10

APPLICATIONS TO NAVIGATION

> The acquisition of the knowledge of navigation has a strange effect on the minds of men.
>
> —Jack London
> The Cruise of the Snark, Macmillan, 1911.

10.1 CHAPTER FOCUS

It is hard to imagine an application for Kalman filtering more fruitful than navigation. Before Kalman filtering, navigation was practiced only by technical specialists with years if training. Today, it is available as a commodity service for the consumer, requiring no more skill than using a smartphone.

When the Kalman filter was published in 1960, it would find immediate applications in many military systems, including inertial and satellite navigation. It would play a major role in the development of satellite navigation, in particular, and in the integration of inertial and satellite navigation systems.

This has spawned a vast area of technology that had been largely classified until the late twentieth century but has since seen many successful commercial applications, illustrating how capabilities of Kalman filtering can be matched to the demands of very complex estimation problems.

The focus here will be on the Kalman filter architectures illustrated in Figure 10.1, and particularly on the contents of the dashed boxes within the two boxes labeled "Kalman filter." We shall not delve too deeply into what happens in the other boxes,

Kalman Filtering: Theory and Practice Using MATLAB®, Fourth Edition.
Mohinder S. Grewal and Angus P. Andrews.

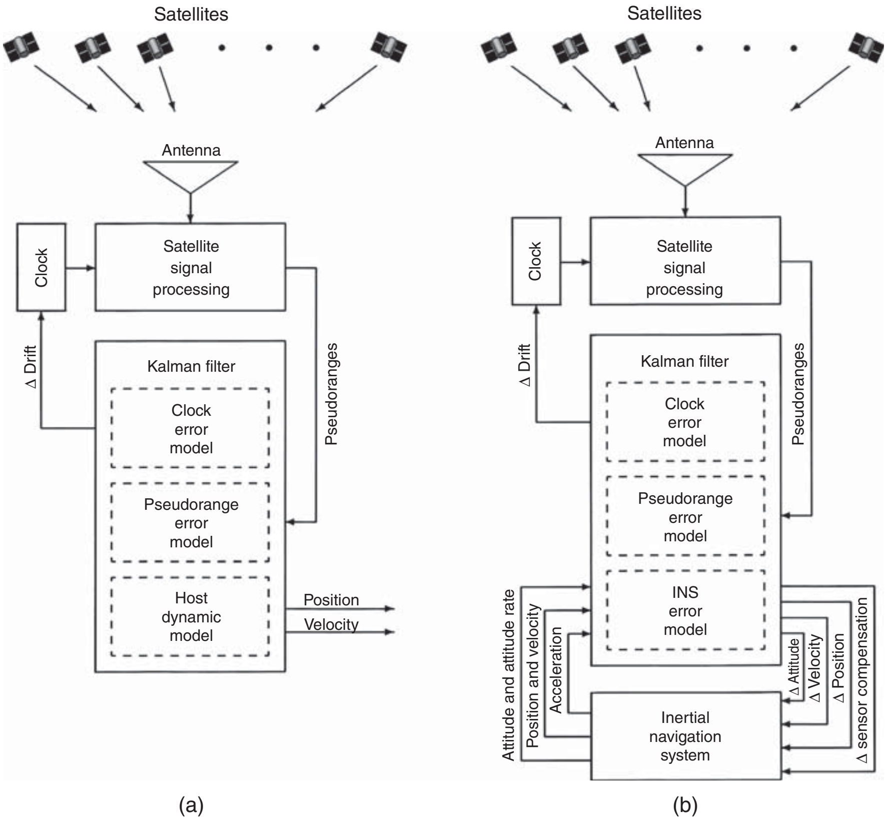


Figure 10.1 Kalman filter architectures for (a) GNSS and (b) GNSS/INS.

but we will derive and demonstrate Kalman filter models to show how it is done. The accompanying software also generates results beyond what can be shown here, just to show the enormous capabilities of the Kalman filtering approach to navigation.

10.2 NAVIGATION OVERVIEW

10.2.1 The Navigation Problem

The purpose of navigation is to direct the movement of a vehicle (wheeled, legged, waterborne, airborne, or spaceborne) so as to arrive at a given destination. An important part of navigation is determining one's location relative to one's destination, and relative to local features related to travel (roads, canals, shipping lanes, etc.).

The solution to the navigation problem generally requires observations or measurements of some kind and being able to use that information to determine your location relative to your destination. The Kalman filter has played a major role in solving the navigation problem.

There are five basic forms of navigation:

1. Pilotage, which essentially relies on recognizing landmarks to know where you are and how you are oriented. It is older than human kind.
2. Dead reckoning, which relies on knowing where you started from, plus some form of heading information (e.g., magnetic compass) and some estimate of distance traveled. Its implementation originally involved plotting on charts using drafting tools, but this operation is now done in software.
3. Celestial navigation, using time and the angles between local vertical and known celestial objects (e.g., sun, moon, planets, stars) to estimate orientation, latitude, and longitude. It depends on clear viewing conditions, and generally requires specialized instruments such as sextants and chronometers.
4. Radio navigation, which relies on radio-frequency sources with known signal characteristics and known locations. Global Navigation Satellite Systems (GNSS) use beacons on satellites for that purpose.
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.

This chapter is about applications of Kalman filtering in solving the navigation problem, and especially combinations of the last two methods in the list above. The development of low cost receivers for GNSS and low cost micro-electro-mechanical systems (MEMS) technologies for inertial navigation systems (INS) have revolutionized the potential cost/performance ratios for high accuracy navigation.

The focus here will be on the design and implementation of Kalman filters for these applications, including design of models used with the associated Riccati equations for predicting the performance of potential sensor system designs. These include examples of Kalman filter models for GNSS and INS errors, and practical Kalman filter implementation architectures for integrated navigation solutions.

There is yet another level of navigation accuracies required for applications such as surveying. These also use Kalman filtering with much the same approach, but with far more detailed modeling.

10.2.2 History of Inertial and Satellite Navigation

Inertial navigation was developed in the mid-twentieth century for missile guidance and matured during the Cold War as a technology for long-range delivery of nuclear weapons. Initial results were aimed at military applications such as self-contained guidance and control systems for ballistic and cruise missiles, submarines carrying ballistic missiles, military ships, and military aircraft, but the technology soon spread to nonmilitary applications such as commercial aircraft navigation.

Because of its stealth and immunity to countermeasures, inertial navigation is very well suited for ballistic missile guidance. Its time of operation during a missile launch is in the order of several minutes, which does not allow much time for any significant build-up of navigation errors. This has helped in achieving acceptable targeting accuracies using inertial navigation technology.

Submarines had long used gyroscopes to keep track of orientation while submerged, and integrating these with acceleration sensor was a natural progression to self-contained stealthy navigation. However, nuclear-missile-carrying submarines would require submerged operation for months, and this was too long to maintain inertial navigation accuracies sufficient for launching ballistic missiles. Some sort of auxiliary navigation information would be necessary for maintaining sufficient navigation accuracies.

Technology for satellite navigation developed soon after the 1957 launch of the world's first artificial satellite. The US Navy was quick to recognize that this could solve the long-range submarine navigation problem. The world's first satellite navigation system (Transit) would be developed and fielded for that specific purpose. After about a quarter century of military-only operation, it would be succeeded by a GPS satellite navigation system with limited civilian access to one of its signals. The impact on civilian navigation worldwide would be revolutionary, and many other countries would develop their own GNSS and augmentations thereof.

Although military satellite navigation systems were designed from the start for integration with INS, commercial development of civilian integrated GNSS/INS was much slower to develop—in part because the additional cost for inertial navigation[1] may not have been justified for the anticipated gain in performance. Among the first nonmilitary applications were such high payoff applications as automated surface mining and grading, but the lowered cost of inertial technologies sufficient for GNSS integration has expanded the number of potential applications.

10.2.3 GNSS Navigation

Satellite signal processing for GNSS navigation produces "pseudoranges" (imputed distances) between the receiver antenna and those of the satellites in view, the locations of which are known at all times. Using these pseudoranges as measurements, the Kalman filter generates estimates of the position and velocity of the GNSS receiver antenna with respect to Earth-fixed coordinates. To do so, it requires stochastic error models for the receiver clock, GNSS pseudorange data, and host vehicle dynamics. The "host vehicle" in this case could be a spacecraft, aircraft, watercraft, wheeled vehicle, or a pack animal (including humans). Each of these has different dynamic statistics, and the respective statistical characteristics can be exploited to improve navigation performance.

10.2.4 Integrated GNSS/INS Navigation

For navigation with an INS and other aiding sensors (e.g., altimeters, airborne radars, star trackers, or GNSS), the INS is the essential keeper of short-term navigation information and the Kalman filter functions to keep that navigation information as accurate as the noise sources will allow.

In the case of GNSS/INS integration, the filter replaces the stochastic model for unpredictable host vehicle dynamics with one for the INS and uses that model to

[1] In the 1970s, when airlines were first required to carry two inertial navigators for over-water flights, the increased equipment cost was in the order of $100,000 per airplane.

estimate, correct, and compensate for errors in the INS implementation. The Kalman filter inputs from the INS its navigation solution (position, velocity, and attitude) plus additional variables required for implementing the INS error model (acceleration and attitude rate), and outputs to the INS updates for its navigation solution (position, velocity, and attitude) plus updates to the parameters used for compensating inertial sensor errors. The result is a profound improvement in INS performance over the long term, as well as the usual short-term performance expected from inertial navigation. A key feature of the integrated navigator is its ability to maintain short-term accuracy when GNSS signals are not available.

10.2.5 Measures of Navigation Performance

10.2.5.1 The Nautical Mile (NMi) Historically, navigators at sea have used the *nautical mile* as the unit of choice when talking distances, and that notation has been passed along to landlubbers, as well. It was originally defined as the distance at sea level equivalent to one arc-minute of latitude change. This made sense as a unit of navigational uncertainty at a time when Earth was thought to be spherical, latitude was determined by measuring the angle of the Pole Star above the horizon at sea, and a minute of arc was the approximate limiting resolution of optical sighting instruments at sea.

However (as Newton surmised) Earth is not quite spherical in shape. As a consequence, the north–south distance equivalent to a variation of one arc-minute of latitude varies by several meters. As a fix, *Le Système international d'unités* (SI) defined the nautical mile as a derived SI unit, equivalent to 1852 m. This would be about 1.15078 US statute miles or 6076.12 US feet.

10.2.5.2 Circle of Equal Probability (CEP) It is also called *circular error probable* (CEP), defined as the radius of a circle centered at an estimated location on the surface of Earth such that it is equally likely that the true location is either inside or outside to that circle.

It is a useful notion in navigation and targeting, giving a precise meaning to a single number characterizing accuracy.

Its implementation in Kalman filtering can be troublesome, however, because one cannot translate estimated position error covariances into CEP without assuming something more about the probability distribution than its mean and covariance. The problem is commonly handled by assuming the probability distributions are Gaussian and approximating CEP as 1.2 times the root-mean-square (RMS) radial horizontal error.

10.2.6 Performance Prediction in Navigation System Design

10.2.6.1 Covariance Analysis The Kalman filter serves a dual role in estimation technology, in that it is not only an optimal estimator, but also serves to predict performance based on the dynamics of the variables being estimated and the statistical characteristics of the sensor system to be used for estimation.

The same Kalman filter models developed for navigation implementation may also be used to design a navigation system to be used on an ensemble of trajectories of the host vehicle carrying the sensors. The same general approach applies to the design of navigation satellite systems, inertial navigators, and integrated GNSS/INS navigation systems.

In the first case (GNSS), the Kalman filter not only functions as a design evaluation tool but also plays an essential role in system integration.

In the second case (INS), parametric INS models are used to represent the possible choices of inertial sensors in the INS, and representative trajectory simulations can be used for comparing relative performances of the design alternatives.

In the case of GNSS/INS integration, the Kalman filter model can be used to assess the performance trade-offs for alternative INS sensor capabilities. In addition, the design problem for integrated GNSS/INS navigation may include ancillary performance objectives such as

1. Minimizing the cost of the onboard integrated GNSS/INS system.
2. Achieving specified "stand-alone" INS-only performance when GNSS signals are lost.
3. Reduction of GNSS signal reacquisition time.
4. Using INS information to improve GNSS signal lock, especially during severe dynamic maneuvers of the host vehicle or periods with excessive satellite signal interference.

10.2.6.2 Test and Evaluation The covariance analysis used for predictive design must eventually be verified by testing and evaluation under controlled conditions. For systems designed for specific applications, the test conditions generally match the intended applications. There are also formal test and evaluation procedures describing what testing is done, what data is taken, how the data is processed, and what specific performance metrics are to be evaluated. For military aircraft navigation systems, this has been done for more than half a century at the Central Inertial and GPS Test Facility (CIGTF) at Holloman Air Force Base near Alamogordo, New Mexico. The facility includes laboratory equipment for evaluating systems under controlled environmental conditions, precision tracking systems for systems under flight or ground testing, formal test and evaluation procedures for specific military applications, and data processing methods approved for evaluating performance relative to the established military standards.

10.2.6.3 Additional Information Sources For a broader historical background on the subject, see, for example, Reference 1. For broader technical coverage of the subject, see, for example, Reference 2. For more in-depth coverage of inertial navigation, see, for example, References 3–5. For better coverage of GNSS receiver technology, see, for example, Reference 6.

Because GNSS and INS technologies are still evolving at a significant rate, technical journals and periodicals on the subject are also good sources for the latest developments. Searches for "CIGTF" may yield more formal descriptions of test and evaluation procedures for INS, GNSS, and integrated GNSS/INS navigation systems.

10.2.7 Dynamic Simulation for Predicting Navigation Performance

Prior to any laboratory or field testing, a common set of simulated dynamic conditions can be used for performance prediction of GNSS navigation, INS navigation, and integrated GNSS/INS navigation. Performance of any developing navigation system can be assessed by testing and evaluation under the intended operating conditions, including dynamic conditions. Assessing expected performance requires some sort of dynamic simulator to generate the essential inputs to the navigation solution under those conditions. As an example, MATLAB®simulators are derived, encoded in m-files on the Wiley web site, and used in demonstrating navigation performance for GNSS, INS, and integrated GNSS/INS navigators.

10.2.7.1 Stationary Testing Checkout of a navigation system usually begins by running it in a known fixed location. For GNSS navigation, this must include a simulator for the satellite locations over time. Even inertial navigation, which is essentially differential, is usually verified first in the laboratory, before dynamic testing. In all cases, the Kalman filter for this is relatively simple and is used to demonstrate the effects of different error sources under "neat" conditions, without the corrupting effects of host vehicle dynamics. For an INS, this sort of testing is used for verifying gyrocompass alignment and for verifying that the INS and its error model show the same behaviors when initial errors are introduced.

10.2.7.2 Racetrack Simulators The common host vehicle dynamic model used for most simulations is for a host vehicle running at 100 kph average speed on a figure-8 track of various lengths. Two track layouts are illustrated in Figure 10.2 for two specific track lengths, although some versions of the track simulator allow both the track length and vehicle speed to be specified at runtime. The limited position excursions of the track are exploited in designing the stochastic dynamic model for the host vehicle used in the associated Kalman filter for by solving the navigation problem.

A vehicle running at 100 kph hardly qualifies as a "racecar," but integrated GNSS/INS navigation is not uncommon on racecars in televised races on closed circuit tracks—not for letting the drivers know where they are, but for letting the broadcast television system know where each racecar is at all times. Onboard navigation solutions are telemetered to the track television recording system and used to generate on-screen graphics to point out individual cars during the race. The accuracy of the navigation solutions is generally in the order of a meter, which is sufficient for this purpose. The heading and pitch angles of the optical axes of track television cameras are used together with camera lens focal lengths to compute where each racecar would appear on the recorded image, and this information is then used to generate text and pointers to locate and identify selected cars on the image during the race. The host vehicle dynamics in this application can be simplified to a two-dimensional model with along-track and cross-track components. Vehicle altitude will always be a known function of these two location components. We will not use this level of sophistication in our demonstrations, even though the figure-8

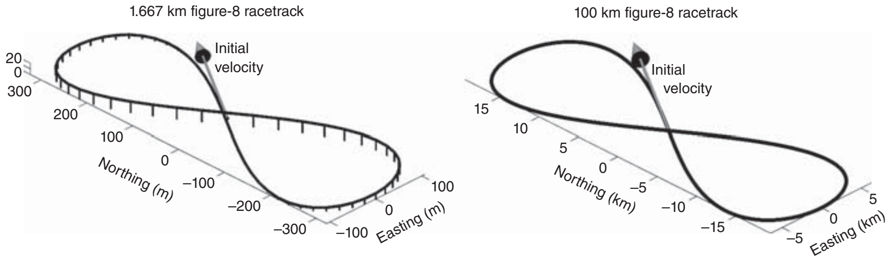


Figure 10.2 Figure-8 track layouts.

track location model uses only along-track position. The m-file `GNSSshootoutNCE.m` demonstrates how this level of modeling can achieve RMS position uncertainties that are 30−50 times smaller than those from using more conventional host vehicle dynamic models.

Statistics of Racetrack Dynamics Average vehicle dynamic statistics during figure-8 track simulations have the values shown in Table 10.1 for the track layouts shown in Figure 10.2. To provide empirical values of dynamic disturbance covariance Q in the Kalman filter models for GNSS navigation, the RMS velocity changes were sampled at the GNSS navigation update intervals on track simulations.

These statistics are matched to the parameters of the stochastic dynamic models used for representing host vehicle dynamics during GNSS navigation.

10.2.7.3 Simulating GNSS Signal Loss GNSS signal availability depends on having an unobstructed line of sight from the transmitting satellite to the receiver antenna and no signal interference at the receiver antenna from sources such as unintended (or intended) interference or multipath reflections. These conditions can fail for a number of reasons. In most cases, signal interruptions are only temporary, however. Still, when GNSS signals are lost, the navigation solution can deteriorate rapidly, with the rate of deterioration generally depending on the relative predictability of host vehicle dynamics. Simulated test conditions for evaluating performance degradation after loss of signal used a modified figure-8 test track simulator with a tunnel covering just a portion of the vehicle trajectory, as illustrated in Figure 10.3. The simulator assumes 100-kph host vehicle speed around a 100-km figure-8 track, which equates to a lap time of 1 h. The tunnel is assumed to cover the track for 1 min of travel, ending 1 min before the undercrossing.

10.3 GLOBAL NAVIGATION SATELLITE SYSTEMS (GNSS)

10.3.1 Historical Background

As we have mentioned in the previous section and in Chapter 1, the beginnings of practical satellite navigation technology occurred soon after America's original

TABLE 10.1 Dynamic Statistics on Figure-8 Track Simulators

FIG8 Track[*]		BIG8 Track	
Statistic	Value	Value	Units
Track length	1.667	100.0	km
Average Speed	100.0	100.0	kph
RMS N−S Position Excursion	230.2564	13,699.7706	m
RMS E−W Position Excursion	76.7521	4,566.5902	m
RMS Vertical. Position Excursion	12.3508	6.1237	m
RMS N−S Velocity	24.3125	23.9139	m/s
RMS E−W Velocity	16.2083	15.9426	m/s
RMS Vertical Velocity	0.74673	0.0061707	m/s
RMS N−S Acceleration	2.525	0.041732	m/s/s
RMS E−W Acceleration	3.3667	0.055643	m/s/s
RMS Vertical Acceleration	0.078846	1.0771_{10}-5	m/s/s
RMS Delta Velocity North[†]	2.5133	0.041732	m/s/s
RMS Delta Velocity East[†]	3.3464	0.055642	m/s/s
RMS Delta Velocity Up[†]	0.077832	1.077_{10}-5	m/s/s
RMS Roll Rate	0.0092193	‡	rad/s
RMS Pitch Rate	0.06351	‡	rad/s
RMS Yaw Rate	0.15311	‡	rad/s
RMS Delta Roll Rate[†]	0.0022114	‡	rad/s/s
RMS Delta Pitch Rate[†]	0.010502	‡	rad/s/s
RMS Delta Yaw Rate[†]	0.023113	‡	rad/s/s
North Position Correlation Time	13.4097	735.6834	s
East Position Correlation Time	7.6696	354.155	s
Vertical Position Correlation Time	9.6786	1,661.1583	s
North Velocity Correlation Time	9.6786	635.1474	s
East Velocity Correlation Time	21.4921	354.155	s
Vertical Velocity Correlation Time	13.4097	735.6834	s
North Acceleration Correlation Time	13.4097	735.6834	s
East Acceleration Correlation Time	7.6696	354.155	s
Vertical Acceleration Correlation Time	9.6786	635.1474	s

∗ Values shown are for strapdown version `Fig8TrackSimRPY.m`.
Track length and speed can be varied through input variables of other versions.

† Sampled at 1-s intervals.

‡ Not computed for gimbaled INS simulation.

“Sputnik moment,” the October 4, 1957 launch of the world’s first artificial satellite[2] by the former Soviet Union. This soon lead to the development by the US Navy of the Transit satellite navigation system, the primary function of which would be as an aid in improving inertial navigation capabilities of military systems—especially for nuclear-missile-carrying submarines. However, due to demands of military secrecy and security, its capabilities would not be made available to the general public. This restriction was eased somewhat in the 1980s for Transit’s eventual replacement, the

[2] “Sputnik” is a transliteration of the Russian word for satellite.

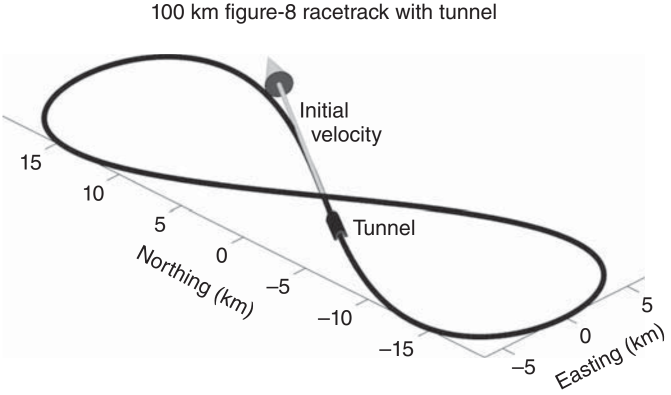


Figure 10.3 Simulator for loss of signal.

Global Positioning System (GPS). On September 1, 1983, inertial navigation errors aboard Korean Air Lines Flight 007 caused it to pass briefly through Soviet airspace, which ended in its being destroyed in international airspace by military aircraft of the former Soviet Union. After that, US President Ronald Reagan signed a directive requiring that the then-developing GPS satellite navigation system be accessible by the general public. The result would revolutionize general navigation and lead to similar efforts by other countries and agencies. To date (2014), the only completely independent functioning alternative to GPS is the Russian GLONASS[3] satellite navigation system, but the European Union (EU) and European Space Agency (ESA) are planning to reach operational status with their own system (Galileo) in 2019, and China plans on expanding its developing regional navigation system (Beidou) into a global system (Compass) by 2020. In addition, France, Japan, and India are developing their own regional augmentation systems.

These systems generally contain ground-based assets (ground segment) integrated with space-based assets (space segment) using Kalman filtering. These Kalman filters are an integral part of the GNSS software infrastructure, but there is also a need for independent Kalman filters in the associated GNSS receivers, for estimating the navigation solution (including the receiver antenna location) based on the satellite “signals in space.”

This section is about receiver-based Kalman filter architectures for obtaining the GNSS navigation solution. There is far more diversity in Kalman filter designs at the receiver level, because performance at that level depends on specifics of the intended application(s).

10.3.2 How Satellite Navigation Works

10.3.2.1 Doppler-based Solution As described in Chapter 1, the Transit navigation satellite system used as its basic measurement variables the Doppler shifts in

[3] Except for an 11-h outage in April Fool’s Day of 2014.

satellite signals as the satellites passed overhead. Its navigation solution method used linearized least-squares fitting (later Kalman filtering) of the antenna position to an observed sequence of Doppler shifts from a satellite with known trajectory. Transit was originally designed for military applications in which the receiver antenna was essentially stationary during the few minutes required for a satellite to pass from horizon to horizon. In the 1970s and 1980s, Transit and Timation (another US Navy satellite system) also served as a testbeds in the development of extremely accurate ground- and space-based clock technologies[4] for the next generation of satellite navigation systems. That would be the US Air Force's GPS, which became operational around the time Transit was ready for retirement in the 1990s. All satellite navigation systems developed thereafter are based on accurate timing technologies.

10.3.2.2 Timing-Based Solution After sufficient timing capabilities had been developed, satellite navigation solutions became "timing based." That is, they use a system of synchronized clocks to measure the time it takes an electromagnetic signal to pass from the transmitting antenna on a satellite to the receiving antenna. Given the transit time Δt and the speed of electromagnetic wave propagation c, this gives a measure of the distance $\rho = c\ \Delta t$ between the two antennas. If the locations of three transmitting antennas on three satellites are known as *functions of time*, this would allow a solution for the location of the receiver antenna, as illustrated in Figure 10.4. That is, given the location in space of the three satellites forming the triangular base of the tetrahedron formed by the four antennas and the three legs ρ_i, there is a unique solution for the location of the receiver antenna. In practice, however, more than three satellites are needed for maintaining clock synchronization.

10.3.3 GNSS Error Sources

A Kalman filter needs error models for all the measurements it uses. For GNSS receivers, the basic measurements are propagation delay times used for computing the distances between the satellite antennas and the receiver antenna. The associated measurement error models have been determined from theoretical and empirical analysis of the GNSS system as it developed.

10.3.3.1 Space Segment Errors A satellite navigation system is infected with all sorts of error sources that have been partially cured by Kalman filtering , a side benefit of which is that the resulting RMS errors of the various contributors to navigation

[4] This was not the first use of accurate timing for navigation. The need for accurate timing to determine longitude at sea was well known as far back as the seventeenth century, when the pendulum clocks of the day would not function at sea. England was then losing more of its naval ships from grounding than from combat, and the disastrous grounding of four Navy ships on the Isles of Scilly in 1704 only emphasized the importance of a speedy solution. After the formation of the United Kingdom in 1707, its Parliament passed the Longitude Act of 1714, establishing substantial prizes to be awarded by Commissioners for the Discovery of the Longitude at Sea (1714–1828) for specific levels of timing accuracy and reliability. The problem was eventually solved by the world's first chronometers, designed by John Harrison (1693–1776). However, awarding of any prizes was thwarted by Nevil Maskeline (1732–1811) and the Commissioners until King George III (1738–1820) interceded on Harrison's behalf [7]

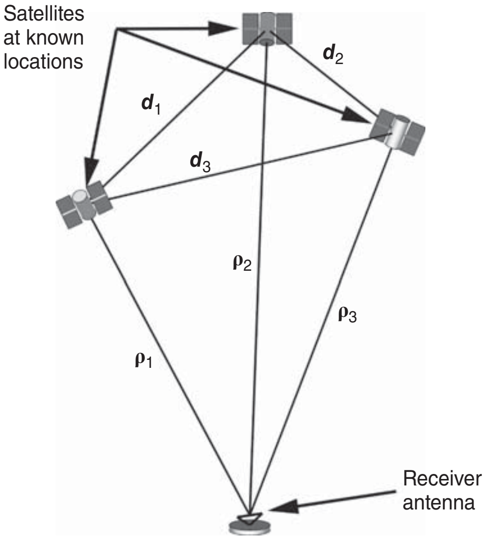


Figure 10.4 Minimal timing-based satellite navigation solution geometry.

errors can be determined from covariance analysis, using the associated solutions of the Riccati equation. For GPS, for example, location errors due to the precise ephemerides (trajectories) of individual satellites (broadcast by all satellites) is in the order of 2 m RMS, as is the error contributed by synchronization of all satellite-based clocks. The current GLONASS K-class space segment performance is comparable.

