CHAPTER 6

Design and performance analysis of Kalman filters

6.1 INTRODUCTION

This chapter seeks to exploit the Kalman filter algorithm to its fullest potential. To do so will encompass a complete depiction of a systematic design procedure, practical aspects of implementation, and development of software tools to provide performance analysis capability for any Kalman filter configuration operating in the real world environment. Extensive examples are the vehicle for developing and emphasizing the essential points of designing efficient, practical filters, and these examples are drawn from the application area that has probably exploited Kalman filtering the most: optimally aided inertial navigation systems. This is by no means the only important applications area, but concentrating attention on one particular problem area will allow a more extensive portrayal of filter design than would a superficial look at many different contexts. For additional applications, see Leondes [28] and Schmidt [47].

6.2 THE REQUISITE OF ENGINEERING JUDGMENT

Chapter 5 may have left the impression that an optimal filter can be generated automatically once a body of applied mathematics has been mastered. Despite the mathematical formalism of the Kalman filter approach, a substantial amount of engineering insight and experience is required to develop an effective operational filter algorithm.

As mentioned in Chapter 5, a mathematical model of both the system structure (state dynamics and output relations) and uncertainties is inherently embodied in the Kalman filter structure. Attaining an *adequate mathematical model upon which to base the filter* is the crux of the design problem. Even once

a particular model is chosen as appropriate for a given application, a considerable effort remains: obtaining appropriate *numerical evaluation* of coefficients (especially covariance matrix elements) within the model. This process is called "tuning" a given Kalman filter, and it involves an iterative search for the coefficient values that yield the best estimation performance possible from that particular filter structure.

Moreover, the design must meet the constraints of online computer time, memory, and wordlength required. These considerations dictate a philosophy of using as simple a filter as possible that yields adequate results (i.e., meeting performance specifications). Consequently, the designer must be able to exploit basic modeling alternatives to achieve a simple but adequate filter, adding or deleting model complexity as the performance needs and practical constraints require.

Evaluation of *true* performance capabilities of simplified, reduced order filters is of critical importance in the design procedure. Although a Kalman filter computes an error covariance internally, this is a valid depiction of the true errors committed by the filter only to the extent that the filter's own system model adequately portrays true system behavior. It is very possible for the filter not to perform as well as it "thinks" it does. If the computed error covariance is inappropriately small, so is the computed gain: the filter weights its internal system model too heavily and discounts the data from the "real world" too much, leading to filter estimates not corresponding to true system performance, with a simultaneous indication by the filter through its computed covariance that all is well, a condition called filter divergence.

Moreover, numerical precision and stability problems can corrupt performance substantially, especially when the filter is implemented on a short wordlength computer. This motivates consideration of alternative algorithm formulations, such as square root filters (to be discussed in Chapter 7).

Evaluation of true performance capabilities involves both covariance analyses and Monte Carlo simulations, which will be discussed in Section 6.8. To achieve a valid portrayal, the designer must fully understand the assumptions that underly a statistical analysis of performance. He must investigate the effects of nonwhite noises, non-Gaussian noises, neglected nonlinearities, and approximations used in achieving a model form compatible with the Kalman filter assumptions. Moreover, testing and simulation experience with the actual digital implementation is crucial before the software is finalized.

In certain applications, the basic Kalman filter structure itself has to be modified. For example, sensor data (rather noisy) is often available more frequently than the established sample rate of the filter algorithm (especially since most sensors are analog devices). In this case, one might consider prefiltering of the data to smooth out the noise, rather than just sample it periodically and neglect the information available between sample instants. One can in fact generate an equivalent single data value that incorporates this information,

but it is not the simple replacement of raw data by an averaged signal that first occurs to an engineer (this will be pursued later in Section 6.5). Other examples would be artificially limiting the filter's "memory" of past data, setting lower bounds on acceptable variance evaluations, discarding measurement samples that fail reasonableness tests, and adaptively setting filter gains.

The design of an effective operational Kalman filter entails an iterative process of proposing alternative designs through physical insights, tuning each, and trading off performance capabilities and computer loading. A systematic procedure for accomplishing such a design, as developed in Section 6.9, will accentuate the need to bring engineering judgment to bear on the overall filter development.

6.3 APPLICATION OF KALMAN FILTERING TO INERTIAL NAVIGATION SYSTEMS

To consider basic aspects of filter implementation in the context of realistic applications, the next sections of this chapter will concentrate upon applying a Kalman filter to inertial navigation systems. Fundamental concepts will be developed in this section, followed by three specific examples in succeeding sections.

A conventional gimbaled inertial measurement unit [2, 4, 25, 40, 41] consists of a platform suspended by a gimbal structure that allows three degrees of rotational freedom, as depicted in Fig. 6.1. From the geometry of the configuration, it is possible to attach the outermost gimbal to the body of some vehicle, and to allow that vehicle to undergo any change in angular orientation while maintaining the platform fixed with respect to some desired coordinate frame. Gyros on the platform sense the angular rate of the platform with respect to inertial space, and their outputs are sent through electronics to the torquer motors on the gimbal structure, commanding them to maintain a desired platform orientation regardless of the orientation of the outermost gimbal. Feedback control loops that keep the gyro outputs nulled will maintain the platform fixed with respect to inertial space; additional computed inputs can be added to these loops to maintain some other orientation, such as north–east–down corresponding to the current location of the vehicle. These feedback loops are such that, in practice, the platform orientation is kept essentially stable regardless of the most violent vehicle maneuvering.

Thus the platform remains aligned with respect to a known reference coordinate system. Accelerometers on the platform then provide vehicle acceleration with respect to that known set of reference coordinates. Actually, accelerometers measure specific force, so local gravity must be computed and subtracted appropriately from these sensor outputs to obtain a measurement of actual vehicle acceleration. The resulting signals can be integrated (or pulses counted

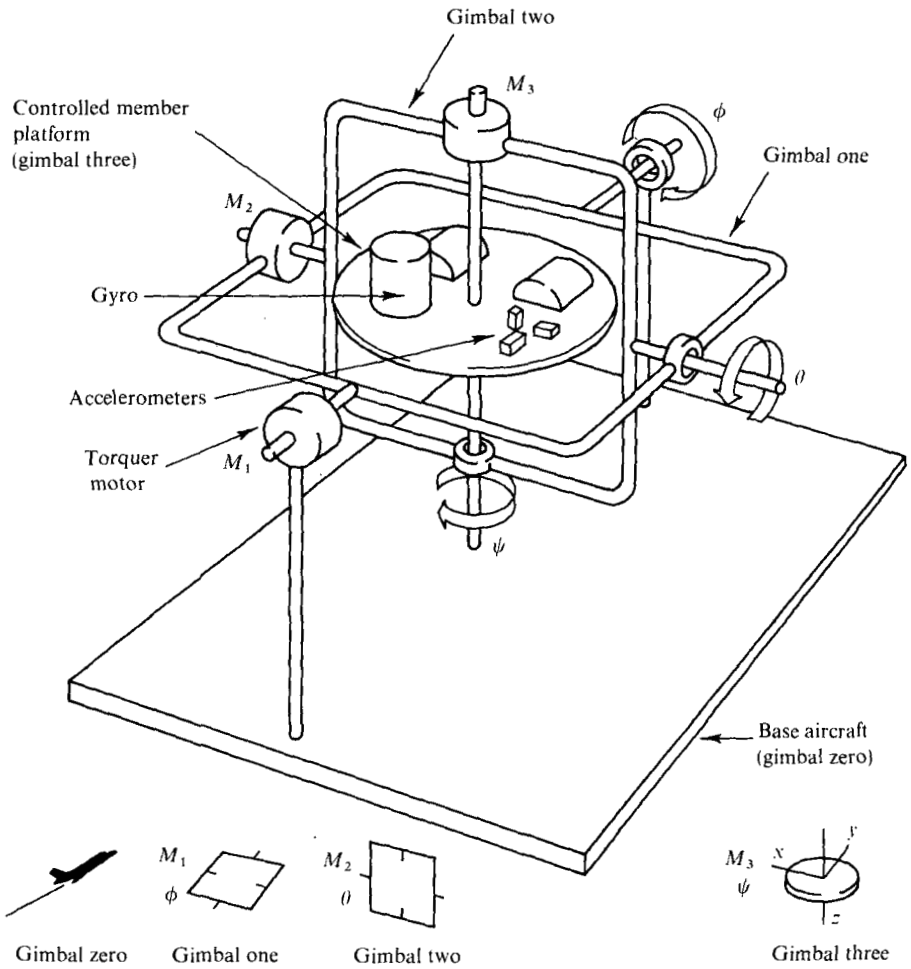


FIG. 6.1 Typical gimbaled inertial measurement unit. Modified from *Gyroscopic Theory, Design and Instrumentation* by Wrigley, Hollister, and Denhard. © 1969. Used with the permission of The M.I.T. Press.

if the signal format is a pulse rate proportional to acceleration) to yield vehicle velocity and position.

Moreover, the gimbal angles, the angles formed between the various members of the gimbal support structure, provide direct readout of the Euler angles to describe the vehicle's angular orientation. Thus, the inertial navigation system (INS) provides attitude information as well as translational information.

The question naturally arises, why does this instrument require optimal aiding by other navigation sensors? Due to the tight control loops just described, an INS provides very good high frequency information. However, because of

gyro characteristics, the system drifts at a slow rate: the long term, or low frequency, content of the data is poor. All inertial systems have position errors that grow slowly with time, and these errors are unbounded. A typical INS might be classed as a "one nautical mile per hour system," meaning that after one hour of operation, the position error standard deviation has grown to one nautical mile.

Some external source of data, such as radio navigation aid position information or Doppler velocity, would naturally be considered as a means of bounding or damping these errors. As opposed to an INS, most other navigation aids provide data which is good on the average (i.e., low frequency), but subject to considerable high frequency noise, due to instrument noise, atmospheric effects, antenna oscillation, unlevel ground effects, and so forth.

Given an INS and other sources of data, one would want to combine the available information in an optimal manner if possible, efficiently providing an estimate of navigation parameters that is best with respect to some criterion. Some earlier navigation systems reset the INS to agree with the other data sources, essentially declaring these other sources perfect and discarding the information previously held in the INS. The Kalman filter approach is instead to use the statistical characteristics of the errors in both the external information and the inertial components to determine this "optimal" combination of information. Actually, the filter *statistically* minimizes the errors in the estimates of the navigation parameters: on an ensemble average basis, no other means of combining the data will outperform it, *assuming* the internal model in the filter is adequate. Once the problem and system model are completely specified, the Kalman filter algorithm systematically provides this optimal estimate.

It will be seen that, although the filter is designed in the time domain, it does in fact weight each information source most heavily in the frequency regime where it provides good information and suppress each in the frequency region where it is most prone to errors. In other words, the filter will use the good low frequency data from the external sources to damp out the slowly growing errors inherent in the INS. Because of these differing sensor characteristics and the existence of well-developed adequate system models, there is substantial benefit to be gained by applying Kalman filtering to aiding INS systems.

