{def}}{=} \frac{\boldsymbol{\rho}}{|\boldsymbol{\rho}|}, \tag{C.115}$$

which was derived in Eq. B.17. That is, for any three-rowed column vector $\mathbf{v}$, $\mathbf{C}(\boldsymbol{\rho})\mathbf{v}$ rotates it through an angle of $|\boldsymbol{\rho}|$ radians about the vector $\boldsymbol{\rho}$.

The form of the matrix in Eq. C.113[2] is better suited for computation when $\theta \approx 0$, but the form of the matrix in Eq. C.112 is useful for computing sensitivities using partial derivatives (used in Chapter 8).

For example, the rotation vector $\boldsymbol{\rho}_{\text{NED}}^{\text{ENU}}$ in Eq. C.108 transforming between ENU and NED has magnitude and direction

$$\theta = \pi \qquad [\sin(\theta) = 0, \cos(\theta) = -1],$$
$$\mathbf{1}_\rho = \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \end{bmatrix},$$

respectively, and the corresponding rotation matrix

$$\begin{aligned} \mathbf{C}_{\text{NED}}^{\text{ENU}} &= \cos(\pi)\mathbf{I}_3 + [1-\cos(\pi)]\mathbf{1}_\rho\mathbf{1}_\rho + \sin(\pi)\begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix} \\ &= -\mathbf{I}_3 + 2\mathbf{1}_\rho\mathbf{1}_\rho^{\text{T}} + 0 \\ &= \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \end{aligned}$$

transforms from ENU to NED coordinates. (Compare this result to Eq. C.73.) Because coordinate transformation matrices are orthogonal matrices and the matrix $\mathbf{C}_{\text{NED}}^{\text{ENU}}$ is also symmetric, $\mathbf{C}_{\text{NED}}^{\text{ENU}}$ is its own inverse. That is,

$$\mathbf{C}_{\text{NED}}^{\text{ENU}} = \mathbf{C}_{\text{ENU}}^{\text{NED}}. \tag{C.116}$$

C.4.2.2 Matrix to Rotation Vector Although there is a unique coordinate transformation matrix for each rotation vector, the converse is not true. Adding multiples of 2π to the magnitude of a rotation vector has no effect on the resulting coordinate transformation matrix. The following approach yields a unique rotation vector with magnitude $|\boldsymbol{\rho}| \leq \pi$.

The trace $\text{tr}(\mathbf{C})$ of a square matrix $\mathbf{M}$ is the sum of its diagonal values. For the coordinate transformation matrix of Eq. C.112,

$$\text{tr}[\mathbf{C}(\boldsymbol{\rho})] = 1 + 2\cos(\theta), \tag{C.117}$$

[2]Linear combinations of the sort $a_1\mathbf{I}_{3\times3} + a_2[\mathbf{1}_\rho\otimes] + a_3\mathbf{1}_\rho\mathbf{1}_\rho^{\text{T}}$, where $\mathbf{1}$ is a unit vector, form a subalgebra of 3×3 matrices with relatively simple rules for multiplication, inversion, etc.

from which the rotation angle

$$|\boldsymbol{\rho}| = \theta \tag{C.118}$$

$$= \arccos\left(\frac{\text{tr}[\mathbf{C}(\boldsymbol{\rho})] - 1}{2}\right), \tag{C.119}$$

a formula that will yield a result in the range $0 < \theta < \pi$, but with poor fidelity near where the derivative of the cosine equals zero at $\theta = 0$ and $\theta = \pi$.

The values of θ near $\theta = 0$ and $\theta = \pi$ can be better estimated using the sine of θ, which can be recovered using the antisymmetric part of $\mathbf{C}(\boldsymbol{\rho})$,

$$\mathbf{A} = \begin{bmatrix} 0 & -a_{21} & a_{13} \\ a_{21} & 0 & -a_{32} \\ -a_{13} & a_{32} & 0 \end{bmatrix} \tag{C.120}$$

$$\stackrel{\text{def}}{=} \tfrac{1}{2}[\mathbf{C}(\boldsymbol{\rho}) - \mathbf{C}^{\text{T}}(\boldsymbol{\rho})] \tag{C.121}$$

$$= \frac{\sin(\theta)}{\theta} \begin{bmatrix} 0 & -\rho_3 & \rho_2 \\ \rho_3 & 0 & -\rho_1 \\ -\rho_2 & \rho_1 & 0 \end{bmatrix}, \tag{C.122}$$

from which the vector

$$\begin{bmatrix} a_{32} \\ a_{13} \\ a_{21} \end{bmatrix} = \sin(\theta)\frac{1}{|\boldsymbol{\rho}|}\boldsymbol{\rho} \tag{C.123}$$

will have magnitude

$$\sqrt{a_{32}^2 + a_{13}^2 + a_{21}^2} = \sin(\theta) \tag{C.124}$$

and the same direction as $\boldsymbol{\rho}$. As a consequence, one can recover the magnitude θ of $\boldsymbol{\rho}$ from

$$\theta = \text{atan2}\left(\sqrt{a_{32}^2 + a_{13}^2 + a_{21}^2},\ \frac{\text{tr}\,[\mathbf{C}(\boldsymbol{\rho})] - 1}{2}\right) \tag{C.125}$$

using the MATLAB function atan2, and then the rotation vector $\boldsymbol{\rho}$ as

$$\boldsymbol{\rho} = \frac{\theta}{\sin(\theta)} \begin{bmatrix} a_{32} \\ a_{13} \\ a_{21} \end{bmatrix} \tag{C.126}$$

when $0 < \theta < \pi$.

***C.4.2.3 Special Cases for* $\sin(\boldsymbol{\theta}) \approx \mathbf{0}$** For $\theta \approx 0$, $\boldsymbol{\rho} \approx 0$, although Eq. C.126 may still work adequately for $\theta > 10^{-6}$, say.

For $\theta \approx \pi$, the symmetric part of $\mathbf{C}(\boldsymbol{\rho})$,

$$\mathbf{S} = \begin{bmatrix} s_{11} & s_{12} & s_{13} \\ s_{12} & s_{22} & s_{23} \\ s_{13} & s_{23} & s_{33} \end{bmatrix} \tag{C.127}$$

$$\stackrel{\text{def}}{=} \tfrac{1}{2}[\mathbf{C}(\boldsymbol{\rho}) + \mathbf{C}^{\mathrm{T}}(\boldsymbol{\rho})] \tag{C.128}$$

$$= \cos(\theta)\mathbf{I}_3 + \frac{1-\cos(\theta)}{\theta_2}\boldsymbol{\rho}\boldsymbol{\rho}^{\mathrm{T}} \tag{C.129}$$

$$\approx -\mathbf{I}_3 + \frac{2}{\theta^2}\boldsymbol{\rho}\boldsymbol{\rho}^{\mathrm{T}} \tag{C.130}$$

and the unit vector

$$\mathbf{1}_{\rho} \stackrel{\text{def}}{=} \frac{1}{\theta}\boldsymbol{\rho} \tag{C.131}$$

satisfies

$$\mathbf{S} \approx \begin{bmatrix} 2u_1^2 - 1 & 2u_1u_2 & 2u_1u_3 \\ 2u_1u_2 & 2u_2^2 - 1 & 2u_2u_3 \\ 2u_1u_3 & 2u_2u_3 & 2u_3^2 - 1 \end{bmatrix}, \tag{C.132}$$

which can be solved for a unique $\mathbf{u}$ by assigning $u_k > 0$ for

$$k = \operatorname{argmax}\left(\begin{bmatrix} s_{11} \\ s_{22} \\ s_{33} \end{bmatrix}\right), \tag{C.133}$$

$$u_k = \sqrt{\tfrac{1}{2}(s_{kk} + 1)} \tag{C.134}$$

then, depending on whether $k = 1$, $k = 2$, or $k = 3$,

$$\left.\begin{array}{lccc} & k=1 & k=2 & k=3 \\ u_1 \approx & \sqrt{\dfrac{s_{11}+1}{2}} & \dfrac{s_{12}}{2u_2} & \dfrac{s_{13}}{2u_3} \\ u_2 \approx & \dfrac{s_{12}}{2u_1} & \sqrt{\dfrac{s_{22}+1}{2}} & \dfrac{s_{23}}{2u_2} \\ u_3 \approx & \dfrac{s_{13}}{2u_1} & \dfrac{s_{23}}{2u_2} & \sqrt{\dfrac{s_{11}+1}{2}} \end{array}\right\} \tag{C.135}$$

and

$$\boldsymbol{\rho} = \theta \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}. \tag{C.136}$$

C.4.2.4 Time Derivatives of Rotation Vectors The mathematical relationships between rotation rates ω_k and the time derivatives of the corresponding rotation vector $\boldsymbol{\rho}$ are fairly complicated, but they can be derived from Eq. C.221 for the dynamics of coordinate transformation matrices.

Let $\boldsymbol{\rho}_{\text{ENU}}$ be the rotation vector represented in earth-fixed ENU coordinates that rotates earth-fixed ENU coordinate axes into vehicle body-fixed RPY axes, and let $\mathbf{C}(\boldsymbol{\rho})$ be the corresponding rotation matrix, so that, in ENU coordinates,

$$\begin{aligned}
\mathbf{1}_E &= [1 \quad 0 \quad 0]^{\mathrm{T}}, \qquad \mathbf{1}_N = [0 \quad 1 \quad 0]^{\mathrm{T}}, \qquad \mathbf{1}_U = [0 \quad 0 \quad 1]^{\mathrm{T}}, \\
\mathbf{C}(\boldsymbol{\rho}_{\text{ENU}})\mathbf{1}_E &= \mathbf{1}_R, \qquad \mathbf{C}(\boldsymbol{\rho}_{\text{ENU}})\mathbf{I}_N = \mathbf{1}_P, \qquad \mathbf{C}(\boldsymbol{\rho}_{\text{ENU}})\mathbf{1}_U = \mathbf{1}_Y, \\
\mathbf{C}_{\text{ENU}}^{\text{RPY}} &= [\mathbf{1}_R \quad \mathbf{1}_P \quad \mathbf{1}_Y], \\
&= [\mathbf{C}(\boldsymbol{\rho}_{\text{ENU}})\mathbf{1}_E \quad \mathbf{C}(\boldsymbol{\rho}_{\text{ENU}})\mathbf{1}_N \quad \mathbf{C}(\boldsymbol{\rho}_{\text{ENU}})\mathbf{1}_U] \\
&= \mathbf{C}(\boldsymbol{\rho}_{\text{ENU}})[\mathbf{1}_E \quad \mathbf{1}_N \quad \mathbf{1}_U] \\
&= \mathbf{C}(\boldsymbol{\rho}_{\text{ENU}}) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{C.137}$$

$$\mathbf{C}_{\text{ENU}}^{\text{RPY}} = \mathbf{C}(\boldsymbol{\rho}_{\text{ENU}}). \tag{C.138}$$

That is, $\mathbf{C}(\boldsymbol{\rho}_{\text{ENU}})$ is the coordinate transformation matrix from RPY coordinates to ENU coordinates. As a consequence, from Eq. C.221,

$$\frac{d}{dt}\mathbf{C}(\boldsymbol{\rho}_{\text{ENU}}) = \frac{d}{dt}\mathbf{C}_{\text{ENU}}^{\text{RPY}} \tag{C.139}$$

$$= \begin{bmatrix} 0 & \omega_U & -\omega_N \\ -\omega_U & 0 & \omega_E \\ \omega_N & -\omega_E & 0 \end{bmatrix} \mathbf{C}_{\text{ENU}}^{\text{RPY}} + \mathbf{C}_{\text{ENU}}^{\text{RPY}} \begin{bmatrix} 0 & -\omega_Y & \omega_P \\ \omega_Y & 0 & -\omega_R \\ -\omega_P & \omega_R & 0 \end{bmatrix}, \tag{C.140}$$

$$\frac{d}{dt}\mathbf{C}(\boldsymbol{\rho}_{\text{ENU}}) = \begin{bmatrix} 0 & \omega_U & -\omega_N \\ -\omega_U & 0 & \omega_E \\ \omega_N & -\omega_E & 0 \end{bmatrix} \mathbf{C}(\boldsymbol{\rho}_{\text{ENU}}) + \mathbf{C}(\boldsymbol{\rho}_{\text{ENU}}) \begin{bmatrix} 0 & -\omega_Y & \omega_P \\ \omega_Y & 0 & -\omega_R \\ -\omega_P & \omega_R & 0 \end{bmatrix}, \tag{C.141}$$

where

$$\boldsymbol{\omega}_{\text{RPY}} = \begin{bmatrix} \omega_R \\ \omega_P \\ \omega_Y \end{bmatrix} \tag{C.142}$$

is the vector of inertial rotation rates of the vehicle body, expressed in RPY coordinates, and

$$\boldsymbol{\omega}_{\text{ENU}} = \begin{bmatrix} \omega_E \\ \omega_N \\ \omega_U \end{bmatrix} \tag{C.143}$$

is the vector of inertial rotation rates of the ENU coordinate frame, expressed in ENU coordinates.