10.3.3.2 Propagation Delay Errors From the GNSS user's standpoint, the biggest error sources are due to variations in signal propagation delays between the satellite antennas and the receiver's antenna. These are dominated by the effects of free electron density in the ionosphere, which varies considerably with solar radiation and space weather. The effect of this variation on the apparent transmission distance is in the order of 10 m RMS—but this can vary up and down by an order of magnitude, depending on levels of solar activity [8]. The dynamics of propagation delay errors at the receiver are also driven by the motions of the satellites in orbit, which drags the path of the signal from satellite to receiver through different parts of the ionosphere over time.

GPS receivers use different methods for mitigating the effects of these errors:

1. Global correction formulas based on tomographic models of the atmosphere, and using measurements made at ground stations at many locations. These stations know where they are, so they can estimate the ionospheric delay along the paths between the station antennas and satellite antennas. There are several such formulas used in existing or proposed augmentation systems associated with different GNSS systems. A formula due to Klobuchar [9] is included in most single-frequency GPS receivers, and it uses parameters broadcast as part

of the GPS signal. This can reduce the contribution to the user location error by a factor of about 2 or 3, but even this can be improved upon by augmenting the state vector of the Kalman filter in the receiver used for estimating the navigation solution. The augmenting state variables in this case will be the residual errors from the Klobuchar model corrections, which are time-correlated with effective correlation times are in the order of minutes to hours [10].

2. A dual-frequency correction method is based on the known relationship between primary signal delay and differential delay between the two different gigahertz carrier frequencies separated by several hundred megahertz. This feature was originally included in GPS as a military-only option, which required a decryption key to access the timing information on the second frequency. However, current GPS upgrades and other GNSS[5] systems include unencrypted civilian signal frequencies (L2C and L5 for GPS) that serve the same purpose. This approach reduces the ionospheric delay contribution to something in the order of 1 m RMS, which is small enough that the residual ionospheric propagation delay errors may not need to be estimated using the receiver's Kalman filter.
3. Differential GNSS navigation using propagation delay values calculated at local ground-fixed receiver stations with known locations and transmitted on a separate channel for use by local GNSS receivers. The US Coast Guard inaugurated such a service for GPS around major waterways and ports in the United States in the 1990s. This nulled out the "Selective Availability" measures the US Department of Defense had designed into the GPS civilian channels to derate RMS navigation performance to around 100 m RMS. Similar services were designed into various Wide Area Augmentation Systems (WAAS) by the United States and other countries. Receivers equipped for differential GPS navigation could then achieve RMS horizontal navigation accuracies of 5 m or less. Selective Availability was removed from GPS in 2000.

There is also a tropospheric propagation delay error and errors due to atmospheric refraction, but their contributions are generally less than 1 m RMS.

Propagation delays also depend on the atmospheric propagation path length, which depends on the slant distance through the atmosphere from the satellite to the receiver antenna. This can make signal propagation delay errors from satellites nearer to the horizon two to three times what they are for satellites directly overhead.

10.3.3.3 Receiver Clock Errors Clocks are among the most analyzed and improved devices we make and use every day, and their behavior over time is fairly well understood and modeled. GNSS receivers can get by with relatively inexpensive quartz resonator clocks, using timing from satellite- and ground-based "atomic clocks" to maintain GNSS system-wide synchronization. This approach

[5] GLONASS is in the process of changing over from a frequency-division multiple access (FDMA) protocol to a code-division multiple access (CDMA) protocol with a common carrier frequency and different spreading codes for different satellites. GPS had been designed from the start to use CDMA with "Gold codes" (designed by Robert Gold) to minimize cross-interference of the spread-spectrum signals.

takes advantage of the superior short-time stability of quartz clocks. An essential part of this is a linear stochastic process model for a receiver clock phase and frequency error and their respective uncertainties as functions of time. The problem is solved by making the clock *bias* (phase error) and *drift* (frequency error) part of the navigation solution.

10.3.3.4 Dilution of Precision (DOP) The minimal essential state variables for a single GNSS location solution are three components of receiver antenna position, plus receiver clock error. An estimate of clock error is essential because it is too big to be ignored. Other "nuisance variables" may be estimated to accommodate other time-correlated error sources, but these four are the minimal set necessary for obtaining a single location solution. How well these four variables can be estimated will be determined by timing measurement noise and satellite geometry. The term *dilution of precision* (DOP) was defined to characterize how relative beacon locations influence LORAN navigation accuracy, and it has been extended to characterize how satellite location geometry affects GNSS navigation accuracy. In both cases, it can be defined in terms of the related measurement information matrices.

GNSS Timing Measurement Sensitivities and Information Matrices As described in Example 8.7, the measurement sensitivity matrix for the timing variation with respect to receiver position is a unit vector u_j in the direction from the receiver antenna to the jth satellite. If clock error is defined as phase lag and multiplied by c to make it an equivalent distance, then the measurement sensitivity matrix for position and clock error for a single timing measurement is

$$H_j = \begin{bmatrix} u_j^{\mathrm{T}} & -1 \end{bmatrix}, \tag{10.1}$$

the measurement sensitivity matrix for N such measurements is

$$H = \begin{bmatrix} u_1^{\mathrm{T}} & -1 \\ u_2^{\mathrm{T}} & -1 \\ u_3^{\mathrm{T}} & -1 \\ \vdots & \vdots \\ u_N^{\mathrm{T}} & -1 \end{bmatrix}, \tag{10.2}$$

and the associated information matrix for the N measurements is

$$Y = H^{\mathrm{T}} R^{-1} H \tag{10.3}$$

$$= R^{-1} \sum_{j=1}^{N} \begin{bmatrix} u_j \\ -1 \end{bmatrix} \begin{bmatrix} u_j \\ -1 \end{bmatrix}^{\mathrm{T}}, \tag{10.4}$$

where the scalar parameter R is the mean-squared measurement error.

Position and Clock Error Estimation Covariance The resulting covariance matrix of state estimation uncertainty will be the inverse of the associated information matrix

$$P = Y^{-1} \tag{10.5}$$

$$= R\mathcal{D} \tag{10.6}$$

$$\mathcal{D} = \left(\sum_{j=1}^{N} \begin{bmatrix} u_j \\ -1 \end{bmatrix} \begin{bmatrix} u_j \\ -1 \end{bmatrix}^{\mathrm{T}} \right)^{-1} \tag{10.7}$$

$$= \begin{bmatrix} d_{11} & d_{12} & d_{13} & d_{14} \\ d_{21} & d_{22} & d_{23} & d_{24} \\ d_{31} & d_{32} & d_{33} & d_{34} \\ d_{41} & d_{42} & d_{43} & d_{44} \end{bmatrix}. \tag{10.8}$$

The 4 × 4 matrix $\mathcal{D}$ represents the multiplying factor between mean-squared measurement noise R and mean-squared uncertainty of estimated location and clock error due to satellite geometry.

The term *dilution of precision* (DOP) refers to the related multiplying effect of RMS (as opposed to mean-squared) measurement errors on RMS estimation errors due to satellite geometry. The overall effect is called *geometric dilution of precision*, and it is characterized by the square root of the matrix trace of $\mathcal{D}$.

Better yet, if the unit vectors u_j to the respective satellites used in the estimation are expressed in east-north-up coordinates, the square roots of the successive diagonal elements of $\mathcal{D}$ characterize the DOP with respect to east position, north position, up position (vertical), and clock error uncertainties. This convention leads to various "sub-DOPs" defined in terms of the respective diagonal elements of $\mathcal{D}$ as

$$\left.\begin{array}{lll} \text{GDOP} & \overset{\text{def}}{=} \sqrt{\operatorname{tr}(\mathcal{D})} & \text{(Geometric DOP)} \\ & \overset{\text{def}}{=} \sqrt{d_{11}+d_{22}+d_{33}+d_{44}} & \\ \text{PDOP} & \overset{\text{def}}{=} \sqrt{d_{11}+d_{22}+d_{33}} & \text{(Position DOP)} \\ \text{HDOP} & \overset{\text{def}}{=} \sqrt{d_{11}+d_{22}} & \text{(Horizontal DOP)} \\ \text{VDOP} & \overset{\text{def}}{=} \sqrt{d_{33}} & \text{(Vertical DOP)} \\ \text{TDOP} & \overset{\text{def}}{=} \sqrt{d_{44}} & \text{(Time DOP)} \end{array}\right\} \tag{10.9}$$

10.3.3.5 Multipath Effects "Multipath" is the name used for GNSS signal distortion from signal reflections off surfaces such as the ground or sea surface. Antenna gain patterns of many GNSS receivers are designed to attenuate signals on or near the horizon, but this will not correct for reflections off mountains, hills, buildings, or vehicle parts protruding above the antenna. This problem can be especially severe in "urban canyons" lined with flat-faced buildings, and several signal processing methods have been developed for contending with the problem [2].

It is not usually treated as a Kalman filtering problem, but as a problem to be solved in the signal processing “upstream” of the Kalman filter used for obtaining the navigation solution.

10.3.3.6 Other Receiver Errors

Receiver Noise Most GNSS receivers have been designed so that all the error contributions from receiver signal processing generally amount to less than 1-m RMS position error.

10.3.4 GNSS Navigation Error Modeling

Navigation errors not tamed by the Kalman filters in the ground segment or space segment of a GNSS system must be tamed by a Kalman filter in the receiver. The Kalman filters in the ground and space segments have already been designed and implemented before a GNSS becomes operational. This section is about those receiver-based models used for navigation and also used for integrating GNSS receivers with other navigation sensors.

10.3.4.1 GNSS Navigation Measurement Model

Pseudoranges The tetrahedral solution shown in Figure 10.4 would work just fine if all the necessary information were error-free—which it is not. Even if the locations of the satellites were know precisely (which is almost true), the ranges from the receiver antenna to satellite antennas determined by timing are called *pseudoranges*, because they do contain errors. The major error sources requiring correction can be modeled in terms of their effects on pseudorange errors.

The real situation is more like that shown in Figure 10.5, where all available satellites above the horizon (shown connected to the receiver by solid lines) are used in the solution, and the actual distances to the respective satellites are more like

$$\begin{aligned}\rho_j = c \times (&t_{j,\text{received}} - t_{j,\text{transmitted}} - \delta t_{j,\text{iono delay modeled}}\\ &-\delta t_{j,\text{iono delay unmmodeled}} - \delta t_{\text{receiver clock bias}}),\end{aligned} \tag{10.10}$$

where the variables are defined as

ρ_j, the physical distance between the antenna on the jth satellite at time $t_{j,\text{transmitted}}$ and the antenna on the GNSS receiver at time $t_{j,\text{received}}$.

$c \overset{\text{def}}{=} 299,792,458$, the speed of light in SI units.

$t_{j,\text{received}}$, the time at which the timing mark on the jth satellite signal was received at the receiver antenna.

$t_{j,\text{transmitted}}$, the time at which the timing mark on the jth satellite signal was transmitted from the jth satellite antenna.

$\delta t_{j,\text{iono delay modeled}}$, the modeled mean propagation delay over the interval $[t_{j,\text{transmitted}}, t_{j,\text{received}}]$, based on

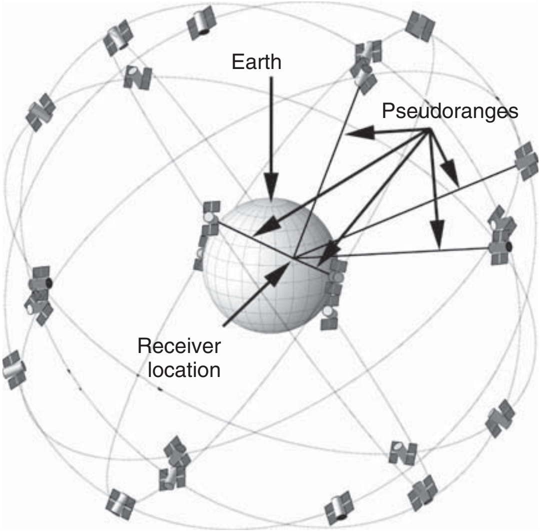


Figure 10.5 Satellite navigation geometry.

(a) a Klobuchar parametric model with parameter values transmitted by the satellites. These parameters are determined in real time from timing measurements made by many receiver ground stations covering much of the land surface, and the resulting model removes most of the ionospheric delay (but not enough),

(b) a formula using signals from the same satellite at two different frequencies, or

(c) differential delay corrections determined at a local auxiliary GNSS receiver at a known fixed location, and broadcast to local GNSS receivers.

$\delta t_{j,\text{iono delay unmmodeled}}$, the unmodeled propagation delay due to limitations of a parametric Klobuchar-like model. This error term may be ignored for the dual-frequency and differential approaches, which may correct enough of the delay that secondary filtering is not necessary.

$\delta t_{\text{receiver clock bias}}$, the bias in the receiver clock at the time the signal was received.

In practice, the last two of these (unmodeled ionospheric delay and receiver clock bias) can be estimated in the same Kalman filter used for estimating the receiver antenna location, and the scaling by c is used to convert all timing error variables to equivalent distance variables. All the other variables are already known from other sources.

This leaves some unknown random variables that require stochastic dynamic models for their time correlation and will add to the number of state variables to be estimated in the Kalman filter.

10.3.4.2 Measurement Sensitivity Matrix Structure The GNSS measurement is a pseudorange with added error variables that corrupt the measurement, but these can also be estimated and compensated for. These error variables include clock errors, and possibly uncompensated ionospheric delay errors (depending on what mitigation is used).

Distributing the factor of c over the right-hand side of Equation 10.10 yields a measurement function in distance units:

$$\begin{aligned}\rho_j = {} & \underbrace{c\ (t_{j,\text{received}} - t_{j,\text{transmitted}} - c\ \delta t_{j,\text{iono delay modeled}})}_{\rho_{0,j}} \\ & \underbrace{-c\ \delta t_{j,\text{iono delay unmmodeled}}}_{\delta_{\text{iono},j}} \underbrace{-\ c\ \delta t_{\text{receiver clock bias}}}_{C_b}, \end{aligned} \tag{10.11}$$

where C_b is the error in the pseudorange measurement attributable to clock bias, $\delta_{\text{iono},j}$ is the error attributable to uncompensated ionospheric propagation delay, and $\rho_{0,j}$ is the (relatively) error-free measurement component—attributable to the physical distance between the satellite antenna and the receiver antenna.

The pseudorange error $\delta_{\text{iono},j}$ due to uncompensated ionospheric delay is a "nuisance variable," in that we need it to improve navigation accuracy, but otherwise do not care what its value is.

Clock bias C_b is a little different, in that it is a variable used in compensating all timing calculations. It is also part of a model used for compensating clock frequency, which is also an essential part of the receiver implementation.

The associated measurement sensitivity to the five state variables (three antenna position components, clock bias, and uncompensated ionospheric delay error) is then

$$H_j \overset{\text{def}}{=} \frac{\partial \rho_j}{\partial x_{\text{antenna}}, \ \dots\ , C_b, \ \dots\ , \delta_{\text{iono},j}} \tag{10.12}$$

$$= \begin{bmatrix} u_j^{\text{T}} & \cdots & -1 & \cdots & -1 & \cdots \end{bmatrix}, \tag{10.13}$$

where the unspecified elements of the row vector H_j are zeros and

x_{antenna} is the location of the receiver antenna.

u_j is a unit vector pointing to the satellite from the receiver antenna.

the first "-1" represents the sensitivity to receiver clock bias.

the second "-1" represents the sensitivity to uncompensated propagation delay for the jth satellite.

10.3.4.3 Dynamic Model Structure These models are all defined by the resulting values of the matrices F (or Φ), Q, H, and R to be used in the Kalman filter.

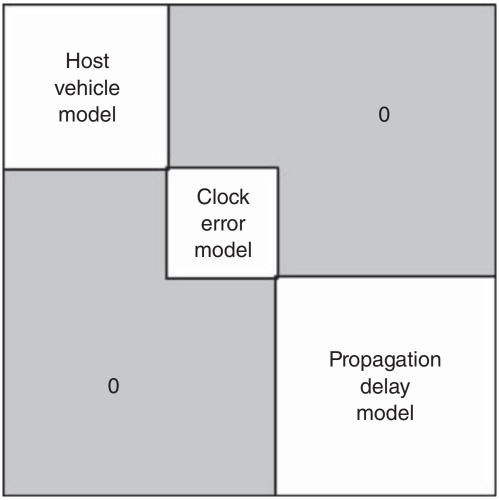


Figure 10.6 Structure of Kalman filter model F-matrix and Q-matrix for GNSS navigation.

The structure of the F and Q matrix will be as shown in Figure 10.6. This block-diagonal structure is useful in practice, because

$$\Phi_{k-1} = \exp \left[\int_{t_{k-1}}^{t_k} F(s)\, dx \right] \tag{10.14}$$

will have the same block diagonal structure as F, with the diagonal subblocks equal to the matrix exponentials of the relevant subblocks of F. This allows the filter designer to treat the dynamic submodels independently, taking exponentials of the time-invariant submatrices only once.

10.3.4.4 Exponentials of Block-Diagonal Matrices There are useful ways to reduce the computational effort required for taking exponentials of matrices with block-diagonal structure such as that shown in Figure 10.6—especially when only a proper subset of the submatrices are time varying. For any matrix with block-diagonal structure

$$M = \begin{bmatrix} M_{11} & 0 & \cdots & 0 \\ 0 & M_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & M_{NN} \end{bmatrix}, \tag{10.15}$$

$$\exp\ (M) = \begin{bmatrix} \exp\ \left(M_{11}\right) & 0 & \cdots & 0 \\ 0 & \exp\ (M_{22}) & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \exp\ (M_{NN}) \end{bmatrix}. \tag{10.16}$$

The number of scalar arithmetic operations required for computation of matrix exponentials grows as the cube of the matrix dimension, so partitioning the matrix in this way saves considerable time and effort.

If the matrix M in this case is the dynamic coefficient matrix F for a linear dynamic system, the associated state-transition matrix is defined by Equation 10.14. Consequently, if any diagonal block F_{ii} of F is time invariant, its exponential

$$\Phi_{(k-1)\ ii} = \exp\ \left(\int_{t_{k-1}}^{t_k} F_{ii}\ dt\right) \tag{10.17}$$

$$= \exp\ (F_{ii}\ \Delta t) \tag{10.18}$$

is also time invariant.

This will be the case for all diagonal submatrices in the dynamic coefficient matrices used for solving the GNSS navigation problem. However, it will not be the case when we get to integrating GNSS with INS. In that case, only the time-varying blocks associated with inertial navigation error propagation need to be recomputed as a function of time. The blocks with time invariant F_{ii} can be left unmolested.

Dimensions of these diagonal subblocks will be determined by which of several possible models is chosen, as will the dimensions of the H and R matrices.

10.3.4.5 Host Vehicle Dynamic Models Timing-based GNSS navigation is defined in terms of having a certain receiver antenna location at the time a satellite signal timing mark is received, and the timing-based navigation correction component for that measurement is always in the direction between the satellite and the receiver location. If the antenna were to remain in that same location long enough for similar corrections to be make for all available satellites, then (assuming good GDOP) that location would be observable from the ensemble of measurements. This may not be a problem in surveying applications, for which the receiver antenna is held at a fixed position on Earth. It is a problem for other applications, however.

The GNSS geometry shown in Figure 10.5 takes into account the motion of the satellite antennas, which is determined by the short-term ephemerides transmitted by the satellites. There is no equivalent predetermined trajectory for the receiver antenna.

Tracking Filters The problem gets a bit more complicated for navigation aboard maneuvering vehicles, in that the location of the receiver antenna now becomes an unknown function of time. This is the same sort of problem faced in the early 1950s, when radar systems were being integrated with real-time computers for detecting and tracking aircraft as part of an air defense system [11], and the GNSS navigation solution can use the same types of "tracking filters" developed around that time. The Kalman filter was not available in the 1950s, but today's tracking filters use both the position and velocity of the receiver antenna[6] as state variables in a Kalman

[6] Air defense jargon uses the term *target* for the object being tracked.

filter. In the case of GNSS navigation, performance can be improved if the parameters of the filter are well matched to the dynamics of the host vehicle carrying the receiver antenna. These parameters include the dynamic coefficient matrix F and the mean-squared dynamic disturbance covariance Q.

Mathematical Formulas Table 10.2 lists some of the parametric models used for modeling a single component of the motion of a host vehicle. These are specified in terms of the dynamic coefficient matrix F and disturbance noise covariance Q for a single axis of motion. Depending on the application, the host vehicle dynamic model may have different models for different axes. Surface ships, for example, may assume constant altitude and only estimate the north and east axes of position and velocity.

Model Descriptions The model parameters listed in Table 10.2 include standard deviations σ and correlation time constants τ of position, velocity, acceleration, and jerk (derivative of acceleration). Those parameters labeled as "independent" can be specified by the designer. Those labeled as "dependent" will depend on the values specified for the independent variables. (See Reference 2 for more details.)

TABLE 10.2 Host Vehicle Dynamic Models

Model	model parameters (each axis)		independent	dependet
No.	F	Q	parameters	parameters
1	0	0	none	none
2	$\begin{bmatrix} 0 & 1 \\ -\dfrac{\sigma_{\text{vel}}^2}{\sigma_{\text{pos}}^2} & \dfrac{-2\,\sigma_{\text{vel}}}{\sigma_{\text{pos}}} \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 0 & \dfrac{4\,\sigma_{\text{vel}}^3}{\sigma_{\text{pos}}} \end{bmatrix}$	σ_{pos}^2 σ_{vel}^2	δ (damping) σ_{acc}^2
3	$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 0 & \sigma_{\text{acc}}^2\,\Delta t^2 \end{bmatrix}$	σ_{acc}^2	$\sigma_{\text{pos}}^2 \to \infty$ $\sigma_{\text{vel}}^2 \to \infty$
4	$\begin{bmatrix} 0 & 1 \\ 0 & -1/\tau_{\text{vel}} \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 0 & \sigma_{\text{acc}}^2\,\Delta t^2 \end{bmatrix}$	σ_{vel}^2 τ_{vel}	$\sigma_{\text{pos}}^2 \to \infty$ σ_{acc}^2
5	$\begin{bmatrix} 0 & 1 & 0 \\ 0 & \dfrac{-1}{\tau_{\text{vel}}} & 1 \\ 0 & 0 & \dfrac{-1}{\tau_{\text{acc}}} \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sigma_{\text{jerk}}^2\,\Delta t^2 \end{bmatrix}$	σ_{vel}^2 σ_{acc}^2 τ_{acc}	$\sigma_{\text{pos}}^2 \to \infty$ τ_{vel} $\rho_{\text{vel, acc}}$ σ_{jerk}^2
6	$\begin{bmatrix} \dfrac{-1}{\tau_{\text{pos}}} & 1 & 0 \\ 0 & \dfrac{-1}{\tau_{\text{vel}}} & 1 \\ 0 & 0 & \dfrac{-1}{\tau_{\text{acc}}} \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sigma_{\text{jerk}}^2\,\Delta t^2 \end{bmatrix}$	σ_{pos}^2 σ_{vel}^2 σ_{acc}^2 τ_{acc}	τ_{pos} τ_{vel} $\rho_{\text{pos, vel}}$ $\rho_{\text{pos, acc}}$ $\rho_{\text{vel, acc}}$ σ_{jerk}^2

The choice of a model and the values of its parameters will depend on the dynamic capabilities of the host vehicle and/or its likely trajectories in the navigation problem being considered.

1. *Stationary Model.* The first model in Table 10.2 is for an object fixed to the earth, such as a GNSS antenna at a fixed location. In these cases, the parameters $F = 0$ and $Q = 0$, and the "host vehicle" could be a building. This model would also apply when GNSS is used in surveying for determining the location of a stationary antenna—usually with respect to another antenna at a known, fixed location. This model has no parameters to adjust.
2. *Quasi-Stationary Model.* The second model in Table 10.2 is that for critically damped harmonic motion with random acceleration excitation. It is used as a model for a host vehicle nominally stationary, but with limited dynamic disturbances. This would apply to ships tied up dockside and to aircraft or land vehicles parked during fueling and loading operations. This type of model is used for INS during initial alignment operations on a quasi-stationary host vehicle, when the inertial sensors can detect small uncontrolled disturbances in acceleration and rotation. A similar model may apply to the vertical dynamics of ships at sea.

   The independent parameters in this model are the RMS position excursion σ_{pos} and RMS velocity σ_{vel}, which would ordinarily be determined from empirical data taken onboard the host vehicle during normal operations. These parameters are related to the critical damping factor δ and mean-squared acceleration excitation σ^2_{acc} by the equations

$$\delta = \frac{2\,\sigma_{\text{vel}}}{\sigma_{\text{pos}}} \tag{10.19}$$

$$\sigma^2_{\text{acc}} = \frac{4\,\sigma^3_{\text{vel}}}{\sigma_{\text{pos}}}. \tag{10.20}$$

   The units of σ^2_{acc} in this case are squared acceleration divided by time, which is consistent with the model in continuous time. The equivalent value in discrete time will depend on the discrete time interval Δt.

   This is a good model for the adaptive suspension systems found in aircraft.

   The critically damped vehicle suspension system model can also be generalized to an overdamped suspension system or an underdamped suspension system with known resonant frequency and damping. Similar models can be used for rotational dynamics, which are important for gyrocompass alignment of medium accuracy strapdown inertial navigation systems.
3. *Type 2 Tracker.* The third model in Table 10.2 is one of the most commonly used in navigation. It is used for tracking GNSS receiver phase and frequency error and is often used for tracking position and velocity of a host vehicle.

   The only adjustable parameter is the mean-squared acceleration noise σ^2_{acc}. In the equivalent discrete-time model, a value for disturbance noise covariance can sometimes be determined empirically as the RMS velocity change over the sample interval.

4. *Modified Type 2 Tracker.* The fourth model in Table 10.2 is a refinement of the type 2 tracker for vehicles with bounded velocity capability. These trackers can perform better when GNSS signals are lost.

   The adjustable parameters in this case include the mean-squared velocity and the velocity correlation time, which can often be determined empirically.

5. *Models for Bounded RMS Velocity and Acceleration.* The fifth model is a further refinement for vehicles with bounded velocity and acceleration. These also perform better when signals are lost, because they take into account the true limitations of the host vehicle.

   The parameter values (mean-squared velocity and acceleration, and the acceleration correlation time) can often be determined empirically. Instrumentation for sampling the empirical data may include three-axis accelerometer clusters, or an INS.

6. *Models for Bounded RMS Position.* The last model in Table 10.2 is for vehicles with bounded position, as well. This model may be appropriate for the limited altitude excursions of ships or land vehicles, including vehicles operating within tightly confined areas.