Typical *external information sources* would include

Position data:

(1) radar—onboard and/or ground based;
(2) radio navigation aids: TACAN, LORAN, OMEGA, VOR/DME;
(3) Global Positioning System (GPS) navigation satellites;
(4) position fixes: star sightings, landmarks;
(5) radiometric area correlation (comparison of a radiometric "picture" of terrain to a stored map);
(6) laser ranging.

Velocity data:

(1) Doppler radar,
(2) indicated airspeed from the air data system.

Altitude data:

(1) barometric altimeter,
(2) radar altimeter,
(3) laser altimeter.

The filter for combining these data sources with an INS is typically a digital computer algorithm that uses sampled data (with sample period on the order of 5–60 sec) to maintain estimates of approximately 10–20 state variables. For instance, the navigation system filter designed for the B-1 bomber (which will serve as a basis for future designs) consists of a 13-state horizontal plane subsection and an independent 4-state vertical channel component, operating with a 6-sec sample period.

There are two very important aspects of implementation of a Kalman filter in conjunction with inertial systems (and other applications): *total state space versus error state space formulation* (also denoted as *direct* versus *indirect* filtering in navigation literature), and *feedforward versus feedback mechanizations* [3]. These aspects will now be discussed.

As the name indicates, in the *total state space* (*direct*) formulation, total states such as vehicle position and velocity are among the state variables in the filter, and the measurements are INS accelerometer outputs and external source signals. In the *error state space* (*indirect*) formulation, the *errors* in the INS-indicated position and velocity are among the estimated variables, and each measurement presented to the filter is the *difference* between INS and external source data.

Consider first the *total state space filter*, as depicted in Fig. 6.2. In this direct configuration, the Kalman filter is *in* the INS loop. The INS accelerometer signals and external source data both feed into the filter, which provides not only the desired navigation information, but also the appropriate commands to the gimbal torquer motors to maintain the platform aligned with the chosen coordinate frame. The angular orientation of the platform then dictates what accelerations will be sensed by each of the accelerometers attached to it.

The benefits of such a configuration are that the available information is weighted optimally rather than operated upon by fixed gains and integrators. Optimal time-varying gains can provide substantial improvement in performance over the classical approaches of resets or fixed gain updating, such as reducing INS gyrocompass (initial alignment) time from about 30 to 7 min while meeting the same precision specification.

However, there is a very serious drawback to this implementation. Being in the INS control loop and using the total state space representation, the filter

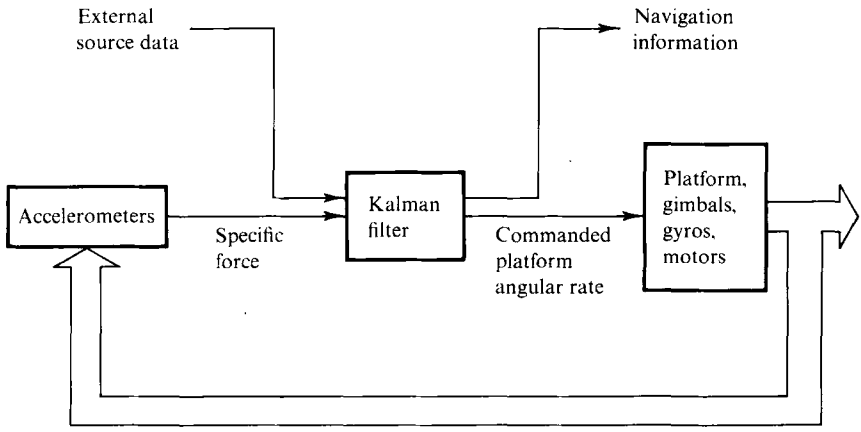


FIG. 6.2 Total state space (direct) Kalman filter.

would have to maintain explicit, accurate awareness of vehicle angular motion as well as attempt to suppress noisy and erroneous data. Sampled data usage requires a sampling rate of at least twice the frequency of the highest frequency signal of interest for adequate reconstruction of the continuous-time system behavior (by the Shannon sampling theorem; engineering practice tends toward five to ten times the highest frequency signal). Admitting aircraft roll rate capability on the order of 400 deg/sec, the filter would need a very fast sample rate and would have to perform all computations within this short sample period. Moreover, in most cases, the Kalman filter is allocated only a small portion of the capabilities of a central processor, and it is run "in the background" at a lower priority than more critical algorithms, such as digital stability and control programs. It is impossible to implement a high order Kalman filter practically on state-of-the-art computers and meet this sample rate requirement.

Not only are the dynamics involved in the total state space description composed of high frequency content, but they are well described only by a nonlinear model. The development of a Kalman filter is predicated upon an adequate linear system model, and such a total state space model does not exist.

Another drawback to this design is that, if the filter should happen to fail (as by a temporary computer failure), the entire navigation system fails: the inertial system *cannot* operate without the filter. From a reliability standpoint, it would be desirable to provide an emergency degraded performance mode in case of such failure.

As a result of these considerations, the direct mechanization is restricted to alignment, calibrations, bias determination in laboratory testing, certain submarine applications (involving slower dynamics), and the like. Section 6.6 will pursue this subject further.

The *error state space (indirect) Kalman filter* estimates the errors in the navigation and attitude information using the difference between INS and external source data. The INS itself follows the high frequency motions of the vehicle very accurately, and there is no need to model these dynamics explicitly in the filter. Instead, the dynamics upon which the filter is based is the set of inertial system error propagation equations, which are relatively well developed, well behaved, low frequency (Schuler 84-min mode dominant), and very adequately represented as linear. Because the filter is out of the INS loop and is based on low frequency linear dynamics, its sample rate can be much lower than that of a direct filter. In fact, an effective indirect filter can be developed with a sample period on the order of half a minute, thereby achieving practicality with respect to the amount of computer time required. For these reasons, the error state space formulation is used in essentially all (except submarine) terrestrial aided inertial navigation systems.

Besides state space differences among Kalman filters, there are two distinct types of implementations, feedforward and feedback. The *indirect feedforward* version is depicted in Fig. 6.3. From this diagram, it can be seen that the filter compares the two sets of data and uses the result to estimate the errors in the inertial system. By subtracting these estimated errors from the inertial data, the onboard computer maintains the optimal estimates of position, velocity, and attitude. The inertial system operates as though there were no aiding: it is "unaware" of the existence of the filter or the external data, so if either should fail, the unaltered INS information would still be available. Herein lies the disadvantage of the feedforward approach though. Acceptable Kalman filter performance depends upon the adequacy of a *linear* dynamics model, so it is necessary for the errors in the inertial system to remain of small magnitude. However, the INS is free to drift with unbounded errors, thereby invalidating this basic assumption.

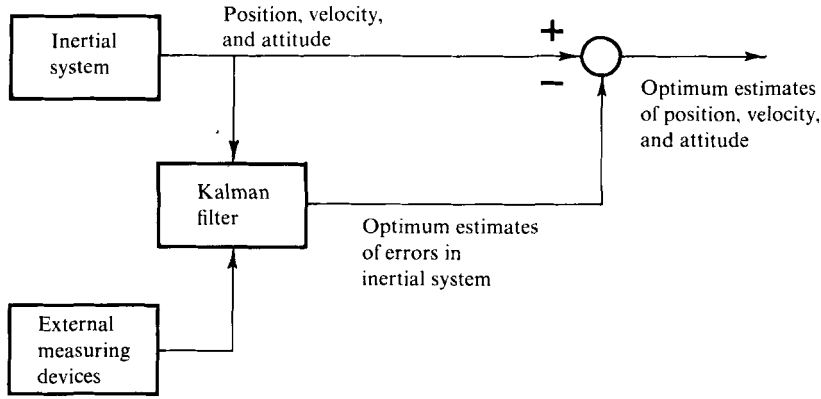


FIG. 6.3 Indirect feedforward Kalman filter.

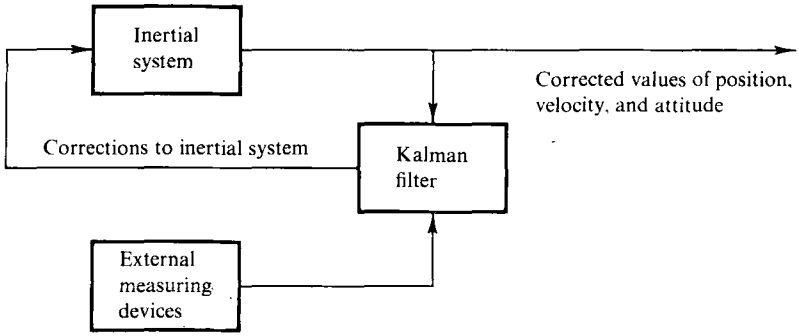


FIG. 6.4 Indirect feedback Kalman filter.

Thus, an *indirect feedback* configuration as in Fig. 6.4 is motivated. The Kalman filter again generates estimates of the errors in the inertial system, but these are fed back into the INS to correct it. In this way, the inertial errors are not allowed to grow unchecked, and the adequacy of a linear model is enhanced. There is a further advantage to this configuration. Since the INS is corrected after each measurement sample, many of the predicted error states at the next sample time will be zero, and thus these components of $\hat{\mathbf{x}}(t_{i+1}^-)$ need not be computed explicitly. With regard to filter or external aid failure, because of the slow sample rate and slow INS error dynamics, such failures could be detected and the corrections to the INS could be removed before much (any) performance deterioration were caused.

The general comments of this section will be developed further in the following sections, which consider two error state space filters and a total state space design.

6.4 INS AIDED BY POSITION DATA: A SIMPLE EXAMPLE

Let us consider the combining of inertial system data with position data provided by radar or a radio navigation aid [21–23, 25, 30, 52]. Conceptually, we will want to weight the INS information heavily in the high frequency range (where it provides good data), and emphasize the position data in the low frequency range. This will be a very simple problem formulation confined to a single direction. More complicated, three-dimensional system models are more realistic, but the two-state model to be studied will allow simple algebra and greater transparency of effects of system model characteristics upon filter performance. The insights gained will be directly extendable to more complex problems. Furthermore, this problem will be formulated with continuous-time measurements to allow direct frequency domain interpretation of the filter's

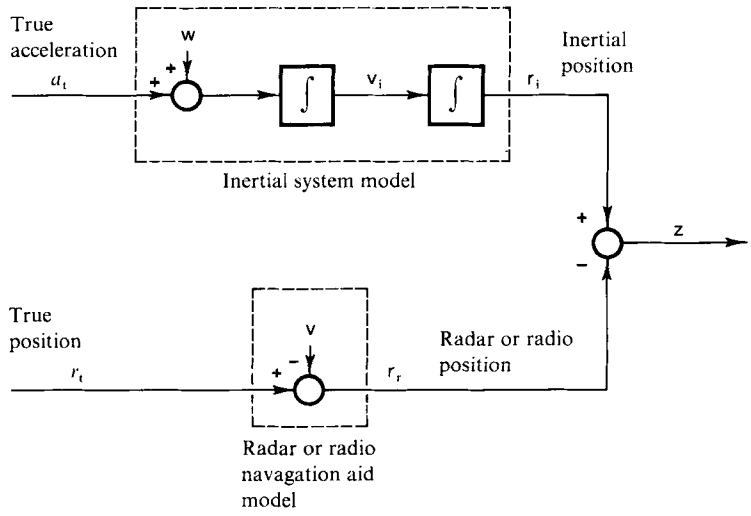


FIG. 6.5 Simple system model for INS aided by position data.

operation. The more realistic case of sampled-data measurements will be considered subsequently.