The 3×3 matrix equation C.141 is equivalent to nine scalar equations:

$$\begin{aligned}
\frac{\partial c_{11}}{\partial \rho_E}\dot{\rho}_E + \frac{\partial c_{11}}{\partial \rho_N}\dot{\rho}_N + \frac{\partial c_{11}}{\partial \rho_U}\dot{\rho}_U &= -c_{1,3}\omega_P + c_{1,2}\omega_Y - c_{3,1}\omega_N + c_{2,1}\omega_U, \\
\frac{\partial c_{12}}{\partial \rho_E}\dot{\rho}_E + \frac{\partial c_{12}}{\partial \rho_N}\dot{\rho}_N + \frac{\partial c_{12}}{\partial \rho_U}\dot{\rho}_U &= c_{1,3}\omega_R - c_{1,1}\omega_Y - c_{3,2}\omega_N + c_{2,2}\omega_U, \\
\frac{\partial c_{13}}{\partial \rho_E}\dot{\rho}_E + \frac{\partial c_{13}}{\partial \rho_N}\dot{\rho}_N + \frac{\partial c_{13}}{\partial \rho_U}\dot{\rho}_U &= -c_{1,2}\omega_R + c_{1,1}\omega_P - c_{3,3}\omega_N + c_{2,3}\omega_U, \\
\frac{\partial c_{21}}{\partial \rho_E}\dot{\rho}_E + \frac{\partial c_{21}}{\partial \rho_N}\dot{\rho}_N + \frac{\partial c_{21}}{\partial \rho_U}\dot{\rho}_U &= -c_{2,3}\omega_P + c_{2,2}\omega_Y + c_{3,1}\omega_E - c_{1,1}\omega_U, \\
\frac{\partial c_{22}}{\partial \rho_E}\dot{\rho}_E + \frac{\partial c_{22}}{\partial \rho_N}\dot{\rho}_N + \frac{\partial c_{22}}{\partial \rho_U}\dot{\rho}_U &= c_{2,3}\omega_R - c_{2,1}\omega_Y + c_{3,2}\omega_E - c_{1,2}\omega_U, \\
\frac{\partial c_{23}}{\partial \rho_E}\dot{\rho}_E + \frac{\partial c_{23}}{\partial \rho_N}\dot{\rho}_N + \frac{\partial c_{23}}{\partial \rho_U}\dot{\rho}_U &= -c_{2,2}\omega_R + c_{2,1}\omega_P + c_{3,3}\omega_E - c_{1,3}\omega_U, \\
\frac{\partial c_{31}}{\partial \rho_E}\dot{\rho}_E + \frac{\partial c_{31}}{\partial \rho_N}\dot{\rho}_N + \frac{\partial c_{31}}{\partial \rho_U}\dot{\rho}_U &= -c_{3,3}\omega_P + c_{3,2}\omega_Y - c_{2,1}\omega_E + c_{1,1}\omega_N, \\
\frac{\partial c_{32}}{\partial \rho_E}\dot{\rho}_E + \frac{\partial c_{32}}{\partial \rho_N}\dot{\rho}_N + \frac{\partial c_{32}}{\partial \rho_U}\dot{\rho}_U &= c_{3,3}\omega_R - c_{3,1}\omega_Y - c_{2,2}\omega_E + c_{1,2}\omega_N, \\
\frac{\partial c_{33}}{\partial \rho_E}\dot{\rho}_E + \frac{\partial c_{33}}{\partial \rho_N}\dot{\rho}_N + \frac{\partial c_{33}}{\partial \rho_U}\dot{\rho}_U &= -c_{3,2}\omega_R + c_{3,1}\omega_P - c_{2,3}\omega_E + c_{1,3}\omega_N,
\end{aligned}$$

where

$$\begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \stackrel{\text{def}}{=} \mathbf{C}\left(\boldsymbol{\rho}_{\text{ENU}}\right)$$

and the partial derivatives

$$\frac{\partial c_{11}}{\partial \rho_E} = \frac{u_E\left(1-u_E^2\right)\left\{2[1-\cos(\theta)]-\theta\sin(\theta)\right\}}{\theta},$$

$$\frac{\partial c_{11}}{\partial \rho_N} = \frac{u_N\left\{-2u_E^2[1-\cos(\theta)]-\theta\sin(\theta)\left(1-u_E^2\right)\right\}}{\theta},$$

$$\frac{\partial c_{11}}{\partial \rho_U} = \frac{u_U\left\{-2u_E^2[1-\cos(\theta)]-\theta\sin(\theta)\left(1-u_E^2\right)\right\}}{\theta},$$

$$\frac{\partial c_{12}}{\partial \rho_E} = \frac{u_N\left(1-2u_E^2\right)[1-\cos(\theta)]+u_Eu_U\sin(\theta)-\theta u_Eu_U\cos(\theta)+\theta u_Nu_E^2\sin(\theta)}{\theta},$$

$$\frac{\partial c_{12}}{\partial \rho_N} = \frac{u_E\left(1-2u_N^2\right)[1-\cos(\theta)]+u_Uu_N\sin(\theta)-\theta u_Nu_U\cos(\theta)+\theta u_Eu_N^2\sin(\theta)}{\theta},$$

$$\frac{\partial c_{12}}{\partial \rho_U} = \frac{-2u_Eu_Nu_U[1-\cos(\theta)]-\left(1-u_U^2\right)\sin(\theta)-\theta u_U^2\cos(\theta)+\theta u_Uu_Nu_E\sin(\theta)}{\theta},$$

$$\frac{\partial c_{13}}{\partial \rho_E} = \frac{u_U\left(1-2u_E^2\right)[1-\cos(\theta)]-u_Eu_N\sin(\theta)+\theta u_Eu_N\cos(\theta)+\theta u_Uu_E^2\sin(\theta)}{\theta},$$

$$\frac{\partial c_{13}}{\partial \rho_N} = \frac{-2u_Eu_Nu_U[1-\cos(\theta)]+\left(1-u_N^2\right)\sin(\theta)+\theta u_N^2\cos(\theta)+\theta u_Uu_Nu_E\sin(\theta)}{\theta},$$

$$\frac{\partial c_{13}}{\partial \rho_U} = \frac{u_E\left(1-2u_U^2\right)[1-\cos(\theta)]-u_Uu_N\sin(\theta)+\theta u_Nu_U\cos(\theta)+\theta u_Eu_U^2\sin(\theta)}{\theta},$$

$$\frac{\partial c_{21}}{\partial \rho_E} = \frac{u_N\left(1-2u_E^2\right)[1-\cos(\theta)]-u_Eu_U\sin(\theta)+\theta u_Eu_U\cos(\theta)+\theta u_Nu_E^2\sin(\theta)}{\theta},$$

$$\frac{\partial c_{21}}{\partial \rho_N} = \frac{u_E\left(1-2u_N^2\right)[1-\cos(\theta)]-u_Uu_N\sin(\theta)+\theta u_Nu_U\cos(\theta)+\theta u_Eu_N^2\sin(\theta)}{\theta},$$

$$\frac{\partial c_{21}}{\partial \rho_U} = \frac{-2u_Eu_Nu_U[1-\cos(\theta)]+\sin(\theta)\left(1-u_U^2\right)+\theta u_U^2\cos(\theta)+\theta u_Uu_Nu_E\sin(\theta)}{\theta},$$

$$\frac{\partial c_{22}}{\partial \rho_E} = \frac{u_E\left\{-2u_N^2[1-\cos(\theta)]-\theta\left(1-u_N^2\right)\sin(\theta)\right\}}{\theta},$$

$$\frac{\partial c_{22}}{\partial \rho_N} = \frac{u_N\left(1-u_N^2\right)\left\{2[1-\cos(\theta)]-\theta\sin(\theta)\right\}}{\theta},$$

$$\frac{\partial c_{22}}{\partial \rho_U} = \frac{u_U\left\{-2u_N^2[1-\cos(\theta)]-\theta\left(1-u_N^2\right)\sin(\theta)\right\}}{\theta},$$

$$\frac{\partial c_{23}}{\partial \rho_E} = \frac{-2u_Eu_Nu_U[1-\cos(\theta)]-\left(1-u_E^2\right)\sin(\theta)-\theta u_E^2\cos(\theta)+\theta u_Eu_Nu_U\sin(\theta)}{\theta},$$

$$\frac{\partial c_{23}}{\partial \rho_N} = \frac{u_U\left(1-2u_N^2\right)[1-\cos(\theta)]+u_Eu_N\sin(\theta)-\theta u_Eu_N\cos(\theta)+\theta u_N^2u_U\sin(\theta)}{\theta},$$

$$\frac{\partial c_{23}}{\partial \rho_U} = \frac{u_N\left(1-2u_U^2\right)[1-\cos(\theta)]+u_Eu_U\sin(\theta)-\theta u_Eu_U\cos(\theta)+\theta u_U^2u_N\sin(\theta)}{\theta},$$

$$\frac{\partial c_{31}}{\partial \rho_E} = \frac{u_U\left(1-2u_E^2\right)[1-\cos(\theta)] + u_E u_N \sin(\theta) - \theta u_E u_N \cos(\theta) + \theta u_U u_E^2 \sin(\theta)}{\theta},$$

$$\frac{\partial c_{31}}{\partial \rho_N} = \frac{-2u_E u_N u_U[1-\cos(\theta)] - \left(1-u_N^2\right)\sin(\theta) - \theta u_N^2 \cos(\theta) + \theta u_U u_N u_E \sin(\theta)}{\theta},$$

$$\frac{\partial c_{31}}{\partial \rho_U} = \frac{u_E\left(1-2u_U^2\right)[1-\cos(\theta)] + u_U u_N \sin(\theta) - \theta u_N u_U \cos(\theta) + \theta u_E u_U^2 \sin(\theta)}{\theta},$$

$$\frac{\partial c_{32}}{\partial \rho_E} = \frac{-2u_E u_N u_U[1-\cos(\theta)] + \left(1-u_E^2\right)\sin(\theta) + \theta u_E^2 \cos(\theta) + \theta u_U u_N u_E \sin(\theta)}{\theta},$$

$$\frac{\partial c_{32}}{\partial \rho_N} = \frac{u_U\left(1-2u_N^2\right)[1-\cos(\theta)] - u_E u_N \sin(\theta) + \theta u_E u_N \cos(\theta) + \theta u_N^2 u_U \sin(\theta)}{\theta},$$

$$\frac{\partial c_{32}}{\partial \rho_U} = \frac{u_N\left(1-2u_U^2\right)[1-\cos(\theta)] - u_E u_U \sin(\theta) + \theta u_E u_U \cos(\theta) + \theta u_U^2 u_N \sin(\theta)}{\theta},$$

$$\frac{\partial c_{33}}{\partial \rho_E} = \frac{u_E\left\{-2u_U^2[1-\cos(\theta)] - \theta\sin(\theta)\left(1+u_U^2\right)\right\}}{\theta},$$

$$\frac{\partial c_{33}}{\partial \rho_N} = \frac{u_N\left\{-2u_U^2[1-\cos(\theta)] - \theta\sin(\theta)\left(1+u_U^2\right)\right\}}{\theta},$$

$$\frac{\partial c_{33}}{\partial \rho_U} = \frac{u_U\left(1-u_U^2\right)\left\{2[1-\cos(\theta)] - \theta\sin(\theta)\right\}}{\theta}$$

for

$$\theta \stackrel{\text{def}}{=} |\boldsymbol{\rho}_{\text{ENU}}|,$$

$$u_E \stackrel{\text{def}}{=} \frac{\rho_E}{\theta}, \qquad u_N \stackrel{\text{def}}{=} \frac{\rho_N}{\theta}, \qquad u_u \stackrel{\text{def}}{=} \frac{\rho_U}{\theta}.$$

These nine scalar linear equations can be put into matrix form and solved in least squares fashion as

$$\mathbf{L}\begin{bmatrix}\dot{\rho}_E\\ \dot{\rho}_N\\ \dot{\rho}_U\end{bmatrix} = \mathbf{R}\begin{bmatrix}\omega_R\\ \omega_P\\ \omega_Y\\ \omega_E\\ \omega_N\\ \omega_U\end{bmatrix}, \tag{C.144}$$

$$\begin{bmatrix}\dot{\rho}_E\\ \dot{\rho}_N\\ \dot{\rho}_U\end{bmatrix} = \underbrace{\left[\mathbf{L}^{\mathrm{T}}\mathbf{L}\right]\backslash\left[\mathbf{L}^{\mathrm{T}}\mathbf{R}\right]}_{\partial\dot{\boldsymbol{\rho}}/\partial\boldsymbol{\omega}}\begin{bmatrix}\boldsymbol{\omega}_{\text{RPY}}\\ \boldsymbol{\omega}_{\text{ENU}}\end{bmatrix}. \tag{C.145}$$

The matrix product $\mathbf{L}^T\mathbf{L}$ will always be invertible because its determinant

$$\det[\mathbf{L}^{\mathrm{T}}\mathbf{L}] = 32\frac{[1-\cos(\theta)]^2}{\theta^4}, \tag{C.146}$$

$$\lim_{\theta\to 0}\det[\mathbf{L}^{\mathrm{T}}\mathbf{L}] = 8, \tag{C.147}$$

and the resulting equation for $\boldsymbol{\rho}_{\mathrm{ENU}}$ can be put into the form

$$\dot{\boldsymbol{\rho}}_{\mathrm{ENU}} = \left[\frac{\partial\dot{\boldsymbol{\rho}}}{\partial\boldsymbol{\omega}}\right]\begin{bmatrix}\boldsymbol{\omega}_{\mathrm{RPY}}\\ \boldsymbol{\omega}_{\mathrm{ENU}}\end{bmatrix}. \tag{C.148}$$

The 3×6 matrix $\partial\boldsymbol{\rho}/\partial\boldsymbol{\omega}$ can be partitioned as

$$\left[\frac{\partial\dot{\boldsymbol{\rho}}}{\partial\boldsymbol{\omega}}\right] = \left[\frac{\partial\dot{\boldsymbol{\rho}}}{\partial\boldsymbol{\omega}_{\mathrm{RPY}}}\middle|\frac{\partial\dot{\boldsymbol{\rho}}}{\partial\boldsymbol{\omega}_{\mathrm{ENU}}}\right] \tag{C.149}$$

with 3×3 submatrices

$$\frac{\partial\boldsymbol{\rho}}{\partial\boldsymbol{\omega}_{\mathrm{RPY}}} = \left[\frac{1}{|\boldsymbol{\rho}|^2} - \frac{\sin(|\boldsymbol{\rho}|)}{2|\boldsymbol{\rho}|[1-\cos(|\boldsymbol{\rho}|)]}\right]\boldsymbol{\rho}\boldsymbol{\rho}^{\mathrm{T}} + \frac{|\boldsymbol{\rho}|\sin(|\boldsymbol{\rho}|)}{2[1-\cos(|\boldsymbol{\rho}|)]}\mathbf{I} + \frac{1}{2}[\boldsymbol{\rho}\otimes] \tag{C.150}$$

$$= \mathbf{1}_\rho\mathbf{1}_\rho^{\mathrm{T}} + \frac{\theta\sin(\theta)}{2[1-\cos(\theta)]}[\mathbf{I} - \mathbf{1}_\rho\mathbf{1}\rho^{\mathrm{T}}] + \frac{\theta}{2}[\mathbf{1}_\rho\otimes], \tag{C.151}$$

$$\lim_{\rho\to 0}\boldsymbol{\rho}\frac{P\partial\dot{\boldsymbol{\rho}}}{\partial\boldsymbol{\omega}_{\mathrm{RPY}}} = \mathbf{I}, \tag{C.152}$$

$$\frac{\partial\dot{\boldsymbol{\rho}}}{\partial\boldsymbol{\omega}_{\mathrm{ENU}}} = -\left[\frac{1}{|\boldsymbol{\rho}|^2} - \frac{\sin(|\boldsymbol{\rho}|)}{2|\boldsymbol{\rho}|[1-\cos(|\boldsymbol{\rho}|)]}\right]\boldsymbol{\rho}\boldsymbol{\rho}^{\mathrm{T}} - \frac{|\boldsymbol{\rho}|\sin(|\boldsymbol{\rho}|)}{2[1-\cos(|\boldsymbol{\rho}|)]}\mathbf{I} + \frac{1}{2}[\boldsymbol{\rho}\otimes] \tag{C.153}$$

$$= -\mathbf{1}_\rho\mathbf{1}_\rho^{\mathrm{T}} - \frac{\theta\sin(\theta)}{2[1-\cos(\theta)]}[\mathbf{I} - \mathbf{1}_\rho\mathbf{1}_\rho^{\mathrm{T}}] + \frac{\theta}{\mathbf{2}}[\mathbf{1}_\rho\otimes]. \tag{C.154}$$

$$\lim_{|\boldsymbol{\rho}|\to 0}\frac{\partial\dot{\boldsymbol{\rho}}}{\partial\boldsymbol{\omega}_{\mathrm{ENU}}} = -\mathbf{I}. \tag{C.155}$$

For locally leveled gimbaled systems, $\boldsymbol{\omega}_{\mathrm{RPH}} = \mathbf{0}$. That is, the gimbals normally keep the accelerometer axes aligned to the ENU or NED coordinate axes, a process modeled by $\boldsymbol{\omega}_{\mathrm{ENU}}$ alone.