Alternative Control-Based Model There is a related controls model for bounding the RMS position, velocity, and acceleration of a vehicle disturbed by white jerk noise:

$$\frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ f_{3,1} & f_{3,2} & f_{3,3} \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ w_{\text{jerk}}(t) \end{bmatrix} \tag{10.21}$$

$$f_{3,1} = -\frac{\sigma^2_{\text{jerk}}\sigma^2_{\text{vel}}}{2(\sigma^2_{\text{pos}}\sigma^2_{\text{acc}} - \sigma^4_{\text{vel}})} \tag{10.22}$$

$$f_{3,2} = -\frac{\sigma^2_{\text{acc}}}{\sigma^2_{\text{vel}}} \tag{10.23}$$

$$f_{3,3} = -\frac{\sigma^2_{\text{pos}}\sigma^2_{\text{jerk}}}{2(\sigma^2_{\text{pos}}\sigma^2_{\text{acc}} - \sigma^4_{\text{vel}})} \tag{10.24}$$

$$\sigma^2_{\text{jerk}} = \text{E}_t\langle w^2_{\text{jerk}}(t)\rangle \tag{10.25}$$

$= \text{mean-squared jerk noise}$

$\sigma_{\text{pos}} = \text{RMS position excursion}$

$\sigma_{\text{vel}} = \text{RMS velocity}$

$$
\begin{aligned}
\sigma_{\text{acc}} &= \text{RMS acceleration} \\
\sigma_{\text{pos}}^2 \sigma_{\text{acc}}^2 &> \sigma_{\text{vel}}^4. \qquad (10.26)
\end{aligned}
$$

This can be verified by solving the associated steady-state Riccati equation,

$$0 = FP + PF^{\mathrm{T}} + Q \tag{10.27}$$

$$Q = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sigma_{\text{jerk}}^2 \end{bmatrix} \tag{10.28}$$

for P, giving the diagonal elements as the steady-state σ_{pos}^2, σ_{vel}^2 and σ_{acc}^2. The result can be solved for F as a function of σ_{pos}^2, σ_{vel}^2, σ_{acc}^2, and σ_{jerk}^2.

Empirical Modeling For most host vehicles, the best instrumentation for determining the required dynamic model statistics is an INS with a data recorder.

10.3.4.6 Reordering Host Vehicle Model States The dynamic coefficient matrices in Table 10.2 are defined for one coordinate axis only, which allows the host vehicle dynamic modeler to use different models for different degrees of freedom. These have been implemented as m-files which return the resulting F (or Φ), Q, and P_0 matrices for a specific axis model.

The resulting state variable order can be transformed to the more conventional order (e.g., as used in inertial navigation error modeling) by an orthogonal matrix. For example, this transformation in the case of the sixth model in Table 10.2 would be

$$\underbrace{\begin{bmatrix} \varepsilon_E \\ \varepsilon_N \\ \varepsilon_U \\ \dot{\varepsilon}_E \\ \dot{\varepsilon}_N \\ \dot{\varepsilon}_U \\ \ddot{\varepsilon}_E \\ \ddot{\varepsilon}_N \\ \ddot{\varepsilon}_U \end{bmatrix}}_{x_d} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{T} \underbrace{\begin{bmatrix} \varepsilon_E \\ \dot{\varepsilon}_E \\ \ddot{\varepsilon}_E \\ \varepsilon_N \\ \dot{\varepsilon}_N \\ \ddot{\varepsilon}_N \\ \varepsilon_U \\ \dot{\varepsilon}_U \\ \ddot{\varepsilon}_U \\ \end{bmatrix}}_{x_a}, \tag{10.29}$$

where the state variable x_a on the right is ordered first by axis, then by derivative order. The state variable x_d on the left is ordered first by derivative order, then by axis.

The transformation matrix T in this case is symmetric and orthogonal, so that

$$T^{-1} = T^{\mathrm{T}} = T.$$

As a consequence, the F, P, and Q matrices of the two types of models are related by the formulas

$$F_d = TF_aT$$

$$P_d = TP_aT$$

$$Q_d = TQ_qT$$

$$F_a = TF_dT$$

$$P_a = TP_dT$$

$$Q_a = TQ_dT$$

because T is a symmetric orthogonal matrix.

This trick is used in the error analysis m-files to convert the per-axis model parameter matrices generated by the m-file `DAMP3ParamsD.m` (used for the figure-8 track dynamic model) into the more conventional order by derivative degree.

10.3.4.7 Clock Error Models Most receiver clocks are relatively inexpensive quartz clocks which are quite stable over periods of time in the order of 0–10 s. This works well for GNSS receiver applications, provided that receivers can use the timing information from hyperaccurate clocks on the GNSS satellites to maintain the required long-term stability and accuracy of their own clocks. The measurement update period for GNSS receivers is generally in the order of 1 s. This allows the receiver to track its own clock errors relatively accurately, which allows receiver designers to use less-expensive clocks.

Clock Phase and Frequency Tracking The most common receiver clock frequency and phase tracking implementation uses the *type 2 tracker* from Table 10.2 to keep the receiver clock synchronized in phase and syntonized in frequency to GNSS satellite clocks. This is essentially a Kalman filter with two state variables:

C_b, the receiver clock bias (i.e., offset from satellite time). The value in seconds can be scaled by the speed of light c to maintain C_b in distance units (e.g., meters).

C_d, the receiver clock drift rate, or time rate of change of bias. It can also be scaled by c to maintain C_d in velocity units.

Clock Dynamic Model in Continuous Time The clock state vector is then two dimensional, with its dynamic model in continuous time being

$$x = \begin{bmatrix} C_b \\ C_d \end{bmatrix} \tag{10.30}$$

$$\dot{x} = \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}_{F} x + w(t) \tag{10.31}$$

$$w(t) \overset{\text{def}}{=} \begin{bmatrix} w_b(t) \\ w_d(t) \end{bmatrix} \tag{10.32}$$

$$Q_t \overset{\text{def}}{=} \mathrm{E}_t \langle w(t) w^{\mathrm{T}}(t) \rangle \tag{10.33}$$

$$= \begin{bmatrix} q_{tbb} & 0 \\ 0 & q_{tdd} \end{bmatrix}. \tag{10.34}$$

That is, the zero-mean white noise processes $w_b(t)$ and $w_d(t)$ are uncorrelated.

This model is a short-term approximation to what is called *flicker noise* in clocks. The power spectral density of flicker noise as a function of frequency f falls off as $1/f$. This behavior cannot be modeled exactly by linear stochastic differential equations, however.

Clock Dynamic Model in Discrete Time The equivalent model in discrete time will be

$$x_k = \Phi(\Delta t) x_{k-1} + w_{k-1} \tag{10.35}$$

$$\Phi(\Delta t) \overset{\text{def}}{=} \exp\ (F\ \Delta t) \tag{10.36}$$

$$= \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \tag{10.37}$$

$$w_{k-1} \overset{\text{def}}{=} \begin{bmatrix} w_{b,\ k-1} \\ w_{d,\ k-1} \end{bmatrix} \tag{10.38}$$

$$Q_{k-1} \overset{\text{def}}{=} \underset{k}{\mathrm{E}} \langle w_{k-1} w_{k-1}^{\mathrm{T}} \rangle \tag{10.39}$$

$$= \Phi(\Delta t) \left[\int_0^{\Delta t} \Phi^{-1}(s) Q_t(s) \Phi^{-\mathrm{T}}(s)\ ds \right] \Phi^{\mathrm{T}}(\Delta t) \tag{10.40}$$

$$= \begin{bmatrix} q_{tbb}\Delta t + q_{tdd}\Delta t^3/3 & q_{tdd}\Delta t^2/2 \\ q_{tdd}\Delta t^2/2 & q_{tdd}\Delta t \end{bmatrix}, \tag{10.41}$$

where Equation 10.40 is from Equation 4.131, which we use to calculate Q_{k-1} from Q_t.

Covariance Propagation in Discrete Time The covariance propagation between updates in discrete time has the form

$$P_k = \Phi(\Delta t) P_{k-1} \Phi^{\mathrm{T}}(\Delta t) + Q_{k-1} \tag{10.42}$$

$$Q_{k-1} = \begin{bmatrix} q_{bb} & q_{bd} \\ q_{db} & q_{dd} \end{bmatrix} \tag{10.43}$$

$$q_{bb} = q_{tbb}\Delta t + q_{tdd}\Delta t^3/3 \tag{10.44}$$

$$q_{bd} = q_{tdd}\Delta t^2/2 \tag{10.45}$$

$$q_{db} = q_{bd} \tag{10.46}$$

$$q_{dd} = q_{tdd}\Delta t. \tag{10.47}$$

Representative Process Noise Covariance Values The values of mean-squared bias noise q_{bb} and drift noise q_{dd} vary with the quality (and price) of the clock used. A common statistic used for the quality of a clock is its RMS relative frequency stability over a stated time period. We show here a method for calculating Q_t and Q_{k-1} from the quoted RMS relative frequency stability for time period equal to the intersample period (≈1 s). The same method can be used for any RMS relative frequency stability and intersample period.

Example 10.1 (Calculating Clock Model Disturbance Noise) Reasonably low cost quartz crystal clocks have frequency stabilities in the order of 10^{-9} to 10^{-6} parts per part over the time between GPS pseudorange measurements (1 s). That is, for clock frequency f, the RMS incremental change in relative frequency $[f(t) - f(t-1)]/f(t-1)$

$$\frac{\sigma_f}{f} \approx 10^{-9}\text{to}10^{-6}. \tag{10.48}$$

We will use the lower value (10^{-9}) to demonstrate a methodology for translating this number to an equivalent process noise covariance. However, the result scales with the square of the clock stability value, so it can be scaled to any stability figure. For example, the value for $\sigma_f/f = 10^{-7}$ will be

$$Q_{10^{-7}} = \left(\frac{10^{-7}}{10^{-9}}\right)^2 Q_{10^{-9}}, \tag{10.49}$$

where $Q_{10^{-9}}$ is the equivalent process noise covariance for $\sigma_f/f = 10^{-9}$.

If we convert the clock model parameters to velocity units, then

$$q_{dd} = (c\ \sigma_f/f)^2 \tag{10.50}$$

$$\approx (3 \times 10^8 \times 10^{-9})^2 \tag{10.51}$$

$$\approx 0.1(\text{m}^2/\text{s}^2) \tag{10.52}$$

for an intersample period of $\Delta t = 1$ s.

From these values of q_{dd} and Δt, one can solve for the equivalent process noise covariance q_{tdd} in continuous time, as

$$q_{tdd} = q_{dd}/\Delta t \tag{10.53}$$

$$= 0.1(\text{m}^2/\text{s}^3). \tag{10.54}$$

The value of frequency drift variance q_{tdd} depends primarily on the quality of the quartz crystal, its temperature control, and the stability of its associated control electronics. The value of the phase noise variance q_{tbb} depends more on the electronics.

In a “balanced” design, the contributions of both to mean-squared timing errors are about equal. If we assume that the contributions to q_{bb} from q_{tbb} and q_{tdd} are about equal, then

$$q_{tbb}\Delta t \approx q_{tdd}\Delta t^3/3 \tag{10.55}$$

$$q_{tbb} \approx q_{tdd}\Delta t^2/3 \tag{10.56}$$

$$\approx 0.03(\mathrm{m}^2/\mathrm{s}) \tag{10.57}$$

$$Q_t \approx \begin{bmatrix} 0.03 & 0 \\ 0 & 0.10 \end{bmatrix}, \tag{10.58}$$

and

$$q_{bb} = q_{tbb}\Delta t + q_{tdd}\Delta t^3/3 \tag{10.59}$$

$$\approx 0.06(\mathrm{m}^2) \tag{10.60}$$

$$q_{bd} = q_{tdd}\Delta t^2/2 \tag{10.61}$$

$$\approx 0.05(\mathrm{m}^2/\mathrm{s}) \tag{10.62}$$

$$Q_{k-1} \approx \begin{bmatrix} 0.06 & 0.05 \\ \\ 0.05 & 0.10 \end{bmatrix} \tag{10.63}$$

in distance and velocity units.

A plot of clock estimation uncertainties versus clock stability is shown in Figure 10.7 for a stationary receiver with good satellite geometry. Under such ideal conditions, clock stability does not severely compromise location uncertainty, but it does compromise clock syntonization (frequency tracking). This tends to corrupt the navigation solution, as well, when the receiver antenna is moving.

Clock Measurement Sensitivity Matrix The sensitivity of any pseudorange ρ to the clock state vector has the form

$$H_{\text{clock}} = \frac{\partial \rho}{\partial\ C_b,\ C_d} \tag{10.64}$$

$$= \begin{bmatrix} -1 & 0 \end{bmatrix}, \tag{10.65}$$

when C_b has units of distance. That is, a clock bias error ε_b is equivalent to a uniform increase of ε_b meters in *all* pseudoranges simultaneously.

10.3.4.8 Uncompensated Iono Delay Error Models These are not likely to require further compensation for dual-frequency receivers, but they are a dominant error source for single-frequency receivers.

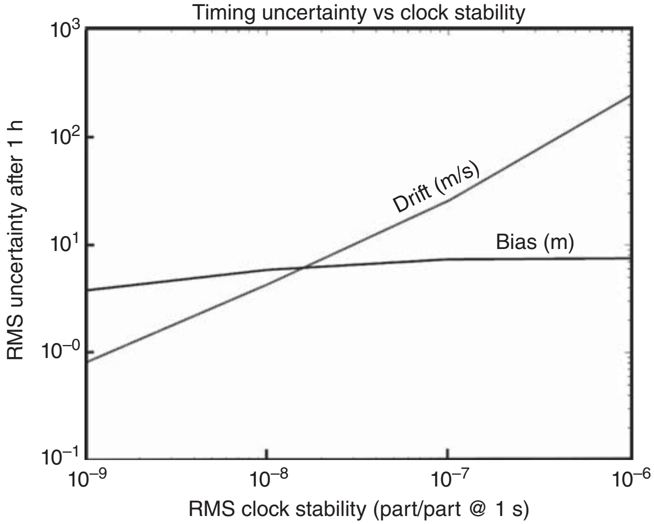


Figure 10.7 Clock bias (synchronization) and drift (syntonization) uncertainties.

Models for Klobuchar Correction Residuals For GPS receivers without two-frequency capability or differential correction, a coarse 3D tomographic model for computing local delay corrections is transmitted as part of the signals from the satellites (see Section 10.3.3.2).

To further reduce the errors due to propagation delays, each receiver can use a Kalman filter to estimate and compensate for the residual time-correlated psuedorange errors in each satellite signal. The delays are approximated as exponentially correlated processes, the model for which in continuous time is

$$\dot{\delta}_{\text{iono},j} = -\delta_{\text{iono},j}/\tau + w(t) \tag{10.66}$$

$$w(t) \in \mathcal{N}(0, Q_t)$$

$$Q_t = 2\sigma^2/\tau \tag{10.67}$$

$$\sigma \approx 10\text{m},$$

$$\tau \approx 60\text{s}.$$

The equivalent pseudorange error model in discrete time is

$$\delta_{\text{iono},j,k(-)} = \Phi\delta_{\text{iono},j,(k-1)(+)} + w_{k-1} \tag{10.68}$$

$$\Phi = \exp\ (-\Delta t/\tau) \tag{10.69}$$

$$\Delta t = \text{discrete time-step}$$

$$\tau \approx 60\text{s}$$

$$\begin{aligned} w_k &\in \mathcal{N}(0, Q) \\ Q &= \sigma^2(1 - \Phi^2) \\ \sigma &\approx 10\text{m}. \end{aligned} \tag{10.70}$$

Measurement Sensitivity Matrix The navigation state vector must be augmented by adding one such state variable for each satellite being used in the navigation solution. This can be implemented by using one such state variable for each GNSS satellite and setting the corresponding row of the measurement sensitivity matrix to zero whenever that satellite is not being used for navigation.

With a little more programming effort, it can also be implemented so that the state vector always has the minimum required dimension. As satellites go out of view, their associated pseudorange error variables can be removed from the state vector. When new satellites are added, their pseudorange error variables can be inserted in the state vector, with associated variances initialized at σ^2. So long as individual satellites remain unused for at least a few correlation time constants, this implementation does not compromise performance.

10.3.5 Performance Evaluations

The real power of Kalman filtering is in its implementations, which provide the greatest benefit from all the available data. A secondary benefit is that its expected performance can be assessed before it is built—using simulations.

We use simulators of the host vehicle dynamics and GNSS satellite motions to evaluate relative performance of different Kalman filter implementations for GNSS. These are described in the following subsections.

10.3.5.1 Host Vehicle Dynamic Simulators

Stationary Receiver Stationary GNSS receivers are often used as clocks. Unless the precise receiver antenna location is already known, it can be estimated along with the navigation solution. The Kalman filter for this is relatively simple and is used to demonstrate the effects of different error sources without the corrupting effects of host vehicle dynamics.

Racetrack Simulators These are described in Section 10.2.7.2.

Racetrack dynamics The statistical parameters used for tuning host vehicle dynamic models of Table 10.2 are summarized in Table 10.1.

10.3.5.2 Efficacy of Vehicle Dynamic Models Figure 10.8 is a plot of simulated GNSS navigation CEPs as a function of time over an 8-h simulation of dynamics on a 100-km figure-8 test track, with GNSS signals being lost for a minute near the end of each hour. The solid line plots the CEP derived from the covariance matrix of estimation uncertainty, which is the solution of a Riccati equation including a stochastic model for host vehicle dynamics. The host vehicle dynamic model is the sixth one in

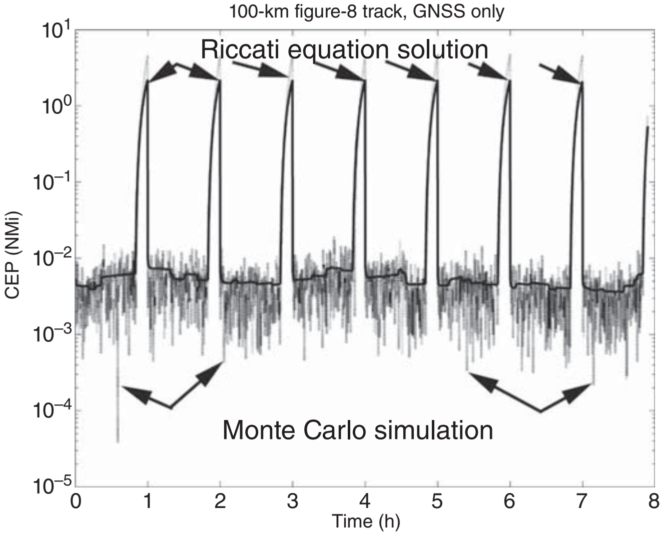


Figure 10.8 Simulated and theoretical performance of GNSS navigation.

Table 10.2, with parameter values taken from Table 10.1. The only model the Kalman filter has for vehicle dynamics is a stochastic one.

The lighter dotted line plots a sample CEP derived from a Monte Carlo simulation with appropriate pseudonoise sources and with the vehicle dynamic conditions dictated by the simulated racetrack geometry.

Agreement between the two solutions is encouraging, given that the Kalman filter model for host vehicle dynamics is based solely on the empirical statistics of the simulator, whereas the Monte Carlo analysis uses the deterministic racetrack dynamic simulator from which the host vehicle stochastic model parameters were derived. In both cases, the CEP values during periods of signal availability are about 10-m RMS.

It is possible to derive a stochastic dynamic model for a vehicle on a test track with only lateral and longitudinal motion, and one would expect better performance with the more accurate dynamic model. However, it is quite remarkable how well a model based on just statistical parameters can do.

10.3.5.3 GNSS Simulation Models

Fixed Satellite Locations These models are used in the MATLAB scripts in `ClockStab.m` (to remove the effects of satellite motion) and `SatelliteGeometry.m` (to demonstrate the influence of satellite geometry on performance) on the companion Wiley web site.

"Real-World" Satellite Motions These GNSS simulations used the GPS satellite configuration from March 2014, as downloaded from the US Coast Guard site

`www.navcen.uscg.gov - /ftp/GPS/almanacs/yuma/.`

There were 30 operational GPS satellites at that time. The MATLAB script `ReadYUMAdata.m` is designed to transform the ASCII text files from that site into MATLAB m-files initializing the satellite simulator. The MATLAB script `YUMAdata.m` was generated by `ReadYUMAdata.m`. It contains the GPS ephemeris information of March 2014, and it produces two global arrays used for GPS simulations. The MATLAB function `HSatSim.m` uses these global arrays to generate pseudorange measurement sensitivity matrices for GNSS receivers at a given latitude, longitude, and time.

Circular Orbit Assumption To simplify the calculations required, the GPS satellite simulator assumes circular orbits. This level of model fidelity is generally permissible in performance analysis, even though it would not be permissible for obtaining the navigation solution.

10.3.6 Quality of Navigation Solutions

In GNSS navigation, the *navigation solution* is an estimate of the location of the antenna of a GNSS receiver.

The “front end” of the GNSS receiver generates the pseudorange measurements to all GNSS satellites being tracked. The digital processors in the “back end” of GNSS receivers use Kalman filters to estimate the navigation solution, given the satellite locations (broadcast in each satellite signal) and the pseudoranges obtained by tracking the satellite signals. The navigation solution always includes the location of the receiver antenna with respect to the earth, but may include the solution for other “nuisance variables” as well. The full navigation solution may include any of the following:

1. The longitude, latitude, and altitude of the receiver antenna with respect to a specified “datum” (reference geoid for the shape of the earth).
2. Universal Time Coordinated (UTC), the international standard time. UTC is referenced to a worldwide collection of atomic clocks, but adjusted occasionally by adding or deleting “leap seconds” to remain phase-locked to the rotation of the earth within $\approx \pm 7.5$ arcsec. But “GNSS time” cannot be subjected to such discrete changes without inflicting the navigation solutions with pseudorange jumps of 299,792,458 m.
3. The receiver clock bias (difference from GNSS clock time), which is generally estimated as part of the navigation solution.
4. The receiver clock frequency error, also generally estimated.
5. The velocity of the host vehicle, which is commonly estimated—unless the receiver antenna is known to be stationary.
6. The acceleration of the host vehicle, which could be of interest in some applications.
7. Time-correlated uncompensated pseudorange errors, due principally to smaller-scale variations in ionospheric delay. The RMS magnitude of these

errors is in the order of 10 m, and the correlation times are in the order of a minute, typically. Estimating these nuisance variables requires one additional Kalman filter state variable for each satellite being used in the navigation solution—which can be in the order of a dozen additional state variables.

The resulting receiver Kalman filter state vector dimension can range from 5 to more than 20, if receiver clock biases and frequencies are always included in the navigation solution. With additional state variables for the propagation delay corrections, the state vector dimension may become 50 or more.

10.3.6.1 Effects of Satellite Geometry The covariance matrix of state estimation uncertainty provides a better measure of GNSS navigation performance than the DOP calculation of Section 10.3.3.4.

The following example uses a simple Kalman filter model with five state variables: three for position and two for clock error corrections. The associated Riccati equation solution is used to show that relative satellite locations influence position uncertainty in the Kalman filter estimate.

Example 10.2 (Effects of Satellite Geometry) This simplified example uses a fixed antenna location and fixed satellite geometry to demonstrate how navigation performance depends on satellite geometry, in terms of estimation uncertainty.

Measured pseudoranges from four satellites are used—the minimum number of satellites required for estimating position and clock errors. The dependence of observability on satellite geometry is demonstrated by using different sets of fixed directions to these four satellites.

The model uses the two-state clock error model from Example 10.4.5 and three antenna position coordinates in locally level coordinates, for a total of five state variables:

$$x^{\mathrm{T}} = \begin{bmatrix} N & E & D & C_b & C_d \end{bmatrix}, \tag{10.71}$$

where

N, north position in meters;

E, east position in meters;

D, downward position in meters;

C_b, receiver clock bias in meters;

C_d, receiver relative clock drift rate in meters per second.

The locally level coordinates $[N,\ E,\ D]$ of antenna position are unknown constants, so that the upper-left 3×3 submatrix of Φ is an identity matrix and the upper-left 3×3 submatrix of Q is zero. The lower right 2×2 submatrix of Q is consistent with a receiver clock with 10^{-8} parts per part RMS frequency stability over $\Delta t = 1$ s. That is,

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6 & 5 \\ 0 & 0 & 0 & 5 & 10 \end{bmatrix} \tag{10.72}$$

$$\Phi = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{10.73}$$

We further assume

- RMS pseudorange measurement error is 15 m,
- Initial RMS antenna position uncertainty is 1 km,
- Initial RMS clock bias is 3 km (10 ms),
- Initial RMS relative frequency uncertainty is 30 m/s (10^{-7} part/part),

so that the covariance matrices

$$R = \begin{bmatrix} 225 & 0 & 0 & 0 \\ 0 & 225 & 0 & 0 \\ 0 & 0 & 225 & 0 \\ 0 & 0 & 0 & 225 \end{bmatrix} \tag{10.74}$$

$$P_0 = \begin{bmatrix} 10^6 & 0 & 0 & 0 & 0 \\ 0 & 10^6 & 0 & 0 & 0 \\ 0 & 0 & 10^6 & 0 & 0 \\ 0 & 0 & 0 & 9 \times 10^6 & 0 \\ 0 & 0 & 0 & 0 & 900 \end{bmatrix}. \tag{10.75}$$

The off-diagonal values of R in Equation 10.74 are zero because there is effectively no correlation between pseudorange errors to different GNSS satellites. Except for receiver clock errors, the pseudorange measurement error mechanisms are effectively statistically independent between one satellite and another. Because the clock errors are part of the state vector, the remaining errors are uncorrelated.

The Kalman filter model in discrete time is completely defined by Φ_k, Q_k, H_k, and R_k, and performance depends on P_0 as well. In this example, H will depend on the directions to the four satellites. If we let the direction to the j^{th} satellite be specified by its azimuth θ_j (measured clockwise from north) and elevation angle ϕ_j (measured upward from the horizon), then

$$H = \begin{bmatrix} -\cos\left(\theta_1\right)\cos\left(\phi_1\right) & -\sin\left(\theta_1\right)\cos\left(\phi_1\right) & \sin\left(\phi_1\right) & 1 & 0 \\ -\cos\left(\theta_2\right)\cos\left(\phi_2\right) & -\sin\left(\theta_2\right)\cos\left(\phi_2\right) & \sin\left(\phi_2\right) & 1 & 0 \\ -\cos\left(\theta_3\right)\cos\left(\phi_3\right) & -\sin\left(\theta_3\right)\cos\left(\phi_3\right) & \sin\left(\phi_3\right) & 1 & 0 \\ -\cos\left(\theta_4\right)\cos\left(\phi_4\right) & -\sin\left(\theta_4\right)\cos\left(\phi_4\right) & \sin\left(\phi_4\right) & 1 & 0 \end{bmatrix}. \qquad (10.76)$$

These equations are programmed in the MATLAB program `SatelliteGeometry.m`, which allows you to enter four directions to GNSS satellites, then computes and plots navigation performance in terms of RMS uncertainties in position (three components) and clock errors (two parameters) versus time for 1 min.