For this example, model the inertial navigation system simply as a double integrator of noise-corrupted acceleration information, as depicted in Fig. 6.5. The noise $\mathsf{w}(\cdot,\cdot)$ is a white Gaussian noise of mean zero and variance kernel

$$E\{\mathsf{w}(t)\mathsf{w}(t+\tau)\} = Q\,\delta(\tau) \tag{6-1}$$

entering at the acceleration level, and it is meant to model the errors corrupting the INS accelerometer outputs (accelerometer biases and noise, platform misalignment, etc.). The noise-corrupted acceleration is integrated once to yield INS-indicated velocity $\mathsf{v}_i(\cdot,\cdot)$, and a second time to obtain inertially-indicated position $\mathsf{r}_i(\cdot,\cdot)$.

Similarly, consider a simple model for the radar or radio navigation aid (a model which is actually incorporated into many operational filters) as the true position r_t, corrupted by noise $\mathsf{v}(\cdot,\cdot)$. The $\mathsf{v}(\cdot,\cdot)$ process is a zero-mean white Gaussian noise of strength R_c:

$$E\{\mathsf{v}(t)\mathsf{v}(t+\tau)\} = R_c\,\delta(\tau) \tag{6-2}$$

meant to model the wideband noise corrupting the data provided by a radar or radio navigation aid. The negative sign in the model in Fig. 6.5 is just for convenience, as will be seen in a moment. This noise is denoted as $\mathsf{v}(\cdot,\cdot)$ to correspond to the notation adopted earlier, and should not be confused with the subscripted velocity variables (subscripts used in this development are t = true, i = inertial system, and r = radar or radio navigation aid).

The two error state variables for this example are

$$\delta r(t) = r_i(t) - r_t(t) \tag{6-3a}$$

$$\delta v(t) = v_i(t) - v_t(t) \tag{6-3b}$$

i.e., the errors in INS-indicated position and velocity. The measurement to be presented to the filter is the difference between the inertially indicated position and that measured by the radar or radio navigation aid; from the figure,

$$\begin{aligned} z(t) &= r_i(t) - r_r(t) && (6\text{-}4)\\ &= [r_t(t) + \delta r(t)] - [r_t(t) - v(t)] \\ &= \delta r(t) + v(t) && (6\text{-}5) \end{aligned}$$

This then is a "measurement" of the *error* $\delta r(t)$, corrupted by noise $v(t)$; the original negative sign on $v(\cdot,\cdot)$ yields the positive sign here.

To establish the state dynamics model for the error states, first consider the total states $r_i(\cdot,\cdot)$ and $v_i(\cdot,\cdot)$. From Fig. 6.5 we can write (in white noise notation)

$$\begin{bmatrix} \dot{r}_i(t) \\ \dot{v}_i(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} r_i(t) \\ v_i(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} [a_t(t) + w(t)] \tag{6-6}$$

However, the true position, velocity, and acceleration are related by

$$\begin{bmatrix} \dot{r}_t(t) \\ \dot{v}_t(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} r_t(t) \\ v_t(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} a_t(t) \tag{6-7}$$

Subtracting (6-7) from (6-6), and using the error state definitions of (6-3), yields the desired relations as

$$\begin{bmatrix} \dot{\delta r}(t) \\ \dot{\delta v}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta r(t) \\ \delta v(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w(t) \tag{6-8a}$$

$$(\dot{\mathbf{x}}(t) = \mathbf{F} \quad \mathbf{x}(t) + \mathbf{G}\ w(t)) \tag{6-8b}$$

In problems for which the total state model is nonlinear rather than as in (6-6), the error state equation would be derived through perturbation techniques. In terms of this state vector notation, (6-5) becomes

$$z(t) = [1 \quad 0] \begin{bmatrix} \delta r(t) \\ \delta v(t) \end{bmatrix} + v(t) \tag{6-9a}$$

$$(z(t) = \mathbf{H} \quad \mathbf{x}(t) + v(t)) \tag{6-9b}$$

Initial conditions are established by modeling $\mathbf{x}(t_0)$ as a Gaussian random variable, with mean zero (appropriate for error states) and covariance $\mathbf{P}_0$:

$$\mathbf{P}_0 = \begin{bmatrix} \sigma_r^2 & r_{rv}\sigma_r\sigma_v \\ r_{rv}\sigma_r\sigma_v & \sigma_v^2 \end{bmatrix} \tag{6-10}$$

The diagonal terms σ_r^2 and σ_v^2 are the variances of uncertainty in knowledge of initial position and velocity, respectively, and the off-diagonal term is the initial cross correlation between position and velocity, with r_{rv} the associated correlation coefficient.

The objective is to use the measured difference

$$[\mathrm{r}_i(t, \omega_j) - \mathrm{r}_r(t, \omega_j)] = [r_i(t) - r_r(t)] \tag{6-11}$$

for all time t of interest, combined with knowledge of the system model and statistical description of initial conditions and the noises and/or uncertainties in both data sources, to generate the optimal estimate of the state realization

$$\mathbf{x}(t, \omega_j) = \begin{bmatrix} \delta \mathrm{r}(t, \omega_j) \\ \delta \mathrm{v}(t, \omega_j) \end{bmatrix} = \begin{bmatrix} \delta r(t) \\ \delta v(t) \end{bmatrix} \tag{6-12}$$

The continuous-time Kalman filter for this problem is given by (5-144) to (5-146) as

$$\begin{aligned} \begin{bmatrix} \dot{\hat{\delta r}}(t) \\ \dot{\hat{\delta v}}(t) \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{\delta r}(t) \\ \hat{\delta v}(t) \end{bmatrix} + \begin{bmatrix} P_{11}(t) & P_{12}(t) \\ P_{12}(t) & P_{22}(t) \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \frac{1}{R_c} [z(t) - \hat{\delta r}(t)] \\ &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{\delta r}(t) \\ \hat{\delta v}(t) \end{bmatrix} + \begin{bmatrix} P_{11}(t)/R_c \\ P_{12}(t)/R_c \end{bmatrix} [z(t) - \hat{\delta r}(t)] \end{aligned} \tag{6-13}$$

where the covariance $\mathbf{P}(\cdot)$ satisfies

$$\begin{aligned} \begin{bmatrix} \dot{P}_{11}(t) & \dot{P}_{12}(t) \\ \dot{P}_{12}(t) & \dot{P}_{22}(t) \end{bmatrix} &= \begin{bmatrix} P_{12}(t) & P_{22}(t) \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} P_{12}(t) & 0 \\ P_{22}(t) & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & Q \end{bmatrix} \\ &\quad - \begin{bmatrix} P_{11}^2(t)/R_c & P_{11}(t)P_{12}(t)/R_c \\ P_{11}(t)P_{12}(t)/R_c & P_{12}^2(t)/R_c \end{bmatrix} \end{aligned} \tag{6-14}$$

starting from the initial conditions

$$\begin{bmatrix} \hat{\delta r}(t_0) \\ \hat{\delta v}(t_0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{6-15a}$$

$$\begin{bmatrix} P_{11}(t_0) & P_{12}(t_0) \\ P_{12}(t_0) & P_{22}(t_0) \end{bmatrix} = \begin{bmatrix} \sigma_r^2 & r_{rv}\sigma_r\sigma_v \\ r_{rv}\sigma_r\sigma_v & \sigma_v^2 \end{bmatrix} \tag{6-15b}$$

A block diagram of the filter is given by Fig. 6.6. The filter is seen to be a device that accepts $z(t)$, the measured difference between INS and radar (or radio) indicated positions, and outputs optimal estimates of $\delta r(t)$ and $\delta v(t)$. Note that the residual difference between $z(t)$ and $\hat{\delta r}(t)$ is put through optimal gains into a model of the system to attain the new best estimates $\hat{\delta r}(t)$ and $\hat{\delta v}(t)$.

The filter gains in (6-13) are time varying, but they converge to steady state values in a short time. By solving $\dot{\mathbf{P}}(t) = \mathbf{0}$, the steady state values can be shown

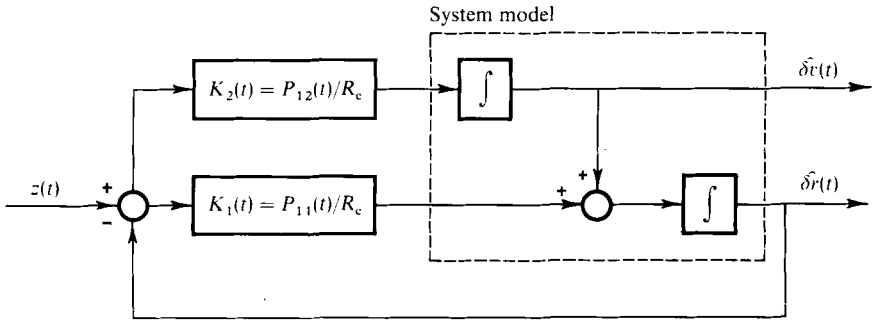


FIG. 6.6 Kalman filter block diagram.

to be covariance $\mathbf{P}$ and gain $\mathbf{K}$, where

$$\mathbf{P} = \begin{bmatrix} \sqrt{2}Q^{1/4}R_c^{3/4} & Q^{1/2}R_c^{1/2} \\ Q^{1/2}R_c^{1/2} & \sqrt{2}Q^{3/4}R_c^{1/4} \end{bmatrix} \tag{6-16a}$$

$$\mathbf{K} = \begin{bmatrix} K_1 \\ K_2 \end{bmatrix} = \begin{bmatrix} P_{11}/R_c \\ P_{12}/R_c \end{bmatrix} = \begin{bmatrix} \sqrt{2}\omega_n \\ \omega_n^2 \end{bmatrix} \tag{6-16b}$$

where ω_n equals $(Q/R_c)^{1/4}$, in radians per second (the reason for denoting it as ω_n will become apparent later). The initial transient behavior of the filter gains depends on $\mathbf{P}_0$, but they are within a few percent of their steady state values (independent of $\mathbf{P}_0$) after $\omega_n t = 2$, so a prediction of time to reach steady state would be approximately $(2/\omega_n)$ sec. Note that if the noise variance kernel (or power spectral density) parameters Q and R_c were determined completely, so would ω_n and the gains.

This filter can be put into feedforward configuration as shown in Fig. 6.7. The optimal estimates of errors committed by the INS, $\delta\hat{r}(t)$ and $\delta\hat{v}(t)$, are subtracted from the INS data, to yield optimally estimated navigation information:

$$\hat{r}(t) = r_i(t) - \delta\hat{r}(t) \tag{6-17a}$$

$$\hat{v}(t) = v_i(t) - \delta\hat{v}(t) \tag{6-17b}$$

For analytical purposes to be considered presently, Fig. 6.7 shows the actual INS and radar (radio navigation aid) systems replaced by the mathematical models used to represent them.