C.4.2.5 Time Derivatives of Matrix Expressions The Kalman filter implementation for integrating GPS with a strapdown INS in Chapter 8 will require derivatives with respect to time of the matrices

$$\frac{\partial \dot{\boldsymbol{\rho}}_{\text{ENU}}}{\partial \boldsymbol{\omega}_{\text{RPH}}} \quad \text{(Eq. C.150)} \quad \text{and} \quad \frac{\partial \dot{\boldsymbol{\rho}}_{\text{ENU}}}{\partial \boldsymbol{\omega}_{\text{ENU}}} \quad \text{(Eq. C.153)}.$$

We derive here a general-purpose formula for taking such derivatives and then apply it to these two cases.

General Formulas There is a general-purpose formula for taking the time derivatives $(d/dt)\mathbf{M}(\boldsymbol{\rho})$ of matrix expressions of the sort

$$\mathbf{M}(\boldsymbol{\rho}) = \mathbf{M}\big(s_1(\boldsymbol{\rho}),\, s_2(\boldsymbol{\rho}),\, s_3(\boldsymbol{\rho})\big) \tag{C.156}$$

$$= s_1(\boldsymbol{\rho})\,\mathbf{I}_3 + s_2(\boldsymbol{\rho})\,[\boldsymbol{\rho}\otimes] + s_3(\boldsymbol{\rho})\boldsymbol{\rho}\boldsymbol{\rho}^{\mathrm{T}}, \tag{C.157}$$

that is, as linear combinations of $\mathbf{I}_3$, $\boldsymbol{\rho}\otimes$, and $\boldsymbol{\rho}\boldsymbol{\rho}^{\mathrm{T}}$ with scalar functions of $\boldsymbol{\rho}$ as the coefficients.

The derivation uses the time derivatives of the basis matrices,

$$\frac{d}{dt}\mathbf{I}_3 = \mathbf{0}_3, \tag{C.158}$$

$$\frac{d}{dt}[\boldsymbol{\rho}\otimes] = [\dot{\boldsymbol{\rho}}\otimes], \tag{C.159}$$

$$\frac{d}{dt}\boldsymbol{\rho}\boldsymbol{\rho}^{\mathrm{T}} = \dot{\boldsymbol{\rho}}\boldsymbol{\rho}^{\mathrm{T}} + \boldsymbol{\rho}\dot{\boldsymbol{\rho}}^{\mathrm{T}}, \tag{C.160}$$

where the vector

$$\dot{\boldsymbol{\rho}} = \frac{d}{dt}\boldsymbol{\rho}, \tag{C.161}$$

and then uses the chain rule for differentiation to obtain the general formula

$$\frac{d}{dt}\mathbf{M}(\boldsymbol{\rho}) = \frac{\partial s_1(\boldsymbol{\rho})}{\partial \boldsymbol{\rho}}\dot{\boldsymbol{\rho}}\mathbf{I}_3 + \frac{\partial s_2(\boldsymbol{\rho})}{\partial \boldsymbol{\rho}}\dot{\boldsymbol{\rho}}[\boldsymbol{\rho}\otimes] + s_2(\boldsymbol{\rho})[\dot{\boldsymbol{\rho}}\otimes],$$

$$+ \frac{\partial s_3(\boldsymbol{\rho})}{\partial \boldsymbol{\rho}}\dot{\boldsymbol{\rho}}\big[\boldsymbol{\rho}\boldsymbol{\rho}^{\mathrm{T}}\big] + s_3(\boldsymbol{\rho})\big[\dot{\boldsymbol{\rho}}\boldsymbol{\rho}^{\mathrm{T}} + \boldsymbol{\rho}\dot{\boldsymbol{\rho}}^{\mathrm{T}}\big], \tag{C.162}$$

where the gradients $\partial s_i(\boldsymbol{\rho})/\partial \boldsymbol{\rho}$ are to be computed as row vectors and the inner products $[\partial s_i(\boldsymbol{\rho})/\partial \boldsymbol{\rho}]\dot{\boldsymbol{\rho}}$ will be scalars.

Equation C.162 is the general-purpose formula for the matrix forms of interest, which differ only in their scalar functions $s_i(\boldsymbol{\rho})$. These scalar functions $s_i(\boldsymbol{\rho})$ are generally rational functions of the following scalar functions (shown in terms of their gradients):

$$\frac{\partial}{\partial \boldsymbol{\rho}} |\boldsymbol{\rho}|^p = p|\boldsymbol{\rho}|^{p-2}\boldsymbol{\rho}^{\mathrm{T}}, \tag{C.163}$$

$$\frac{\partial}{\partial \boldsymbol{\rho}} \sin(|\boldsymbol{\rho}|) = \cos(|\boldsymbol{\rho}|)|\boldsymbol{\rho}|^{-1}\boldsymbol{\rho}^{\mathrm{T}}, \tag{C.164}$$

$$\frac{\partial}{\partial \boldsymbol{\rho}} \cos(|\boldsymbol{\rho}|) = -\sin(|\boldsymbol{\rho}|)|\boldsymbol{\rho}|^{-1}\boldsymbol{\rho}^{\mathrm{T}} \tag{C.165}$$

Time Derivative of $\partial\dot{\boldsymbol{\rho}}_{\mathrm{ENU}}/\partial\boldsymbol{\omega}_{\mathrm{RPY}}$ In this case (Eq. C.150).

$$s_1(\boldsymbol{\rho}) = \frac{|\boldsymbol{\rho}|\sin(|\boldsymbol{\rho}|)}{2[1-\cos(|\boldsymbol{\rho}|)]}, \tag{C.166}$$

$$\frac{\partial s_1(\boldsymbol{\rho})}{\partial \boldsymbol{\rho}} = -\frac{1-|\boldsymbol{\rho}|^{-1}\sin(|\boldsymbol{\rho}|)}{2[1-\cos(|\boldsymbol{\rho}|)]}\boldsymbol{\rho}^{\mathrm{T}}, \tag{C.167}$$

$$s_2(\boldsymbol{\rho}) = \tfrac{1}{2}, \tag{C.168}$$

$$\frac{\partial s_2}{\partial \boldsymbol{\rho}} = \mathbf{0}_{1\times 3}, \tag{C.169}$$

$$s_3(\boldsymbol{\rho}) = \left[\frac{1}{|\boldsymbol{\rho}|^2} - \frac{\sin(|\boldsymbol{\rho}|)}{2|\boldsymbol{\rho}|[1-\cos(|\boldsymbol{\rho}|)]}\right], \tag{C.170}$$

$$\frac{\partial s_3(\boldsymbol{\rho})}{\partial \boldsymbol{\rho}} = \frac{1+|\boldsymbol{\rho}|^{-1}\sin(|\boldsymbol{\rho}|) - 4|\boldsymbol{\rho}|^{-2}[1-\cos(|\boldsymbol{\rho}|)]}{2|\boldsymbol{\rho}|^2[1-\cos(|\boldsymbol{\rho}|)]}\boldsymbol{\rho}^{\mathrm{T}}, \tag{C.171}$$

$$\begin{aligned}\frac{d}{dt}\frac{\partial\dot{\boldsymbol{\rho}}_{\mathrm{ENU}}}{\partial\boldsymbol{\omega}_{\mathrm{RPY}}} &= \frac{\partial s_1(\boldsymbol{\rho})}{\partial \boldsymbol{\rho}}\dot{\boldsymbol{\rho}}\mathbf{I}_3 + \frac{\partial s_2(\boldsymbol{\rho})}{\partial \boldsymbol{\rho}}\dot{\boldsymbol{\rho}}[\boldsymbol{\rho}\otimes] + s_2(\boldsymbol{\rho})[\dot{\boldsymbol{\rho}}\otimes] \\ &\quad + \frac{\partial s_3(\boldsymbol{\rho})}{\partial \boldsymbol{\rho}}\dot{\boldsymbol{\rho}}\left[\boldsymbol{\rho}\boldsymbol{\rho}^{\mathrm{T}}\right] + s_3(\boldsymbol{\rho})\left[\dot{\boldsymbol{\rho}}\boldsymbol{\rho}^{\mathrm{T}} + \boldsymbol{\rho}\dot{\boldsymbol{\rho}}^{\mathrm{T}}\right], \end{aligned} \tag{C.172}$$

$$\begin{aligned} &= -\left(\frac{1-|\boldsymbol{\rho}|^{-1}\sin(|\boldsymbol{\rho}|)}{2[1-\cos(|\boldsymbol{\rho}|)]}\right)\left(\boldsymbol{\rho}^{\mathrm{T}}\dot{\boldsymbol{\rho}}\right)\mathbf{I}_3 + \frac{1}{2}[\dot{\boldsymbol{\rho}}\otimes], \\ &\quad + \left(\frac{1+|\boldsymbol{\rho}|^{-1}\sin(|\boldsymbol{\rho}|) - 4|\boldsymbol{\rho}|^{-2}[1-\cos(|\boldsymbol{\rho}|)]}{2|\boldsymbol{\rho}|^2[1-\cos(|\boldsymbol{\rho}|)]}\right) \times \left(\boldsymbol{\rho}^{\mathrm{T}}\dot{\boldsymbol{\rho}}\right)\left[\boldsymbol{\rho}\boldsymbol{\rho}^{\mathrm{T}}\right], \\ &\quad + \left(\frac{1}{|\boldsymbol{\rho}|^2} - \frac{\sin(|\boldsymbol{\rho}|)}{2|\boldsymbol{\rho}|[1-\cos(|\boldsymbol{\rho}|)]}\right)\left[\dot{\boldsymbol{\rho}}\boldsymbol{\rho}^{\mathrm{T}} + \boldsymbol{\rho}\dot{\boldsymbol{\rho}}^{\mathrm{T}}\right]. \end{aligned} \tag{C.173}$$

Time Derivative of $\partial\dot{\boldsymbol{\rho}}_{\text{ENU}}/\partial\boldsymbol{\omega}_{\text{ENU}}$ In this case (Eq. C.153),

$$s_1(\boldsymbol{\rho}) = -\frac{|\boldsymbol{\rho}|\sin(|\boldsymbol{\rho}|)}{2[1-\cos(|\boldsymbol{\rho}|)]}, \tag{C.174}$$

$$\frac{\partial s_1(\boldsymbol{\rho})}{\partial \boldsymbol{\rho}} = \frac{1-|\boldsymbol{\rho}|^{-1}\sin(|\boldsymbol{\rho}|)}{2[1-\cos(|\boldsymbol{\rho}|)]}\boldsymbol{\rho}^{\mathrm{T}}, \tag{C.175}$$

$$s_2(\boldsymbol{\rho}) = \tfrac{1}{2}, \tag{C.176}$$

$$\frac{\partial s_2}{\partial \boldsymbol{\rho}} = \mathbf{0}_{1\times 3}, \tag{C.177}$$

$$s_3(\boldsymbol{\rho}) = -\left[\frac{1}{|\boldsymbol{\rho}|^2} - \frac{\sin(|\boldsymbol{\rho}|)}{2|\boldsymbol{\rho}|[1-\cos(|\boldsymbol{\rho}|)]}\right], \tag{C.178}$$

$$\frac{\partial s_3(\boldsymbol{\rho})}{\partial \boldsymbol{\rho}} = -\frac{1+|\boldsymbol{\rho}|^{-1}\sin(|\boldsymbol{\rho}|) - 4\,|\boldsymbol{\rho}|^{-2}[1-\cos(|\boldsymbol{\rho}|)]}{2|\boldsymbol{\rho}|^2[1-\cos(|\boldsymbol{\rho}|)]}\boldsymbol{\rho}^{\mathrm{T}}, \tag{C.179}$$

$$\begin{aligned}\frac{d}{dt}\frac{\partial\dot{\boldsymbol{\rho}}_{\text{ENU}}}{\partial\boldsymbol{\omega}_{\text{ENU}}} &= \frac{\partial s_1(\boldsymbol{\rho})}{\partial\boldsymbol{\rho}}\dot{\boldsymbol{\rho}}\mathbf{I}_3 + \frac{\partial s_2(\boldsymbol{\rho})}{\partial\boldsymbol{\rho}}\dot{\boldsymbol{\rho}}[\boldsymbol{\rho}\otimes] + s_2(\boldsymbol{\rho})[\dot{\boldsymbol{\rho}}\otimes],\\ &\quad + \frac{\partial s_3(\boldsymbol{\rho})}{\partial\boldsymbol{\rho}}\dot{\boldsymbol{\rho}}\left[\boldsymbol{\rho}\boldsymbol{\rho}^{\mathrm{T}}\right] + s_3(\boldsymbol{\rho})\left[\dot{\boldsymbol{\rho}}\boldsymbol{\rho}^{\mathrm{T}} + \boldsymbol{\rho}\dot{\boldsymbol{\rho}}^{\mathrm{T}}\right] \qquad \text{(C.180)}\\ &= \left(\frac{1-|\boldsymbol{\rho}|^{-1}\sin(|\boldsymbol{\rho}|)}{2[1-\cos(|\boldsymbol{\rho}|)]}\right)\left(\boldsymbol{\rho}^{\mathrm{T}}\dot{\boldsymbol{\rho}}\right)\mathbf{I}_3 + \frac{1}{2}(\boldsymbol{\rho})[\dot{\boldsymbol{\rho}}\otimes],\\ &\quad - \left(\frac{1+|\boldsymbol{\rho}|^{-1}\sin(|\boldsymbol{\rho}|) - 4\,|\boldsymbol{\rho}|^{-2}[1-\cos(|\boldsymbol{\rho}|)]}{2|\boldsymbol{\rho}|^2[1-\cos(|\boldsymbol{\rho}|)]}\right)\times\left(\boldsymbol{\rho}^{\mathrm{T}}\dot{\boldsymbol{\rho}}\right)\left[\boldsymbol{\rho}\boldsymbol{\rho}^{\mathrm{T}}\right],\\ &\quad - \left(\frac{1}{|\boldsymbol{\rho}|^2} - \frac{\sin(|\boldsymbol{\rho}|)}{2|\boldsymbol{\rho}|[1-\cos(|\boldsymbol{\rho}|)]}\right)\left[\dot{\boldsymbol{\rho}}\boldsymbol{\rho}^{\mathrm{T}} + \boldsymbol{\rho}\dot{\boldsymbol{\rho}}^{\mathrm{T}}\right]. \qquad \text{(C.181)}\end{aligned}$$