Figures 10.9 and 10.10 were generated using this program. These are plots of the RMS uncertainty in the clock bias (C_b) and clock drift (C_d) versus time. The first of these uses a good satellite geometry, and the second, a bad satellite geometry.

The "good" satellite geometry has three satellites equally space 120° apart in the horizontal plane, which is sufficient by itself to determine horizontal position *and* receiver clock time error. A receiver clock time error is equivalent to lengthening or shortening all pseudoranges the same amount, and the three horizontal pseudoranges are sufficient for determining horizontal position and clock time error simultaneously with this geometry. The fourth satellite is overhead, the pseudorange of which is sensitive only to altitude and clock time error. However, clock time error is already uniquely determinable from the first three pseudoranges.

The "bad" satellite geometry has all four satellites at 45° elevation, equally spaced 90° apart in azimuth. In that configuration, a clock timing error is not distinguishable from an antenna altitude error.

10.3.6.2 Influence of Host Vehicle Models The MATLAB script `GNSSshootout-NCE.m`[7] on the companion Wiley web site compares side-by-side four different host vehicle dynamic models for the same navigation problem, which is a figure-8 track model described in Section 10.2.7 with time-correlated random vehicle velocity variations. The three host vehicle dynamic models are

(a) [MODL3,] which uses model number 3 (type 2 tracker) from Table 10.2 with parameter values from the track simulator statistics. This vehicle model has six state variables.

(b) [MODL5], which uses model number 5 (bounded RMS velocity) from Table 10.2 with parameter values from the track simulator statistics. This vehicle model has six state variables.

(c) [MODL6], which uses model number 6 (bounded RMS position) from Table 10.2 with parameter values from the track simulator statistics. This vehicle model has nine state variables.

[7] The "NCE" stands for "No Clock Errors."

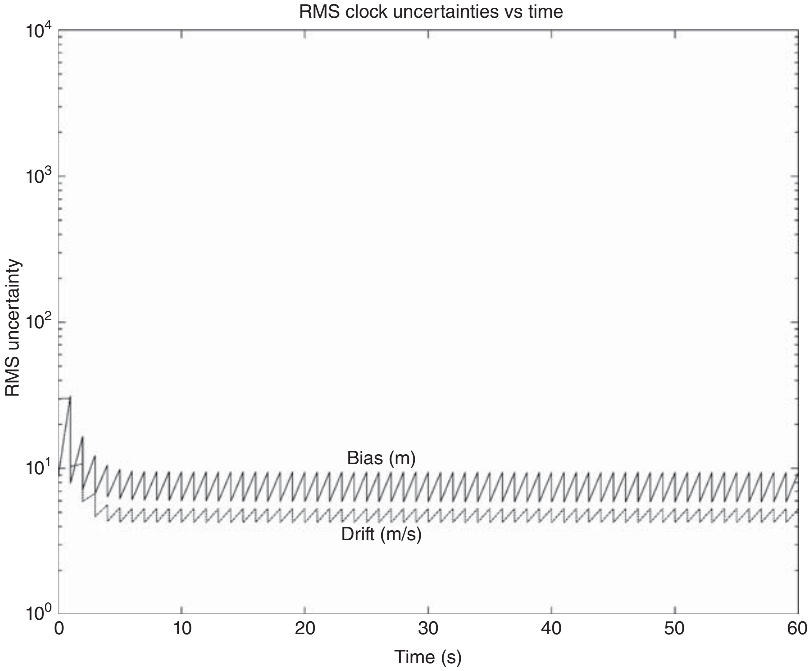


Figure 10.9 Clock parameter uncertainties with good satellite geometry.

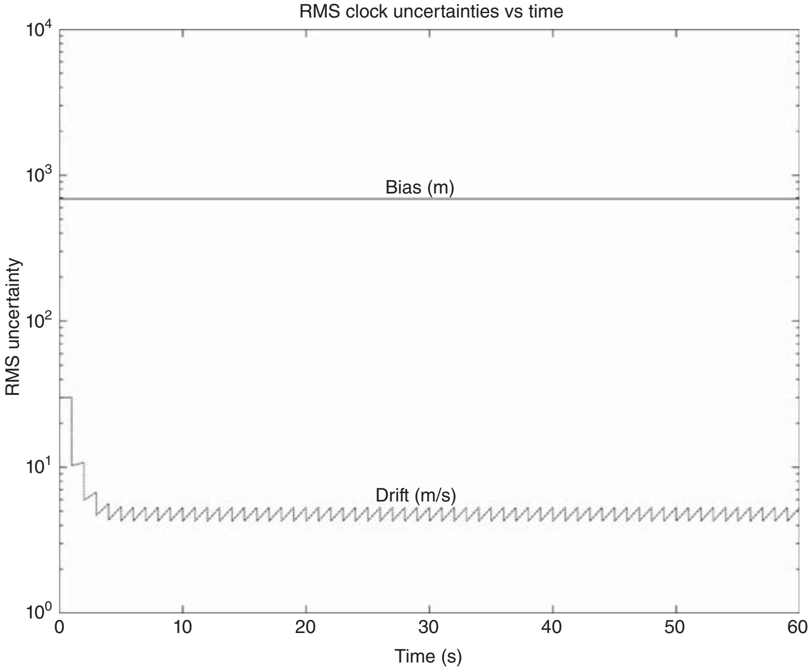


Figure 10.10 Clock parameter uncertainties with bad satellite geometry.

(d) [FIG8], which uses a specialized 1D model for the figure-8 track dynamics, using only along-track distances, velocities, and accelerations. It has three state variables. The model is coded in `GNSSshootoutNCE.m`.

Each model was "tuned" to the simulated host vehicle dynamics by setting its independent model parameters to match the values listed in Table 10.1. In all cases, the full Kalman filter model includes state variables for the time-correlated propagation delay errors, *but does not include clock errors* (two state variables).

Clock errors were excluded to better demonstrate the influence of host vehicle dynamic modeling on estimating the dynamic state variables. The respective achievable performance values after taking into account the receiver clock errors will depend on the quality of the clock, but will generally be worse in all cases. All results do include simulated time-correlated propagation delay errors and realistic GPS satellite trajectories.

The resulting respective RMS position and velocity errors from `GNSSshootout-NCE.m` are listed in Table 10.3.

The values given in Table 10.3 for the number of state variables is for the vehicle dynamic model only. The number of additional state variables required for propagation delay errors was the number of satellites in view (≥ 15° above the horizon). Clock error modeling was not included, but it would add two more state variables to the model.

These results clearly indicate that host vehicle dynamic modeling can be important, although the differences would have been less dramatic if clock errors had been included.

10.3.6.3 Single-Frequency Versus Dual-Frequency GNSS Navigation The main difference between single-frequency GNSS navigation and navigation with a dual-frequency receiver (or with differential correction) is that the state variables in the single-frequency case include the uncompensated ionospheric delays, and number of state variables is significantly greater in the single-frequency case. As a consequence, the Kalman filter must expend some of its information in solving for the added state variables, and it has less information left for the navigation variables.

The m-file `GPS2FreqOnly.m` solves the Riccati equation with dynamic coefficient matrix structure like that in the lower left of Figure 10.29, generating plots of the various RMS navigational uncertainties, including the RMS horizontal position uncertainty shown plotted in Figure 10.11.

TABLE 10.3 GNSS Navigation with Different Vehicle Models and No Clock Errors

vehicle dynamic model	number of state variables	rms position error, m			rms velocity error, m/s		
		north	east	vertical	north	east	vertical
MODL3	6	49.3	34.6	5.7	18.1	16.1	0.4
MODL5	6	27.5	17.2	3.5	20.1	15.0	0.4
MODL6	9	8.4	7.9	3.3	22.2	16.7	0.4
FIG8	3	1.2	0.7	0.02	0.11	0.16	0.002

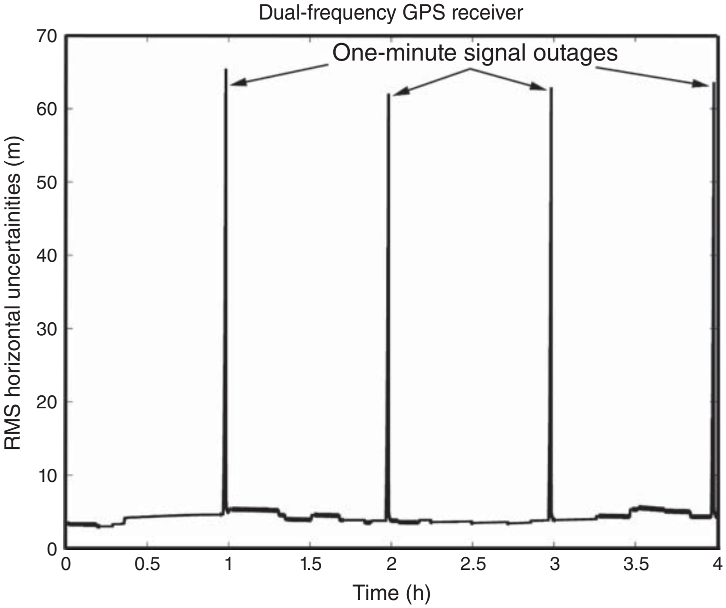


Figure 10.11 RMS horizontal position uncertainty using dual-frequency GPS.

By comparison, the m-file `GPS1FreqOnly.m` solves the Riccati equation with F and Q matrix structures like that in the upper left of Figure 10.29, but with the common parts of the model parameters having the same values. A plot of the resulting RMS horizontal position uncertainty is shown in Figure 10.12.

As expected, the RMS navigation errors are a bit worse for the single-frequency receiver.

10.3.7 Schmidt–Kalman Filtering for Residual Iono Corrections

Single-frequency GPS receivers can use the parametric Klobuchar model built into the GPS satellite message structure for removing much (but not enough) of the ionospheric propagation delays of the received signals. The rest of the correction requires additional Kalman filter state variables for the residual iono errors of each satellite used. This implementation may either assign one state variable for each satellite in the GPS constellation or use a state-swapping strategy to keep just enough state variables for the number of satellites currently being used. Either way, this adds from 4 (minimal solution) to 30+ (maximum solution) nuisance variables to the Kalman filter state vector.

The Schmidt–Kalman filter (described in Chapter 9) is designed to reduce filter computational requirements by ignoring nuisance variables—thereby sacrificing some estimation performance. The resulting filter performance after reducing the number of state variables is taken into account in the associated Riccati equation

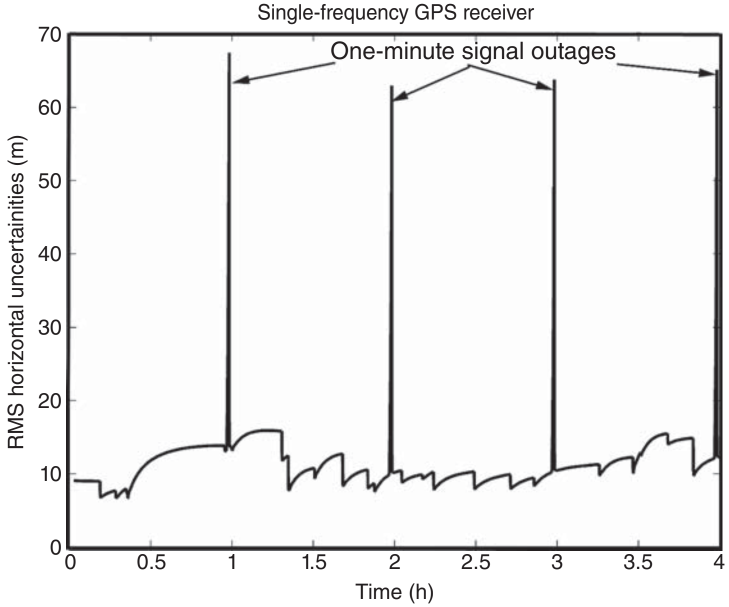


Figure 10.12 RMS horizontal position uncertainty using single-frequency GPS.

implementation, however. This allows the filter designer to assess the performance degradation to be taken into account in the trade-offs between alternative implementations.

These sorts of trade-offs can be important in system processor selection, when the Schmidt–Kalman approach could make the difference between a more expensive processor and a less-expensive but slower alternative.

The MATLAB m-file `SchmidtKalmanTest.m` on the companion Wiley web site evaluates the performance degradation in terms of the penalty in RMS position error added by the Schmidt–Kalman filter during a simulated dynamic test. The alternative Kalman filter implementation in this comparison used one state variable for each satellite in the constellation, although only those more than 15° above the horizon were being tracked. Other details of the simulated conditions are given in the MATLAB script. The result is plotted in Figure 10.13, in terms of the "un-RMSed" difference between the straight Kalman filter implementation and the Schmidt–Kalman implementation,

$$\sigma_{\text{unRMS}} = \sqrt{\sigma^2_{\text{SKF}} - \sigma^2_{\text{KF}}}.$$

Figure 10.13(a) is the additional position error contribution from the Schmidt–Kalman approximation, and Figure 10.13(b) is the number of satellites being tracked. This shows very well the transient performance degradation each time a satellite is added to or deleted from the list of those being tracked. These transients amount to an additional error contribution in the order of several meters to more than 10 m, with

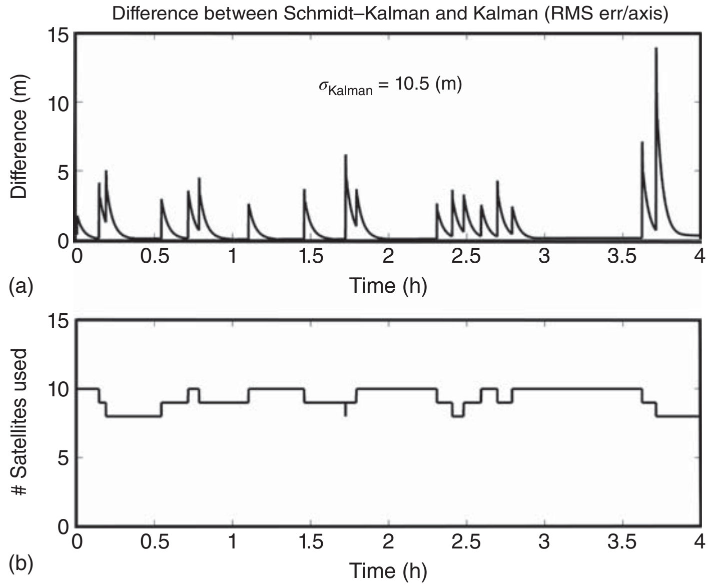


Figure 10.13 Performance Degradation with Schmidt–Kalman filtering.

the transient effect vanishing after several minutes. In this particular example, the degradation is rather small, considering that the baseline Kalman filter navigation performance is in the order of 10-m RMS.

Any decision about whether to use the Schmidt–Kalman filter should be based on the intended implementation conditions. It is also important in making this sort of trade-off decision to quantify the computational savings associated with the performance loss.

10.3.8 Using Pseudorange Differences

By exploiting known error characteristics in the signals, GNSS users have devised the following methods (and many more) to improve GNSS navigation performance. We mention them here to identify alternative methods available to system implementers.

10.3.8.1 Time Differences GNSS navigation solutions can use the difference between successive pseudorange measurements to the same satellite as a measure of the velocity difference between the satellite and the receiver antenna. Because satellite velocities are known very accurately and the differences are relatively insensitive to signal propagation delays, these measurements are useful for estimating host vehicle velocities.

10.3.8.2 Spatial Differences Pseudoranges from separate antennas are also used in GNSS navigation. Receiver antennas at known fixed locations are used to estimate pseudorange errors for each available satellite due to uncompensated propagation delays. The propagation delays do not vary significantly over distances of tens or hundreds of kilometers. The estimated pseudorange corrections are broadcast to nearby receivers and used to improve the psuedorange measurement accuracies for each satellite. This approach can provide position accuracies in the order of centimeters if carrier phase tracking is used.

10.3.8.3 Derived Attitude Estimates Arrays of antennas rigidly mounted on a common base are used to estimate attitude. The GNSS receivers in this application commonly use carrier phase interferometry to provide a better measurement of attitude than pseudoranges alone. Antenna separations in the order of a meter can provide RMS attitude accuracies of a milliradian or less.

Although GNSS pseudorange measurements at a single antenna are insensitive to antenna attitude, the estimated antenna velocity and acceleration are commonly used to generate secondary estimates of heading, pitch angle and roll angle. Here, these secondary estimates will assume the vehicle roll axis is aligned with the direction of velocity, and that turns are "coordinated." That is, the sensed accelerations are kept quasi-parallel to the vehicle yaw axis.

More sophisticated models might include the modeled effects of slide-slip (the angle between the velocity vector and the vehicle roll axis due to accelerations along the vehicle pitch axis), aerodynamic disturbances, and suspension reactions. Here, these other effects are ignored, except perhaps as part of the RMS uncertainties in the secondary estimates.

The simplified attitude estimation model is

$$\tan(\widehat{\theta}_Y) = \frac{\widehat{V}_E}{\widehat{V}_N}\text{(heading angle)} \tag{10.77}$$

$$\sin(\widehat{\theta}_P) = \frac{-\widehat{V}_D}{\sqrt{\widehat{V}_N^2 + \widehat{V}_E^2 + \widehat{V}_D^2}}\text{(pitch angle)} \tag{10.78}$$

$$\sin(\widehat{\theta}_R) = \frac{U_D}{\cos(\widehat{\theta}_P)}\text{(roll angle)} \tag{10.79}$$

$$U = \frac{\widehat{V} \otimes \left[I - \frac{\widehat{V}\widehat{V}^{\mathrm{T}}}{\widehat{V}^{\mathrm{T}}\widehat{V}}\right]\widehat{A}}{\left|\widehat{V} \otimes \left[I - \frac{\widehat{V}\widehat{V}^{\mathrm{T}}}{\widehat{V}^{\mathrm{T}}\widehat{V}}\right]\widehat{A}\right|}, \tag{10.80}$$

where $\widehat{V}$ is the estimated velocity vector and $\widehat{A}$ is the estimated sensed acceleration vector (i.e., not including gravitational acceleration) in north-east-down (NED) coordinates, and $\widehat{\theta}_Y$, $\widehat{\theta}_P$, and $\widehat{\theta}_R$ are the estimated vehicle heading, pitch, and roll angles.

This model is only good so long as the magnitude of velocity is much greater than the RMS velocity uncertainty.

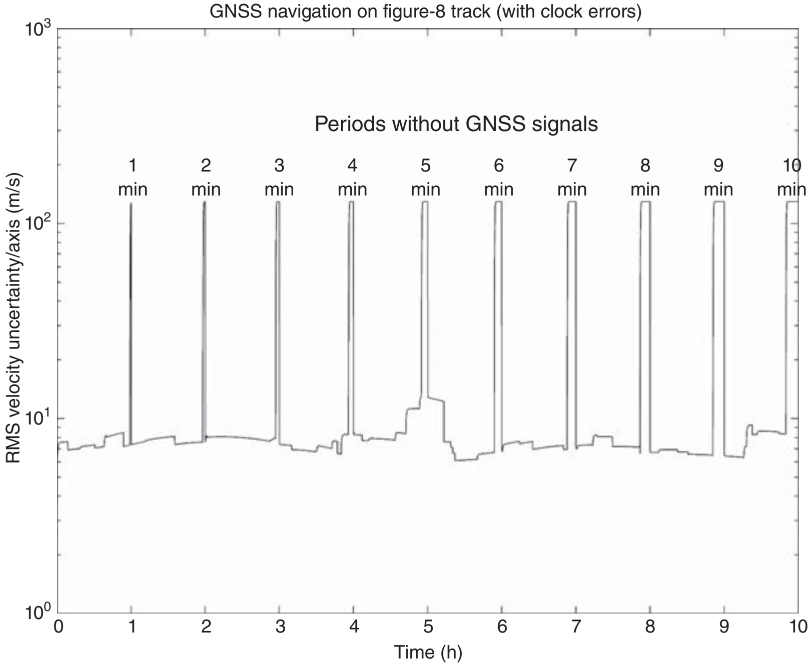


Figure 10.14 RMS velocity uncertainty for GNSS navigation on figure-8 track.

Results from a GNSS-only simulation on a figure-8 track are plotted in Figure 10.14. The simulated velocity magnitude $|V| \approx 25$ m/s and its standard deviation $\sigma_V \approx 7.7$ m/s per axis for those periods of time when GNSS signals are available. The equivalent RMS derived angle uncertainties due to velocity uncertainties only would be in the order of $\sigma_\theta \approx 0.3$ rad $\approx 17°$. This is much worse than the short-term RMS tilt uncertainties in INS navigation, and not very useful for dead reckoning either.

However, the angular uncertainties scale roughly as the RMS velocity uncertainty and the inverse of speed. Given the same GNSS velocity estimation uncertainties, vehicles on a freeway with $|V| \approx 100$ m/s would have attitude uncertainties in the order of 4–5. For high performance aircraft with $|V| \approx 300$ m/s, the RMS attitude uncertainties in this approximation would be in the order of 1–2. However, these vehicles would not be using the same dynamic model (MODL6) used in the track simulation, and their RMS velocity uncertainties would probably differ.

10.4 INERTIAL NAVIGATION SYSTEMS (INS)

The purpose of this section is to derive, implement, and demonstrate INS navigation error models for Kalman filters used in GNSS/INS integration. Early military "stand-alone" (i.e., unaided) inertial navigators did not use Kalman filters for navigation because INS development began before the Kalman filter was available, but inertial navigation and airborne radar would be instrumental in the development of flightworthy computers that would later be useful for Kalman filter implementations.

There are only a couple of operational GNSS systems, with perhaps one or two more on the way. Their error models at the receiver level have a lot in common.

Inertial navigation is far more diverse. There are perhaps hundreds or thousands of different INS designs. Their error models are much more diverse at the GNSS/INS integration level. We will select a few example error models to demonstrate the general principles exploited by Kalman filtering, but keep in mind that these are but a few of the many.

The main problem with inertial navigation is that its errors increase with time unless other navigation sources are used to keep them in check. This is not necessarily a serious problem for ballistic missiles, because their time of active inertial navigation is only for the few minutes of launch. But it is a problem for inertial navigation of land vehicles, aircraft, and ships.

Soon after the introduction of the Kalman filter, it was used to assess the relative advantages of using various forms of "external" navigation solutions (including GNSS) for keeping INS errors in check. Models developed for these studies would play a major role in the initial design, development, and implementation of satellite-based navigation systems.

10.4.1 A Little Background

Inertial navigation has also been called *Newtonian navigation*. The basic idea comes from Newton's calculus: *The second integral of acceleration is position.* Given that and sensors capable of measuring the three components of acceleration over time, plus initial values for position and velocity, turning this into a navigation system[8] would appear to be a relatively straightforward integration problem. Reducing it to practice would prove to be a bit more complicated, however.

The oldest known INSs are of the type you carry around in your head. There are two of them, and they are part of the *vestibular systems* located in the bony mass behind each ear. They have been evolving since the time your ancestors were fish. Each includes three rotational acceleration sensors (*semicircular canals*) and two 3-axis accelerometers (*otolith organs*). Their primary function is to compensate your vision system when you move your head. They are not accurate enough for long-distance navigation, but they do enable you to balance and navigate in total darkness for short distances.

10.4.1.1 History Enabling technologies for practical long-distance inertial navigation would not emerge until the twentieth century, at the beginning of which precision

[8] German entrepreneur Johann M. Boykow (1878–1935) was one who foresaw the possibility and led much of the pioneering German pre-WWII inertial sensor development at the gyroscope company Kreiselgeräte GMBH in Berlin.

gyroscopes were being developed to replace the magnetic compass—which was not very useful aboard iron ships.

After the 1919 Treaty of Versailles had forbidden long-range artillery development in Germany, efforts there shifted to longer-range cruise missiles (V-1) and ballistic missiles (V-2). These require guidance and control systems to direct them to their targets, which would be done using rotation sensors (gyroscopes) and acceleration sensors (accelerometers). Rotation sensors were used to keep track of where the acceleration sensors were pointing relative to their intended targets.

The end of World War II (WWII) saw the beginning of the "atomic age," the Cold War between the Soviet Union and NATO allies, and the race to develop long-range delivery systems for nuclear weapons. Building on German WWII experience (and with the aid of some of its personnel), inertial navigation development in the United States and the Union of Soviet Socialist Republics was rather rapid. Much of this development was (and still is) highly classified, so technical details are not always easily come by.

By 1953, an MIT-designed INS[9] could successfully navigate a WWII-era B-29 bomber from Bedford, Massachusetts to Los Angeles, California. The INS alone weighed more than a tonne and was about the size of a Smart car. By the time of the dissolution of the Soviet Union in 1991, the technology for inertial navigation had become quite mature. Inertial systems were then very much more accurate and much smaller but still quite expensive.

At about that time (the early 1990s), the United States was transitioning from its first-generation satellite navigation system (Transit) to its second-generation system (GPS) Both had been developed for integration with inertial navigation systems, which greatly increases INS operational time and range, and reduces inertial sensor performance requirements for achieving a given level of navigation accuracy.

Inertial navigation systems were initially designed as stand-alone, self-contained, secure navigation systems that need not rely on any external aids except for initialization—which was ideal for its initial application to guidance and control of long-range ballistic missiles. While stationary, it is capable of self-alignment with respect to Earth-fixed coordinates and self-determination of its latitude but requires input of its initial longitude and altitude from other sources.

When the Kalman filter first appeared, it first replaced most of the methods used for calibrating inertial sensors and for initial self-alignment. However, it was soon applied to using any available auxilliary information to improve INS functionality and performance. This came to include airborne radar (for aircraft) and (for high altitude flight) star trackers. It also came to include using Transit, LORAN, and GPS for navigation aiding and using sensed acceleration and rotation matching for initial alignment with respect to a carrier-borne INS.

Development of MEMS technologies in the 1970s and 1980s had significantly reduced the cost of inertial sensors at the lowered performance levels required for integrated GNSS/INS navigation. MEMS sensors are not only inexpensive but also

[9] Developed under technical leadership by INS pioneer Charles Stark Draper (1901–1987) and named "Space Inertial Reference Equipment," (SPIRE).

extremely small and light and require very little power. These attributes have combined to enable a significant expansion of the applications markets for integrated GNSS/INS systems.

10.4.1.2 Basic Concepts

Inertia is the propensity of rigid bodies to maintain constant translational and rotational velocity, unless disturbed by forces or torques, respectively (Newton's first law or 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. They are not necessarily the same as the *navigation coordinates*, which are typically dictated by the navigation problem at hand. For example, "locally level" coordinates used for navigation on or near the surface of the earth are rotating (with the earth) and accelerating (to counter gravity). Such rotations and accelerations must be taken into account in the practical implementation of inertial navigation.

Inertial sensors measure inertial accelerations and rotations, both of which are vector-valued variables.