Let us consider steady state filter operation. Under these conditions, Laplace transform techniques can be used to provide a frequency domain interpretation of the filter, and Bode amplitude ratio plots can characterize the transfer functions between INS noise $w(t)$ or radar noise $v(t)$ and the outputs $\delta\hat{r}(t)$ or $\delta\hat{v}(t)$.

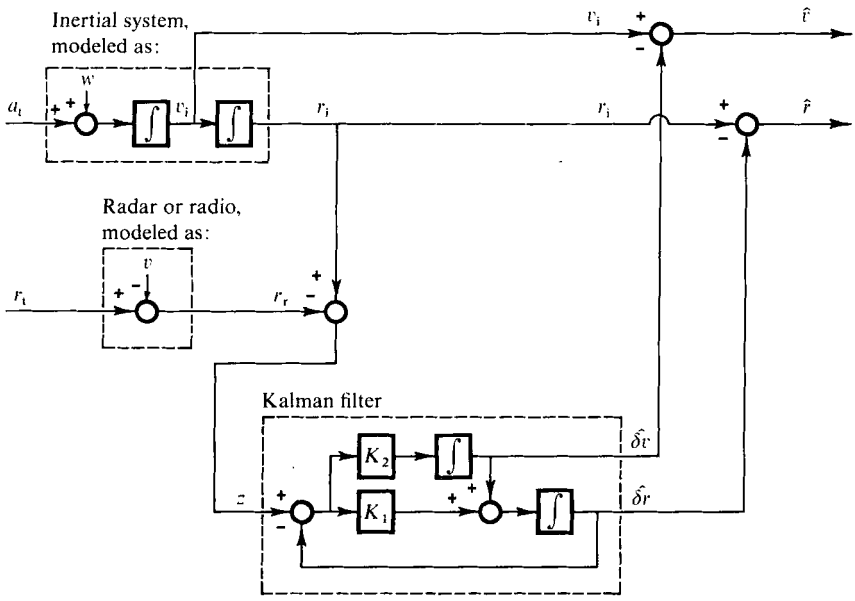


FIG. 6.7 Feedforward filter configuration.

As an intermediate step, the filter transfer function $[\hat{\delta r}(s)/z(s)]$ is given as

$$\frac{\hat{\delta r}(s)}{z(s)} = \frac{[K_1 + (K_2/s)][1/s]}{1 + [K_1 + (K_2/s)][1/s]} = \frac{K_1[s + (K_2/K_1)]}{s^2 + K_1 s + K_2}$$
$$= \frac{\sqrt{2}\omega_n[s + (\omega_n/\sqrt{2})]}{s^2 + \sqrt{2}\omega_n s + \omega_n{}^2} \tag{6-18}$$

Thus, the filter is a second order system with undamped natural frequency ω_n (motivating the original notation ω_n) and damping ratio $\sqrt{2}/2$: the "optimal" damping ratio that provides minimum settling time for a second order system. Furthermore, from Fig. 6.7 it is apparent that $\hat{r}(s)/r_r(s) = -\hat{r}(s)/v(s)$ and

$$\frac{\hat{r}(s)}{v(s)} = -\frac{\hat{\delta r}(s)}{v(s)} = -\frac{\hat{\delta r}(s)}{z(s)} = \frac{-\sqrt{2}\omega_n[s + (\omega_n/\sqrt{2})]}{s^2 + \sqrt{2}\omega_n s + \omega_n{}^2} \tag{6-19}$$

Thus, the transfer function from radar (radio) noise to position estimate is represented by Fig. 6.8: the filter operates as a low-pass filter on the radar, attenuating this signal at frequencies above ω_n with a 20 dB/decade rolloff, as desired. By a similar procedure, INS-caused errors are attenuated at frequencies below ω_n with a 20 dB/decade rolloff, as desired. For example, components of error at the Schuler frequency ω_s would be attenuated by the factor $(\omega_s/\omega_n)^2$.

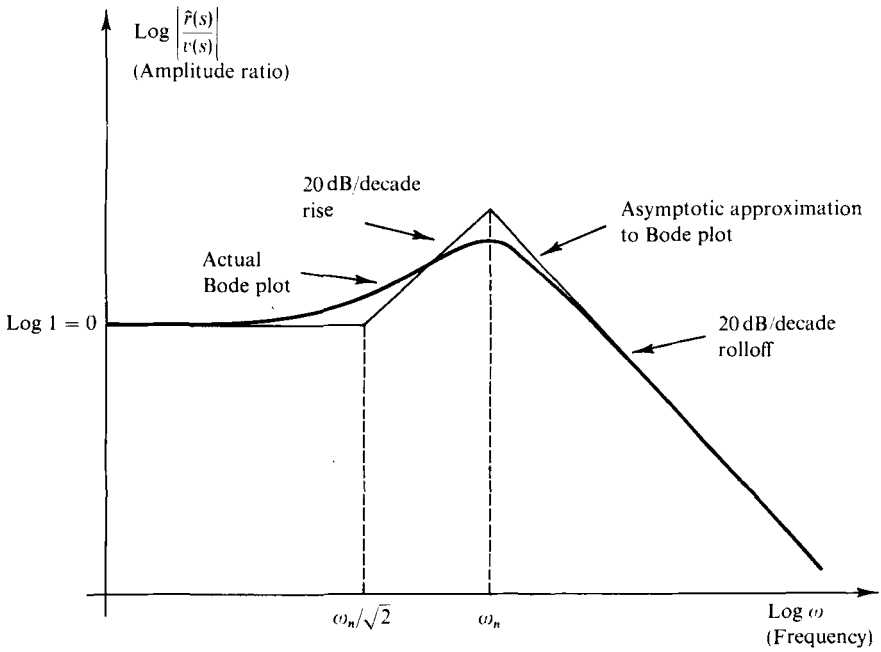


FIG. 6.8 Bode amplitude ratio plot of transfer function $[\hat{r}(s)/v(s)]$.

Figure 6.7 reveals that optimal estimates of both position and velocity are obtained from the position-type measurement *without requiring a differentiation.* This is extremely advantageous, since differentiation of a noisy signal accentuates the noise and generally yields unsatisfactory results.

Because the Kalman filter is based upon the assumed validity of a linear system model, a feedback configuration is preferable to feedforward. Perhaps the most straightforward means of generating the feedback implementation is to write the system and filter equations in terms of corrected INS states. Define

$$\hat{r}(t) = r_i(t) - \hat{\delta r}(t) \tag{6-20a}$$

$$\hat{v}(t) = v_i(t) - \hat{\delta v}(t) \tag{6-20b}$$

as the outputs of an INS corrected by feedback from the filter, and write the difference of (6-6) and (6-13) (for stochastic processes or for their realizations as done here) as:

$$\begin{bmatrix} \dot{r}_i(t) - \dot{\hat{\delta r}}(t) \\ \dot{v}_i(t) - \dot{\hat{\delta v}}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} r_i(t) - \hat{\delta r}(t) \\ v_i(t) - \hat{\delta v}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} [a_t(t) + w(t)]$$

$$- \begin{bmatrix} K_1(t) \\ K_2(t) \end{bmatrix} [z(t) - \hat{\delta r}(t)]$$

By writing the residual as

$$[z(t) - \hat{\delta r}(t)] - [r_i(t) - \hat{\delta r}(t) - r_r(t)]$$

and using (6-20), this becomes

$$\begin{bmatrix} \dot{\hat{r}}(t) \\ \dot{\hat{v}}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{r}(t) \\ \hat{v}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} [a_t(t) + w(t)] - \begin{bmatrix} K_1(t) \\ K_2(t) \end{bmatrix} [\hat{r}(t) - r_r(t)] \tag{6-21}$$

Figure 6.9a is a block diagram representation of these equations, from which the INS, radar (radio), and filter become readily apparent. Portrayed in Fig.

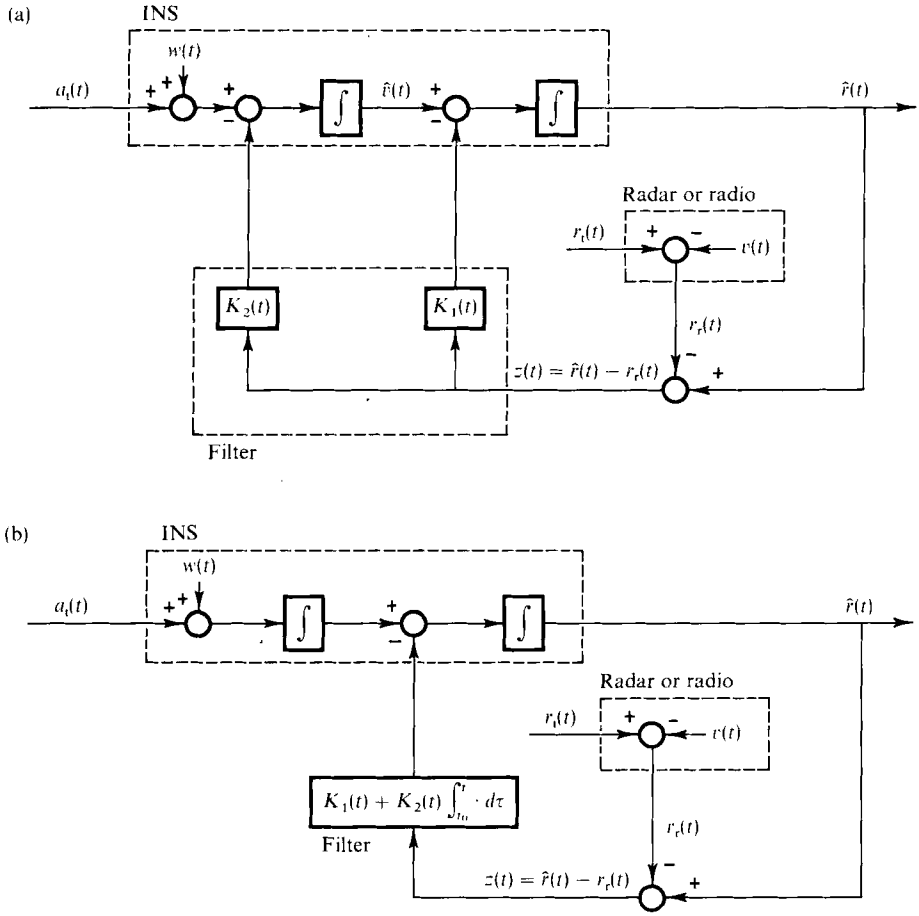


FIG. 6.9 Feedback filter configuration.

6.9b is an equivalent form obtained by replacing the feedback path through $K_2(t)$ and the first INS integrator by $K_2(t)$ and an integrator in the filter itself. Thus, with grossly approximate models for the INS and radar (radio), the filter achieved in Fig. 6.9b in steady state becomes a proportional plus integral filter with transfer function

$$G(s) = K_1 + (K_2/s) = (K_1 s + K_2)/s \tag{6-22}$$

This is precisely the form that has been suggested by classical filtering techniques for this application. Unlike the classical approach of essentially guessing an appropriate filter transfer function form, however, optimal estimation theory *dictates* the *form* of the filter once an adequate system model has been established.