C.4.2.6 Partial Derivatives with Respect to Rotation Vectors Calculation of the dynamic coefficient matrices $\mathbf{F}$ and measurement sensitivity matrices $\mathbf{H}$ in linearized or extended Kalman filtering with rotation vectors $\boldsymbol{\rho}_{\text{ENU}}$ as part of the system model state vector requires taking derivatives with respect to $\boldsymbol{\rho}_{\text{ENU}}$ of associated vector-valued $\mathbf{f}$- or $\mathbf{h}$-functions, as

$$\mathbf{F} = \frac{\partial\mathbf{f}\left(\boldsymbol{\rho}_{\text{ENU}}, \mathbf{v}\right)}{\partial\boldsymbol{\rho}_{\text{ENU}}}, \tag{C.182}$$

$$\mathbf{H} = \frac{\partial\mathbf{h}\left(\boldsymbol{\rho}_{\text{ENU}}, \mathbf{v}\right)}{\partial\boldsymbol{\rho}_{\text{ENU}}}, \tag{C.183}$$

where the vector-valued functions will have the general form

$$\begin{aligned}&\mathbf{f}\left(\boldsymbol{\rho}_{\text{ENU}}, \mathbf{v}\right) \text{ or } \mathbf{h}\left(\boldsymbol{\rho}_{\text{ENU}}, \mathbf{v}\right)\\ &\quad = \left\{s_0\left(\boldsymbol{\rho}_{\text{ENU}}\right)\mathbf{I}_3 + s_1\left(\boldsymbol{\rho}_{\text{ENU}}\right)\left[\boldsymbol{\rho}_{\text{ENU}}\otimes\right] + s_2\left(\boldsymbol{\rho}_{\text{ENU}}\right)\boldsymbol{\rho}_{\text{ENU}}\boldsymbol{\rho}_{\text{ENU}}^{\mathrm{T}}\right\}\mathbf{v},\end{aligned} \tag{C.184}$$

and s_0, s_1, s_2 are scalar-valued functions of $\boldsymbol{\rho}_{\text{ENU}}$ and $\mathbf{v}$ is a vector that does not depend on $\boldsymbol{\rho}_{\text{ENU}}$. We will derive here the general formulas that can be used for taking the partial derivatives $\partial\mathbf{f}(\boldsymbol{\rho}_{\text{ENU}}, \mathbf{v})/\partial\boldsymbol{\rho}_{\text{ENU}}$ or $\partial\mathbf{h}(\boldsymbol{\rho}_{\text{ENU}}, \mathbf{v})/\partial\boldsymbol{\rho}_{\text{ENU}}$. These formulas can all be derived by calculating the derivatives of the different factors in the functional forms and then using the chain rule for differentiation to obtain the final result.

Derivatives of Scalars The derivatives of the scalar factors s_0, s_1, s_2 are

$$\frac{\partial}{\partial\boldsymbol{\rho}_{\text{ENU}}}s_i(\boldsymbol{\rho}_{\text{ENU}}) = \left[\frac{\partial s_i(\boldsymbol{\rho}_{\text{ENU}})}{\partial\rho_E}\frac{\partial s_i(\boldsymbol{\rho}_{\text{ENU}})}{\partial\rho_N}\frac{\partial s_i(\boldsymbol{\rho}_{\text{ENU}})}{\partial\rho_U}\right], \tag{C.185}$$

a row vector. Consequently, for any vector-valued function $\mathbf{g}(\boldsymbol{\rho}_{\text{ENU}})$ by the chain rule, the derivatives of the vector-valued product $s_i(\boldsymbol{\rho}_{\text{ENU}})\,\mathbf{g}(\boldsymbol{\rho}_{\text{ENU}})$ are

$$\frac{\partial\{s_i(\boldsymbol{\rho}_{\text{ENU}})\mathbf{g}(\boldsymbol{\rho}_{\text{ENU}})\}}{\partial\boldsymbol{\rho}_{\text{ENU}}} = \underbrace{\mathbf{g}(\boldsymbol{\rho}_{\text{ENU}})\frac{\partial s_i(\boldsymbol{\rho}_{\text{ENU}})}{\partial\boldsymbol{\rho}_{\text{ENU}}}}_{3\times3\text{ matrix}} + s_i(\boldsymbol{\rho}_{\text{ENU}})\underbrace{\frac{\partial\mathbf{g}(\boldsymbol{\rho}_{\text{ENU}})}{\partial\boldsymbol{\rho}_{\text{ENU}}}}_{3\times3\text{ matrix}}, \tag{C.186}$$

the result of which will be the 3×3 Jacobian matrix of that subexpression in $\mathbf{f}$ or $\mathbf{h}$.

Derivatives of Vectors The three potential forms of the vector-valued function $\mathbf{g}$ in Eq. C.186 are

$$\mathbf{g}(\boldsymbol{\rho}_{\text{ENU}}) = \begin{cases} \mathbf{I}\mathbf{v} = \mathbf{v}, \\ \boldsymbol{\rho}_{\text{ENU}} \otimes \mathbf{v}, \\ \boldsymbol{\rho}_{\text{ENU}}\boldsymbol{\rho}_{\text{ENU}}^T\mathbf{v}, \end{cases} \tag{C.187}$$

each of which is considered independently:

$$\frac{\partial\mathbf{v}}{\partial\boldsymbol{\rho}_{\text{ENU}}} = 0_{3\times3}, \tag{C.188}$$

$$\frac{\partial\boldsymbol{\rho}_{\text{ENU}} \otimes \mathbf{v}}{\partial\boldsymbol{\rho}_{\text{ENU}}} = \frac{\partial[-\mathbf{v} \otimes \boldsymbol{\rho}_{\text{ENU}}]}{\partial\boldsymbol{\rho}_{\text{ENU}}}, \tag{C.189}$$

$$= -[\mathbf{v}\otimes], \tag{C.190}$$

$$= -\begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}, \tag{C.191}$$

$$\frac{\partial\boldsymbol{\rho}_{\text{ENU}}\boldsymbol{\rho}_{\text{ENU}}^{\text{T}}\mathbf{v}}{\partial\boldsymbol{\rho}_{\text{ENU}}} = \left(\boldsymbol{\rho}_{\text{ENU}}^{\text{T}}\mathbf{v}\right)\frac{\partial\boldsymbol{\rho}_{\text{ENU}}}{\partial\boldsymbol{\rho}_{\text{ENU}}} + \boldsymbol{\rho}_{\text{ENU}}\frac{\partial\boldsymbol{\rho}_{\text{ENU}}^{\text{T}}\mathbf{v}}{\partial\boldsymbol{\rho}_{\text{ENU}}}, \tag{C.192}$$

$$= \left(\boldsymbol{\rho}_{\text{ENU}}^{T}\mathbf{v}\right)\mathbf{I}_{3\times3} + \boldsymbol{\rho}_{\text{ENU}}\mathbf{v}^{\text{T}}. \tag{C.193}$$

General Formula Combining the above formulas for the different parts, one can obtain the following general-purpose formula:

$$\begin{aligned}
&\frac{\partial}{\partial \boldsymbol{\rho}_{\mathrm{ENU}}}\left\{s_0(\boldsymbol{\rho}_{\mathrm{ENU}})\mathbf{I}_3 + s_1(\boldsymbol{\rho}_{\mathrm{ENU}})[\boldsymbol{\rho}_{\mathrm{ENU}}\otimes] + s_2(\boldsymbol{\rho}_{\mathrm{ENU}})\boldsymbol{\rho}_{\mathrm{ENU}}\boldsymbol{\rho}_{\mathrm{ENU}}^{T}\right\}\mathbf{v} \\
&\quad = \mathbf{v}\left[\frac{\partial s_0(\boldsymbol{\rho}_{\mathrm{ENU}})}{\partial \rho_E}\ \frac{\partial s_0(\boldsymbol{\rho}_{\mathrm{ENU}})}{\partial \rho_N}\ \frac{\partial s_0(\boldsymbol{\rho}_{\mathrm{ENU}})}{\partial \rho_U}\right] \\
&\qquad + [\boldsymbol{\rho}_{\mathrm{ENU}}\otimes\mathbf{v}]\left[\frac{\partial s_1(\boldsymbol{\rho}_{\mathrm{ENU}})}{\partial \rho_E}\ \frac{\partial s_1(\boldsymbol{\rho}_{\mathrm{ENU}})}{\partial \rho_N}\ \frac{\partial s_1(\boldsymbol{\rho}_{\mathrm{ENU}})}{\partial \rho_U}\right] \\
&\qquad - s_1(\boldsymbol{\rho}_{\mathrm{ENU}})[\mathbf{v}\otimes] \\
&\qquad + (\boldsymbol{\rho}_{\mathrm{ENU}}^{\mathrm{T}}\mathbf{v})\boldsymbol{\rho}_{\mathrm{ENU}}\left[\frac{\partial s_2(\boldsymbol{\rho}_{\mathrm{ENU}})}{\partial \rho_E}\ \frac{\partial s_2(\boldsymbol{\rho}_{\mathrm{ENU}})}{\partial \rho_N}\ \frac{\partial s_2(\boldsymbol{\rho}_{\mathrm{ENU}})}{\partial \rho_U}\right] \\
&\qquad + s_2(\boldsymbol{\rho}_{\mathrm{ENU}})\left[(\boldsymbol{\rho}_{\mathrm{ENU}}^{\mathrm{T}}\mathbf{v})\mathbf{I}_{3\times 3} + \boldsymbol{\rho}_{\mathrm{ENU}}\mathbf{v}^{\mathrm{T}}\right],
\end{aligned} \tag{C.194}$$

applicable for any differentiable scalar functions s_0, s_1, s_2.

C.4.3 Direction Cosines Matrix

We have demonstrated in Eq.C.12 that the coordinate transformation matrix between one orthogonal coordinate system and another is a matrix of direction cosines between the unit axis vectors of the two coordinate systems,

$$\mathbf{C}_{XYZ}^{UVW} = \begin{bmatrix} \cos(\theta_{XU}) & \cos(\theta_{XV}) & \cos(\theta_{XW}) \\ \cos(\theta_{YU}) & \cos(\theta_{YV}) & \cos(\theta_{YW}) \\ \cos(\theta_{ZU}) & \cos(\theta_{ZV}) & \cos(\theta_{ZW}) \end{bmatrix}. \tag{C.195}$$

Because the angles do not depend on the order of the direction vectors (i.e., $\theta_{ab} = \theta_{ba}$), the inverse transformation matrix

$$\mathbf{C}_{UVW}^{XYZ} = \begin{bmatrix} \cos(\theta_{UX}) & \cos(\theta_{UY}) & \cos(\theta_{UZ}) \\ \cos(\theta_{VX}) & \cos(\theta_{VY}) & \cos(\theta_{VZ}) \\ \cos(\theta_{WX}) & \cos(\theta_{WY}) & \cos(\theta_{WX}) \end{bmatrix}, \tag{C.196}$$

$$= \begin{bmatrix} \cos(\theta_{XU}) & \cos(\theta_{XV}) & \cos(\theta_{XW}) \\ \cos(\theta_{YU}) & \cos(\theta_{YV}) & \cos(\theta_{YW}) \\ \cos(\theta_{ZU}) & \cos(\theta_{ZV}) & \cos(\theta_{ZW}) \end{bmatrix}^{\mathrm{T}}, \tag{C.197}$$

$$= \left(\mathbf{C}_{XYZ}^{UVW}\right)^{\mathrm{T}}. \tag{C.198}$$

That is, the inverse coordinate transformation matrix is the transpose of the forward coordinate transformation matrix. This implies that the coordinate transformation matrices are orthogonal matrices.

C.4.3.1 Rotating Coordinates Let "rot" denote a set of rotating coordinates, with axes X_{rot}, Y_{rot}, Z_{rot}, and let "non" represent a set of non-rotating (i.e., inertial) coordinates, with axes X_{non}, Y_{non}, Z_{non}, as illustrated in Fig. C.14.

Any vector $\mathbf{v}_{rot}$ in rotating coordinates can be represented in terms of its nonrotating components and unit vectors parallel to the nonrotating axes, as

$$\mathbf{v}_{rot} = v_{x,non}\mathbf{1}_{x,non} + v_{y,non}\mathbf{1}_{y,non} + v_{z,non}\mathbf{1}_{z,non} \tag{C.199}$$

$$= [\mathbf{1}_{x,non} \quad \mathbf{1}_{y,non} \quad \mathbf{1}_{z,non}] \begin{bmatrix} v_{x,non} \\ v_{y,non} \\ v_{z,non} \end{bmatrix} \tag{C.200}$$

$$= \mathbf{C}_{rot}^{non}\mathbf{v}_{non}, \tag{C.201}$$

where $v_{x,non}, v_{y,non}, v_{z,non}$ are nonrotating components of the vector, $\mathbf{1}_{x,non}, \mathbf{1}_{y,non}, \mathbf{1}_{z,non}$ = unit vectors along $X_{non}, Y_{non}, Z_{non}$ axes, as expressed in rotating coordinates
$\mathbf{v}_{rot}$ = vector $\mathbf{v}$ expressed in RPY coordinates
$\mathbf{v}_{non}$ = vector $\mathbf{v}$ expressed in ECI coordinates,
$\mathbf{C}_{rot}^{non}$ = coordinate transformation matrix from nonrotating coordinates to rotating coordinates

and

$$\mathbf{C}_{rot}^{non} = [\mathbf{1}_{x,non} \quad \mathbf{1}_{y,non} \quad \mathbf{1}_{z,non}]. \tag{C.202}$$

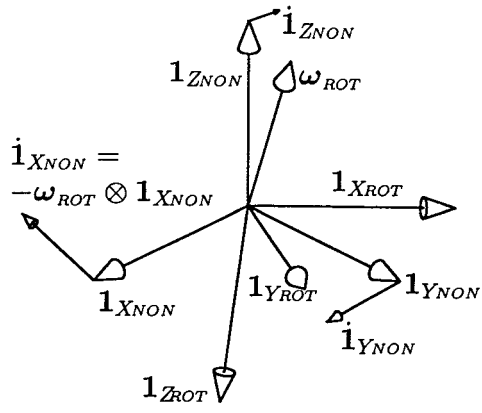


Fig. C.14 *Rotating coordinates.*

The time derivative of $\mathbf{C}_{\text{rot}}^{\text{non}}$, as viewed from the non-rotating coordinate frame, can be derived in terms of the dynamics of the unit vectors $\mathbf{1}_{x,\text{non}}$, $\mathbf{1}_{y,\text{non}}$ and $\mathbf{1}_{z,\text{non}}$ in rotating coordinates.