- *Gyroscopes* (commonly shortened to "gyros") are sensors for measuring rotation. *Rate gyros* measure rotation rate, and *displacement gyros* (also called *whole-angle gyros*) measure accumulated rotation angle. Inertial navigation depends on gyros for maintaining knowledge of how the accelerometers are oriented in inertial and navigational coordinates. There are many basic physical mechanisms for sensing rotation, including
  - Momentum-wheel gyroscopes use the law of conservation of angular momentum (in direction and magnitude) of a spinning mass to establish an inertial reference direction—the rotation axis of the spinning mass. The Foucault gyroscope is a momentum-wheel gyroscope.
  - Lightwave gyroscopes, which use the constant speed of light in opposite directions around a closed path to detect rotations in the plane of that path. Two basic designs are
    - Laser gyroscopes, in which the light is propagated through lasing cavities and reflected off mirrors to complete the loop. Rotations are detected by interferometry of the relative phases in the counter-rotating light paths. Laser gyroscopes are rate integrating.
    - Fiber-optic gyroscopes (FOG), in which light from a laser source is coupled into an optical fiber with many windings around a closed path. Rotation is detected using interferometry between the laser source and the light exiting the loop, based on a phenomenon called the *Sagnac effect.*
  - Vibrating Coriolis gyroscopes , which detect the out-of-plane forces on a body vibrating in a plane due to rotation of the trajectory (Coriolis effect). There are many different designs. "Tuning fork" gyroscopes, for example,

detect rotational vibrations of a tuning fork due to the Corilis effect on the vibrating tines. Hemispherical resonator gyroscopes (also called *wine glass* gyros) detect shifts in the phasing of mechanical vibrational modes caused by the Coriolis effect during rotation. A microscale MEMS design by the Charles Stark Draper Laboratory (now Draper Laboratory) detects out-of-plane motions of a vibrating free layer of silicon supported by silicon "wires" and powered by an electrostatic "comb drive" developed at the Berkeley Sensor and Actuator Center (BSAC) of the University of California.

- *Accelerometers* are sensors for measuring inertial acceleration, also called *specific force* to distinguish it from what we call "gravitational acceleration." *Accelerometers do not measure gravitational acceleration*, which is perhaps more accurately modeled as a warping of the space-time continuum in a gravitational field. An accelerometer in free fall (e.g., in orbit) in a gravitational field has no detectable input. What accelerometers measure is modeled by Newton's second law as $a = F/m$, where F is the physically applied force (not including gravity), m is the mass it is applied to, and *specific force* is the ratio F/m. This all has a profound effect on inertial navigation error propagation. Basic types of acceleration sensors include
  - Pendulous integrating gyroscopic accelerometers (PIGA[10]) in which the center of support of a momentum-wheel gyroscope is offset axially from its center of mass, which creates a precession torque proportional to the component of acceleration orthogonal to the mass offset. These are *integrating accelerometers*, because the output accumulated precession angle is proportional to the integral of input acceleration. They are also the most accurate (and expensive) accelerometers.
  - Proof-mass accelerometers, which measure the force necessary to keep a mass centered in its instrument housing. There are many methods for measuring this force, leading to many different accelerometer designs.
- The *input axis* of an inertial sensor defines which vector component of acceleration or rotation rate it measures. Multiaxis sensors measure more than one component.

An *inertial sensor assembly* (ISA) is an ensemble of inertial sensors rigidly mounted to a common base to maintain the same relative orientations, as illustrated in Figure 10.15. Inertial sensor assemblies used in inertial navigation contain at least three accelerometers (represented by blocks with "A" on each face) and three gyroscopes (represented by blocks with "G" on each face), as shown in the figure, but they may contain more and they may use one multiaxis sensor in place of an equivalent number of single-axis sensors. Reasons for using more than three sensor input axes for an ISA include

[10] Invented by Fritz K. Mueller (1907–2001) in Germany in the 1930s and brought (with Mueller) to the United States in 1945. It was originally used in Germany for controlling ballistic missile range by integrating the total sensed thrust acceleration and signaling for engine cut-off when a predetermined range was attained. It soon became the dominant high accuracy accelerometer design.

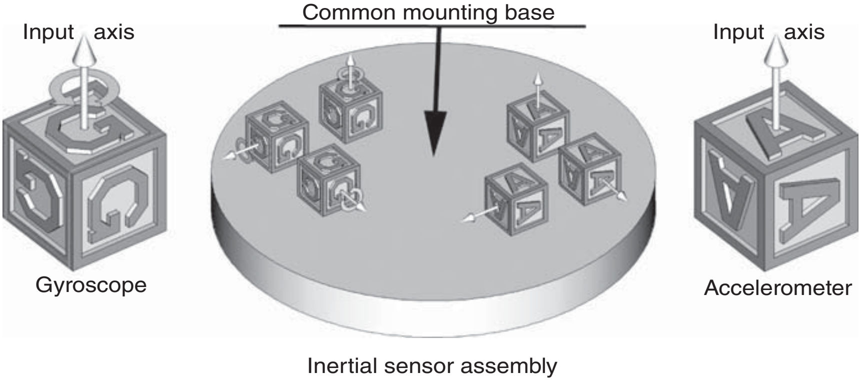


Figure 10.15 Inertial sensor assembly (ISA) components.

- Added sensor redundancy for coping with single-senor failures.
- Reducing effective sensor errors in certain input directions.

The term *inertial reference unit* (IRU) sometimes refers to an inertial sensor system for attitude information only (i.e., using only gyroscopes). Other terms used for the ISA are *instrument cluster* and (for gimbaled systems) *stable element, inertial platform,* or *stable platform.*

An *inertial measurement unit* (IMU) includes an ISA and its associated support electronics for calibration and control of the ISA (possibly including temperature control). These can be small enough to be packaged in a wristwatch.

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

- *Navigation computers* (one or more) to calculate the gravitational acceleration (not measured by accelerometers) and process the outputs of the accelerometers and gyroscopes from the IMU to maintain an estimate of the position of the IMU. Intermediate results of the implementation method usually include estimates of velocity, attitude, and attitude rates of the IMU.
- *User interfaces*, such as display consoles for human operators and analog and/or digital data interfaces for vehicle guidance[11] and control functions.
- *Power supplies* and/or raw power conditioning for the complete INS.

Some INS designs include temperature control.

Navigation Reference An INS estimates the position of its ISA, just as a GNSS receiver estimates the position of its antenna. When these two systems are integrated together, the offset between these two reference points (ISA and antenna) must be taken into account.

[11] "Guidance" generally includes the generation of command signals for controlling the motion and attitude of a vehicle to follow a specified trajectory or to arrive at a specified destination.

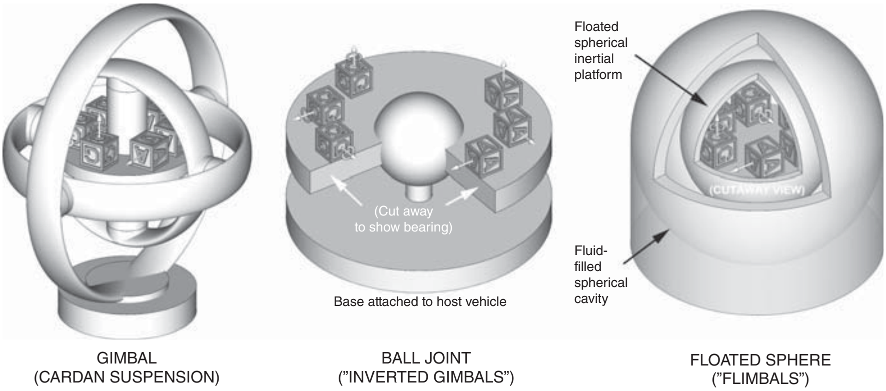

Floated spherical inertial platform
(Cut away to show bearing)
(CUTAWAY VIEW)
Fluid-filled spherical cavity
Base attached to host vehicle
GIMBAL (CARDAN SUSPENSION)
BALL JOINT ("INVERTED GIMBALS")
FLOATED SPHERE ("FLIMBALS")

Figure 10.16 ISA attitude isolation methods. (a) Gimbal (cardian suspension), (b) ball joint ("inverted gimbals"), and (c) floated sphere ("flimbals").

10.4.1.3 Sensor Configurations There are two basic configurations for implementing inertial navigation, and some variations within each of the following:

- *Gimbaled systems* use gimbals (also called a *Cardan*[12] *suspension*) to isolate the ISA from inertial rotations or equivalent configurations as shown in Figure 10.16. The orientation of the ISA may also be controlled to align it with inertial coordinates, navigation coordinates, or rotating navigation coordinates. For navigation in the near-Earth environment, these may include locally level orientations in which one ISA axis is slaved to point in the direction of the local vertical (gravity), leaving the other two axes free to assume directions such as
  - One axis east and one axis north (impossible at the poles).
  - An "alpha-wander" system with a known angle α between a locally level axis and an Earth-fixed direction, which solves the problem at the poles.
  - Orientations rotating about the vertical, either continuously ("carouseling") or discretely ("indexing"), either of which cancels out certain types of navigation errors.
- *Strapdown systems* , in which the ISA is "quasi-hard" mounted (i.e., with possibly some shock and vibration isolation) to the carrier vehicle. These systems essentially have "software gimbals" using the gyro outputs to keep track of the orientation of the ISA at all times. An incentive for strapdown configurations is that software is generally much less expensive than hardware. MEMS inertial navigators are generally strapdown, and some designs allow more than one independent and orthogonal input axes in the plane of the MEMS substrate.

[12] Named for Gerolamo Cardano (1501–1576), a Renaissance polymath who did not claim he invented gimbals. The earliest known descriptions of gimbals date from the third century BCE, in Greece and in China.

- Hybrid strapdown systems with a single gimbal axis for rotating the ISA (either continuously or discretely) about a "near-vertical" axis. (Most manned vehicles, e.g., try to keep their passengers oriented with their tall axes near vertical.) This has nearly the same benefits as carouseling or indexing, without the additional cost of other gimbals.

10.4.2 Navigation Solution

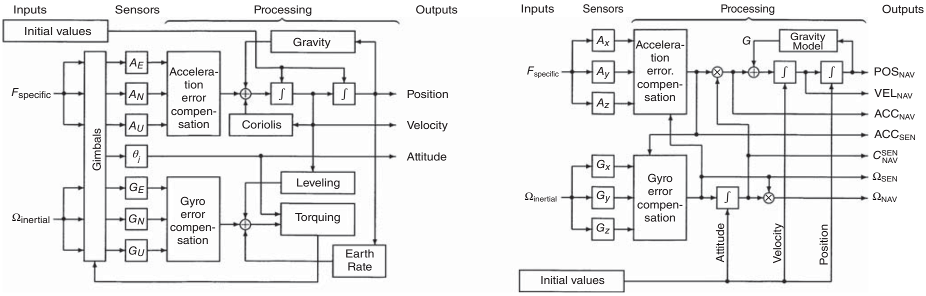
As a minimum, any navigation solution must include the current position of the host vehicle in navigation coordinates. For inertial navigation, the intermediate results must also include velocity and attitude. These ancillary variables may also be useful for other aspects of the mission, including

- Attitude, which is useful for driving aircraft cockpit displays for heading (compass card), and roll and pitch angles (artificial horizon). It is also useful for determining lateral wind speed (e.g., for decrabbing in cross-wind landings).
- Attitude rate, which can be used in the control loops for autopilots or missile steering, and for aiding antenna switching and phase locking for navigation satellite signals in GNSS/INS integration.
- Velocity, which is useful for determining time of arrival, and in control loops for autopilots and targeting. It can also be used to compensate for Doppler shift during GNSS signal acquisition and tracking.
- Acceleration, which is useful as an input to control loops for vehicle guidance and control, and as an aid in phase tracking of satellite signals in GNSS/INS integration.

10.4.2.1 Gimbaled Implementation For gimbaled systems, the attitude solution is maintained by the gimbals, which must include gimbal angle encoders for determining the relative attitude of the host vehicle.

The fundamental processing functions for a gimbaled INS are shown as a flowchart in Figure 10.17(a). The boxes in this flowchart represent the major processing functions, including

- The function of the gimbals to keep the input axes of the inertial sensors at known directions in navigation coordinates. In this particular configuration, the accelerometers and gyroscopes have their input axes pointed east, north, and up. The gimbal angles (θ_j) are used for driving attitude displays.
- Integration of acceleration to get velocity.
- Integration of velocity to get position.
- Compensation of the sensor outputs for errors determined during calibration.
- Generation of predicted gravitational acceleration of navigation coordinates (locally level).
- Generation of predicted locally level coordinate rotation rates and Coriolis accelerations for velocities over a curved Earth surface.



(a) (b)

Figure 10.17 INS data flow. (a) Gimbaled and (b) strapdown.

- Compensation for Earth rotation rate.
- Compensating for the Coriolis effect in rotating coordinates.
- Generation of gimbal torquing commands for maintaining the ISA aligned with navigation coordinates.

Acceleration and attitude sensors in gimbaled systems are isolated from large rotation rates, which has some advantages in sensor design.

10.4.2.2 Strapdown Implementation For strapdown systems, attitude information is maintained in memory, and there are no gimbals to isolate the inertial sensors from rotations of the host vehicle. This eliminates the use of gyroscopic (e.g., PIGA) accelerometers, because they are also rotation-sensitive. It also places greater demands on rotation sensors to operate over wider dynamic ranges. Otherwise the fundamental implementation functions are much the same as in gimbaled systems.

The fundamental processing functions are shown in Figure 10.17(b).

The fundamental differences between gimbaled and strapdown implementations are

1. Gimbals are a hardware solution to the attitude problem. For strapdown systems, the solution must be implemented in the software.
2. Strapdown systems expose their sensors to much greater dynamic ranges of rotation rates, which places greater demands on their design—and on the algorithms used for maintaining the "attitude solution."
3. The "attitude solution" is characterized by $C^{\mathrm{NAV}}_{\mathrm{SEN}}$, the coordinate transformation matrix from sensor-fixed coordinates to navigation coordinates. A factor of $C^{\mathrm{NAV}}_{\mathrm{SEN}}$ is the only difference between the error models for strapdown and gimbaled systems.

10.4.3 Initializing the Navigation Solution

As mentioned at the beginning of Section 10.4.1, inertial navigation requires initial values for position, velocity, and attitude of its ISA. Historically, some of this was done using only internal resources—if the ISA could be kept motionless with respect to the earth's surface for several minutes to a few hours (depending on the accuracy required). When the Kalman filter came along in the early 1960s, methods were developed to allow initialization "on the fly" in much less time.

10.4.3.1 Gyrocompass Alignment This is an "almost complete" initialization method using only the inertial sensors.

Alignment is the process of determining the orientation of the ISA with respect to navigation coordinates.

Gyrocompass alignment is a method for determining the initial orientation of the ISA with respect to earth-fixed coordinates. This could be done with the INS held motionless with respect to the surface of Earth, in which case its initial velocity in

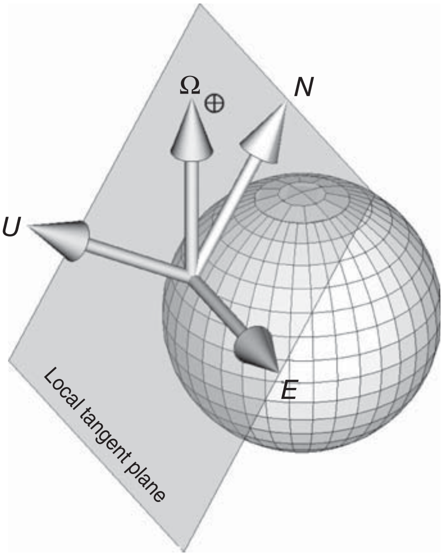


Figure 10.18 Gyrocompass alignment geometry.

Earth-fixed coordinates is zero and its position is known. That leaves the problem of determining its orientation. Figure 10.18 illustrates how it is done:

1. The direction of sensed acceleration is upward, which establishes that direction in ISA coordinates.
2. Except at the poles, the sensed Earth rotation vector has a north component. That is sufficient to establish the orientation of the ISA with respect to east-north-up (ENU) coordinates. Gyrocompass alignment does not work at the poles.
3. The angle between the vertical and the Earth rotation axis equals 90° minus latitude, from which latitude can be determined.

The only missing initial conditions are longitude and altitude. Given latitude and the magnitude of sensed acceleration, one might get a very crude estimated of altitude—but generally not good enough for navigation. Before GNSS or its equivalent, these data had to be entered manually or from local wireless sources.

Gimbaled Implementation Alignment of a gimbaled INS while the ISA remains in a stationary position with respect to Earth can be implemented by rotating the gimbals to force the outputs of the east accelerometer, north accelerometer, and east gyro to be zero. These three constraints are sufficient to solve the alignment problem—except at the poles.

Milli Earth Rate Units (MERUs) The gyroscopes in gimbaled systems are primarily used for maintaining the ISA aligned with navigation coordinates. As such, the dominating input rate will be the Earth rotation rate of about 15° per hour. At that rate, a relative scale factor error of 10^{-3} would be equivalent to 0.015/h, or 1 " MERU" (an acronym for "milli Earth rate unit"), defined at the MIT Instrumentation Laboratory (now the Draper[13] Laboratory) as being about the gyro bias accuracy required for gyrocompass alignment to 1 mrad of accuracy.

Gyro Bias and Scale Factor Requirements Gyro bias is the gyro output offset error. Gyro bias requirements for medium accuracy navigation (1 NMi/h CEP rate) are usually specified to be significantly less than 1 MERU, in the order of 10^{-3} to 10^{-2} degrees/h, and scale factor requirements for gimbaled systems are often specified to be in the order of 10^{-4} parts per part (just to add a little cushion). This has given some much-needed relief in scale factor requirements for gyros in gimbaled systems. Scale factor requirements for that level of performance in strapdown systems, on the other hand, will depend on the expected body rotation rates of the host vehicle, which can be hundreds of degrees per second—or millions of MERUs. In that case, scale factor stability requirements for strapdown inertial navigation could be a millionth of that required for gimbaled systems. Because of this, the savings from eliminating gimbals in strapdown systems can be partially eaten up by the added cost of better gyroscopes.

Kalman Filter Implementations Gyrocompass alignment is a bit more difficult aboard parked aircraft and docked ships, which can be subjected to external disturbances due to winds, tides, and waves, and to internal disturbances during fueling, cargo loading, or passenger loading. In these cases, gyrocompass alignment can still be done using a Kalman filter with a stochastic dynamic model for these disturbances. These generally include damped harmonic oscillator models tuned to match the support or suspension systems (including rotational dynamics) of the host vehicle during such operations.

Initializing Position Although gyrocompass alignment can estimate INS latitude, it has become common practice to use external sources—including GNSS—for initializing longitude, latitude, and altitude. However, in integrated GNSS/INS implementations, it is possible to estimate attitude, position, and velocity when the host vehicle is under way. The GNSS solution also estimates the direction of travel, which can be used as an initial estimate of heading.

10.4.3.2 Alignment Using Auxilliary Information There are a number of methods for speeding up alignment by using other sources of information, such as

[13] When asked to give some general notation of the magnitude of 1 MERU, Charles Stark Draper is alleged to have replied "It's about how fast you have to twist my arm to get me to take a drink."

1. An INS-equipped vehicle carried by another host vehicle with its own INS can use information from the host INS to initialize its own INS in a Kalman filtering procedure called *velocity matching*. Carrier-based aircraft, for example, can use the sensed roll and pitch of the ship and the known direction of the launch catapult action relative to the ship. This requires a Kalman filter similar in some ways to that used in gyrocompassing, but generally initializes much faster due to the additional information.
2. *Transfer alignment* uses a Kalman filter to initializing the navigation solutions of tactical missiles carried by an aircraft with its own INS, by matching their velocities during maneuvering. The same Kalman filter model can be used offline to select those types of maneuvers most useful for speeding up alignment and for determining when such initialization is adequate for the intended mission.
3. Integrated GNSS/INS navigation is capable of initializing the INS "on the fly," using the GNSS solutions.

10.4.4 INS Navigation Error Sources

Inputs to inertial navigation systems include

- Initial conditions for the navigation solution, including position, velocity, and orientation of its ISA.
- Outputs of acceleration and rotation sensors on the ISA, used for carrying the initial navigation solution forward during dynamic disturbances of the ISA.
- Outputs from internal models for unsensed rotations and accelerations of its navigation coordinates. For navigation with respect to the surface of the earth, for example, these will include Earth rotation rate, gravitational acceleration, and centrifugal forces and Coriolis effects from velocities with respect to a rotating earth.
- Any navigation aids used for keeping inertial navigation errors in check. These may include radio navigation fixes, such as those from navigation satellites.

Each of these is a potential error source, as are any modeling errors and roundoff errors in the computer software implementation.

10.4.4.1 Core Error Variables The minimal number of navigation error variables for keeping track of inertial navigation errors during operation is nine. The "core nine" include

1–3: Three components of position error.

4–6: Three components of velocity error.

7–9: Three components of attitude error.

10.4.4.2 Dynamic Noise Sources One source of effective dynamic disturbance noise in inertial navigation is noise on the outputs of the inertial sensors (gyroscopes and accelerometers), including roundoff errors for digital sensors. These are not treated like sensor noise in a Kalman filter implementation but like dynamic disturbance noise on the core variables of the navigation solution. There are at least six such sources, one for each input axis of each sensor:

- Three or more "angular random walk" noise sources from three or more rate gyroscopes. Even "whole-angle" gyroscopes can exhibit noise, but it is more likely to be modeled as angular white noise.
- Three or more acceleration noise sources from three or more accelerometers.

The respective output noise characteristics can be determined empirically by observing sensor outputs with constant sensor input. If the noise is not white (uncorrelated), then additional state variables may be needed for sensor noise modeling.

10.4.4.3 Sensor Compensation Errors Each of the six (or more) inertial sensors has input/output errors due to the limits of manufacturing precision and calibration uncertainty, and these can be nonconstant errors that drift slowly over time due to intrinsic and environmental factors. They can usually be modeled as "slow variables" such as random walk processes or exponentially correlated random processes, each of which requires adding one state variable to the Kalman filter state vector. The simplest and most common sensor compensation errors are (i) output bias and (ii) input/output scale factor, but there may be other nonlinear sources as well. Just the biases and scale factors alone could total 12 or more in number (2 variables for each of 6 or more sensors), but high precision sensors can have even more error characteristics requiring continual updating from the sensor-integrating Kalman filter.

Gimbaled versus Strapdown Models Although the dynamics of INS navigation errors are modeled here in locally level coordinates, the same dynamic model applies to any INS configuration—gimbaled or strapdown. However, the models for how sensor noise and sensor compensation errors couple into navigation errors will be different for different gimbaled implementations (e.g., ENU versus carouseled) and for strapdown implementations. The models used here and in the next section are for a gimbaled ENU, but their transformations for other implementations is straightforward, as described in Sections 10.4.5.1, 10.4.5.4, and 10.4.6.7.

10.4.4.4 INS Error Budgeting *Error budgeting* is the process of allocating subsystem performances in order to achieve some specified system-level performance. For INS, the dominant subsystems in this breakout are the sensors. It is an important part of system design because it allows reallocation of subsystem performance levels to minimize system cost, for example, or to accommodate derating of one component by tightening the performance requirements of others. It has been extremely important in the development of GNSS.

In the case of inertial navigation, system-level performance may be specified in different ways, depending on the application. Performance for intercontinental ballistic missiles, for example, is commonly specified in terms of RMS miss distance or CEP at the target. Performance for applications in cruise navigation, where navigational accuracy tends to deteriorate over time, is commonly specified in terms of how fast RMS navigational uncertainties grow with time. Performance in the latter case is often specified in units of nautical miles per hour CEP rate.

In some cases, the INS error budgeting process may only require evaluating expected performance with different choices of available accelerometers and gyros. In other cases, it may involve determining the required performance specifications for a sensor being developed for a specific application or to asses the expected performance improvement from proposed improvements in subsystem performance. Sensor subsystem performances are generally specified in terms of the RMS values of sensor error sources such as those described above.

In any case, the fundamental tool in error budgeting is the Riccati equation of the Kalman filter. This requires a Kalman filter model for these error sources, specified in terms of the fundamental parameters of the Kalman filter model. These are the matrices F (or Φ), Q, H, and R. Formulas for these parameters are derived in the following sections.

Furthermore, because the impact of INS error sources such as scale factor errors on navigation errors depends on the sensor inputs, the solution of the Riccati equation requires simulating representative dynamic conditions for the intended INS applications.

10.4.5 INS Navigation Error Dynamics

Although it is simple in concept, inertial navigation has rather complex error propagation models. Be aware that INS navigation error analysis can get fairly complicated. Perhaps the only book on the subject currently in print is the one by Britting [12], written about half a century ago. More recent derivations for the inertial navigation error propagation models presented below can be found in Chapter 11 of Reference 2. These are models for navigation on or near the surface of Earth, although they would apply as well—with appropriate changes of parameter values—to any near-spherical or ellipsoidal planet with a well-defined surface.

10.4.5.1 Navigation Error Coordinates

ENU Coordinates The dominant error mechanisms for navigating in the near-earth environment depend on the direction of the (unmeasured) gravitational acceleration (downward) and the direction of the Earth rotation axis (polar). These have fixed directions with respect to "locally level" coordinates, either NED or ENU. Navigators of vehicles with well-defined roll, pitch, and yaw coordinates (Figure 10.19) may prefer roll–pitch–yaw (RYP) coordinates, which align with NED coordinates when the vehicle is right-side-up and headed North. Navigators who feel more comfortable with the vertical axis in the direction of increasing altitude and the others in the directions of increasing longitude and latitude, on the other hand, might prefer

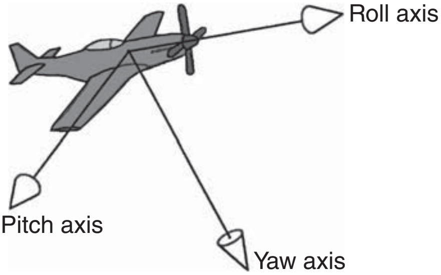


Figure 10.19 Roll–pitch–yaw coordinates.

ENU coordinates. Both GNSS and inertial navigation use altitude for the near-Earth environment, so derivations here will use ENU coordinates.

Sensor Coordinates For strapdown systems, the sensor input axes will have fixed directions with respect to vehicle-fixed RPY coordinates. An integral part of the INS installation and alignment process is the determination of the coordinate transformation matrix from sensor-fixed to vehicle-fixed coordinates,

$$C^{\mathrm{SEN}}_{\mathrm{RPY}}.$$

An integral part of the strapdown inertial navigation solution is the determination of the coordinate transformation matrix from sensor-fixed coordinates to navigation coordinates,

$$C^{\mathrm{SEN}}_{\mathrm{NAV}},$$

which is also used in the transformation of gimbaled inertial navigation error models to equivalent strapdown inertial navigation error models.

Vehicle attitude in navigation coordinates is then characterized by the coordinate transformation matrix

$$C^{\mathrm{NAV}}_{\mathrm{RPY}} = C^{\mathrm{SEN}}_{\mathrm{RPY}}\left(C^{\mathrm{SEN}}_{\mathrm{NAV}}\right)^{\mathrm{T}} \tag{10.81}$$

from navigation to vehicle-fixed coordinates.

10.4.5.2 First-Order Dynamic Modeling Inertial navigation errors are generally so small compared to the navigation solution that first-order approximations can be used in modeling their dynamics during navigation. However, this assumption may not hold for some methods of INS initialization using auxiliary sensors, for which initial orientation uncertainties can be much larger.