This has been an extremely simple example to demonstrate the mechanization of an optimally aided inertial navigation system and to reveal certain facets of the Kalman filter incorporated in the system integration. However, the same error state space formalism and fundamental estimation concepts are directly applied in practice to more realistic system models. By replacing the two-state INS error model with the Pinson 9-state (three position errors, three velocity errors, and three platform misalignment angles) error model [40, 41, 52], and replacing the single white noise $w(\cdot, \cdot)$ by appropriate disturbances generated from numerous shaping filters, an operational discrete-time filter can readily be developed based on the basic insights gained from this example.

A tradeoff of algorithm simplicity and estimation performance is involved in the choice of state space model: one wants to portray the dominant effects of the error dynamics well enough to attain desired accuracy for the overall navigation system, while meeting constraints of computer time, memory, wordlength, and the like.

Furthermore, once the basic model form is chosen, "best" values for noise statistics must be attained. In this problem, if Q and R_c were not known completely, engineering judgment could be employed to select values to achieve a good break frequency ω_n for the steady state filter. Iterative fine tuning of a filter will be discussed in detail in Section 6.8.

6.5 DOPPLER-AIDED INS

The functional diagram of a Doppler-aided inertial navigation system [19, 21, 25, 32, 40, 45, 52] in one axis is given in Fig. 6.10. The arrow denoted "platform orientation" is meant to indicate a physical connection rather than an electrical signal connection: what the accelerometer measures is dictated by the angular orientation of its sensitive axis, fixed to the INS platform. As is typical of terrestrial navigation systems, it will be assumed that the platform is aligned to a local-level coordinate frame (north–east–down, wander azimuth,

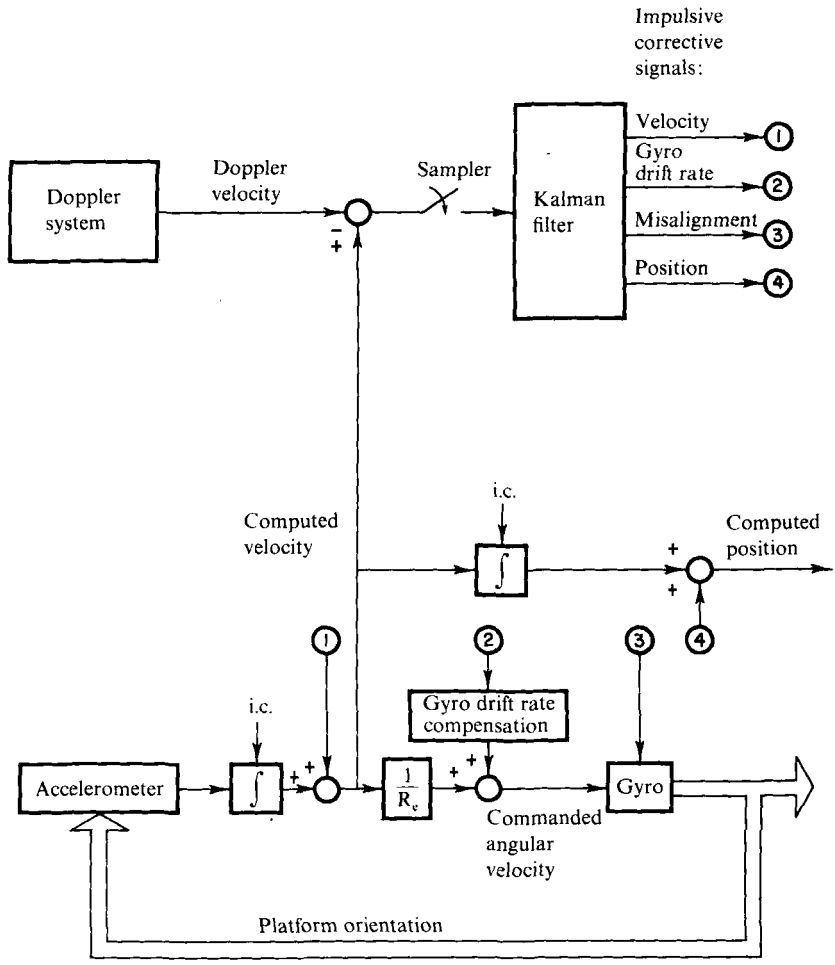


FIG. 6.10 Functional diagram of Doppler-aided INS in one axis.

etc.) and the accelerometer in Fig. 6.10 is one whose sensitive axis is nominally horizontal. (Thus, there is no computed gravity term subtracted from its output.)

The difference between INS-indicated velocity and Doppler-indicated velocity is sampled to provide discrete-time measurements to the Kalman filter, implemented in a digital computer. As can be seen from the diagram, this is an *indirect feedback* configuration: each time the measurement is sampled and $\hat{\mathbf{x}}(t_i^+)$ is computed, the filter outputs four discrete-time corrective signals which are fed back into the INS. The four parameters to be corrected are position, velocity, platform misalignment angle, and gyro drift rate.

Position and velocity integrators (computer registers) can be corrected essentially instantaneously. Similarly, the drift rate compensation register can be altered impulsively, yielding a compensation signal (analog, or pulse rate if the INS is a pulse-torque loop design) that is held constant over the ensuing sample period. We will *assume* that instantaneous corrections to the platform misalignment are also achievable. In fact, the gimbal torquer motors are commanded to remove estimated misalignments at maximum rate, which yields a response time very short compared to the filter sample period. In some inertial systems, the outputs are resolved through a direction cosine matrix which *can* be changed instantaneously, while the gimbal motors simultaneously zero out the estimated error from the physical platform orientation, thus enhancing the validity of the assumed impulsive correction capability.

The filter is often implemented as a software program in a general purpose digital computer, inherently requiring sampled data measurements. Typically, the two velocities are sampled and differenced every 5 to 30 sec, and thus discrete-time corrections are applied to the INS with this same periodicity. For this example, we will assume a 30-sec sample period. Note that it is the use of the error state space model with its slow, linear dynamics that allows such a slow filter algorithm iteration rate to be employed.

First let us formulate the mathematical system model. Since we seek coupled first order linear differential equations, an error state space formulation will be used. Neglecting accelerometer bias (which is justifiable in many practical systems since other error sources dominate the bias effects), there are four variables of primary interest:

δr = error in INS-indicated position,
δv = error in INS-indicated velocity,
ψ = platform misalignment angle (or attitude error, tilt, or "correction to the vertical"),
ε = gyro drift rate.

In terms of these variables, the state differential equation becomes:

$$\begin{bmatrix} \dot{\delta r}(t) \\ \dot{\delta v}(t) \\ \dot{\psi}(t) \\ \dot{\varepsilon}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & g & 0 \\ 0 & -1/R_e & 0 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta r(t) \\ \delta v(t) \\ \psi(t) \\ \varepsilon(t) \end{bmatrix} \tag{6-23a}$$

$$(\dot{\mathbf{x}}(t) = \mathbf{F} \quad \mathbf{x}(t)) \tag{6-23b}$$

where g is the magnitude of the gravity vector at the earth surface and R_e is the radius of the earth (effects of vehicle altitude on g and the difference of the true geoid shape from an assumed sphere are included in navigation equations but neglected in the error state space filter). The first component of (6-23) simply says that the time rate of change of position error is equal to velocity error. To

interpret the second component, recall that a local-level inertial system is used: nominally the accelerometer sensitive axis is orthogonal to the gravitational field, but if there is a platform misalignment angle $\psi(t)$, the accelerometer senses $g \sin \psi(t)$, which for a small misalignment angle is approximately $g\psi(t)$. The time rate of change of velocity error equals the error in sensed acceleration, or $g\psi(t)$. If the vehicle were traveling at velocity $v(t)$ over an earth of radius R_e, the correct angular rate at which to command the platform to maintain it aligned to the local level would be $v(t)/R_e$, so an error in knowledge of $v(t)$ would yield an inappropriate rate command of $\delta v(t)/R_e$. The time rate of change of platform misalignment angle equals this plus gyro drift rate, with the signs in the third component of (6-23) determined by angle sign conventions. Finally, assuming gyro drift rate to be adequately modeled as an unknown constant yields the last entry in (6-23). If a random walk model of bias were used, or equivalently if a pseudonoise were to be added to reflect the fact that an unknown constant were not a totally adequate model, then (6-23) would become

$$\begin{bmatrix} \dot{\delta r}(t) \\ \dot{\delta v}(t) \\ \dot{\psi}(t) \\ \dot{\varepsilon}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & g & 0 \\ 0 & -1/R_e & 0 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta r(t) \\ \delta v(t) \\ \psi(t) \\ \varepsilon(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} w(t) \tag{6-24}$$

where $w(\cdot, \cdot)$ is a zero-mean white Gaussian noise of appropriate strength Q. One objective of this example is to investigate the difference in filter performance caused by the two bias models.

The state initial condition is modeled as a Gaussian random variable, $\mathbf{x}(t_0)$. It is estimated that the system is error-free initially (since deterministic errors would be compensated):

$$E\{\mathbf{x}(t_0)\} = \hat{\mathbf{x}}_0 = [0 \quad 0 \quad 0 \quad 0]^T \tag{6-25}$$

The initial covariance matrix $\mathbf{P}_0$ provides a statistical measure of confidence that the states truly are error-free:

$$E\{[\mathbf{x}(t_0) - \hat{\mathbf{x}}_0][\mathbf{x}(t_0) - \hat{\mathbf{x}}_0]^T\} = \mathbf{P}_0 = \begin{bmatrix} \sigma_{r0}^2 & 0 & 0 & 0 \\ 0 & \sigma_{v0}^2 & 0 & 0 \\ 0 & 0 & \sigma_{\psi 0}^2 & 0 \\ 0 & 0 & 0 & \sigma_{\varepsilon 0}^2 \end{bmatrix} \tag{6-26}$$

In this matrix, the diagonal terms represent variances, or mean squared errors, in knowledge of the initial conditions. $\mathbf{P}_0$ is often assumed diagonal for lack of sufficient statistical information to evaluate its off-diagonal terms; this infers that initially the four states are uncorrelated, or independent since $\mathbf{x}(t_0)$ is assumed to be Gaussian.