As seen by an observer fixed with respect to the nonrotating coordinates, the nonrotating coordinate directions will appear to remain fixed, but the external inertial reference directions will appear to be changing, as illustrated in Fig. C.14. Gyroscopes fixed in the rotating coordinates would measure three components of the inertial rotation rate vector

$$\boldsymbol{\omega}_{\text{rot}} = \begin{bmatrix} \omega_{x,\text{rot}} \\ \omega_{y,\text{rot}} \\ \omega_{z,\text{rot}} \end{bmatrix} \tag{C.203}$$

in rotating coordinates, but the non-rotating unit vectors, as viewed in rotating coordinates, appear to be changing in the opposite sense, as

$$\frac{d}{dt}\mathbf{1}_{x,\text{non}} = -\boldsymbol{\omega}_{\text{rot}} \otimes \mathbf{1}_{x,\text{non}}, \tag{C.204}$$

$$\frac{d}{dt}\mathbf{1}_{y,\text{non}} = -\boldsymbol{\omega}_{\text{rot}} \otimes \mathbf{1}_{y,\text{non}}, \tag{C.205}$$

$$\frac{d}{dt}\mathbf{1}_{z,\text{non}} = -\boldsymbol{\omega}_{\text{rot}} \otimes \mathbf{1}_{z,\text{non}}, \tag{C.205}$$

as illustrated in Fig. C.14. The time-derivative of the coordinate transformation represented in Eq. C.202 will then be

$$\begin{aligned} \frac{d}{dt}\mathbf{C}_{\text{rot}}^{\text{non}} &= \left[\frac{d}{dt}\mathbf{1}_{x,\text{non}} \quad \frac{d}{dt}\mathbf{1}_{y,\text{non}} \quad \frac{d}{dt}\mathbf{1}_{z,\text{non}}\right] && \text{(C.207)} \\ &= [-\boldsymbol{\omega}_{\text{rot}} \otimes \mathbf{1}_{x,\text{non}} \quad -\boldsymbol{\omega}_{\text{rot}} \otimes \mathbf{1}_{y,\text{non}} \quad -\boldsymbol{\omega}_{\text{rot}} \otimes \mathbf{1}_{z,\text{non}}] \\ &= -[\boldsymbol{\omega}_{\text{rot}}\otimes][\mathbf{1}_{x,\text{non}}] \quad \mathbf{1}_{y,\text{non}} \quad \mathbf{1}_{z,\text{non}}] \\ &= -[\boldsymbol{\omega}_{\text{rot}}\otimes]\mathbf{C}_{\text{rot}}^{\text{non}}, && \text{(C.208)} \end{aligned}$$

$$[\boldsymbol{\omega}_{\text{rot}}\otimes] \stackrel{\text{def}}{=} \begin{bmatrix} 0 & -\omega_{z,\text{rot}} & \omega_{y,\text{rot}} \\ \omega_{z,\text{rot}} & 0 & -\omega_{x,\text{rot}} \\ -\omega_{y,\text{rot}} & \omega_{x,\text{rot}} & 0 \end{bmatrix}. \tag{C.209}$$

The inverse coordinate transformation

$$\begin{aligned} \mathbf{C}_{\text{non}}^{\text{rot}} &= \left(\mathbf{C}_{\text{rot}}^{\text{non}}\right)^{-1} && \text{(C.210)} \\ &= \left(\mathbf{C}_{\text{rot}}^{\text{non}}\right)^{\text{T}}, && \text{(C.211)} \end{aligned}$$

the transpose of $\mathbf{C}_{\text{rot}}^{\text{non}}$, and its derivative

$$\frac{d}{dt}\mathbf{C}_{\text{non}}^{\text{rot}} = \frac{d}{dt}\left(\mathbf{C}_{\text{rot}}^{\text{non}}\right)^{\text{T}} \tag{C.212}$$

$$= \left(\frac{d}{dt}\mathbf{C}_{\text{rot}}^{\text{non}}\right)^{\text{T}} \tag{C.213}$$

$$= \left(-\left[\boldsymbol{\omega}_{\text{rot}}\otimes\right]\mathbf{C}_{\text{rot}}^{\text{non}}\right)^{\text{T}} \tag{C.214}$$

$$= -\left(\mathbf{C}_{\text{rot}}^{\text{non}}\right)^{\text{T}}\left[\boldsymbol{\omega}_{\text{rot}}\otimes\right]^{\text{T}}, \tag{C.215}$$

$$= \mathbf{C}_{\text{non}}^{\text{rot}}\left[\boldsymbol{\omega}_{\text{rot}}\otimes\right]. \tag{C.216}$$

In the case that "rot" is "RPY" (roll-pitch-yaw coordinates) and "non" is "ECI" (earth centered inertial coordinates), Eq. C.216 becomes

$$\frac{d}{dt}\mathbf{C}_{\text{ECI}}^{\text{RPY}} = \mathbf{C}_{\text{ECI}}^{\text{RPY}}\left[\boldsymbol{\omega}_{\text{RPY}}\otimes\right], \tag{C.217}$$

and in the case that "rot" is "ENU" (east-north-up coordinates) and "non" is "ECI" (earth centered inertial coordinates), Eq. C.208 becomes

$$\frac{d}{dt}\mathbf{C}_{\text{ENU}}^{\text{ECI}} = -\left[\boldsymbol{\omega}_{\text{ENU}}\otimes\right]\mathbf{C}_{\text{ENU}}^{\text{ECI}}, \tag{C.218}$$

and the derivative of their product

$$\mathbf{C}_{\text{ENU}}^{\text{RPY}} = \mathbf{C}_{\text{ENU}}^{\text{ECI}}\mathbf{C}_{\text{ECI}}^{\text{RPY}}, \tag{C.219}$$

$$\frac{d}{dt}\mathbf{C}_{\text{ENU}}^{\text{RPY}} = \left[\frac{d}{dt}\mathbf{C}_{\text{ENU}}^{\text{ECI}}\right]\mathbf{C}_{\text{ECI}}^{\text{RPY}} + \mathbf{C}_{\text{ENU}}^{\text{ECI}}\left[\frac{d}{dt}\mathbf{C}_{\text{ECI}}^{\text{RPY}}\right] \tag{C.220}$$

$$= \left[\left[-\boldsymbol{\omega}_{\text{ENU}}\otimes\right]\mathbf{C}_{\text{ENU}}^{\text{ECI}}\right]\mathbf{C}_{\text{ECI}}^{\text{RPY}} + \mathbf{C}_{\text{ENU}}^{\text{ECI}}\left[\mathbf{C}_{\text{ECI}}^{\text{RPY}}\left[\boldsymbol{\omega}_{\text{RPY}}\otimes\right]\right]$$

$$= \left[-\boldsymbol{\omega}_{\text{ENU}}\otimes\right]\underbrace{\mathbf{C}_{\text{ENU}}^{\text{ECI}}\mathbf{C}_{\text{ECI}}^{\text{RPY}}}_{\mathbf{C}_{\text{ENU}}^{\text{RPY}}} + \underbrace{\mathbf{C}_{\text{ENU}}^{\text{ECI}}\mathbf{C}_{\text{ECI}}^{\text{RPY}}}_{\mathbf{C}_{\text{ENU}}^{\text{RPY}}}\left[\boldsymbol{\omega}_{\text{RPY}}\otimes\right],$$

$$\frac{d}{dt}\mathbf{C}_{\text{ENU}}^{\text{RPY}} = -\left[\boldsymbol{\omega}_{\text{ENU}}\otimes\right]\mathbf{C}_{\text{ENU}}^{\text{RPY}} + \mathbf{C}_{\text{ENU}}^{\text{RPY}}\left[\boldsymbol{\omega}_{\text{RPY}}\otimes\right]. \tag{C.221}$$

Equation C.221 was originally used for maintaining vehicle attitude information in strapdown INS implementations, where the variables

$$\begin{aligned}\boldsymbol{\omega}_{\text{RPY}} &= \text{vector of inertial rates measured by the gyroscopes,}\\ \boldsymbol{\omega}_{\text{ENU}} &= \boldsymbol{\omega}_{\text{earthrate}} + \boldsymbol{\omega}_{v_E} + \boldsymbol{\omega}_{v_N},\end{aligned} \tag{C.222}$$

$$\boldsymbol{\omega}_{\oplus} = \omega_{\oplus}\begin{bmatrix} 0 \\ \cos(\phi_{\text{geodetic}}) \\ \sin(\phi_{\text{geodetic}}) \end{bmatrix}, \tag{C.223}$$

$$\boldsymbol{\omega}_{v_E} = \frac{v_E}{r_T + h}\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \tag{C.224}$$

$$\boldsymbol{\omega}_{v_N} = \frac{v_N}{r_M + h}\begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}, \tag{C.225}$$

and

where $\omega_{\oplus}$ = earth rotation rate
ϕ_{geodetic} = geodetic latitude
v_E = the east component of velocity with respect to the surface of the earth
r_T = transverse radius of curvature of the ellipsoid (Eq. 6.41)
v_N = north component of velocity with respect to the surface of the earth
r_M = meridional radius of curvature of the ellispoid (Eq. 6.38)
h = altitude above (+) or below (−) the reference ellipsoid surface (≈ mean sea level)

Unfortunately, Eq. C.221 was found to be not particularly well suited for accurate integration in finite-precision arithmetic. This integration problem was eventually solved using quaternions.

C.4.4 Quaternions

The term *quaternions* is used in several contexts to refer to sets of four. In mathematics, it refers to an algebra in four dimensions discovered by the Irish physicist and mathematician Sir William Rowan Hamilton (1805–1865). The utility of quaternions for representing rotations (as points on a sphere in four dimensions) was known before strapdown systems, they soon became the standard representation of coordinate transforms in strapdown systems, and they have since been applied to computer animation.

C.4.4.1 Quaternion Matrices For people already familiar with matrix algebra, the algebra of quaternions can be defined by using an isomorphism between 4×1 *quaternion vectors* $\mathbf{q}$ and real 4×4 *quaternion matrices* $\mathbf{Q}$:

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \leftrightarrow \mathbf{Q} = \begin{bmatrix} q_1 & -q_2 & -q_3 & -q_4, \\ q_2 & q_1 & -q_4 & q_3, \\ q_3 & q_4 & q_1 & -q_2, \\ q_4 & -q_3 & q_2 & q_1 \end{bmatrix} \tag{C.226}$$

$$= q_1\mathcal{Q}_1 + q_2\mathcal{Q}_2 + q_3\mathcal{Q}_3 + q_4\mathcal{Q}_4, \tag{C.227}$$

$$\mathcal{Q}_1 \stackrel{\text{def}}{=} \begin{bmatrix} 1 & 0 & 0 & 0, \\ 0 & 1 & 0 & 0, \\ 0 & 0 & 1 & 0, \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{C.228}$$

$$\mathcal{Q}_2 \stackrel{\text{def}}{=} \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{C.229}$$

$$\mathcal{Q}_3 \stackrel{\text{def}}{=} \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}, \tag{C.230}$$

$$\mathcal{Q}_4 \stackrel{\text{def}}{=} \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \tag{C.231}$$

in terms of four 4×4 *quaternion basis matrices*, $\mathcal{Q}_1,\ \mathcal{Q}_2,\ \mathcal{Q}_3,\ \mathcal{Q}_4$, the first of which is an identity matrix and the rest of which are antisymmetric.

C.4.4.2 Addition and Multiplication Addition of quaternion vectors is the same as that for ordinary vectors. Multiplication is defined by the usual rules for matrix multiplication applied to the four quaternion basis matrices, the multiplication table for which is given in Table C.1. Note that, like matrix multiplication, *quaternion multiplication is noncommutative*. That is, the result depends on the order of multiplication.

Using the quaternion basis matrix multiplication Table (C.1), the ordered product **AB** of two quaternion matrices

$$\mathbf{A} = a_1\mathcal{Q}_1 + a_2\mathcal{Q}_2 + a_3\mathcal{Q}_3 + a_4\mathcal{Q}_4, \tag{C.232}$$

$$\mathbf{B} = b_1\mathcal{Q}_1 + b_2\mathcal{Q}_2 + b_3\mathcal{Q}_3 + b_4\mathcal{Q}_4 \tag{C.233}$$

TABLE C.1 Multiplication of Quaternion Basis Matrices

First Factor	Second Factor			
	$\mathcal{Q}_1$	$\mathcal{Q}_2$	$\mathcal{Q}_3$	$\mathcal{Q}_4$
$\mathcal{Q}_1$	$\mathcal{Q}_1$	$\mathcal{Q}_2$	$\mathcal{Q}_3$	$\mathcal{Q}_4$
$\mathcal{Q}_2$	$\mathcal{Q}_2$	$-\mathcal{Q}_1$	$\mathcal{Q}_4$	$-\mathcal{Q}_3$
$\mathcal{Q}_3$	$\mathcal{Q}_3$	$-\mathcal{Q}_4$	$-\mathcal{Q}_1$	$\mathcal{Q}_2$
$\mathcal{Q}_4$	$\mathcal{Q}_4$	$\mathcal{Q}_3$	$-\mathcal{Q}_2$	$-\mathcal{Q}_1$

can be shown to be

$$\begin{aligned}\mathbf{AB} = &\left(a_1b_1 - a_2b_2 - a_3b_3 - a_4b_4\right)\mathcal{Q}_1 \\ &+ \left(a_2b_1 + a_1b_2 - a_4b_3 + a_3b_4\right)\mathcal{Q}_2 \\ &+ \left(a_3b_1 + a_4b_2 + a_1b_3 - a_2b_4\right)\mathcal{Q}_3 \\ &+ \left(a_4b_1 - a_3b_2 + a_2b_3 + a_1b_4\right)\mathcal{Q}_4\end{aligned} \tag{C.234}$$

in terms of the coefficients a_k, b_k and the quaternion basis matrices.