Because an INS is self-contained, errors in one part of the navigation solution easily couple into other parts of the solution. This is illustrated by the process flow diagram in Figure 10.20, in which some boxes are numbered according to the error coupling mechanisms they represent:

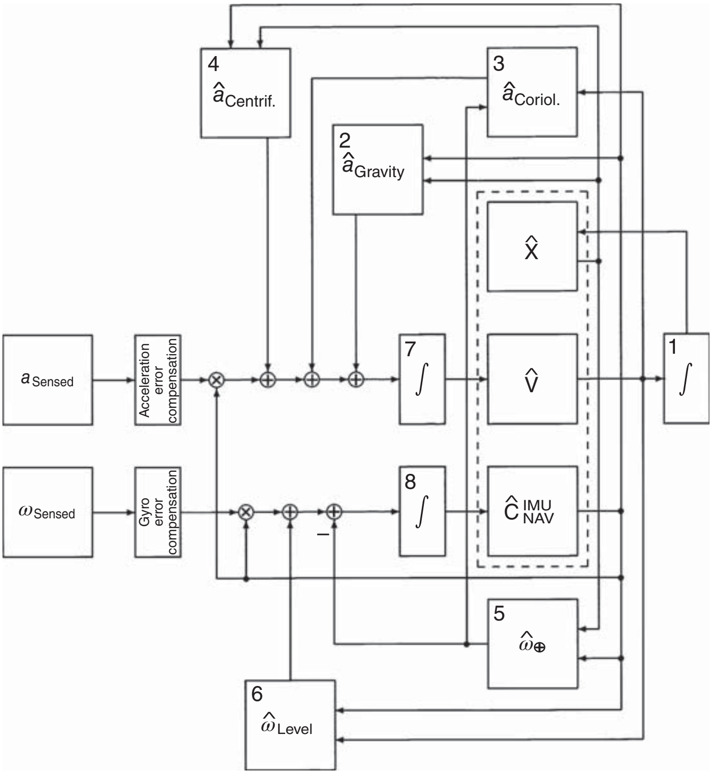

4
$\hat{a}_{\text{Centrif.}}$
3
$\hat{a}_{\text{Coriol.}}$
2
$\hat{a}_{\text{Gravity}}$
$\hat{x}$
a_{Sensed}
Acceleration error compensation
7
$\int$
$\hat{v}$
1
$\int$
ω_{Sensed}
Gyro error compensation
8
$\int$
$\hat{C}^{\text{IMU}}_{\text{NAV}}$
−
5
$\hat{\omega}_{\oplus}$
6
$\hat{\omega}_{\text{Level}}$

Figure 10.20 INS processing data flows.

1. Velocity errors couple into position errors through integration.
2. Position and attitude errors couple into acceleration errors through gravity compensation.
3. Velocity and Earthrate modeling errors couple into attitude rate errors through Coriolis compensation.
4. Position and attitude errors couple into acceleration errors through centrifugal compensation.
5. Position and attitude errors couple into attitude rate errors through Earthrate compensation.
6. Velocity and attitude errors couple into attitude rate errors through leveling.
7. Acceleration errors couple into velocity errors through integration.
8. Attitude rate errors couple into attitude errors through integration.
9. Acceleration errors couple into velocity errors through integration.
10. Attitude rate errors couple into attitude errors through integration.

For a more detailed treatment of INS error modeling, see References 12, 13, and 35 or Chapter 11 of Reference 2.

10.4.5.3 Nine-State Dynamic Model The nine state variables of inertial navigation error are the three components of position error, three components of velocity error, and three components of attitude error. In locally level "ENU" coordinates, these will be

ε_E, INS easting error (m)
ε_N, INS northing error (m)
ε_U, INS altitude error (m)
$\dot{\varepsilon}_E$, INS east velocity error (m/s)
$\dot{\varepsilon}_N$, INS north velocity error (m/s)
$\dot{\varepsilon}_U$, INS vertical velocity error (m/s)
ρ_E, INS misalignment about east axis (rad)
ρ_N, INS misalignment about north axis (rad)
ρ_U, INS misalignment about vertical axis (rad),

which can be represented as the components of an error dynamic model state vector

$$\varepsilon_{\mathrm{NAV}} \overset{\mathrm{def}}{=} \begin{bmatrix} \varepsilon_E & \varepsilon_N & \varepsilon_U & \dot{\varepsilon}_E & \dot{\varepsilon}_N & \dot{\varepsilon}_U & \rho_E & \rho_N & \rho_U \end{bmatrix}^{\mathrm{T}}. \tag{10.82}$$

The resulting 9 × 9 dynamic coefficient matrix for the nine INS error variables can then be partitioned into block form with 3 × 3 blocks as

$$\frac{d\varepsilon_{\mathrm{NAV}}(t)}{dt} = \mathbf{F}(t)\varepsilon_{\mathrm{NAV}}(t) + w(t) \tag{10.83}$$

$$\mathbf{F} = \begin{bmatrix} 0_{3\times3} & I_3 & 0_{3\times3} \\ F_{21} & F_{22} & F_{23} \\ F_{21} & F_{22} & F_{23} \end{bmatrix} \tag{10.84}$$

$$F_{21} = \begin{bmatrix} 0 & 2\dfrac{\Omega_\oplus \sin\,\phi v_U}{\overline{R}_\oplus} + 2\dfrac{\Omega_\oplus \cos\,\phi v_N}{\overline{R}_\oplus} & 0 \\ 0 & -2\dfrac{\Omega_\oplus \cos\,\phi v_E}{\overline{R}_\oplus} + \Omega_\oplus{}^2 \sin{}^2\phi - \Omega_\oplus{}^2 \cos{}^2\phi & -\Omega_\oplus{}^2 \sin\,\phi \cos\,\phi \\ 0 & -2\dfrac{\Omega_\oplus \sin\,\phi v_E}{\overline{R}_\oplus} - 2\Omega_\oplus{}^2 \sin\,\phi \cos\,\phi & 2\dfrac{\mathrm{GM}_\oplus}{\overline{R}_\oplus{}^3} + \Omega_\oplus{}^2 \cos{}^2\phi \end{bmatrix} \tag{10.85}$$

$$F_{22} = \begin{bmatrix} 0 & 2\Omega_\oplus \sin\,\phi & -2\Omega_\oplus \cos\,\phi \\ -2\Omega_\oplus \sin\,\phi & 0 & 0 \\ 2\Omega_\oplus \cos\,\phi & 0 & 0 \end{bmatrix} \tag{10.86}$$

$$F_{23} = \tag{10.87}$$

$$\left[\begin{matrix} 2\Omega_\oplus \sin\,\phi v_U + 2\Omega_\oplus \cos\,\phi v_N \\ a_U + \dfrac{\mathrm{GM}_\oplus}{\overline{R}_\oplus{}^2} - 2\Omega_\oplus \cos\,\phi v_E - \Omega_\oplus{}^2 \cos\,{}^2\phi \overline{R}_\oplus \\ -a_N - 2\Omega_\oplus \sin\,\phi v_E - \Omega_\oplus{}^2 \sin\,\phi \cos\,\phi \overline{R}_\oplus \end{matrix}\right.$$

$$\left.\begin{matrix} -a_U - \dfrac{\mathrm{GM}_\oplus}{\overline{R}_\oplus{}^2} + \Omega_\oplus{}^2 \cos\,{}^2\phi \overline{R}_\oplus & a_N + \Omega_\oplus{}^2 \sin\,\phi \cos\,\phi \overline{R}_\oplus \\ 2\Omega_\oplus \sin\,\phi v_U & -a_E - 2\Omega_\oplus \cos\,\phi v_U \\ a_E - 2\Omega_\oplus \sin\,\phi v_N & 2\Omega_\oplus \cos\,\phi v_N \end{matrix}\right]$$

$$F_{31} = \begin{bmatrix} 0 & 0 & \dfrac{v_N}{\overline{R}_\oplus{}^2} \\ 0 & \dfrac{\Omega_\oplus \sin\,\phi}{\overline{R}_\oplus} & -\dfrac{v_E}{\overline{R}_\oplus{}^2} \\ 0 & -\dfrac{\Omega_\oplus \cos\,\phi}{\overline{R}_\oplus} & 0 \end{bmatrix} \tag{10.88}$$

$$F_{32} = \begin{bmatrix} 0 & -\overline{R}_\oplus{}^{-1} & 0 \\ \overline{R}_\oplus{}^{-1} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{10.89}$$

$$F_{33} = \begin{bmatrix} -\dfrac{v_U}{\overline{R}_\oplus} & -\Omega_\oplus \sin\,\phi & \Omega_\oplus \cos\,\phi \\ \Omega_\oplus \sin\,\phi & -\dfrac{v_U}{\overline{R}_\oplus} & 0 \\ -\Omega_\oplus \cos\,\phi + \dfrac{v_E}{\overline{R}_\oplus} & \dfrac{v_N}{\overline{R}_\oplus} & 0 \end{bmatrix}, \tag{10.90}$$

where the INS error model dynamic coefficient matrix is defined in terms of the following parameters and variables:

$$\left.\begin{array}{llll}
\overline{R}_{\oplus} & \overset{\text{def}}{=} & \text{mean Earth radius} & \approx\ 0.6371009\times 10^{7}(\text{m}) \\
\text{GM}_{\oplus} & \overset{\text{def}}{=} & \text{Earth gravity constant} & \approx\ 0.3986004\times 10^{15}(\text{m}^{3}/\text{s}^{2}) \\
\Omega_{\oplus} & \overset{\text{def}}{=} & \text{Earth rotation rate} & \approx\ 0.7292115\times 10^{-4}(\text{rad/s}) \\
\widehat{\phi} & = & \phi+\varepsilon_{N}/\overline{R}_{\oplus} & \text{latitude (rad)} \\
\widehat{\theta} & = & \theta+\varepsilon_{E}/\left(\overline{R}_{\oplus}\cos\,\phi\right) & \text{longitude (rad)} \\
\hat{\mathrm{E}} & = & E+\varepsilon_{E} & \text{easting with respect to INS (m)} \\
\widehat{N} & = & N+\varepsilon_{N} & \text{northing with respect to INS (m)} \\
\hat{h} & = & h+\varepsilon_{U} & \text{altitude (m)} \\
\widehat{v}_{E} & = & v_{E}+\dot{\varepsilon}_{E} & \text{east INS velocity (m/s)} \\
\widehat{v}_{N} & = & v_{N}+\dot{\varepsilon}_{N} & \text{north INS velocity (m/s)} \\
\widehat{v}_{U} & = & v_{U}+\dot{\varepsilon}_{U} & \text{vertical INS velocity (m/s).}
\end{array}\right\} \quad (10.91)$$

Here, the "hatted" $(\widehat{\ })$ variables are provided by the INS and updated by a Kalman filter in GNSS/INS integration.

These equations are implemented in the MATLAB function `Fcore9.m` on the accompanying Wiley web site. More detailed derivations can be found in Chapter 11 of Reference 2.

10.4.5.4 Universality of the Core Model The nine-state model for inertial navigation errors is fundamental to all types of INS implementations—gimbaled and strapdown. The only differences between gimbaled and strapdown models will be a factor of the transformation matrix

$$C_{\text{NAV}}^{\text{SEN}}$$

defined in Section 10.4.5.1. The same applies for carouseled or indexed gimbal implementations.

10.4.5.5 Seven-State Dynamic Model The seven-state model is for an INS in which altitude and altitude rate are not part of the INS solution but are determined independently from other sources—such as altimeter outputs. The error model omits the vertical components of position (altitude) and velocity (altitude rate) in the nine-sate model. This avoids the problem of vertical channel instability, and this model is commonly used for predicting CEP rates .

Formulas for the resulting 7 × 7 dynamic coefficient matrix are implemented in the MATLAB function `Fcore7.m` on the accompanying Wiley web site.

10.4.5.6 Common Characteristics of INS Errors When inertial navigation technology was being developed in the United States after WWII, much of the practical and theoretical error modeling had already been done by German engineers and scientists working on aircraft and missile inertial instrumentation before and during the War.

Schuler Oscillations The coupling of position errors into gravity compensation errors in the INS implementation creates an oscillation of position errors with the same period as a satellite at the same altitude as the INS, called the *Schuler period.* It is named after Maximilian Schuler (1882–1972), who discovered this phenomenon in his 1923 analysis [14] of a gyrocompass designed by his cousin Hermann Anschütz-Kaempfe (1832–1931). The gyrocompass design uses a gyroscope held in a fixed orientation with respect to local vertical by a pendulum linkage. Schuler was able to show that the effective pendulum period of the supporting linkage would have to be around 84.4 min to keep lateral accelerations from interfering with its operation. When the same oscillation period was found in inertial navigation errors, they came to be called *Schuler oscillations*.

INS implementations are usually tested in the laboratory by initializing with velocity errors and looking for the characteristic Schuler oscillation that should result. The same test also serves to verify INS simulation software. Figure 10.21 is a plot of simulated INS errors generated using the MATLAB INS Toolbox from GPSoft [15]. This shows how an initial northward velocity error of 0.1 m/s excites Schuler oscillations in the INS navigation errors, and how Coriolis accelerations rotate the direction of oscillation—like a Foucault pendulum with Schuler period. The total simulated time is about 14 h, time enough for 10 Schuler oscillation periods.

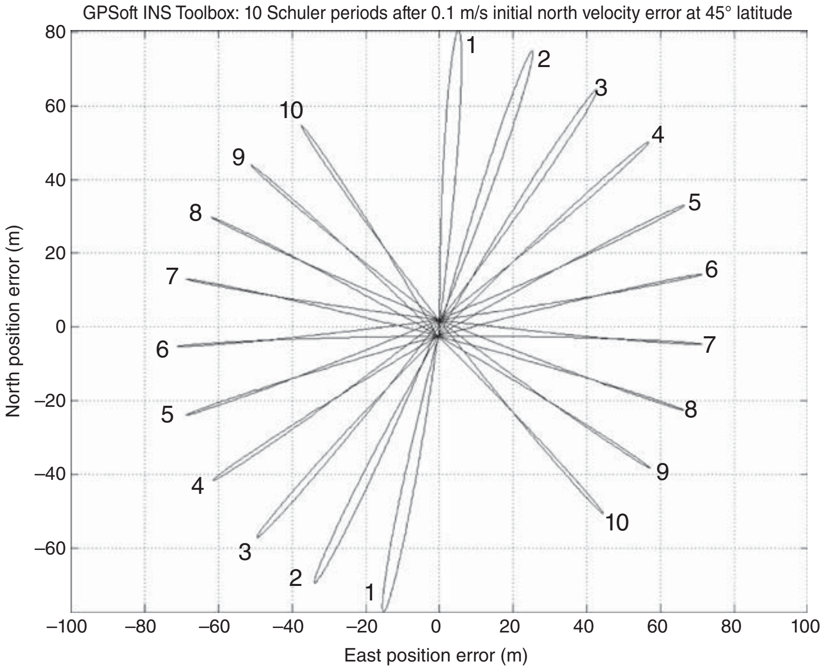


Figure 10.21 GPSoft INS Toolbox simulation of Schuler oscillations.

Vertical Channel Instability This is another INS error vulnerability that had been discovered before inertial navigation was reduced to practice. In 1946, when serious technical development of nuclear weapon delivery systems was just getting under way in the United States, Russian-born American scientist George Gamow (1904–1968) was part of a government technology advisory committee. He advised against military support for development of inertial navigation on the grounds that it was inherently unstable in the vertical direction. Gamow was correct, but he was eventually outmaneuvered by Charles Stark Draper, founder of the MIT Instrumentation Laboratory and considered to be the "father" of inertial navigation. Draper would demonstrate that the problem could be solved for aircraft navigation by aiding with a barometric altimeter, already standard equipment aboard most aircraft. This was perhaps the first time additional sensors were used for aiding INS navigation.

It is the theoretical gravity gradient model that causes the instability. In essence, having a positive altitude error causes an error in the calculated downward gravitational acceleration, which leads to exponential growth of the altitude error with an exponential time constant in the order of several minutes at sea level.

Performance Degradation over Time Inertial navigation without aiding (except in the vertical channel) is called *free-inertial* navigation. Other than horizontal Schuler oscillation and vertical channel instability, most free-inertial navigation errors do tend to degrade with time. This is not terribly serious for ballistic missile guidance, in which the total INS navigation/guidance phase lasts only for the first few minutes of launch. It is much more serious for "cruise" applications such as for navigation of aircraft and watercraft operating on or near Earth's surface.

CEP Rate For cruise applications, free-inertial navigation performance classes have been defined in terms of the rate at which navigation errors grow with time. The established measure of performance has been the rate of growth of the CEP (defined in Section 10.2.5.2). The unit of distance used is the *nautical mile* (defined in Section 10.2.5.1)), and the rate of growth is specified in *nautical miles per hour.*

INS Performance Classes In the 1970s the US Air Force defined three classes of INS performance:

High accuracy INS navigation accuracy degrades at 0.1 NMi/h CEP rate. This originally included systems designed for strategic military operations, although most of these systems eventually surpassed the specification.

Medium accuracy INS navigation accuracy degrades at 1 NMi/h CEP rate. This applied to most aircraft, including commercial aircraft—before GPS.

Low accuracy INS navigation accuracy degrades at 10 NMi/h CEP rate. This applied to short-range tactical missiles, for example.

All this would change when satellite navigation became available.

10.4.6 INS Sensor Compensation Parameter Errors

10.4.6.1 Inertial Sensor Calibration When the Kalman filter was introduced in the 1960s, it was soon put to use in determining parameters of the formulas used for compensating the sensors used in inertial navigation. It was already known that gimbaled systems could be calibrated in the laboratory (and in missile silos) by using their gimbals to orient the ISA in certain directions with respect to the local vertical (to control input acceleration) and Earth rotation axis (to control input rotation rates). Covariance analysis using the Riccati equation could then quantify INS performance as a function of sensor error characteristics and calibration procedures. This approach was soon extended to include self-calibration using many types of external navigation aids.

10.4.6.2 Sensor Compensation Parameter Errors Stanley F. Schmidt, who might be considered to be the "father" of INS integration with other sensor systems, was instrumental in developing a navigation system integrating INS and airborne radar for the C5A aircraft [16] in the mid-1960s.[14]

Starting in 1946, the Northrop Corporation[15] developed a series of cruise missiles with gimbaled INS, some of them with star trackers mounted on the stable element. The earliest designs used the star tracker to servo the stable element to keep it horizontal and aligned to true north. Later designs—including the navigator for the U-2 spycraft—would use a Kalman filter to correct sensor errors, as well.

Some of the work done in secrecy in the 1960s and 1970s would determine which types of auxiliary sensors could be used for keeping INS navigation errors in check by recalibrating the INS sensors on the fly. With the availability of a civilian channel on the GPS navigation satellites in the early 1990s, much of this technology would migrate to the public domain.

10.4.6.3 INS Sensor Calibration and Compensation An integral part of GNSS/INS integration is the ability to estimate and correct drifting values of the parameters used in compensating for sensor errors in inertial navigation. This capability reduces sensor stability requirements and costs. The following is a brief tutorial, including derivations of the models used for that purpose.

Sensor Calibration Because manufacturing tolerances on inertial sensors could not be made tight enough for the demands of navigation, it soon became standard practice to get the last few bits of accuracy by compensating the outputs of the sensors for any known residual error characteristics. This required a parametric model for the expected errors as a function of the sensor outputs and a calibration procedure in

[14] Most military applications of Kalman filtering during this era were classified, but this may have been the first use of the Kalman filter and other sensors to correct INS instrument errors during navigation. The integration of the Navy Transit navigation satellite system with ship-board INS systems also occurred in this time frame, but technical details are scant.

[15] The third corporation founded by aerospace technology pioneer Jack Northrop (1895–1981) and bearing his name.

which the outputs were measured with known inputs. A Kalman filter could then be used to estimate the parameters of the model, given the input/output data.

Affine Compensation Parameters Among the more common sensor errors requiring calibration and compensation in this way were terms such as

- Sensor bias, which equals the sensor output when there is zero sensor input.
- Sensor scale factor, which is the first-order variation of output with input.
- Sensor input axis misalignment errors.

These, it turns out, can all be modeled as an affine transformation at the level of three, nominally orthogonal sensors (gyroscopes or accelerometers) as

$$z_{\text{output}} = Mz_{\text{input}} + b, \tag{10.92}$$

where M is a 3×3 matrix and b is a column 3-vector. The values of M and b can be determined by least-squares or Kalman filter fitting of input–output pairs. Once M and b have been determined, the affine transformation is easily inverted to obtain the *error compensation formula* as an affine transformation:

$$z_{\text{input}} = Nz_{\text{output}} + d \tag{10.93}$$

$$N = M^{-1} \tag{10.94}$$

$$d = -Nb, \tag{10.95}$$

or the compensation model parameters N and d can be determined directly by least-squares fitting from the same set of input–output pairs.

Affine transformations are just the zeroth-order (bias) and first-order (scale factor and misalignment) terms in the power series expansion of the input/output relationship, and this same modeling and compensation procedure can be extended to any order. For many sensor designs and applications, however, the zeroth-order and first-order terms are the dominant errors.

There may also be other sensor output errors modeled as sensitivities of gyroscopes to acceleration, accelerometers to rotation rates, or both to temperature. These, too are generally represented by parametric models, the parameters of which can be calibrated under controlled conditions.

10.4.6.4 Drifting Compensation Parameters

Parameter Drift Inertial sensors are subject to all sorts of errors, many of which can be modeled, calibrated, and compensated. Sensor failure modes may include abrupt changes in input/output characteristics due to subcomponent failures. Some high reliability INS designs have included redundant sensors, and software to detect suspected failures of this sort and switch to using the remaining sensors. Beyond that, there is a class of slowly creeping input/output characteristics attributed to "compensation parameter drift."

Causes and Remedies Many sensor error compensation parameters (especially those of the more expensive sensors) have relatively stable values, but some may tend to drift slowly but unpredictably over time. This drifting may be due to a number of factors, including ambient temperature variations. Temperature sensitivities can be compensated to some degree by calibrating the effect as a function of sensor temperature and compensating for the error by using one or more temperature sensors on the ISA during operation. But some drift phenomena are not so easily understood and modeled, and are often simply attributed to "aging" of sensors. This may be due, in part, to slow creep of materials due to internal stresses from manufacturing processes, but such processes are generally not sufficiently predictable to be modeled and compensated for. Among those sensor error compensation parameters commonly subject to such "aging" are biases and scale factors. In the next section, we will use scale factor and bias parameters to demonstrate how GNSS/INS integration can actually correct for drift in the compensation model parameter values. For that purpose, we will need to model these parameters in terms of how they couple into navigation solution error.

Drift Dynamic Modeling The best models come from analysis of the actual drift data from sensors under operating conditions. It would be useful, for example, to know whether the actual drift resembles a truly constant parameter, an exponentially correlated process, or a random walk, and to know in the latter cases what is the approximate driving noise covariance and correlation time. For example, for some sensor technologies, input axis directions tend to be relatively stable compared to biases and scale factors (especially for sensors with limited temperature control).

Random Walk Models If we assume random walk models for just the biases and scale factors, for example, the dynamic models for the compensation parameter errors will have the form

$$\varepsilon_{ca} \overset{\text{def}}{=} \begin{bmatrix} \varepsilon_{saE} \\ \varepsilon_{saN} \\ \varepsilon_{saU} \\ \varepsilon_{baE} \\ \varepsilon_{baN} \\ \varepsilon_{baU} \end{bmatrix} \tag{10.96}$$

$$\frac{d}{dt}\varepsilon_{ca} = w_{ca}(t) \tag{10.97}$$

$$\underset{W_{ca}}{\mathrm{E}} \langle w_{ca}(t) \rangle = 0 \tag{10.98}$$

$$\underset{W_{ca}}{\mathrm{E}} \langle w_{ca} w_{ca}^{\mathrm{T}} \rangle = Q_{ca} \tag{10.99}$$

$$\varepsilon_{c\omega} \overset{\text{def}}{=} \begin{bmatrix} \varepsilon_{s\omega E} \\ \varepsilon_{s\omega N} \\ \varepsilon_{s\omega U} \\ \varepsilon_{b\omega E} \\ \varepsilon_{b\omega N} \\ \varepsilon_{b\omega U} \end{bmatrix} \tag{10.100}$$

$$\frac{d}{dt}\varepsilon_{c\omega} = w_{c\omega}(t) \tag{10.101}$$

$$\underset{W_{c\omega}}{\mathrm{E}} \langle w_{c\omega}(t) \rangle = 0 \tag{10.102}$$

$$\underset{W_{c\omega}}{\mathrm{E}} \langle w_{c\omega} w_{c\omega}^{\mathrm{T}} \rangle = Q_{c\omega}, \tag{10.103}$$

$$\tag{10.104}$$

where the subscripts

$_{ca}$ refers to accelerometer compensation parameter errors,
$_{sa}$ refers to accelerometer scale factor errors,
$_{ba}$ refers to accelerometer bias errors,
$_{c\omega}$ refers to rate gyro compensation parameter errors,
$_{s\omega}$ refers to rate gyro scale factor errors,
$_{b\omega}$ refers to rate gyro bias errors,

and the subscript post indices $_E$, $_N$, and $_U$ refer to the individual input axes. This choice would imply an ENU gimbaled system, but the same sort of indexing can be done using host/carrier-fixed axes for a strapdown system, or similar axes for carouseled systems.

Determining Model Parameters The covariance matrices of dynamic disturbance noise Q_{ca} and $Q_{c\omega}$ will be diagonal if the random walk processes are independent for the different sensor compensation parameters. However, because performance is generally improved if the model parameters are close to correct, it is usually worth verifying independence by analyzing the actual drift histories. Intermediate integrated GNSS/INS navigation results can provide the drift histories for such analysis.

10.4.6.5 Compensation Error Coupling into Navigation Errors Figure 10.22 illustrates how errors in scale factor and bias parameters create navigation errors. Figure 10.22(a) is for accelerometers, and Figure 10.22(b) is for gyroscopes. The resulting errors in acceleration and attitude rate would then be represented in algebraic form as

$$\varepsilon_a = \underbrace{\begin{bmatrix} D_a & I_3 \end{bmatrix}}_{F_{24}} \begin{bmatrix} \varepsilon_{sa} \\ \varepsilon_{ba} \end{bmatrix} \tag{10.105}$$

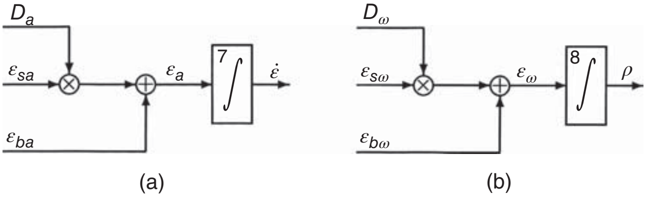


Figure 10.22 Sensor scale factor and bias compensation error flows.

$$D_a = \mathrm{diag}[a_E,\ a_N,\ a_U] \tag{10.106}$$

$$\varepsilon_\omega = \underbrace{\begin{bmatrix} D_\omega & I_3 \end{bmatrix}}_{F_{35}} \begin{bmatrix} \varepsilon_{s\omega} \\ \varepsilon_{b\omega} \end{bmatrix} \tag{10.107}$$

$$D_\omega = \mathrm{diag}[\omega_E,\ \omega_N,\ \omega_U], \tag{10.108}$$

where the significance of F_{24} and F_{35} is explained below.