The measurement to be used as the input to the filter is the sampled difference between INS and Doppler velocities. By the definition of $\delta v(t_i)$, the INS-

indicated velocity at time t_i is modeled as

$$\mathsf{v}_{\text{INS}}(t_i) = \mathsf{v}_{\text{true}}(t_i) + \delta\mathsf{v}(t_i) \tag{6-27}$$

For this problem, the Doppler velocity indication is modeled as the true velocity corrupted by the discrete-time white Gaussian noise $\mathsf{v}(\cdot,\cdot)$, of mean zero and variance R:

$$\mathsf{v}_{\text{Doppler}}(t_i) = \mathsf{v}_{\text{true}}(t_i) - \mathsf{v}(t_i) \tag{6-28}$$

Thus, the measurement can be modeled as

$$\mathsf{z}(t_i) = \mathsf{v}_{\text{INS}}(t_i) - \mathsf{v}_{\text{Doppler}}(t_i) \tag{6-29}$$

$$= \delta\mathsf{v}(t_i) + \mathsf{v}(t_i) \tag{6-30}$$

In terms of the error state vector notation, this becomes

$$\mathsf{z}(t_i) = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta\mathsf{r}(t_i) \\ \delta\mathsf{v}(t_i) \\ \psi(t_i) \\ \varepsilon(t_i) \end{bmatrix} + \mathsf{v}(t_i) \tag{6-31a}$$

$$(\mathsf{z}(t_i) = \quad \mathbf{H} \quad \mathbf{x}(t_i) \quad + \mathsf{v}(t_i)) \tag{6-31b}$$

Based upon the system model of (6-23) or (6-24), (6-25), (6-26), and (6-31), the Kalman filter can now be delineated. First consider propagation from sample time t_{i-1} to time t_i. It is assumed that when the measurement is sampled, the update computations are performed and the corrective signals applied to the INS, at which point the optimal error state estimate becomes

$$\hat{\mathbf{x}}(t_{i-1}^{+c}) = \begin{bmatrix} \hat{\delta r}(t_{i-1}^{+c}) \\ \hat{\delta v}(t_{i-1}^{+c}) \\ \hat{\psi}(t_{i-1}^{+c}) \\ \hat{\varepsilon}(t_{i-1}^{+c}) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \hat{\psi}(t_{i-1}^{+}) \\ 0 \end{bmatrix}$$

where the additional superscript c denotes after the control is fed back to the inertial system. Moreover, the estimated tilt is zeroed out as quickly as possible, so that by the next sample time (and well before that time in the sample period), before the next measurement is processed, the estimated state is

$$\hat{\mathbf{x}}(t_i^{-}) = \begin{bmatrix} \hat{\delta r}(t_i^{-}) \\ \hat{\delta v}(t_i^{-}) \\ \hat{\psi}(t_i^{-}) \\ \hat{\varepsilon}(t_i^{-}) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{6-32}$$

Consequently, there is no need to compute $\hat{\mathbf{x}}(t_i^{-})$ explicitly onboard.

If (6-23) is used as the state dynamics model, the covariance propagation relation is

$$\mathbf{P}(t_i^-) = \mathbf{\Phi}(t_i, t_{i-1})\mathbf{P}(t_{i-1}^+)\mathbf{\Phi}^{\mathrm{T}}(t_i, t_{i-1}) \tag{6-33}$$

where $\mathbf{\Phi}(t_i, t_{i-1})$ is the state transition matrix associated with $\mathbf{F}$. Equivalently, $\mathbf{P}(t_i^-)$ can be found by integrating

$$\dot{\mathbf{P}}(t/t_{i-1}) = \mathbf{F}\mathbf{P}(t/t_{i-1}) + \mathbf{P}(t/t_{i-1})\mathbf{F}^{\mathrm{T}} \tag{6-34}$$

forward to time t_i from $\mathbf{P}(t_{i-1}/t_{i-1}) = \mathbf{P}(t_{i-1}^+)$ at t_{i-1}. If a random walk bias model and (6-24) were used instead, these would become, respectively,

$$\mathbf{P}(t_i^-) = \mathbf{\Phi}(t_i, t_{i-1})\mathbf{P}(t_{i-1}^+)\mathbf{\Phi}^{\mathrm{T}}(t_i, t_{i-1}) + \int_{t_{i-1}}^{t_i} \mathbf{\Phi}(t_i, \tau)\mathbf{G}\mathbf{Q}\mathbf{G}^{\mathrm{T}}\mathbf{\Phi}^{\mathrm{T}}(t_i, \tau)\, d\tau \tag{6-35}$$

and

$$\dot{\mathbf{P}}(t/t_{i-1}) = \mathbf{F}\mathbf{P}(t/t_{i-1}) + \mathbf{P}(t/t_{i-1})\mathbf{F}^{\mathrm{T}} + \mathbf{G}\mathbf{Q}\mathbf{G}^{\mathrm{T}} \tag{6-36}$$

To update the estimate at a measurement sample time, the filter gain is calculated as

$$\begin{aligned}\mathbf{K}(t_i) &= \mathbf{P}(t_i^-)\mathbf{H}^{\mathrm{T}}[\mathbf{H}\mathbf{P}(t_i^-)\mathbf{H}^{\mathrm{T}} + \mathbf{R}]^{-1} \\ &= \begin{bmatrix} K_1(t_i) \\ K_2(t_i) \\ K_3(t_i) \\ K_4(t_i) \end{bmatrix} = \frac{1}{P_{22}(t_i^-) + R} \cdot \begin{bmatrix} P_{12}(t_i^-) \\ P_{22}(t_i^-) \\ P_{32}(t_i^-) \\ P_{42}(t_i^-) \end{bmatrix}\end{aligned} \tag{6-37}$$

Since $\hat{\mathbf{x}}(t_i^-)$ is zero, the usual state estimate update,

$$\hat{\mathbf{x}}(t_i^+) = \hat{\mathbf{x}}(t_i^-) + \mathbf{K}(t_i)[\mathbf{z}_i - \mathbf{H}\hat{\mathbf{x}}(t_i^-)]$$

simplifies to

$$\hat{\mathbf{x}}(t_i^+) = \mathbf{K}(t_i)\mathbf{z}_i = \begin{bmatrix} K_1(t_i) \\ K_2(t_i) \\ K_3(t_i) \\ K_4(t_i) \end{bmatrix} [v_{\mathrm{INS}}(t_i) - v_{\mathrm{Doppler}}(t_i)] \tag{6-38}$$

Finally, the covariance update is

$$\mathbf{P}(t_i^+) = \mathbf{P}(t_i^-) - \mathbf{K}(t_i)\mathbf{H}\mathbf{P}(t_i^-) \tag{6-39}$$

An error analysis (performance analysis) can be conducted *before* any actual hardware is built, because, under our assumptions, the estimation error covariance matrix is *not* a function of the actual measurement values. A *first step* in such an analysis would be to propagate the filter covariance equations. However, one must beware of a *misinterpretation* of these results that has been committed in some past filter design efforts. The filter-propagated error covariance is a good representation of the actual performance to be expected

only to the extent that the filter system model is an adequate representation of the "real world" environment. A more thorough and valid performance analysis will be discussed subsequently in Section 6.8.

Assume first that a filter is based upon the noise-free dynamics, (6-23), so that the filter error covariance time propagation is given by (6-33) or (6-34). Let the strength of the measurement corruption noise be

$$R = 0.25 \quad \text{ft}^2/\text{sec}^2 \tag{6-40}$$

In other words, the Doppler is modeled as a device with wideband (white) noise contributing an rms (root mean square) error of 0.5 ft/sec. Finally, let the initial state covariance be

$$\mathbf{P}_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \text{ meru}^2 \end{bmatrix} \tag{6-41}$$

where a meru is a milli-earth-rate-unit, equal to 0.015 deg/hr. This $\mathbf{P}_0$ infers that we are absolutely certain of all initial conditions except for the value of gyro drift rate. By itself, this would be rather unrealistic, but this can be viewed as one step in establishing an "error budget" to indicate overall system errors due to a *single* source of uncertainty; such a concept will be pursued further in Section 6.8.

Figure 6.11 depicts the rms errors in INS position and velocity indications and the rms platform misalignment for the case of a "pure" inertial system, running free without any Doppler aiding. These plots were generated by taking the square root of the diagonal terms of the filter-computed covariance matrix. From plot (a), it can be seen that if our model were an adequate representation of the INS, then this is about a $1\frac{1}{2}$ nautical mile per hour inertial system. Also apparent from the plots is the Schuler mode oscillation [rectified in plot (c)] with a period of 84 min, as expected from the characteristic equation associated with the $\mathbf{F}$ in (6-23):

$$|\lambda \mathbf{I} - \mathbf{F}| = 0 = \lambda^2(\lambda^2 + g/R_e) = \lambda^2(\lambda^2 + \omega_s{}^2) \tag{6-42}$$

Now assume that the difference between INS and Doppler velocities is sampled and processed by the filter every 30 sec. Figure 6.12 presents the corresponding rms errors in position, velocity, platform misalignment, and gyro drift rate estimates provided by the Doppler-aided inertial system. The performance improvement is impressive: the position error plot indicates achievement of an aided system with performance on the order of 1/50 nautical mile per hour. But this is a realistic prediction of true performance only if the simple four-state model adequately models true system behavior. If it were valid, one would expect about 68% of the cases of real system operation to lie within this envelope, 95% to be within an envelope of twice the magnitude,

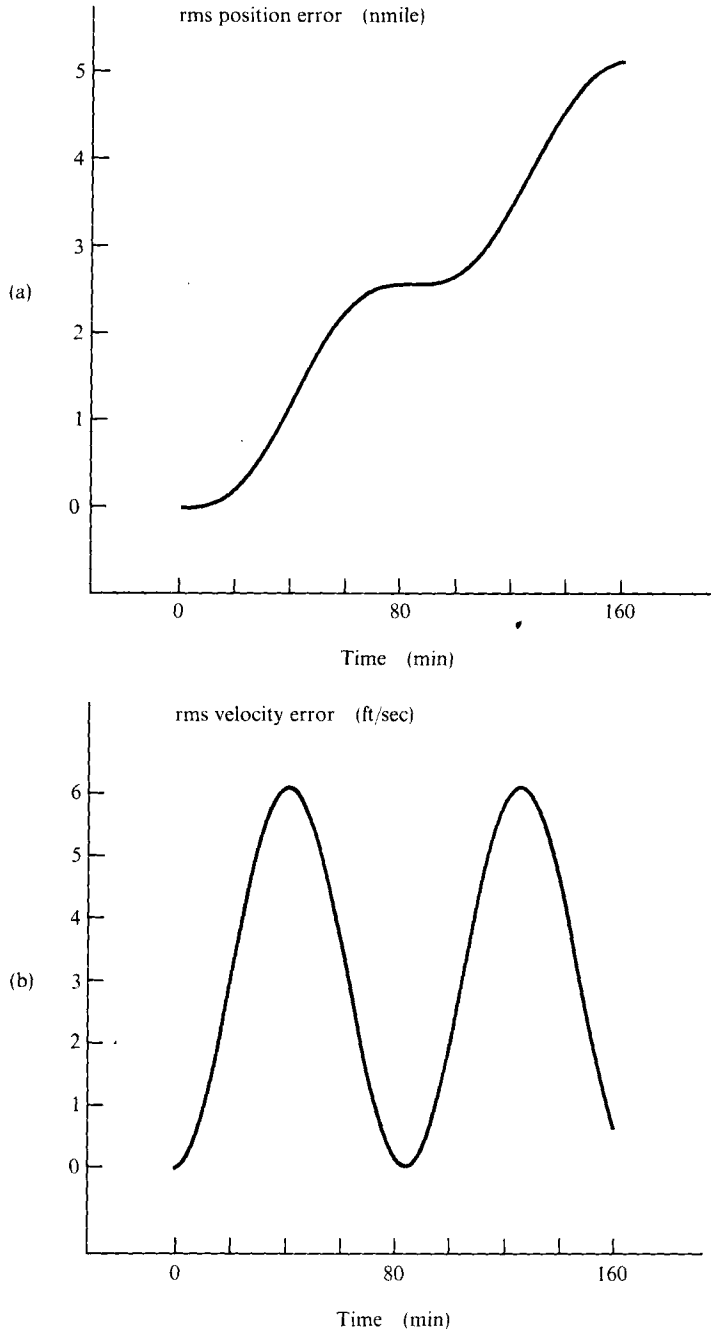


FIG 6.11a,b Pure inertial system. (a) rms position error. (b) rms velocity error.