C.4.4.3 Conjugation Conjugation of quaternions is a unary operation analogous to conjugation of complex numbers, in that the real part (the first component of a quaternion) is unchanged and the other parts change sign. For quaternions, this is equivalent to transposition of the associated quaternion matrix

$$\mathbf{Q} = q_1\mathcal{Q}_1 + q_2\mathcal{Q}_2 + q_3\mathcal{Q}_3 + q_4\mathcal{Q}_4, \tag{C.235}$$

so that

$$\mathbf{Q}^{\mathrm{T}} = q_1\mathcal{Q}_1 - q_2\mathcal{Q}_2 - q_3\mathcal{Q}_3 - q_4\mathcal{Q}_4 \tag{C.236}$$

$$\leftrightarrow \mathbf{q}^*, \tag{C.236}$$

$$\mathbf{Q}^{\mathrm{T}}\mathbf{Q} = \left(q_1^2 + q_2^2 + q_3^2 + q_4^2\right)\mathcal{Q}_1 \tag{C.238}$$

$$\leftrightarrow \mathbf{q}^*\mathbf{q} = |\mathbf{q}|^2. \tag{C.239}$$

C.4.4.4 Representing Rotations The problem with rotation vectors as representations for rotations is that the rotation vector representing successive rotations $\boldsymbol{\rho}_1$, $\boldsymbol{\rho}_2$, $\boldsymbol{\rho}_3, \ldots,$ $\boldsymbol{\rho}_n$ is not a simple function of the respective rotation vectors.

This representation problem is solved rather elegantly using quaternions, such that the quaternion representation of the successive rotations is represented by the quaternion product $\mathbf{q}_n \times \mathbf{q}_{n-1} \times \cdots \times \mathbf{q}_3 \times \mathbf{q}_2 \times \mathbf{q}_1$. That is, each successive rotation can be implemented by a single quaternion product.

The quaternion equivalent of the rotation vector $\boldsymbol{\rho}$ with $|\boldsymbol{\rho}| = \theta$,

$$\boldsymbol{\rho} \stackrel{\text{def}}{=} \begin{bmatrix}\rho_1 \\ \rho_2 \\ \rho_3\end{bmatrix} \stackrel{\text{def}}{=} \theta \begin{bmatrix}u_1 \\ u_2 \\ u_3\end{bmatrix} \tag{C.240}$$

(i.e., where **u** is a unit vector), is

$$\mathbf{q}(\boldsymbol{\rho}) \stackrel{\text{def}}{=} \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ \frac{\rho_1 \sin(\theta/2)}{\theta} \\ \frac{\rho_2 \sin(\theta/2)}{\theta} \\ \frac{\rho_3 \sin(\theta/2)}{\theta} \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ u_1 \sin\left(\frac{\theta}{2}\right) \\ u_2 \sin\left(\frac{\theta}{2}\right) \\ u_3 \sin\left(\frac{\theta}{2}\right) \end{bmatrix}, \tag{C.241}$$

and the vector **w** resulting from the rotation of any three-dimensional vector

$$\mathbf{v} \stackrel{\text{def}}{=} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

through the angle θ about the unit vector **u** is implemented by the quaternion product

$$\mathbf{q}(\mathbf{w}) \stackrel{\text{def}}{=} \mathbf{q}(\boldsymbol{\rho})\mathbf{q}(\mathbf{v})\mathbf{q}^*(\boldsymbol{\rho}) \tag{C.242}$$

$$\stackrel{\text{def}}{=} \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ u_1 \sin\left(\frac{\theta}{2}\right) \\ u_2 \sin\left(\frac{\theta}{2}\right) \\ u_3 \sin\left(\frac{\theta}{2}\right) \end{bmatrix} \times \begin{bmatrix} 0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} \times \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ -u_1 \sin\left(\frac{\theta}{2}\right) \\ -u_2 \sin\left(\frac{\theta}{2}\right) \\ -u_3 \sin\left(\frac{\theta}{2}\right) \end{bmatrix} \tag{C.243}$$

$$= \begin{bmatrix} 0 \\ w_1 \\ w_2 \\ w_3 \end{bmatrix}, \tag{C.244}$$

$$w_1 = \cos(\theta)v_1 + [1-\cos(\theta)][u_1(u_1v_1 + u_2v_2 + u_3v_3)] + \sin(\theta)(u_2v_3 - u_3v_2), \tag{C.245}$$

$$w_2 = \cos(\theta)v_2 + [1-\cos(\theta)][u_2(u_1v_1 + u_2v_2 + u_3v_3)] + \sin(\theta)(u_3v_1 - u_1v_3), \tag{C.246}$$

$$w_3 = \cos(\theta)v_3 + [1-\cos(\theta)][u_3(u_1v_1 + u_2v_2 + u_3v_3)] + \sin(\theta)(u_1v_2 - u_2v_1), \tag{C.247}$$

or

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \mathbf{C}(\boldsymbol{\rho}) \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}, \tag{C.248}$$

where the rotation matrix $\mathbf{C}(\boldsymbol{\rho})$ is defined in Eq. C.113 and Eq. C.242 implements the same rotation of $\mathbf{v}$ as the matrix product $\mathbf{C}(\boldsymbol{\rho})\mathbf{v}$. Moreover, if

$$\mathbf{q}(\mathbf{w}_k) \stackrel{\text{def}}{=} \mathbf{v} \tag{C.249}$$

and

$$\mathbf{q}(\mathbf{w}_k) \stackrel{\text{def}}{=} \mathbf{q}(\boldsymbol{\rho}_k)\mathbf{q}(\mathbf{w}_{k-1})\mathbf{q}^*(\boldsymbol{\rho}_k) \tag{C.250}$$

for $k = 1, 2, 3, \ldots, n$, then the nested quaternion product

$$\mathbf{q}(\mathbf{w}_n) = \mathbf{q}(\boldsymbol{\rho}_n) \cdots \mathbf{q}(\boldsymbol{\rho}_2)\mathbf{q}(\boldsymbol{\rho}_1)\mathbf{q}(\mathbf{v})\mathbf{q}^*(\boldsymbol{\rho}_1)\mathbf{q}^*(\boldsymbol{\rho}_2) \cdots \mathbf{q}^*(\boldsymbol{\rho}_n) \tag{C.251}$$

implements the succession of rotations represented by the rotation vectors $\boldsymbol{\rho}_1$, $\boldsymbol{\rho}_2$, $\boldsymbol{\rho}_3$, $\ldots$, $\boldsymbol{\rho}_n$, and the single quaternion

$$\mathbf{q}_{[n]} \stackrel{\text{def}}{=} \mathbf{q}(\boldsymbol{\rho}_n)\mathbf{q}(\boldsymbol{\rho}_{n-1}) \cdots \mathbf{q}(\boldsymbol{\rho}_3)\mathbf{q}(\boldsymbol{\rho}_2)\mathbf{q}(\boldsymbol{\rho}_1) \tag{C.252}$$

$$= \mathbf{q}(\boldsymbol{\rho}_n)\mathbf{q}_{[n-1]} \tag{C.253}$$

then represents the net effect of the successive rotations as

$$\mathbf{q}(\mathbf{w}_n) = \mathbf{q}_{[n]}\mathbf{q}(\mathbf{w}_0)\mathbf{q}_{[n]}^*. \tag{C.254}$$

The initial value $\mathbf{q}_{[0]}$ for the rotation quaternion will depend upon the inital orientation of the two coordinate systems. The initial value

$$\mathbf{q}_{[0]} \stackrel{\text{def}}{=} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{C.255}$$

applies to the case that the two coordinate systems are aligned. In strapdown system applications, the initial value $\mathbf{q}_{[0]}$ is determined during the INS alignment procedure.

Equation C.252 is the much-used quaternion representation for successive rotations, and Eq. C.254 is how it is used to perform coordinate transformations of any vector $\mathbf{w}_0$.

This representation uses the four components of a unit quaternion to maintain the transformation from one coordinate frame to another through a succession of rotations. In practice, computer roundoff may tend to alter the magnitude of the alegedly unit quaternion, but it can easily be rescaled to a unit quaternion by dividing by its magnitude.

Global Positioning Systems, Inertial Navigation, and Integration,
Mohinder S. Grewal, Lawrence R. Weill, Angus P. Andrews

Print ISBN 0-471-35032-X Electronic ISBN 0-471-20071-9

Glossary

$\propto$ Proportional to.

$\otimes$ Vector cross-product.

$\omega_{\oplus}$ Inertial rotation rate of the earth ($\approx 7292115167 \times 10^{-14}$ rads/s).

A posteriori Pertaining to the probability distribution of the *corrected value* of a variable. The term applies to the expected value (mean) and variance (or covariance matrix).

A priori Pertaining to the probability distribution of the *predicted value* of a variable.

argmax The argument of a function where it achieves its maximum value, e.g., $\text{argmax}(\sin) = \pi/2$.

Argument of perigee The angle measured in the orbital plane from the ascending node, in the direction of satellite motion, to the perigee (closest approach to the earth) of the satellite orbit.

ARINC Aeronautical Radio, Inc. An organization providing technical staff for various international airline committees.

ARINC 24 Format from Aeronautical Radio, Inc. (ARINC) for the transfer of data from one data base to another.

Ascending node The ascending node of a satellite in orbit about the earth is the direction from the center of mass of the earth to the point where the satellite crosses from the Southern Hemisphere into the Northern Hemisphere. The opposite node is called the *descending node*.

bps Bits per second.

BPSK Binary phase-shift keying. A carrier modulation scheme using zero and 180° phase shifts.

c Symbol used for the speed of light in a vacuum, 299,792,458 m/s. This value of *c* is exact in the International System of Units (SI).

CDM Code division multiplexing. A signaling protocol for sharing a common bandwidth allocation amoung several users by using independent spread-spectrum modulations (i.e., spreading codes).

CEP Circle of equal probability. The radius of a circle centered at the mean of a distribution such that the probabilities of being inside or outside the circle are equal.

Datum In geodesy, cartography (mapping), surveying, navigation, and astronomy, a *datum*[1] is a set of parameter values defining a model for the size and shape of the earth and physical control points defining the origin and orientation of a coordinate system for mapping the earth. A datum used for navigation may also include the rotation rate and gravity field of the earth.

Descending node See *ascending node*.

DGPS Differential GPS. Employing a reference station at a known location to determine timing corrections for each satellite in view.

DME Distance-measuring Equipment. A radionavigation aid giving the receiver user his or her direction and range from a transmitting station with known location.

DOP Dilution of precision. A measure of degradation of the navigation estimate due to satellite geometry.

Earthrate The inertial rotation rate of the earth, approximately $7{,}292{,}115 \times 10^{-11}$ rad/s, or 15.04109° per hour.

Easting Distance eastward from a reference point. In global easting–northing coordinates, positive easting is measured eastward along the equator from the prime meridian.

ECEF Earth centered, earth fixed. Coordinates centered in the earth and rotating with the earth.

ECI Earth centered inertial. Inertial (i.e., nonrotating) coordinates centered in the earth.

EGM Earth Gravity Model. A designation used by the U.S. National Imaging and Mapping Agency to designated geoid models of the gravitational field of the earth. EGM 96 is the name used for the geoid model based on 1996 data.

Epoch The instant of time at which a given data set (e.g., satellite ephemeris) is valid.

[1]*Data* and *datum* are both derived from the Latin verb *dare*, "to give," the passive voice perfect participle of which means "given." Depending on the grammatical gender, number, and case of the noun to which it refers (i.e., what is given), its Latin spelling could be *data, datae, datam, datarum, datas, dati, datis, dato, datorum, datos, datum,* or *datus.* In the nominative case, neuter gender, *datum* is singular and *data* is plural. In Engish usage, either may refer to something singular or plural, and *datum* can be pluralized as *datums.*

Expected value The mean of a probability distribution or a function of a distributed variate.

FAR False-alarm rate. Equal to the expected number of false detections in a specified time period, based on the detection threshold used.

FSLF Free-space loss factor. A formula (Eq. 3.11) that accounts for the decrease in signal power density with distance.

g Unit of acceleration, approximately equal to gravitational acceleration at the surface of the earth, or $9.80665\,\mathrm{m/s^2}$.

Geoid A model for an equipotential surface (mean sea height, usually) of the gravitational field of the earth.

GLONASS Global Orbiting Navigation Satellite System. Developed by the former Soviet Union and currently maintained by the Russian Republic.

Inclination The inclination of a satellite orbit is the dihedral angle between the equatorial plane and the satellite orbit plane.

L_1 GPS L-band signal with carrier center frequency at 1575.42 MHz.

L_2 GPS L-band signal with carrier center frequency at 1227.6 MHz.

L-band The 1–2 GHz frequency range.

Line of nodes The line of intersection of an orbit plane and the equatorial plane.

MEMS Micro-Electro-Mechanical Systems. Including electromechanical device design and fabrication technologies derived from semiconductor processing technology.

ML Maximum likelihood. A statistical estimation method based on likelihood functions (as opposed to probability functions) and on maximizing the likelihood of the estimate, rather than minimizing some expected loss function (e.g., minimum mean-squared error).

NMI Nautical mile ($=1852\,\mathrm{m}$).

Northing Distance northward from a reference point. In global easting–northing coordinates, northing is measured from the equator.

PLGR Personal low-cost GPS receiver. A militarized hand-held GPS receiver with L_2 signal access.

ppm Parts per million.

PRN Pseudorandom noise or pseudorandom number. Also used to designate the C/A spreading code number (1–31) to designate a GPS satellite ID, and used synonymously with the SVN (space vehicle number).

PSD Power spectral density. Power distribution density in the frequency domain.

Pseudorange The apparent range to a GPS satellite from the receiver antenna, calculated from the time of signal transmission (encoded in the received signal), time of signal reception (including receiver clock errors), and speed of signal propagation (c).

RAAN Right ascension of ascending node. Essentially, the celestial longitude (measured from the vernal equinox) of the ascending node of a satellite orbit. The

ascending node of a satellite in orbit about the earth is the direction from the center of mass of the earth to the point where the satellite crosses from the Southern Hemisphere into the Northern Hemisphere. Its right ascension is the angle measured in the equatorial plane from the direction of the vernal equinox to the ascending node.

SAVVAN Système Automatique de Vérification en Vol des Aides a la Navigation. An automatic in-flight navigation aids checking system.

Selective Availability Intentional introduction of errors into GPS signals to deny full accuracy capability to unauthorized users.

SNR Signal-to-noise ratio. The unitless ratio of signal power to noise power, or (when a function of frequency) signal power spectral density to noise power spectral density.

SVN Space vehicle number. A unique identification number assigned to each operating GPS satellite. Synonymous with PRN.

TLM Telemetry word in GPS message subframe.

TTFF Time to first fix. A GPS receiver performance characteristic defined as the average time between receiver turn-on and the first four-dimensional (position and time) navigation solution.

Variate Random variable.

Vernal Equinox The direction from the center of mass of the earth to the center of mass of the sun at the instant when the sun is passing through the equatorial plane from the Southern Hemisphere to the Northern Hemisphere.

VOR VHF OmniRange. A radio navigation aid providing users with range-only information relative to transmitters with known locations.

W-code Secure code used in encrypting P-code to produce Y-code.