Accelerometer compensation errors couple into acceleration errors, which would be input to the box numbered “7” in Figure 10.20. Rate gyro compensation errors couple into attitude rate errors, which would be input to the box numbered “8” in Figure 10.20.

10.4.6.6 Augmented Dynamic Coefficient Matrix Adding just the 12 state variables for bias and scale factor drift on each of the six sensors (three for acceleration and three for attitude rate) to the nine core navigation error state variables would yield a 21 × 21 augmented dynamic coefficient matrix:

$$F_{\mathrm{aug}} = \begin{bmatrix} 0_{3\times3} & I_3 & 0_{3\times3} & 0_{3\times6} & 0_{3\times6} \\ F_{21} & F_{22} & F_{23} & F_{24} & 0_{3\times6} \\ F_{31} & F_{32} & F_{33} & 0_{3\times6} & F_{35} \\ 0_{6\times3} & 0_{6\times3} & 0_{6\times3} & 0_{6\times6} & 0_{6\times6} \\ 0_{6\times3} & 0_{6\times3} & 0_{6\times3} & 0_{6\times6} & 0_{6\times6} \end{bmatrix} \tag{10.109}$$

$$F_{24} = \begin{bmatrix} a_E & 0 & 0 & 1 & 0 & 0 \\ 0 & a_N & 0 & 0 & 1 & 0 \\ 0 & 0 & a_U & 0 & 0 & 1 \end{bmatrix} \tag{10.110}$$

$$F_{35} = \begin{bmatrix} \omega_E & 0 & 0 & 1 & 0 & 0 \\ 0 & \omega_N & 0 & 0 & 1 & 0 \\ 0 & 0 & \omega_U & 0 & 0 & 1 \end{bmatrix} \tag{10.111}$$

for the random walk parameter drift model. As we have mentioned before, the acceleration and attitude rate vector components must be in sensor input-axis coordinates for this to work. The “ENU” coordinates are only correct for a locally level ENU gimbaled implementation. Otherwise, the input accelerations and attitude rates must be transformed from navigation coordinates to sensor input axis coordinates, then back to navigation coordinates. In practice, the outputs of the sensors would work for the first part, but the resulting acceleration and rotation rate errors need to be transformed back to navigation coordinates (which, in this case, are locally level ENU).

10.4.6.7 Strapdown and Carouseled Model Differences The above derivations of compensation drift error models are for gimbaled systems in which ISA-fixed coordinates are east, north, and up, in which case the inputs of the sensors are in navigation (NAV) coordinates.

For carouseled or α-wander gimbaled implementations, the ISA axes are rotated through a known angle α about the vertical axis, in which case the ISA-to-NAV coordinate transformation matrix is

$$C_{\text{ISA}\to\text{NAV}} = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{10.112}$$

In that case, the only difference will be in the definitions of the submatrices F_{24} (Equation 10.109) and F_{35} (Equation 10.110) of the system error dynamic coefficient matrix as

$$F_{24} = C_{\text{ISA}\to\text{NAV}} \begin{bmatrix} a_E & 0 & 0 & 1 & 0 & 0 \\ 0 & a_N & 0 & 0 & 1 & 0 \\ 0 & 0 & a_U & 0 & 0 & 1 \end{bmatrix} \tag{10.113}$$

$$F_{35} = C_{\text{ISA}\to\text{NAV}} \begin{bmatrix} \omega_E & 0 & 0 & 1 & 0 & 0 \\ 0 & \omega_N & 0 & 0 & 1 & 0 \\ 0 & 0 & \omega_U & 0 & 0 & 1 \end{bmatrix}, \tag{10.114}$$

where the acceleration a in D_a and ω in D_ω are now in sensor-fixed ISA coordinates, which can be implemented using the compensated outputs of the accelerometers (in ISA coordinates) before their transformation to NAV coordinates. The required value of $C_{\text{ISA}\to\text{NAV}}$ is already part of the navigation solution.

For strapdown implementations, the sensor-fixed inputs are also in sensor-fixed (ISA) coordinates, which are no longer the NAV coordinates. But the ISA-to-NAV coordinate transformation matrix $C_{\text{ISA}\to\text{NAV}}$ is still part of the navigation solution. In this case, that will be the value of $C_{\text{ISA}\to\text{NAV}}$ to be used in Equations 10.112 and 10.113.

10.4.7 MATLAB Implementations

10.4.7.1 Initial Navigation Uncertainties Initial navigation errors do have a significant impact of the ensuing CEP rates determined during testing. Standard INS performance testing typically includes the approved initial alignment and navigation initialization for the system under test, which we have not defined here.

As an alternative, we have assumed a standard, relatively accurate, initial navigation solution in all cases. The RMS uncertainties for this initialization error budget are listed in Table 10.4.

TABLE 10.4 RMS Initial Navigation Uncertainties

Navigation Variable	RMS Value	Units
Easting	2	(m)
Northing	2	(m)
Altitude	2	(m)
East Velocity	10^{-1}	(m/s)
North Velocity	10^{-1}	(m/s)
Altitude Rate	10^{-1}	(m/s)
East Tilt	10^{-8}	(rad)
North Tilt	10^{-8}	(rad)
Heading	10^{-8}	(rad)

These have been made quite small so that the resulting estimates of CEP rates are influenced more by INS error budgets and less by initialization errors. That luxury is generally not allowed in practice, however.

10.4.7.2 Taming Vertical Channel Instability This implementation uses a barometric altimeter to stabilize the altitude of an INS otherwise operating in "free-inertial" mode. That is, the INS navigates in 3D but uses altimeter measurements to estimate and correct its vertical position and velocity errors. The barometer is assumed to be corrected for local air pressure variations by using another barometer at a fixed location to telemeter the local correction. The resulting RMS altitude bias error is assumed to be 2 m with a correlation time constant of 1 h, plus RMS white noise of 0.1 m. The INS is assumed to be of "medium accuracy," having a CEP rate in this augmented mode of 1 NMi/h—as shown in Figure 10.23.

The full 22-state error model (9 navigation errors, plus 12 sensor compensation errors, plus 1 altimeter bias error) used in the Riccati equation for assessing performance with and without the barometer is listed in the m-file `MediumAccuracyINS.m` on the Wiley web site. This evaluates RMS vertical channel errors during a simulated test on a 100-km figure-8 track, the results of which are plotted in Figure 10.24. This shows the resulting RMS altitude uncertainty over 4 h, with and without barometer aiding. The aided altitude uncertainty settles at around 2 m, whereas the RMS uncertainty for the unaided case climbs to more than 10^4 km in 4 h. The horizontal error also grows unreasonably without vertical channel stabilization.

With vertical channel stabilization, the computed CEP grows as shown in Figure 10.23, a least-squares straight-line fit to which yields an estimated CEP rate of 0.96 NMi/h for this particular simulated test trajectory with the assumed initial navigation uncertainties. This plot also exhibits a significant Schuler frequency component, which one might expect from initial navigation errors.

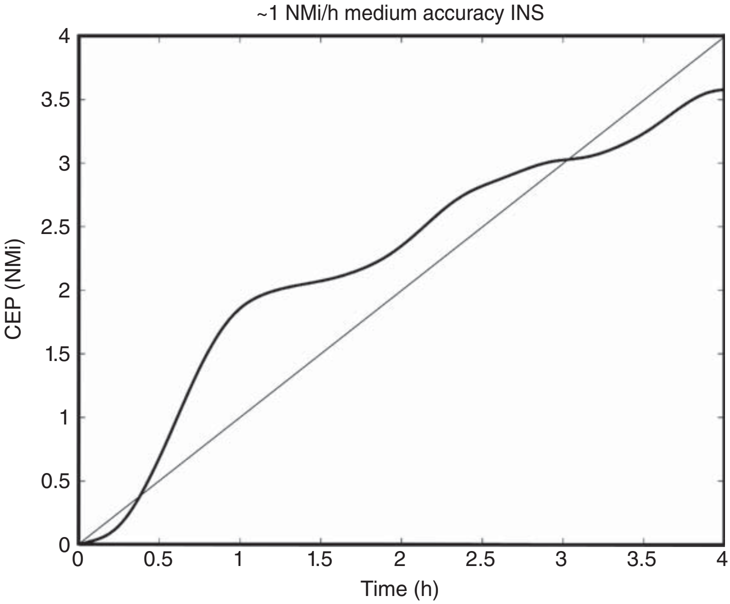


Figure 10.23 Medium accuracy INS: horizontal uncertainties versus time.

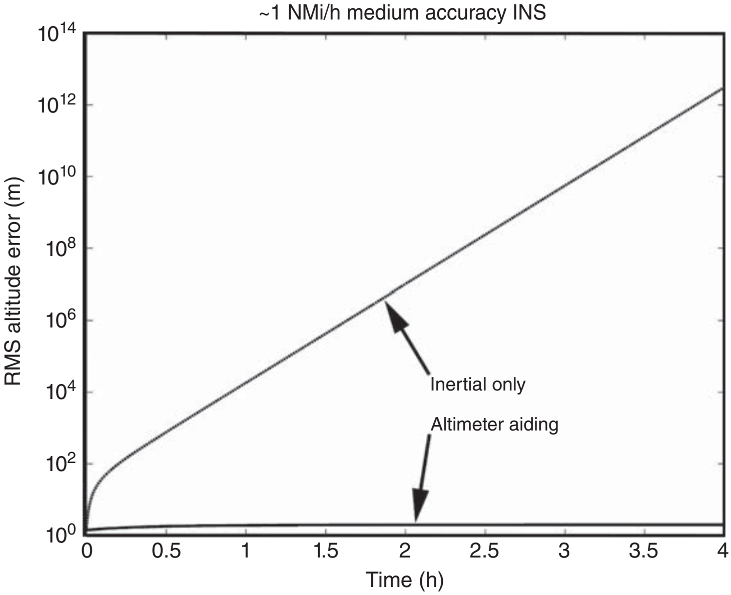


Figure 10.24 Medium accuracy INS: RMS altitude uncertainties versus time.

10.4.7.3 INS Error Budgeting Using Covariance Analysis Table 10.5 lists tentative error budgets[16] for what might be considered “high” (0.1 NMi/h CEP rate), “medium” (1 NMi/h CEP rate), and “low” accuracy (10 NMi/h CEP rate), inertial navigators, along with performance specifications for the barometric altimeter system used for stabilizing vertical navigation. The middle error budget has been used for demonstrating the efficacy of this form of vertical channel stabilization, shown in Figure 10.24.

The three m-files

`INSErrBud0point1NMiPerHr.m`	(≈ 0.1 NMi/h CEP Rate)
`INSErrBud1NMiPerHr.m`	(≈ 1 NMi/h CEP Rate)
`INSErrBud10NMiPerHr.m`	(≈ 10 NMi/h CEP Rate)

on the Wiley web site are designed to demonstrate how varying individual parameters in the corresponding INS error budgets will influence the resulting INS CEP rate in a specific dynamic simulation.

The common dynamic simulation is a 4-h run at 100 kph on a 100-km figure-8 test track, using a barometric altimeter for stabilizing vertical navigation.

Outputs include plots of CEP versus time and results of scaling individual error budget terms up and down by an order of magnitude. These error budgets include seven that are specific to the INS sensors, three related to the barometric altimeter used for altitude stabilization, and one (the intersample time period) related to simulation conditions. The standard values used in the m-files are listed in Table 10.5 for the three error budgets. The standard files used for saving the respective error budget as mat-files are also listed at the top of the table. These files can then be downloaded for use by other m-files.

The analysis in this case consists of estimating the resulting up- or down-scaling of the CEP rate when each INS error budget term is scaled up and down by an order of magnitude. Only the first seven error budget values are varied in the analysis, because the other values are not related to the INS. The eighth is the intersample interval and the rest are related to the auxilliary barometric altimeter used for vertical channel stabilization. Gyro-related parameters are specified in degrees per hour in the table but converted to radians per second internally for the simulations.

Example 10.3 (Error Budget Analysis for Medium Accuracy INS) This example uses the medium accuracy error budget from `INSErrBud1NMiPerHr.m`, which yields the CEP history shown in Figure 10.23 and the error budget analysis shown in Figure 10.25, which is a semi-log plot of the relative scaling effects on CEP as a function of parameter number.

[16] These error budgets are not based on actual INSs. They are only hypothetical examples which just happen to have the approximate CEP rates of high, medium, and low accuracy performance on the 100-km figure-8 test track simulator.

TABLE 10.5 **Baseline INS Error Budgets for 0.1, 1, and 10 NMi/h CEP Rates**

Parameter Sequence No.	Parameter Description	Units	CEP Rate		
			0.1	1	10
			`INSEBHiAcc.mat`	`INSEBMedAcc.mat`	`INSEBLoAcc.mat`
1	RMS acceleration noise	m/s/sqrt(s)	10^{-6}	10^{-4}	10^{-4}
2	RMS gyro noise	deg/hr/sqrt(s)	10^{-4}	10^{-2}	10^{-2}
3	Sensor compensation error correlation time	s	15	600	1800
4	RMS Accel. Bias Err.	m/s/s	9.8×10^{-5}	9.8×10^{-5}	9.8×10^{-4}
5	RMS acceleration scale factor error	part/part	10^{-5}	10^{-5}	10^{-4}
6	RMS gyro bias error	deg/hr	10^{-4}	0.03	0.19
7	RMS gyro scale factor error	part/part	10^{-4}	10^{-4}	10^{-3}
8	intersample interval	s	1	1	1
9	RMS altimeter error	m	2	2	2
10	altimeter bias error correlation time	s	3600	3600	3600
11	RMS altimeter noise	m/sqrt(s)	0.1	0.1	0.1

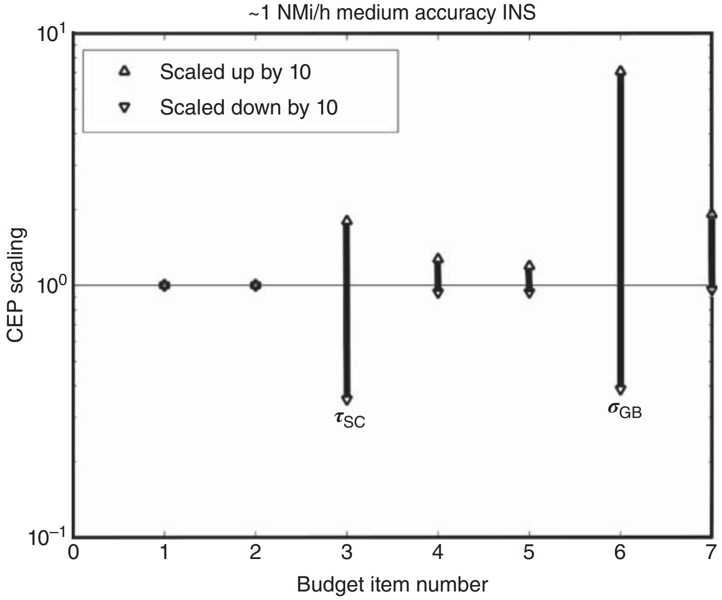


Figure 10.25 Relative CEP scaling from varying INS error budget parameters.

Varying the first two INS error budget parameters,

1. RMS Accelerometer Noise
2. RMS Gyro Noise

up and down by an order of magnitude has essentially no impact on CEP rate. This would indicate that these are already so small that they have little impact on CEP rate. From this, one might surmise that they can be derated by an order of magnitude or more with little or no impact on performance.

Varying the third parameter (sensor compensation error correlation time, τ_{SC}) up or down by a factor of 10 has a significant impact on CEP rate in both directions. Error correlation times are not that easy to change, however. Variation of performance with correlation time is not necessarily monotonic, either. Figure 10.26 is a plot of CEP rate as a function of τ_{SC} over several orders of magnitude. It shows that there are two values of τ_{SC} at which the resulting CEP rate is 1 NMi/h, with a peak in between. This type of dependence of behavior on correlation time is not that unusual.

The next two INS error budget parameters,

1. RMS accelerometer bias error
2. RMS accelerometer scale factor error,

show more downside performance degradation from loosening their values, but little upside gain from tightening them.

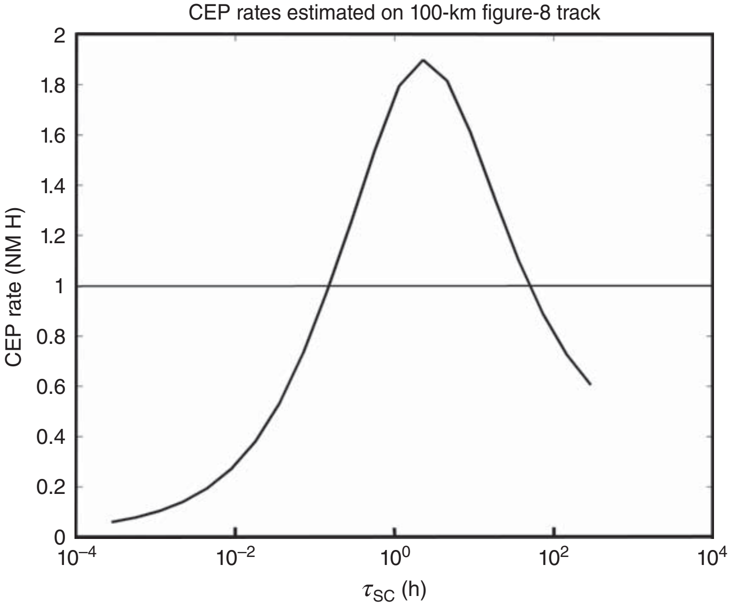


Figure 10.26 CEP rate versus t_{SC}.

The sixth parameter (RMS gyro bias error, σ_{GB}) shows about the same relative gain from tightening the requirement as τ_{SC}, and considerably more downside performance degradation from loosening it. This is perhaps the dominating error source, and the one to which one would look first to improve overall performance.

The final parameter, RMS gyro scale factor error, shows more downside gain from loosening it than upside gain from tightening it.

Error budget sensitivities for the higher and lower accuracy INS systems can be studied by running `INSErrBud0point1NMiPerHr.m` and `INSErrBud10NMiPerHr.m`, respectively.

These sensitivities of performance to error budget variations provide useful insight to the system designer, in terms of

1. What are the critical sensor performance parameters, in terms of their impact on expected INS CEP rate?
2. Where can sensor performance requirements be relaxed without much impact on navigation performance?
3. Where can tightening sensor performance requirements make the greatest improvement in navigation performance?
4. How can system cost be lowered without sacrificing performance?
5. How can performance be improved with the least cost?

These are always critical design considerations in the development of inertial sensors and systems.

10.5 GNSS/INS INTEGRATION

Stochastic models for INS and GNSS navigation errors were derived in the previous two sections. Integrated GNSS/INS navigation uses Kalman filters with models of those types plugged into the relevant parameter matrices F (or Φ), Q, H, and R. But we first need to establish some nomenclature used in GNSS/INS integration technologies.

10.5.1 Background

10.5.1.1 A Very Short History The earliest attempts at GNSS/INS integration were for the Navy Transit navigation satellite system used by Navy submarines in the 1960s, but much of this work had been cloaked in secrecy and lost to history. At around the same time, INS systems aboard military aircraft were being integrated with auxiliary sensors such as airborne radar, star trackers, LORAN, or Omega.

The earliest hardware demonstrations with Transit's successor began in the 1980s with GPS, before it was fully operational. In order to maintain lower technical and schedule risks, these began by using existing inertial navigators with minimal hardware modifications to demonstrate potential improvements in Air Force weapon delivery accuracies. It was less risky to proceed by stages, going from low technical complexity and cost to higher technical complexity and cost. Test results did confirm what had already been demonstrated in computer simulations with much more sophisticated implementation equations. Military applications of GPS/INS integration were already well established before there were opportunities for civilian applications.

10.5.1.2 Loosely and Tightly Coupled Implementations GNSS/INS integration architectures have been loosely labeled as "loosely coupled" or "tightly coupled" according to the degree to which the Kalman filter implementation alters the internal workings of each subsystem from its stand-alone configuration, as illustrated in Figure 10.27.

The most loosely coupled implementations treat the standard outputs from each subsystem (i.e., GNSS or INS) as a measurement. The integrating Kalman filter combines their outputs by treating each as an independent sensor. It may improve navigation accuracy while GNSS is available, but it defaults to free-inertial accuracy when it is not available.

The more tightly coupled implementations alter the internal workings of the INS and GNSS receiver in a number of ways:

- Using nonstandard outputs of either system, such as the pseudoranges from GNSS receivers or the sensed accelerations from an INS.

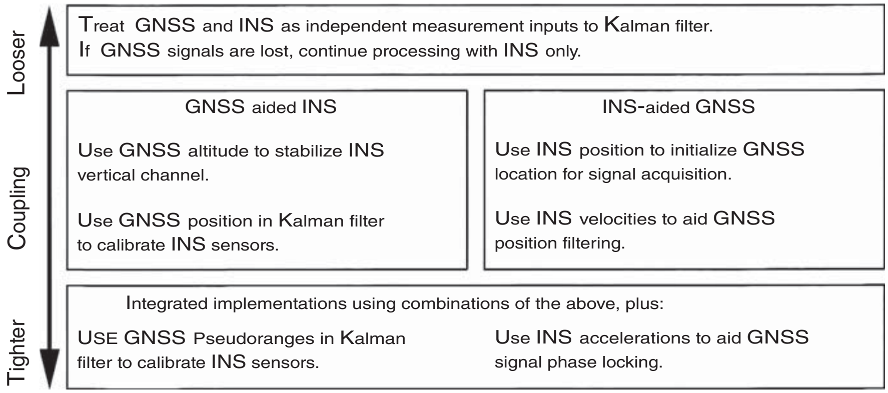

Treat GNSS and INS as independent measurement inputs to Kalman filter.
If GNSS signals are lost, continue processing with INS only.
GNSS aided INS
Use GNSS altitude to stabilize INS vertical channel.
Use GNSS position in Kalman filter to calibrate INS sensors.
INS-aided GNSS
Use INS position to initialize GNSS location for signal acquisition.
Use INS velocities to aid GNSS position filtering.
Integrated implementations using combinations of the above, plus:
USE GNSS Pseudoranges in Kalman filter to calibrate INS sensors.
Use INS accelerations to aid GNSS signal phase locking.
Looser
Coupling
Tighter

Figure 10.27 Loose/tight GSNN/INS integration strategies.

- Using nonstandard inputs, such as for controlling slew rates of Doppler-tracking frequency sources in the receiver by using sensed acceleration from the INS or altering the internal sensor compensation in the INS by using estimates of inertial sensor calibration errors derived from the integrating Kalman filter. The latter allows continuous recalibration of the INS using GNSS, so that the navigation solution is still good whenever GNSS signals are lost, and GNSS signal reacquisition is faster when signals do become available. Also, satellite selection can be improved by using the attitude and attitude rate information from the INS to predict which satellites will be within the GNSS receiver antenna pattern as the host vehicle maneuvers.

The main advantage of the more tightly coupled implementations is that they generally perform better than more loosely coupled implementations. One reason for this is that the Kalman filter models for tightly coupled implementations are generally more faithful representations of the actual hardware.

Another factor favoring more tightly coupled implementations is that their performance is generally less sensitive to the quality (and cost) of INS sensors. Historically, the INS had been the more expensive subsystem, so this can make a big difference in total system cost. On the other hand, the more tightly coupled implementations generally require redesign of the GNSS receiver and/or INS to supply and/or use the additional information needed for integration. Any additional costs for such design changes must be included in the cost/benefit analysis.

10.5.1.3 Loosely Coupled Implementations These are mostly of historical interest today, because tightly coupled implementations offer the same benefits—and more.

1. *INS Position Initialization.* Medium and high accuracy inertial navigators are capable of sensing their latitude, but not their longitude. Manual entry of initial longitude is discouraged, because consequences of entry errors can be severe. GNSS-derived position is more reliable.

2. *INS Altitude Stabilization.* From the very beginning of inertial navigation, barometric altimeters have been used for stabilizing altitude error, as illustrated in Example 10.4.7.2. The function of the altimeter in this implementation can be replaced by a GNSS receiver, and the resulting performance can be modeled and analyzed in much the same way as for the altimeter. Both auxiliary sensors (altimeter or GNSS) have time-correlated errors, but GNSS errors tend to have shorter correlation times.
3. *INS-Aided GNSS Satellite Signal Phase Tracking.* GNSS receivers need to correct for satellite signal Doppler shifts, due to relatively high satellite velocities.[17] In GNSS stand-alone receivers, frequency-lock loops and phase-lock loops are used to maintain signal lock-on. These types of control loops have limited dynamic capabilities, and they can easily loose signal lock during violent maneuvers of high-g vehicles such as missiles or unmanned autonomous vehicles. In integrated GNSS/INS implementations, accelerations sensed by the INS can be used as an aid in maintaining signal lock-on during such maneuvers. This type of GNSS receiver aiding can significantly improve signal lock-on during such maneuvers. Because attitude maneuvers of the host vehicle could place the direction to a satellite outside the antenna pattern of a GNSS antenna mounted on the surface of a host vehicle, highly maneuverable vehicles generally require more than one antenna to maintain signal lock. In that case, attitude and attitude rate information from the INS can also aid the antenna-switching implementation.

Example 10.4 (INS Vertical Channel Damping with GNSS) We can model the dynamics of altitude error ε_h in inertial navigation systems as

$$\frac{d}{dt}\varepsilon_h = \dot{\varepsilon}_h \tag{10.115}$$

$$\frac{d}{dt}\dot{\varepsilon}_h = \frac{1}{\tau_S}\varepsilon_h + \varepsilon_a(t) \tag{10.116}$$

$$\frac{d}{dt}\varepsilon_a(t) = \frac{-1}{\tau_a}\varepsilon_a(t) + w(t) \tag{10.117}$$

$$\tau_S \approx 806.4 \text{ s}, \tag{10.118}$$

where we have added an exponentially correlated random disturbance noise term $\varepsilon_a(t)$ to represent the slow drift of vertical accelerometer bias and scale factor. The correlation time constant of $\varepsilon_a(t)$ is τ_a, and the variance of the zero-mean white noise process $w(t)$ will be

$$Q_{ta} = \frac{2\,\sigma_a^2}{\tau_a}, \tag{10.119}$$

where σ_a^2 is the mean-squared accelerometer error.

[17] Some few GNSS satellites may be in geosynchronous orbits. In order to maintain global coverage, however, most of the satellites will be in lower orbits with relatively high inclination angles to the equator.

The state-space model for the vertical channel in continuous time is then

$$x = \begin{bmatrix} \varepsilon_h \\ \dot{\varepsilon}_h \\ \varepsilon_a \end{bmatrix} \tag{10.120}$$

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{1}{\tau_S} & 0 & 1 \\ 0 & 0 & \frac{-1}{\tau_a} \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ w(t) \end{bmatrix} \tag{10.121}$$

for the zero-mean white noise process $w(t) \in \mathcal{N}(0, Q_{ta})$.