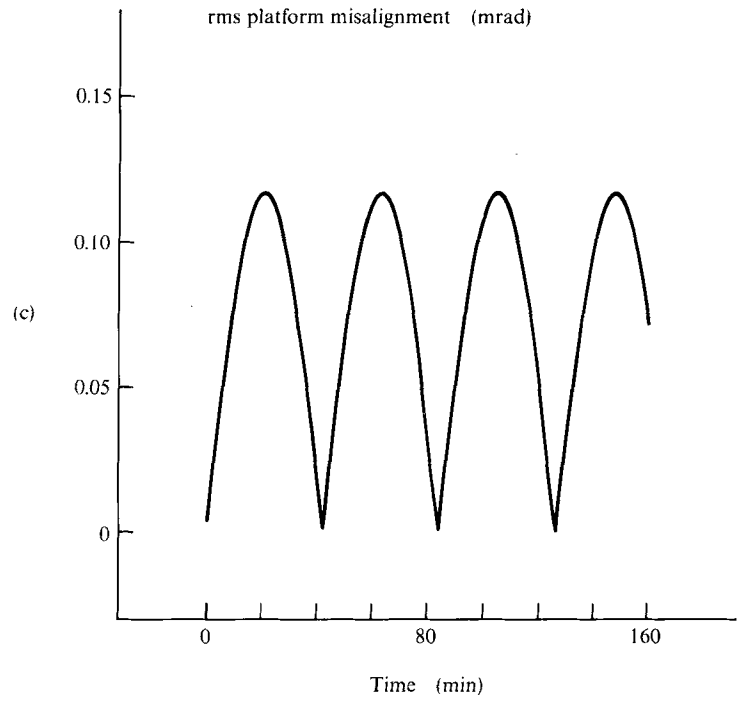


FIG. 6.11c Pure inertial system. rms platform misalignment. From Schmidt [45] with permission of author.

and so forth. Practical experience with Doppler-aided inertial systems would discount this validity significantly.

Recall that these plots pertain to the random constant bias model: there is no dynamic driving noise $w(\cdot,\cdot)$. From Fig. 6.12d it can be seen that the filter "thinks" that the error in the gyro drift rate estimate asymptotically goes to zero, and it turns out that the Kalman gain for the drift rate estimate, K_4, also goes to zero. From the postulated mathematical model, this is appropriate: with no noise $w(\cdot,\cdot)$, the model "tells" the filter that you are uncertain of the initial value of drift rate, but that you are *sure* it is a constant value. Therefore, the filter estimates the drift rate using early measurement data, sends out the appropriate correction signals, and essentially ignores later measurements (because it "knows" once a good estimate is obtained, it is good for all time, since drift rate *is* a *constant* according to the model).

However, rarely are you so sure of drift rate (or other "biases" or parameters) being a true constant in time that you would be willing to cease estimation of its value after an initial transient period. Furthermore, this would preclude the possibility of sensor failure detection as discussed in Section 5.4. Thus, the dynamics model (6-24) is motivated, leading to filter equations as in (6-35) or

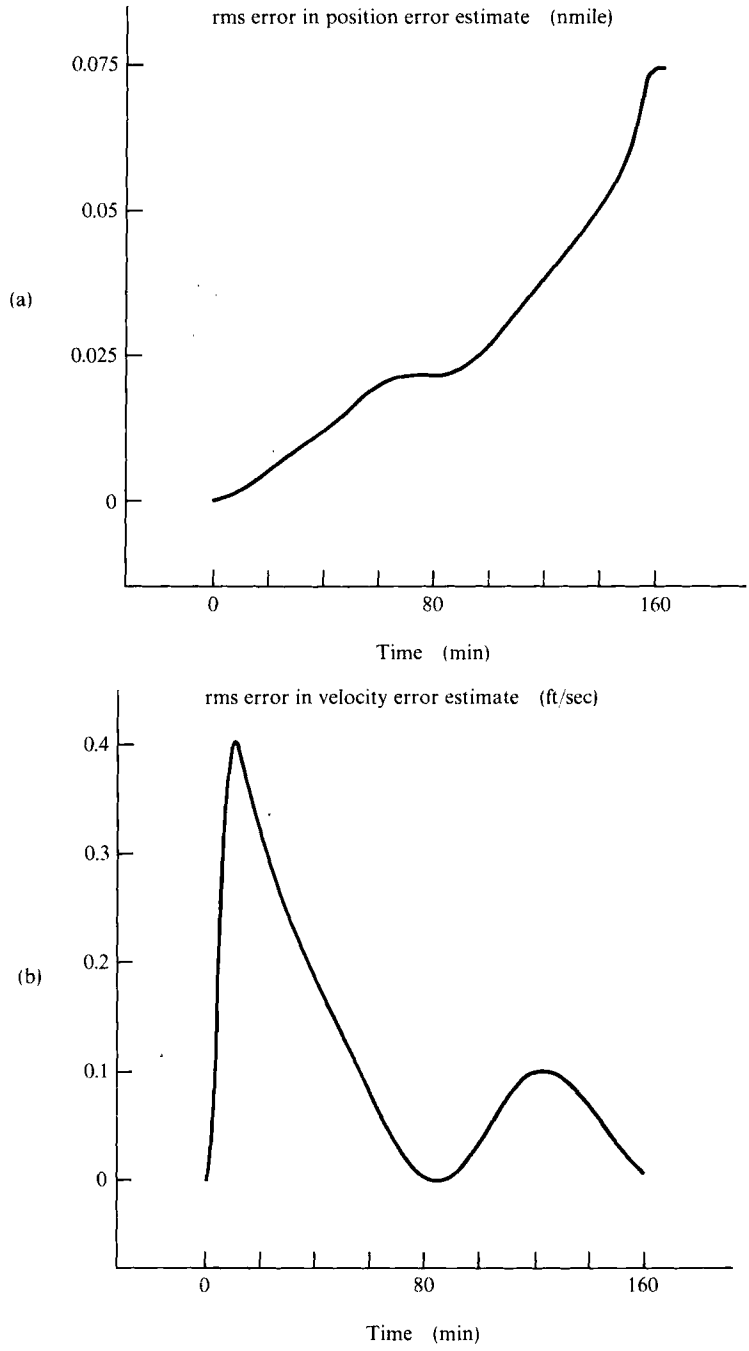


FIG. 6.12a,b Doppler-aided inertial system (a) rms position error. (b) rms velocity error.

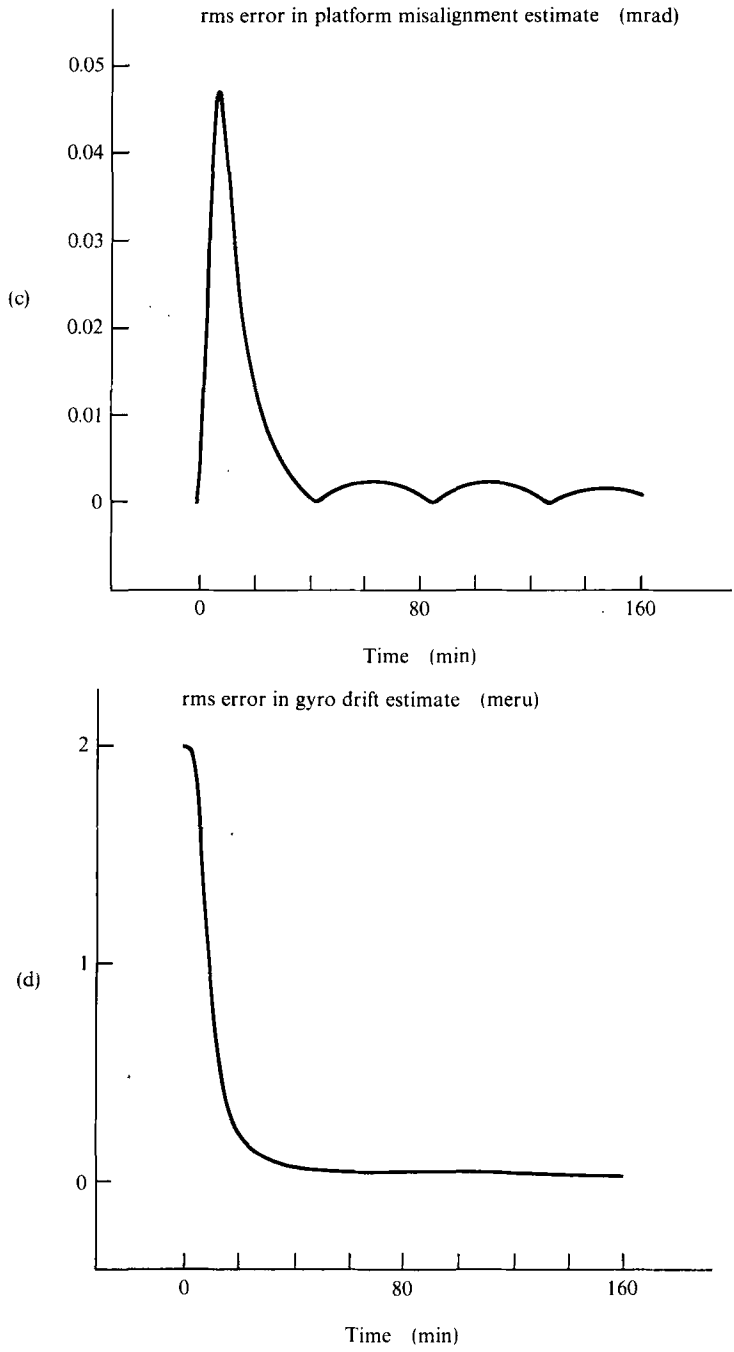


FIG. 6.12c,d Doppler-aided inertial system. (c) rms platform misalignment. (d) rms gyro drift rate error. From Schmidt [45] with permission of author.

(6-36): a "pseudonoise" is added, of strength appropriate to how quickly you think the "bias" might actually change in time. As a result, the diagonal terms of $\mathbf{P}(t_i^-)$ and $\mathbf{P}(t_i^+)$ and the filter gains $\mathbf{K}(t_i)$ converge to nonzero values, and a valid estimate of drift rate is maintained by using measurement data over the entire time of filter operation.