WGS 84 World Geodetic System 1984. Datum for the earth, defining a gravitational constant, a rotation rate, origin and orientation of ECEF coordinates, and a reference ellipsoid.

WN Week number (0–1023). Referenced to last rollover date (when WN last reset to zero). (Rollover occurred at GPS time zero on August 22, 1999.)

Y-code Encrypted P-code in the L_2 channel.

Global Positioning Systems, Inertial Navigation, and Integration,
Mohinder S. Grewal, Lawrence R. Weill, Angus P. Andrews

Print ISBN 0-471-35032-X Electronic ISBN 0-471-20071-9

References

1. R. Ahmadi, G. S. Becker, S. R. Peck, F. Choquette, T. F. Gerard, A. J. Mannucci, B. A. Iijima, and A. W. Moore, "Validation Analysis of the WAAS GIVE and UIVE Algorithms," in *Proceedings of the Institute of Navigation, ION '98*, (Santa Monica CA), ION, Alexandria, VA, Jan. 1998.
2. D. W. Allan, "The Measurement of Frequency and Frequency Stability of Precision Oscillators", NBS Technical Note 669, pp. 1–27, 1975.
3. D. W. Allan, "Fine-Tuning Time in the Space Age," *IEEE Spectrum*, Vol. 35, No. 3, pp. 43–51, 1998.
4. D. W. Allan, "Time-Domain Spectrum of GPS SA," in *Proceedings of the 6th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION) GPS-93*, (Salt Lake City, UT), Sept. 22–24, 1993, pp. 129–136, ION, Alexandria, VA, 1993.
5. A. Andrews, "A Square Root Formulation of the Kalman Covariance Equations," *AIAA Journal*, Vol. 6, pp. 1165–1166, 1968.
6. T. Barnes, "Selective Availability via the Levinson Predictor," in *Proceedings of the Institute of Navigation (ION), GPS '95*, (Palm Springs, CA), Sept. 12–15, ION, Alexandria, VA, 1995.
7. R. R. Bate, D. D. Mueller, and J. E. White, *Fundamentals of Astrodynamics*, Dover, New York, 1971.
8. R. H. Battin, *Astronautical Guidance*, McGraw-Hill, New York, 1964.
9. Commandant Benoit, "Sur une méthode de résolution des équations normales provenant de l'application de la méthode des moindes carrés a un système d'équations linéaires en numbre inférieur a celui des inconnues—application de la méthode a la resolution d'un système defini d'équations linéaires, (Procédé du Commandant Cholesky)," *Bulletin Géodêsique et Géophysique Internationale*, Toulouse, pp. 67–77, 1924.

10. G. L. Bierman, *Factorization Methods for Discrete Sequential Estimation*, *Mathematics in Science and Engineering*, Vol. 128, Academic, New York, 1977.
11. Å. Björck, "Solving Least Squares Problems by Orthogonalization," *BIT*, Vol. 7, pp. 1–21, 1967.
12. F. R. Bletzacker, D. H. Eller, T. M. Gorgette, G. L. Seibert, J. L. Vavrus, and M. D. Wade, "Kalman Filter Design for Integration of Phase III GPS with an Inertial Navigation System," in *Proceedings of the Institute of Navigation*, (Santa Barbara , CA) Jan. 26–29, 1988, pp. 113–129, ION, Alexandria, VA, 1988.
13. G. Blewitt, "An Automatic Editing Algorithm for GPS Data," *Geophysical Research Letters*, Vol. 17, No. 3, pp. 199–202, 1990.
14. M. S. Braasch, "Improved Modeling of GPS Selective Availability," in *Proceedings of The Institute of Navigation (ION) Annual Technical Meeting*, ION, Alexandria, VA, 1993.
15. M. S. Braasch, "A Signal Model for GPS," *NAVIGATION: Journal of the Institute of Navigation*, Vol. 37, No. 4, pp. 363–379, 1990.
16. E. A. Bretz, "X Marks the Spot, Maybe," *IEEE Spectrum*, Vol. 37, No. 4, pp. 26–36, 2000.
17. R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, 2nd ed., Wiley, New York, 1992.
18. R. G. Brown and P. Y. C. Hwhang, *Introduction to Random Signals and Applied Kalman Filtering: With Matlab Exercises and Solutions*, 3rd ed. Wiley, New York, 1997.
19. R. S. Bucy and P. D. Joseph, *Filtering for Stochastic Processes with Applications to Guidance*, Chelsea, New York, 1968 (republished by the American Mathematical Society).
20. N. A. Carlson, "Fast Triangular Formulation of the Square Root Filter," *AIAA Journal*, Vol. 11, No. 9, pp. 1259–1265, 1973.
21. Y. Chao and B. W. Parkinson, "The Statistics of Selective Availability and Its Effects on Differential GPS," in *Proceedings of the 6th International Technical Meeting of the Satellite Division of (ION) GPS-93*, (Salt Lake City, UT) Sept. 22–24, 1993, ION; Alexandria, VA, 1993. pp. 1509–1516.
22. A. B. Chatfield, *Fundamentals of High Accuracy Inertial Navigation,* American Institute of Aeronautics and Astronautics, New York, 1997.
23. S. Cooper and H. Durrant-Whyte, "A Kalman Filter Model for GPS Navigation of Land Vehicles," *Intelligent Robots and Systems*, IEEE, Piscataway, NJ, 1994.
24. J. P. Costas, "Synchronous Communications," *Proceedings of the IRE*, Vol. 45, pp. 1713–1718, 1956.
25. E. Copros, J. Spiller, T. Underwood, and C. Vialet, "An Improved Space Segment for the End-State WAAS and EGNOS Final Operational Capability," in *Proceedings of the Institute of Navigation*, ION GPS-96 (Kansas City, MO), pp. 1119–1125 ION, Alexandria, VA, Sept. 1996.
26. G. G. Coriolis, "Sur les équations du mouvement relatif des systèmes de corps," Ecole Polytechnique, Paris, 1835.
27. C. C. Counselman, III "Multipath-Rejecting GPS Antennas," *Proceedings of the IEEE*, Vol. 87, No. 1, pp. 86–91, 1999.
28. P. Daum, J. Beyer, and T. F. W. Köhler, "Aided Inertial LAnd NAvigation system

(ILANA) with a Minimum Set of Inertial Sensors," in *Proceedings of IEEE Position, Locations and Navigation Conference*, Las Vegas, 1994.

29. A. J. Dierendonck, "Understanding GPS Receiver Terminology: A Tutorial," *GPS WORLD* January 1995, pp. 34–44.
30. Mani Djodat, "Comparison of Various Differential Global Positioning Systems," California State University, Fullerton Masters' Thesis, Fullerton, CA, 1996.
31. R. M. du Plessis, "Poor Man's Explanation of Kalman Filtering, or How I Stopped Worrying and Learned to Love Matrix Inversion," Autonetics Technical Note, Anaheim, CA, 1967, republished by Taygeta Scientific Incorporated, Monterey, CA, 1996.
32. P. Dyer and S. McReynolds, "Extension of Square-Root Filtering to Include Process Noise," *Journal of Optimization Theory and Applications*, Vol. 3, pp. 444–458, 1969.
33. M. B. El-Arini, El-Arini, Robert S. Conker, Thomas W. Albertson, James K. Reagan, John A. Klobuchar, and Patricia H. Doherty, "Comparison of Real-Time Ionospheric Algorithms for a GPS Wide-Area Augmentation System," *Navigation*, Vol. 41, no. 4, pp. 393–413, Winter 1994/1995.
34. J. A. Farrell and M. Barth, *The Global Positioning System & Inertial Navigation*, McGraw-Hill, New York, 1998.
35. Federal Aviation Administration (U.S.A.), *FAA* Specification WAAS FAA-E-2892 B, Oct. 1997.
36. C. M. Feit, "GPS Range Updates in an Automatic Flight Inspection System: Simulation, Static and Flight Test Results," in *Proceedings of The Institute of Navigation (ION) GPS-92*, pp. 75–86 ION, Alexandria, VA, Sept. 1992.
37. W. A. Feess and S. G. Stephens, "Evaluation of GPS Ionospheric Time Delay Algorithm for Single Frequency Users," in *Proceedings of the IEEE Position, Location, and Navigation Symposium (PLANS '86)* (Las Vegas, NV), Nov. 4–7, 1986, pp. 206–213. New York, NY, 1986.
38. M. E. Frerking, "Fifty Years of Progress in Quartz Crystal Frequency Standards," *Proceedings of the 1996 IEEE International Frequency Control Symposium*, IEEE, New York, 1996, pp. 33–46.
39. L. Garin, F. van Diggelen, and J. Rousseau, "Strobe and Edge Correlator Multipath Mitigation for Code," in *Proceedings of ION GPS-96, the Ninth International Technical Meeting of the Satellite Division of the Institute of Navigation*, (Kansas City, MO), ION, Alexandria, VA, 1996. pp. 657–664.
40. A. Gelb (Editor). *Applied Optimal Estimation*, MIT Press, Cambridge, MA, 1974.
41. G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed., Johns Hopkins University Press, Baltimore, MD, 1989.
42. GPS Interface Control Document ICD-GPS-200, Rockwell International Corporation, Satellite Systems Division, Revision B, July 3, 1991.
43. R. L. Greenspan, "Inertial Navigation Technology from 1970–1995," *Navigation, The Journal of the Institute of Navigation*, Vol. 42, No. 1, pp. 165–186, 1995.
44. M. S. Grewal, "GEO Uplink Subsystem (GUS) Clock Steering Algorithms Performance and Validation Results," in *Proceedings of 1999 ION Conference, Vision 2010: Present & Future, National Technical Meeting*, (San Diego, CA) Jan. 25–27, pp. 853–859 ION, Alexandria, VA, 1999.

45. M. S. Grewal and A. P. Andrews, *Application of Kalman Filtering to GPS, INS, & Navigation*, Short Course Notes, Kalman Filtering Consulting Associates, Anaheim, CA, June 2000.

46. M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice*, 2nd ed., Wiley, New York, 2000.

47. M. S. Grewal, W. Brown, S. Evans, P. Hsu, and R. Lucy, "Ionospheric Delay Validation Using Dual Frequency Signal from GPS at GEO Uplink Subsystem (GUS) Locations," *Proceedings of ION GPS '99, Satellite Division of the Institute of Navigation Twelfth International Technical Meeting*, Session C4, Atmospheric Effects, (Nashville, TN) September 14–17, ION, Alexandria, VA, 1999.

48. M. S. Grewal, W. Brown, and R. Lucy, "Test Results of Geostationary Satellite (GEO) Uplink Sub-System (GUS) Using GEO Navigation Payloads," *Monographs of the Global Positioning System: Papers Published in Navigation ("Redbook")*, Vol. VI, Institute of Navigation, pp. 339–348, ION, Alexandria, VA, 1999.

49. M. S. Grewal, W. Brown, P. Hsu, and R. Lucy, "GEO Uplink Subsystem (GUS) Clock Steering Algorithms Performance, Validation and Test Results," in *Proceedings of Thirty-First Annual Precise Time and Time Interval (PTTI) Systems and Applications Meeting*, (Dana Point, CA), December 7–9, 1999. Time Services Dept., US Naval Observatory, Washington, DC.

50. M. S. Grewal, N. Pandya, J. Wu, and E. Carolipio, "Dependence of User Differential Ranging Error (UDRE) on Augmentation Systems—Ground Station Geometries," in *Proceedings of the Institute of Navigation's (ION) 2000 National Technical Meeting*, (Anaheim, CA) January 26–28, 2000, pp. 80–91, ION Alexandria, VA, 2000.

51. L. Hagerman, "Effects of Multipath on Coherent and Noncoherent PRN Ranging Receiver," Aerospace Report No. TOR-0073(3020-03)-3, Aerospace Corporation, Development Planning Division, El Segundo, CA, May 15, 1973.

52. B. Hassibi, A. H. Sayed, and T. Kailath, *Indefinite Quadratic Estimation and Control: A Unified Approach to* $\mathbf{H}^2$ *and* $\mathbf{H}^\infty$ *Theories*, SIAM, Philadelphia, PA, 1998.

53. H. V. Henderson and S. R. Searle, "On Deriving the Inverse of a Sum of Matrices," *SIAM Review*, Vol. 23, pp. 53–60, 1981.

54. T. A. Herring, "The Global Positioning System," *Scientific American*, Feb. 1996, pp. 44–50.

55. D. Herskovitz, "A Sampling of Global Positioning System Receivers," *Journal of Electronic Defense*, pp. 61–66, May 1994.

56. B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *GPS: Theory and Practice*, Springer-Verlag, Vienna, 1997.

57. P. Y. C. Hwang, "Recommendations for Enhancement of RTCM-104 Differential Standard and Its Derivatives," in *Proceedings of the Sixth International Technical Meeting of the Satellite Division of the The Institute of Navigation (ION) GPS-93*, (Salt Lake City, UT) pp. 1501–1508, ION, Alexandria, VA, Sept. 1993.

58. Institute of Navigation, *Monographs of the Global Positioning System: Papers Published in NAVIGATION ("Redbook")*, Vol. I, ION, Alexandria, VA, 1980.

59. Institute of Navigation, *Monographs of the Global Positioning System: Papers Published in NAVIGATION ("Redbook")*, Vol. II, ION, Alexandria, VA, 1984.

60. Institute of Navigation, *Monographs of the Global Positioning System: Papers Published*

in NAVIGATION ("Redbook"), with Overview by R. Kalafus, Vol. III, ION, Alexandria, VA, 1986.