The state-transition matrix for discrete time intervals Δt is then

$$\Phi = \exp\ (F\ \Delta t) \tag{10.122}$$

$$= \begin{bmatrix} \phi_{1,1} & \phi_{1,2} & \phi_{1,3} \\ \phi_{2,1} & \phi_{2,2} & \phi_{2,3} \\ 0 & 0 & \lambda_a \end{bmatrix} \tag{10.123}$$

$$\phi_{1,1} = \frac{\lambda_S^2 + 1}{2\ \lambda_S} \tag{10.124}$$

$$\phi_{1,2} = \frac{\tau_S(\lambda_S^2 - 1)}{2\ \lambda_S} \tag{10.125}$$

$$\phi_{1,3} = \frac{\tau_a \tau_S{}^2(\tau_S + \tau_a - 2\ \tau_a \lambda_a \lambda_S + \tau_a \lambda_S^2 - \tau_S \lambda_S^2)}{2\ \lambda_S(\tau_a^2 - \tau_S^2)} \tag{10.126}$$

$$\phi_{2,1} = \frac{(\lambda_S - 1)(\lambda_S + 1)}{2\ \tau_S \lambda_S} \tag{10.127}$$

$$\phi_{2,2} = \frac{\lambda_S^2 + 1}{\lambda_S} \tag{10.128}$$

$$\phi_{2,3} = \frac{\tau_S \tau_a(-\tau_S - \tau_a + 2\ \tau_S \lambda_a \lambda_S + \tau_a \lambda_S^2 - \tau_S \lambda_S^2)}{2\ \lambda_S(\tau_a^2 - \tau_S^2)} \tag{10.129}$$

$$\lambda_S = \exp\ (\Delta t / \tau_S) \tag{10.130}$$

$$\lambda_a = \exp\ (-\Delta t / \tau_a). \tag{10.131}$$

The corresponding process noise covariance in discrete time,

$$Q_a \approx \Delta t\ Q_{ta} \tag{10.132}$$

$$\approx \frac{2\ \Delta t\ \sigma_a^2}{\tau_a}. \tag{10.133}$$

The *k*th altitude outputs from the INS and GNSS receiver will be

$$\hat{h}_k^{[\mathrm{INS}]} = h_k^{[\mathrm{TRUE}]} + \varepsilon_h \tag{10.134}$$

$$\hat{h}_k^{[\mathrm{GNSS}]} = h_k^{[\mathrm{TRUE}]} + w_k, \tag{10.135}$$

respectively, where $h_k^{[\mathrm{TRUE}]}$ is the true altitude and w_k is the GNSS receiver altitude error.

The difference

$$z_k = \hat{h}_k^{[\mathrm{INS}]} - \hat{h}_k^{[\mathrm{GNSS}]} \tag{10.136}$$

$$= \varepsilon_h - w_k \tag{10.137}$$

$$= \underbrace{\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}}_{H} \hat{x}_k - w_k \tag{10.138}$$

can then be used as a "pseudomeasurement" in a Kalman filter to estimate x. The true altitude estimate will be

$$\hat{h}_k^{[\mathrm{TRUE}]} = \hat{h}_k^{[\mathrm{INS}]} - \hat{x}_{k,\,1}, \tag{10.139}$$

where $\hat{x}_{k,\,1}$ is the first component of the state vector $\hat{x}_k$ of Equation 10.120.

Characterizing Altitude Error Versus Accelerometer Error The variance of the altitude estimate obtained in this way will be the element $p_{1,1}$ of the covariance matrix P of the Riccati equation in the Kalman filter. The steady-state solution of the continuous form of the Riccati equation can also be solved for value of $p_{1,1}$. The resulting steady-state RMS altitude uncertainty is plotted in Figure 10.28 as a function of RMS accelerometer error, for RMS GNSS altitude error equal to 20 m and its correlation time constant equal to 5 min. This shows the trade-off between altitude error and the quality (and cost) of the vertical channel accelerometer.

The MATLAB m-file `AltStab.m` on the companion Wiley web site simulates and plots the time histories of RMS-stabilized and "unstable" (i.e., unaided by GNSS) altitude error for the same range of RMS accelerometer noise levels as in Figure 10.28. These are Monte Carlo simulations, so the results should be different each time it is run.

10.5.1.4 Tightly Coupled Implementations These use a full INS error propagation model, including the effects of sensor compensation errors. As a minimum, they estimate and correct any changes in the navigation solution, as well as any changes in the sensor compensation parameters.

The stochastic model for unpredictable vehicle dynamics is replaced by the INS, which measures vehicle dynamics. However, the full navigation solution now includes INS attitude and sensor compensation parameters, which add state variables to be estimated. GNSS/INS integration generally increases the computational load.

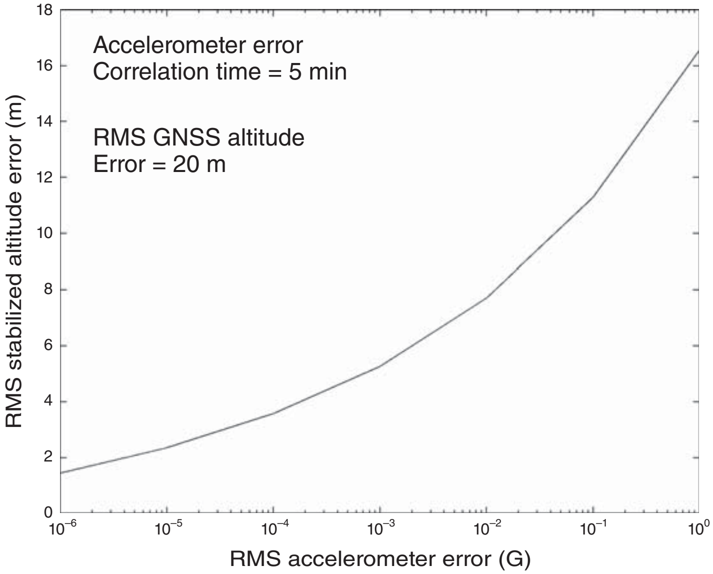


Figure 10.28 GNSS-aided INS altitude uncertainty versus RMS accelerometer error.

Keep in mind that there is no universal one-size-fits-all model for integrated GNSS/INS navigation, because there is no universal GNSS or INS. Although the underlying navigation principles may be universal, each physical realization has its own error characteristics, and each has its own models for those error characteristics. There are fundamental differences between single-frequency and dual-frequency GNSS receivers, for example, and there are fundamental differences between inertial navigators with different sensor suites. For example, Figure 10.29 shows some possible structures of dynamic coefficient matrices in Kalman filters for GNSS-only navigation (left column) and integrated GNSS/INS navigation (right column), and for single-frequency GNSS receivers (top row) and dual-frequency GNSS receivers (bottom row). Furthermore, the particulars of the F-matrix for integrated GNSS/INS navigation will depend very much on the sensor qualities and layout (e.g., gimbaled or strapdown) of the INS.

10.5.1.5 Benefits of INS Integration INS performance during GNSS outages can be improved by recalibrating the INS whenever GNSS is available. The approach shown here treats each subsystem (GNSS and INS) as an independent sensor, but with nonstandard outputs:

1. The GNSS receiver measures pseudoranges, which are sensitive to
   (a) True position of the vehicle antenna.
   (b) Uncompensated signal propagation delays (mostly due to the ionosphere).
   (c) Receiver clock time errors.

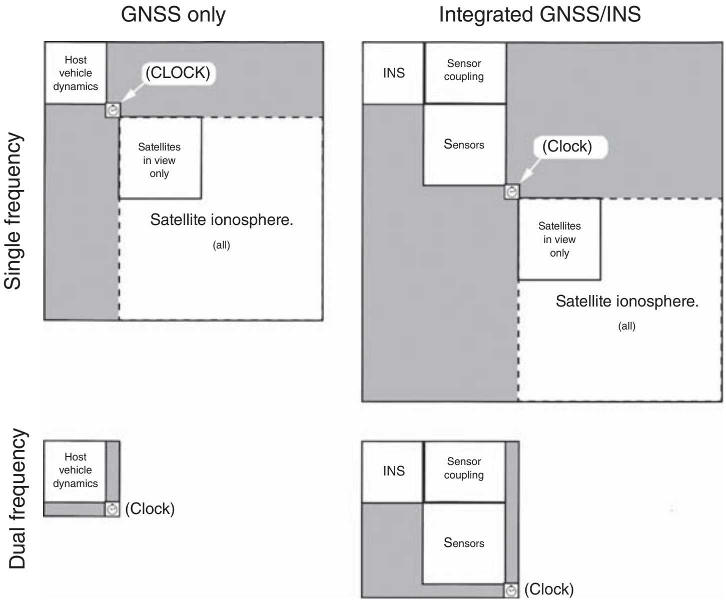


Figure 10.29 Dynamic coefficient matrices for GNSS and integrated GNSS/INS.

2. The INS measures the "vehicle state," which includes three components each of position, velocity, acceleration, attitude, and attitude rate. However, what the INS is actually "measuring" is the sum of true vehicle state plus INS navigation errors and sensor errors. Its outputs are then sensitive to

   (a) The true vehicle state, including the true position of the GNSS antenna.

   (b) INS navigation errors, which have known dynamics.

   (c) INS sensor errors, which cause navigation errors.

   All three of these are coupled together dynamically in ways that can be exploited by Kalman filtering. The filter can use the correlations caused by dynamic coupling to estimate causes (e.g., sensor errors) as well as their effects (e.g., navigation errors) by using independent measurements from GNSS. This enables the Kalman filter to recalibrate an INS in a properly integrated GNSS/INS implementation.

All these are combined into one Kalman filter representing the stochastic dynamics of all the component parts, and how they interact.

10.5.2 MATLAB Implementations

In order to illustrate the general approach to integrated GNSS/INS navigation, we present here some specific examples, based on the models derived in Sections 10.3 and 10.4, using the dynamic simulators described in Section 10.2.7.

10.5.2.1 Gimbaled INS/GNSS Integration

Example 10.5 (Gimbaled INS and Dual-Frequency GPS Receiver) This example uses the same INS and GNSS models used for INS-only and GPS-only navigations with the same 100-km figure-8 test track simulator, so that the three navigation modes can be compared.

The MATLAB m-file `GNSSINS1.m` uses a matrix structure like that shown in the lower right in Figure 10.29 for the F matrix of its dynamic model, and the same process noise values used for medium accuracy INS and dual-frequency GPS in previous examples. The resulting plot of RMS horizontal position uncertainty for a 4-h excursion on the 100-km figure-8 test track simulator is shown in Figure 10.30. The uncertainty level shifts during periods when GNSS signals are available are likely due to DOP changes as satellites come into and go from view.

For comparison, Figure 10.31 is a multiplot of RMS horizontal position uncertainties from all three navigation modes:

1. Dual-frequency GPS only.
2. Medium accuracy INS.
3. Both the above, integrated.

In all three cases, there was one minute without GPS signals near the end of each hour. The integrated GPS/INS shows a reduced effect from those 1-min outages.

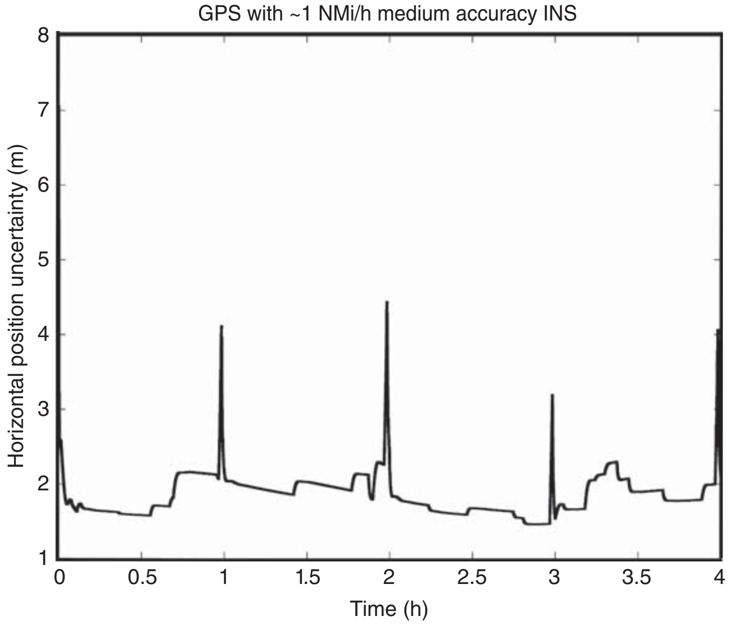


Figure 10.30 Integrating 1 NMi/h INS with dual-frequency GPS.

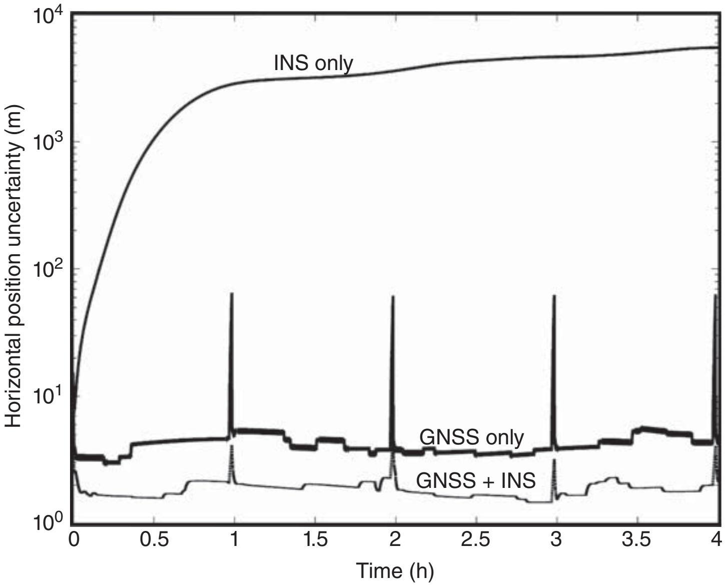


Figure 10.31 Comparing three navigation modes.

Note that the outages result in horizontal errors in the order of 5-m RMS horizontal error build-up for GNSS/INS integration, but in the order of 60-m RMS horizontal error build-up for GNSS-only navigation—an improvement of more than an order of magnitude. In the GNSS-only case, it is the host vehicle dynamics model that determines how fast navigation errors deteriorate when signals are lost.

10.5.2.2 INS-Aided GNSS Satellite Signal Phase Tracking

Satellite Signal Tracking GNSS receivers need to correct for satellite signal Doppler shifts, due to relatively high satellite velocities.[18]

Receiver Tracking Loops In GNSS stand-alone receivers, frequency-lock loops, and phase-lock loops are used to maintain signal lock-on. These types of control loops have limited dynamic capabilities, and they can easily loose signal lock during violent maneuvers of high-g vehicles such as missiles or unmanned autonomous vehicles.

INS-Aided Signal Tracking In integrated GNSS/INS implementations, accelerations sensed by the INS can be used as an aid in maintaining signal lock-on during such maneuvers. This type of GNSS receiver aiding can significantly improve signal lock-on during such maneuvers.

[18] Some few GNSS satellites may be in geostationary orbits. In order to maintain global coverage, however, most of the satellites will be in lower orbits with relatively high inclination angles to the equator.

GNSS Receiver Antenna Switching Because attitude maneuvers of the host vehicle could place the direction to a satellite outside the antenna pattern of a GNSS antenna mounted on the surface of a host vehicle, highly maneuverable vehicles generally require more than one antenna to maintain signal lock. In that case, attitude and attitude rate information from the INS can also aid the antenna switching implementation.

10.5.2.3 Model Modifications for Strapdown INS/GNSS Integration We have used a gimbaled INS with sensor axes aligned with ENU navigation coordinates in the derivations of INS error models for GNSS/INS integration.

As we have mentioned in Sections 10.4.4.3, 10.4.5.1, and 10.4.5.4, the error model modifications required for other INS implementations are related to the INS ENU models by the coordinate transformation matrix

$$C_{\text{NAV}}^{\text{SEN}}$$

from sensor-fixed coordinates to navigation coordinates—which is computed as an essential part of any INS implementation.

The dynamic coefficient matrix F_{CORE} for navigation error propagation remains the same as before, as does the dynamic coefficient matrix F_{SC} for errors in the sensor compensation parameters.

The only changes occur in the submatrices F_{24} and F_{35} of the dynamic coupling matrix between sensor compensation parameter errors and navigation errors, as illustrated in Figure 10.32.

These changes will depend on the particulars of the sensor compensation parameters. For the bias and scale factor compensation parameters used in the derivations

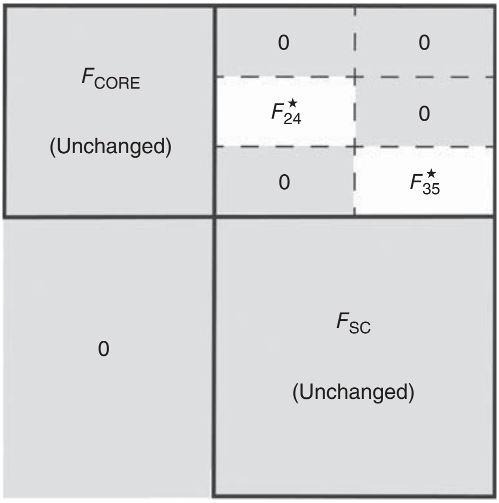


Figure 10.32 Changes to dynamic coefficient matrix for strapdown model. * Changed.

above, the changes are relatively simple. In that case, the new values are

$$F_{24}^{\star} = \begin{bmatrix} C_{\text{NAV}}^{\text{SEN}} & C_{\text{NAV}}^{\text{SEN}} D_{a^{\star}} \end{bmatrix} \tag{10.140}$$

$$a^{\star} = a_{\text{SEN}} \tag{10.141}$$

$$F_{35}^{\star} = \begin{bmatrix} C_{\text{NAV}}^{\text{SEN}} & C_{\text{NAV}}^{\text{SEN}} D_{\omega^{\star}} \end{bmatrix} \tag{10.142}$$

$$\omega^{\star} = \omega_{\text{SEN}}, \tag{10.143}$$

where a_{SEN} and ω_{SEN} are the compensated outputs of the respective (acceleration and rotation rate) sensors, and D_v represents a diagonal matrix with diagonal entries specified by the vector v.

For dynamic simulation in which the acceleration (a) and rotation rate (ω) are simulated in navigation coordinates, the simulated values

$$a_{\text{SEN}} = (C_{\text{NAV}}^{\text{SEN}})^{\text{T}} a_{\text{NAV}} \tag{10.144}$$

$$\omega_{\text{SEN}} = (C_{\text{NAV}}^{\text{SEN}})^{\text{T}} \omega_{\text{NAV}}, \tag{10.145}$$

where a_{NAV} and ω_{NAV} are the simulated dynamic conditions and the matrix $C_{\text{NAV}}^{\text{SEN}}$ (or its transpose) is generated as part of the attitude simulation.

As a rule, performance of GNSS/INS integrated navigation will be different for strapdown and gimbaled inertial systems with the same sensor performance specifications. The major differences are due to the different inputs the sensors will experience in the two (strapdown or gimbaled) implementations. The potential dynamic range of rotational inputs is generally much greater for strapdown systems, and strapdown systems also have more opportunities to sense part of the supporting forces used to counter gravity (the major sensible acceleration in many applications). As a consequence, strapdown systems may have greater observability of sensor compensation errors when a GNSS navigation solution is available.

Also, due to particulars of compensation parameter error dynamic coupling, carouseled or indexed gimbaled systems generally have better performance than systems maintaining a north alignment.

10.6 SUMMARY

1. Kalman filters have had an enormous impact on the accuracy, efficiency, and efficacy of navigation. Many of the newer navigation systems (e.g., GNSS or integrated GNSS/INS) were designed to use Kalman filters and were designed using Kalman filters.
2. When a satellite navigation system is being designed for a specific type of host vehicle (e.g., cargo ship, automobile, and fighter aircraft), its Kalman filter design can improve navigation performance by using a stochastic dynamic model tailored to the maneuvering capabilities of the intended host vehicle.

3. The design possibilities for integrating satellite navigation with inertial navigation are only limited by the available inputs and outputs of the GNSS receiver and INS and by the extent to which the integration designer can alter the inner workings of each subsystem (GNSS receiver or INS). This has led to a variety of integration approaches:
   (a) *Loosely coupled* implementations are more conservative in terms of how much the integration mechanization alters the standard inputs and outputs of each subsystem.
   (b) *Tightly coupled* implementations can alter the more intimate details of the subsystem implementations and generally perform better than loosely coupled implementations.
4. We have provided detailed, illustrative examples of linear stochastic system models for GNSS and INS error dynamics and shown how these models behave under representative dynamic conditions.
5. We have derived some example models for specific types of inertial sensor errors. These do not begin to cover the range of possible error types encountered in inertial sensors, but they should serve as examples of how model derivations are done.
6. An error budget for any sensor system (including inertial navigators) is an allocation of the individual performances of its various subsystems so as to meet the overall performance of the system. This process allows some trade-offs, in which making some components less accurate can be compensated by making others more accurate. There may be overall performance standards for a navigation system, but this does not specify how these allocations of performances of subsystems is to be decided. This allows some freedom for minimizing costs or improving other performance metrics.
7. Real error budgets are based on real hardware with known, empirically derived error models. The example error budgets used here are strictly hypothetical, however. Sensor vendors and developers are better sources for sensor error characteristics, and well-vetted commercial MATLAB toolboxes for GNSS and INS navigation are likely sources for more reliable software.
8. The Riccati equation of the Kalman filter is an indispensable tool for the design of navigation systems, especially those integrating sensors with very different error characteristics such as GNSS and INS. The Riccati equation allows us to predict the performance of a particular system design as a function of subsystem and component error characteristics. Designers can then search for the combination of components to satisfy a specified set of performance characteristics at minimum system cost.
9. The performance of integrated GNSS/INS navigation systems is not very sensitive to the stand-alone performance of the INS, which means that relatively low cost INS technologies can be used to bring down the total integrated system cost. This technology development is currently driving down the cost for high accuracy integrated GNSS/INS systems for applications such as pilotless aircraft, driverless automobiles and trucks, automated surface mining, automated

control of farm equipment, and automated grading for roads. The potential economic payoff for this capability is substantial, and Kalman filtering will play an important part in this development.

10. GNSS and INS navigation problems provide good examples of Kalman filter applications to real-world physical systems, for which the Kalman filter models must be derived from the physics of the systems they represent, demonstrating Kalman's maxim that "Once you get the physics right, the rest is mathematics."

PROBLEMS

10.1 Equation 10.9 has formulas for vertical and horizontal dilution of precision. How would you break down horizontal dilution of precision (HDOP) into components for east dilution of precision (EDOP) and north dilution of precision (NDOP)?

10.2 Formulate the GPS plant model with three position errors, three velocity errors, and three acceleration errors and the corresponding measurement model with pseudorange and delta pseudorange as measurements.

10.3 Run the MATLAB program `ClockStab.m` with clock stability values of 10^{-9}, 10^{-8}, 10^{-7}, and 10^{-6}. This is for the best of conditions: a stationary receiver with good satellite geometry. How is the estimate of clock drift C_d affected by clock stability under the best of conditions?

10.4 Run the MATLAB program `SatelliteGeometry.m` with the following two sets of satellite directions:

Set	Satellite	Elevation	Azimuth
1	1	0	0
	2	0	120
	3	0	240
	4	90	0
2	1	45	0
	2	45	90
	3	45	180
	4	45	270

Which has better performance? Can you explain why?

10.5 Answer the first three questions at the end of Example 10.3.

10.6 Why would gyrocompass alignment fail at the north and south poles?

10.7 Run the MATLAB m-file `AltStab.m` for simulating GNSS-aided altitude stabilization. Follow the instructions displayed at the end to display the roots of the tenth-order polynomial equation for the steady-state mean-squared altitude estimation uncertainty as a function of RMS accelerometer noise. Successive columns represent $10^{-6},\ 10^{-5},\ 10^{-6},\ \dots,\ 10^{0}$ g RMS accelerometer noise, and the rows are the polynomial roots. Can you guess which roots are the “correct” values? If so, plot their square roots (i.e., the steady-state RMS altitude uncertainty) versus the RMS accelerometer noise σ_a. How do these results compare to the final plot displayed?

10.8 Add clock errors to the model in `GNSSshootoutNCE.m`. Assume 10^{-9} relative clock stability over one second. (See how it is done in Example 10.1.) Compare your results with those in Table 10.3.

10.9 Solve Equation 10.27 for P as a function of the elements of F and Q.

10.10 In Figure 10.11, RMS horizontal position error is bounded whenever GNSS signals are lost. Can you explain why?

REFERENCES

[1] L. J. Levy, “The Kalman filter: navigation’s integration workhorse,” *GPS World*, Vol. 8, No. 9, pp. 65–71, 1997.

[2] M. S. Grewal, A. P. Andrews, and C. G. Bartone, *Global Positioning Systems, Inertial Navigation and Integration*, 3rd ed., John Wiley & Sons, Inc., New York, 2013.

[3] A. Lawrence, *Modern Inertial Technology: Navigation, Guidance, and Control*, 2nd ed., Springer, New York, 1998.

[4] D. H. Titterton and J. L. Weston, *Strapdown Inertial Navigation Technology*, 2nd ed. IEE and Peter Peregrinus, UK, 2004.

[5] W. S. Widnall and P. A. Grundy, *Inertial Navigation System Error Models*, Technical Report TR-03-73, Intermetrics, Cambridge, MA, 1973.

[6] J. B.Y. Tsui, *Fundamentals of Global Positioning System Receivers: A Software Approach*, 2nd ed., John Wiley & Sons, Inc., New York, 2004.

[7] D. Sobel, *Longitude: The True Story of a Lone Genius Who Solved the Greatest Scientific Problem of His Time*, Penguin, New York, 1995.

[8] D. J. Allain and C. N. Mitchell, “Ionospheric delay corrections for single-frequency GPS receivers over Europe using tomographic mapping,” *GPS Solutions*, Vol. 13, No. 2, pp. 141–151, 2009.

[9] J. Klobuchar, “Ionospheric time-delay algorithms for single-frequency GPS users,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-23, No. 3, pp. 325-331, 1987.

[10] H. Dekkiche, S. Kahlouche, and H. Abbas, “Differential ionosphere modelling for single-reference long-baseline GPS kinematic positioning,” *Earth Planets Space*, Vol. 62, 915–922, 2010.

[11] K. C. Redmond and T. M. Smith, *From Whirlwind to MITRE: The R&D Story of the SAGE Air Defense Computer*, MIT Press, Cambridge, MA, 2000.

[12] K. R. Britting, *Inertial Navigation Systems Analysis*, Artech House, Norwood, MA, 2010.

[13] P. G. Savage, *Strapdown Analytics*, Vol. 2, Strapdown Associates, Maple Plain, MN, 2000.

[14] M. Schuler, "Die Störung von Pendel-und Kreiselapparaten durch die Beschleunigung der Fahrzeuges," *Physicalische Zeitschrift*, Vol. B, p. 24, 1923.

[15] GPSoft, *GPSoft Inertial Navigation System Toolbox for Matlab*, Version 3.0, GPSoft, Athens, OH, 2007.

[16] S. F. Schmidt, "The Kalman filter—its recognition and development for aerospace applications," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 4, No. 1, pp. 4–7, 1981.