61. Institute of Navigation, *Monographs of the Global Positioning System: Papers Published in NAVIGATION ("Redbook")*, with Overview by R. Hatch, Institute of Navigation, Vol. IV, ION, Alexandria, VA, 1993.
62. Institute of Navigation, *Monographs of the Global Positioning System: Papers Published in NAVIGATION ("Redbook")*, Vol. V, ION, Alexandria, VA, 1998.
63. Institute of Navigation, *Global Positioning System, Selected Papers on Satellite Based Augmentation Systems (SBASs) ("Redbook")*, Vol. VI, ION Alexandria, VA, 1999.
64. K. Ito, K. Hoshinoo, and M. Ito, "Differential Positioning Experiment Using Two Geostationary Satellites," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 35, No. 3, pp. 866–878, 1999.
65. H. W. Janes, R. B. Langley, and S. P. Newby, "Analysis of Tropospheric Delay Prediction Models: Comparisons with Ray-Tracing and Implications for GPS Relative Positioning," *Bulletin Geodisique*, Vol. 65, No. 3, pp. 151–161, 1991.
66. J. M. Janky, "Clandestine Location Reporting by a Missing Vehicle," U.S. Patent 5,629,693, May 13, 1997.
67. A. H. Jazwinski, *Stochastic Processes and Filtering Theory*, Academic, San Diego, CA, 1970.
68. G. E. Johnson, "Constructions of Particular Random Processes," *Proceedings of the IEEE*, Vol. 82, No. 2, 1994.
69. T. Kailath, A. H. Sayed, and B. Hassibi, "Kalman Filtering Techniques," in *Wiley Encyclopedia of Electrical and Electronics Engineering*, Wiley, New York, 1999.
70. R. Kalafus, A. J. Van Dierendonck, and N. Pealer, "Special Committee 104 Recommendations for Differential GPS Service," in *Global Positioning System*, Vol. III, Institute of Navigation, ION, Alexandria, VA, pp. 101–116, 1986.
71. R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *ASME Transactions, Series D: Journal of Basic Engineering*, Vol. 82, pp. 35–45, 1960.
72. R. E. Kalman and R. S. Bucy, "New Results in Linear Filtering and Prediction Theory," *ASME Transactions, Series D: Journal of Basic Engineering*, Vol. 83, pp. 95–108, 1961.
73. P. G. Kaminski, "Square Root Filtering and Smoothing for Discrete Processes," Ph.D. Thesis, Stanford University, Stanford, CA, 1971.
74. E. D. Kaplan, *Understanding GPS Principles and Applications*, Artech House, Boston, 1996.
75. M. Kayton and W. L. Fried, *Avionics Navigation Systems*, 2nd ed. Wiley, New York, 1997.
76. J. A. Klobuchar,"Ionospheric Time Delay Corrections for Advanced Satellite Ranging Systems," NATO AGARD Conference Proceedings No. 209, in *Propagation Limitations of Navigation and Positioning Systems*, NATO AGARD, Paris, France, 1976.
77. D. T. Knight,"Demonstration of a New, Tightly-Coupled GPS/INS," in *Proceedings of the Sixth International Technical Meeting, Institute of Navigation, ION GPS-93*, (Salt Lake City, UT), September 22–24, ION, Alexandria, VA, 1993.
78. D. T. Knight, "Rapid Development of Tightly-Coupled GPS/INS Systems," *IEEE AES Systems Magazine*, Feb. 1997, pp. 14–18.

79. D. Kügler, "Integration of GPS and Loran-C/Chayka: A European Perspective, *Navigation: Journal of the Institute of Navigation*, Vol. 46, No. 1, pp. 1–13, 1999.
80. A. Leick, "Appendix G," *GPS: Satellite Surveying*, 2nd ed., Wiley, New York, 1995, pp. 534–537.
81. R. B. Langley, "A GPS Glossary," *GPS World*, Oct. 1995, pp. 61–63.
82. R. B. Langley, "The GPS Observables," *GPS World*, Apr. 1993, pp. 54–59.
83. A. Lawrence, *Modern Inertial Technology: Navigation, Guidance, and Control*, 2nd ed., Springer-Verlag, New York, NY, 1993.
84. T. Logsdon, *The NAVSTAR Global Positioning System*, Van Nostrand Reinhold, New York, NY pp. 1–90, 1992.
85. *Loran-C User Handbook*, Department of Transportation, U.S. Coast Guard, Commandant Instruction M16562.3, Washington, DC, May 1990.
86. P. F. MacDoran, inventor, "Method and Apparatus for Calibrating the Ionosphere and Application to Surveillance of Geophysical Events," U.S. Patent No. 4,463,357, July 31, 1984.
87. J. O. Mar and J.-H. Leu, "Simulations of the Positioning Accuracy of Integrated Vehicular Navigation Systems," *IEE Proceedings-Radar Sonar Navigation*, Vol. 143, No. 2, pp. 121–128, 1996.
88. M. B. May, "Inertial Navigation and GPS," *GPS World*, Sept. 1993, pp. 56–65.
89. P. S. Maybeck, *Stochastic Models, Estimation and Control* (3 vol.), Academic, New York, NY, 1979.
90. G. McGraw and M. Braasch, "GNSS Multipath Mitigation Using Gated and High Resolution Correlator Concepts," *Proceedings of the 1999 National Technical Meeting and Nineteenth Biennial Guidance Test Symposium*, Institute of Navigation, San Diego, CA, 1999, pp. 333–342.
91. J. C. McMillan and D. A. G. Arden, "Sensor Integration Options for Low Cost Position and Attitude Determination," in *Proceedings of IEEE Position, Location and Navigation Conference*, (PLANS '94) (Las Vegas), IEEE, New York, NY, 1994.
92. R. Moreno and N. Suard, "Ionospheric Delay Using Only L1: Validation and Application to GPS Receiver Calibration and to Inter-Frequency Bias Estimation," in *Proceedings of The Institute of Navigation (ION)*, Jan. 25–27, 1999. pp. 119–129, ION, Alexandria, VA, 1999.
93. M. Morf and T. Kailath, "Square Root Algorithms for Least Squares Estimation," *IEEE Transactions on Automatic Control*, Vol. AC-20, pp. 487–497, 1975.
94. NAVSYS Corporation, "WAAS Dual Frequency Ranging and Timing Analysis Design Report," NAVSYS Corp. Report, Nov. 11, 1996, Colorado Springs, CO.
95. N. Pandya, "Dependence of GEO UDRE on Ground Station Geometries," "WAAS Engineering Notebook," Raytheon Systems Company, Fullerton, CA, Dec. 1, 1999.
96. B. W. Parkinson, and J. J. Spilker, Jr. Vol. 1, (Eds.), *Global Positioning System: Theory and Applications*, Progress in Astronautics and Aeronautics, American Institute of Aeronautics and Astronautics, Washington, DC, 1996.
97. B. W. Parkinson and J. J. Spilker, Jr. (Eds.) *Global Positioning System: Theory and Applications, Vol. 2*, Progress in Astronautics and Aeronautics, American Institute of Aeronautics and Astronautics, Washington, DC, 1996.

98. B. W. Parkinson, M. L. O'Connor and K. T. Fitzgibbon, "Aircraft Automatic Approach and Landing Using GPS," Chapter 14, *Global Positioning System: Theory & Applications Vol. II*, B. W. Parkinson and J. J. Spilker, Jr. eds, from Progress in Astronautics and Aeronautics, Vol. 164, Paul Zarchan, ed-in-chief, American Institute of Aeronautics and Astronautics, Washington, DC, pp. 397–425, 1995.
99. S. Peck, C. Griffith, V. Reinhardt, W. Bertiger, B. Haines, and G. M. R. Winkler, "WAAS Network Time Performance and Validation Results," Proceedings of the Institute of Navigation, (Santa Monica CA), ION, Alexandria VA, Jan. 1998.
100. J. E. Potter and R. G. Stern, "Statistical Filtering of Space Navigation Measurements, in *Proceedings of the 1963 AIAA Guidance and Control Conference*, American Institute of Aeronautics and Astronautics, Washington, DC, 1963.
101. K. D. Rao and L. Narayana, "An Approach for a Faster GPS Tracking Extended Kalman Filter," *NAVIGATION: Journal of the Institute of Navigation*, Vol. 42, No. 4, pp. 619–630, 1995/1996.
102. H. E. Rauch, F. Tung, and C. T. Streibel, "Maximum Likelihood Estimates of Linear Dynamic Systems," *AIAA Journal*, Vol. 3, pp. 1445–1450, 1965.
103. Raytheon Systems Company, "GEO Uplink System (GUS) Clock Steering Algorithms," ENB, Fullerton, CA. April 13, 1998.
104. D. Roddy, *Satellite Communications*, 2nd ed., McGraw-Hill, New York, 1989.
105. P. Ross and R. Lawrence, "Averaging Measurements to Minimize SA Errors," Internet communication on AOL, September 18, 1995.
106. RTCA, "Minimum Operational Performance Standard for Global Positioning System/Wide Area Augmentation System Airborne Equipment," RTCA/DO-229, Jan. 16, 1996, and subsequent changes, Appendix A, "WAAS System Signal Specification," RTCA, Washington, DC.
107. O. Salychev, *Inertial Systems in Navigation and Geophysics*, Bauman Moscow State Technical University Press, Moscow, Russia, 1998.
108. S. F. Schmidt, "Application of State Space Methods to Navigation Problems," in C. T. Leondes (Ed.), *Advances in Control Systems*, Vol. 3, Academic, New York, 1966.
109. K. P. Schwartz, M. Wei, and M. Van Gelderen, "Aided Versus Embedded: A Comparison of Two Approaches to GPS/INS Integration," in *Proceedings of IEEE Position, Locations and Navigation Conference*, (Las Vegas) NV, IEEE, New York, NY, 1994.
110. G. Seeber, *Satellite Geodesy: Foundation, Methods, and Applications*, Walter de Gruyter, Berlin, 1993.
111. J. Sennott, I.-S. Ahn and D. Pietraszewski, "Marine Applications," U.S. Government, Chapter 11, Global Positioning Systems: Theory & Applications vol. II, (see Ref. 98) pp. 303–325.
112. T. Stansell, Jr., "RTCM SC-104 Recommended Pseudolite Signal Specification,"*Global Positioning System*, Vol. III, Institute of Navigation, 1986, pp. 117–134.
113. X. Sun, C. Xu and Y. Wang, "Unmanned Space Vehicle Navigation by GPS," *IEEE AES Systems Magazine*, Vol. 11 pp. 31–33, July 1996.
114. P. Swerling, "First Order Error Propagation in a Stagewise Smoothing Procedure for Satellite Observations," *Journal of the Astronomical Society*, Vol. 6, pp. 46–52, 1959.
115. S. Thomas, *The Last Navigator: A Young Man, An Ancient Mariner, the Secrets of the Sea*, McGraw-Hill, New York, 1997.

116. C. L. Thornton, "Triangular Covariance Factorizations for Kalman Filtering," Ph.D. Thesis, University of California at Los Angeles, School of Engineering, 1976.

117. C. L. Thornton and G. J. Bierman, "Gram-Schmidt Algorithms for Covariance Propagation," *International Journal of Control*, Vol. 25, No. 2, pp. 243–260, 1977.

118. D. H. Titterton and J. L. Weston, *Strapdown Inertial Navigation Technology*, Peter Peregrinus, Stevenage, United Kingdom, 1997.

119. B. Townsend and P. Fenton, "A Practical Approach to the Reduction of Pseudorange Multipath Errors in a L1 GPS Receiver," in *Proceedings of ION GPS-94, the Seventh International Technical Meeting of the Satellite Division of the Institute of Navigation*, (Salt Lake City, UT), Alexandria, VA, 1994, pp. 143–148.

120. B. Townsend, D. J. R. Van Nee, P. Fenton, and K. Van Dierendonck, "Performance Evaluation of the Multipath Estimating Delay Lock Loop," in *Proceedings of the National Technical Meeting*, Institute of Navigation, Anaheim, CA, 1995, pp. 277–283.

121. A. J. Van Dierendonck, P. Fenton, and T. Ford, "Theory and Performance of Narrow Correlator Spacing in a GPS Receiver," in *Proceedings of the National Technical Meeting*, Institute of Navigation, San Diego, CA, 1992, pp. 115–124.

122. A. J. Van Dierendonck, S. S. Russell, E. R. Kopitzke, and M. Birnbaum, "The GPS Navigation Message," in *Global Positioning System*, Vol. I, ION, Alexandria, VA, 1980.

123. H. L. Van Trees, *Detection, Estimation and Modulation Theory*, Part 1, Wiley, New York, 1968.

124. M. Verhaegen and P. Van Dooren, "Numerical Aspects of Different Kalman Filter Implementations," *IEEE Transactions on Automatic Control*, Vol. AC-31, pp. 907–917, 1986.

125. A. Wald, *Sequential Analysis*, Wiley, New York, 1947.

126. L. Weill, "C/A Code Pseudoranging Accuracy—How Good Can it Get?" *Proceedings of ION GPS-94, the Seventh International Technical Meeting of the Satellite Division of the Institute of Navigation*, (Salt Lake City, UT,) 1994, pp. 133–141. ION, Alexandria, VA.

127. L. Weill, "GPS Multipath Mitigation by Means of Correlator Reference Waveform Design," in *Proceedings of the National Technical Meeting*, Institute of Navigation, (Santa Monica, CA) 1997, pp. 197–206. ION, Alexandria, VA, pp. 97–207.

128. L. Weill, "Application of Superresolution Concepts to the GPS Multipath Mitigation Problem," in *Proceedings of the National Technical Meeting*, Institute of Navigation, (Long Beach, CA), 1998, ION, Alexandria, VA, pp. 673–682.

129. L. Weill, "Achieving Theoretical Accuracy Limits for Pseudoranging in the Presence of Multipath," in *Proceedings of ION GPS-95, the Eighth International Technical Meeting of the Satellite Division of the Institute of Navigation*, (Palm Springs, CA) 1995, ION, Alexandria, VA, pp. 1521–1530.

130. L. Weill and B. Fisher, "Method for Mitigating Multipath Effects in Radio Ranging Systems," U.S. Patent No. 6,031,881, February 29, 2000.

131. J. D. Weiss and D. S. Kee, "A Direct Performance Comparison Between Loosely Coupled and Tightly Coupled GPS/INS Integration Techniques," in *Proceedings of the Fifty-First Annual Meeting, Institute of Navigation*, (Colorado Springs, CO), June 5–7, 1995, pp. 537–544. ION, Alexandria, VA, 1995

132. J. D. Weiss, "Analysis of an Upgraded GPS Internal Kalman Filter," *IEEE AES Systems Magazine*, Jan. 1996, pp. 23–26.

133. *WGS 84 Implementation Manual*, Version 2.4, European Organization for the Safety of Air Navigation (EUROCONTROL), Brussels, Feb. 1998.

134. C. E. Wheatley III, C. G. Mosley, and E. V. Hunt, "Controlled Oscillator Having Random Variable Frequency," U. S. Patent No. 4,646,032, February 24, 1987.

135. J. E. D. Williams, *From Sails to Satellites: The Origin and Development of Navigational Science*, Oxford University Press, 1992.

136. C. Yinger, W. A. Feess, R. DiEsposti, A. Chasko, B. Cosentino, D. Syse,, B. Wilson, and B. Wheaton, "GPS Satellite Interfrequency Biases," in *Proceedings of The Institute of Navigation (ION)*, (Cambridge, MA) June 28–30, 1999. pp 347–355, ION, Alexandria, VA.

137. T. Yunk et al., "A Robust and Efficient New Approach to Real Time Wide Area Differential GPS Navigation for Civil Aviation," NASA/JPL Internal Report JPL D-12584, 1995, Pasadena, CA.

Global Positioning Systems, Inertial Navigation, and Integration,
Mohinder S. Grewal, Lawrence R. Weill, Angus P. Andrews

Print ISBN 0-471-35032-X Electronic ISBN 0-471-20071-9

Index