One of the most effective operational Doppler/INS navigation Kalman filters (as implemented on the F-111 aircraft) is a direct extension of this example [32]. It consists of 12 states: two horizontal positions, two horizontal velocities, three platform misalignments, three gyro drift rates, and two Doppler error states. Vertical position and velocity are not maintained in the filter, but are filtered classically with barometric altimeter data; the third axis of misalignment angle, i.e., azimuth error, is modeled directly because it is critical to system performance. Essentially, the first ten states just apply the one-dimensional example to the three-dimensional case, with a slightly more complicated $\mathbf{F}(\cdot)$ to account for cross-coupling effects. The final two states are shaping filter states to allow more accurate depiction of Doppler error characteristics: like the gyro drift rate shaping filters, these are simply noise-driven integrators to generate a bias plus random walk. For use over land, they correspond to Doppler scale factor and boresight error, whereas for use over water (in which case errors caused by the water surface effects dominate) they correspond to sea motion error states in the two horizontal directions.

The filter operates in a partial feedback mode, with a sample period of 8 sec. No feedback is provided to the Doppler because the additional computation and system complexity required to do so was not warranted by performance benefits. Since Doppler errors do not exhibit the unbounded growth characteristic of INS errors, it is not critical for system model validity to provide such feedback. Similarly, gyro drift rate compensation feedback was removed after a tradeoff analysis indicated that the resulting hardware simplification could be gained with no appreciable performance degradation. Thus, not all components of $\hat{\mathbf{x}}(t_i^-)$ can be assumed to be zeros, and some state estimate time propagation computations are required.

Doppler velocity is available more often than the 8-sec filter sample period, so some smoothing or prefiltering of the velocity difference signal could enhance system performance. If one were to average the signal over an 8-sec period (with a dumped integrator), noise would in fact be smoothed out, but the resulting average would be a good representation of the signal at the *midpoint* of the integration interval, not at the end of the interval. Simply averaging the signal and inputting the result into the Kalman filter would thus introduce a 4-sec "delay" in the measurement data. To account for this directly, additional integrator states could be added to the filter model, indicating that $z(t_i)$ is not a velocity difference signal but an average (time integral over a sample period, divided by the period) of such a signal. Such an increased-dimension filter was evaluated against the 12-state design that accounted for the averaging to some

degree with a modified $\mathbf{H}(t_i)$ and $\mathbf{R}(t_i)$, and the simpler design was chosen because performance still met specifications while computer loading was decreased.

The filter also employs reasonableness checking via residual monitoring as discussed in Section 5.4. If the observed residual exceeds 2.83 times the computed standard deviation, the data point is rejected and the operator is alerted.

In early operational tests, the pure inertial mode unfortunately seemed to outperform the Doppler velocity-aided inertial mode. This was indicative of an overly pessimistic model of errors committed by the INS and/or an overly optimistic model of errors in the external velocity signal. By gathering extensive sensor performance data (unavailable at time of initial design), better values of noise strengths $\mathbf{Q}$ and $\mathbf{R}$ were established, the filter retuned, and performance improved substantially.

The point of this discussion is that the simple four-state filter discussed in this section provides the essence of a practical, proven design.

6.6 INS CALIBRATION AND ALIGNMENT USING DIRECT KALMAN FILTER

Kalman filters are exploited in the initial calibration and alignment of inertial systems as well as optimal aiding during the navigation mode of operation. A direct filter can process *external* information in order first to estimate the system misalignments and miscalibrations and then to command appropriate control signals to remove the estimated errors. The external information may simply be the fact that the vehicle is sitting still at a known location (i.e., preflight), or it may be position and/or velocity data from other sources (i.e., inflight alignment, with TACAN data, for example). In fact, experience has shown the necessity of a Kalman filter to achieving inflight alignments rapidly and precisely enough to meet most system specifications. In this section, however, the simpler problem of preflight calibration and alignment will be discussed. A rudimentary problem will be considered first, progressing in complexity to the filter form as essentially implemented for the Apollo spacecraft program.

Basically, in a preflight procedure, the accelerometers are calibrated by the known gravity acceleration magnitude, and the gyros by the known inertial rate of rotation of the gravity vector, at a point on the surface of the earth. The platform is approximately aligned to a local-level coordinate frame, and then the gyros alone are used to generate the commands appropriate to maintain an inertially fixed platform orientation. Since the estimates of platform misalignments and gyro drift rates will depend upon the measurement by the accelerometers of the rotation of the gravity vector with respect to the "inertial" frame as instrumented by the gyros, critical disturbances will be accelerometer

quantization errors and the motion of the vehicle (for instance, wind-induced sway for a missile on a launch pad). Therefore, the eventual filter design must account for these effects.

First consider a simplified single-axis problem formulation, with continuous-time measurements [42], as depicted in Fig. 6.13. The gimbal motors are driven by a known command torque, T_{com} (zero if an inertially stable platform is desired) and by gyro drift rate ε. The accelerometer sensitive axis is parallel to the platform, so if it is misaligned from local-level by an angle ψ, the accelerometer output is $g \sin \psi \cong g\psi$, corrupted by a wideband noise, modeled as white noise $v_c(\cdot, \cdot)$ of strength R_c. The purpose of the calibration filter is to accept the accelerometer data and to estimate the misalignment angle $\psi(t)$ and gyro drift rate $\varepsilon(t)$.

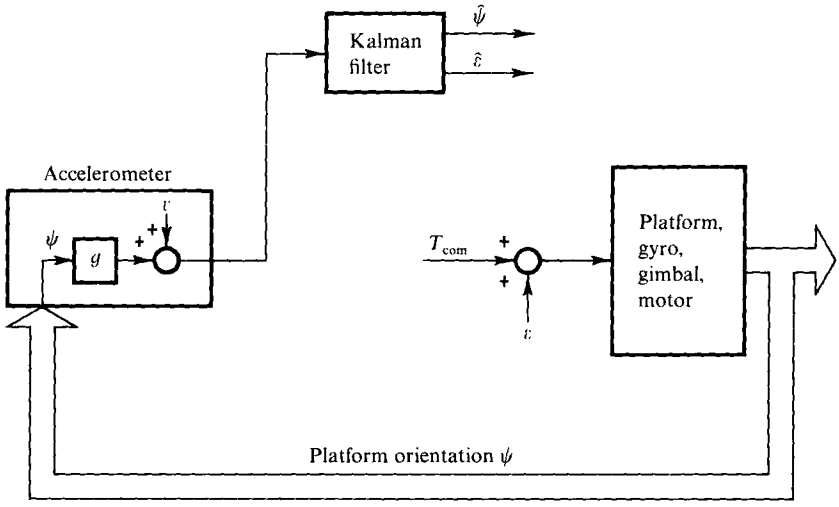


FIG. 6.13 Simplified functional diagram of INS calibration.

A very simple dynamics model would be

$$\begin{bmatrix} \dot{\psi}(t) \\ \dot{\varepsilon}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \psi(t) \\ \varepsilon(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} T_{com}(t) \tag{6-43a}$$

$$(\dot{\mathbf{x}}(t) = \mathbf{F} \quad \mathbf{x}(t) + \mathbf{B} \quad u(t)) \tag{6-43b}$$

in which the gyro drift rate is modeled as an unknown constant. The measurement from the accelerometer can be modeled as the continuous-time relation

$$z(t) = [g \quad 0] \begin{bmatrix} \psi(t) \\ \varepsilon(t) \end{bmatrix} + v_c(t) \tag{6-44a}$$

$$(z(t) = \mathbf{H} \quad \mathbf{x}(t) + v_c(t)) \tag{6-44b}$$

Based on this model, the filter equations for calibration are

$$\mathbf{K}(t) = \mathbf{P}(t)\mathbf{H}^{\mathrm{T}}\mathbf{R}_{\mathrm{c}}^{-1} = \begin{bmatrix} gP_{11}(t)/R_{\mathrm{c}} \\ gP_{12}(t)/R_{\mathrm{c}} \end{bmatrix} = \begin{bmatrix} K_1(t) \\ K_2(t) \end{bmatrix} \tag{6-45}$$

$$\begin{bmatrix} \dot{\hat{\psi}}(t) \\ \dot{\hat{\varepsilon}}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{\psi}(t) \\ \hat{\varepsilon}(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} T_{\mathrm{com}}(t) + \begin{bmatrix} K_1(t) \\ K_2(t) \end{bmatrix} [z(t) - g\hat{\psi}(t)] \tag{6-46a}$$

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{F}\hat{\mathbf{x}}(t) + \mathbf{B}u(t) + \mathbf{K}(t)[z(t) - \mathbf{H}\hat{\mathbf{x}}(t)] \tag{6-46b}$$

$$\dot{\mathbf{P}}(t) = \mathbf{F}\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}^{\mathrm{T}} - \mathbf{P}(t)\mathbf{H}^{\mathrm{T}}\mathbf{R}_{\mathrm{c}}^{-1}\mathbf{H}\mathbf{P}(t) \tag{6-47}$$

integrated forward from the initial conditions $\hat{\mathbf{x}}(t_0) = \hat{\mathbf{x}}_0$, $\mathbf{P}(t_0) = \mathbf{P}_0$. Note that because there is no dynamic driving noise, $\mathbf{P}(t)$ will converge to $\mathbf{0}$, but it is the transient performance that is exploited in this application. The structure of the filter given by (6-46) is diagramed in Fig. 6.14a.

Now consider alignment: assume that calibration has been completed and a drift rate estimate $\hat{\varepsilon}_{\mathrm{a}}$ is available, and that the platform has been torqued so as to zero out the estimated misalignment. Thus, at the time when alignment is initiated, t_{a}, $\hat{\varepsilon}(t_{\mathrm{a}}) = \hat{\varepsilon}_{\mathrm{a}}$, and $\hat{\psi}(t_{\mathrm{a}}) = 0$. Now it is desired to maintain this local-level orientation, using the filter to aid in generation of the required command torque $T_{\mathrm{com}}(t)$. Since

$$\dot{\psi}(t) = T_{\mathrm{com}}(t) + \varepsilon(t) \tag{6-48}$$

according to our model, in order to regulate $\psi(t)$ to zero, the appropriate command would be

$$T_{\mathrm{com}}(t) = -\hat{\psi}(t) - \hat{\varepsilon}(t) \tag{6-49}$$

Closing the feedback loop on the filter yields the result portrayed in Fig. 6.14b. This is essentially the same filter as used in calibration, except that the system itself provides the feedback of $\hat{\psi}(t)$ for a continuous Kalman filter design. The initial conditions for this filter are

$$\hat{\mathbf{x}}(t_{\mathrm{a}}) = \begin{bmatrix} 0 \\ \hat{\varepsilon}_{\mathrm{a}} \end{bmatrix} \tag{6-50a}$$

$$\mathbf{P}(t_{\mathrm{a}}) = \begin{bmatrix} P_{11}(t_{\mathrm{a}}) & P_{12}(t_{\mathrm{a}}) \\ P_{12}(t_{\mathrm{a}}) & P_{22}(t_{\mathrm{a}}) \end{bmatrix} \tag{6-50b}$$

Note that $\mathbf{P}(t_{\mathrm{a}})$ is taken from the calibration filter and that $P_{11}(t_{\mathrm{a}})$ is *not* zero:

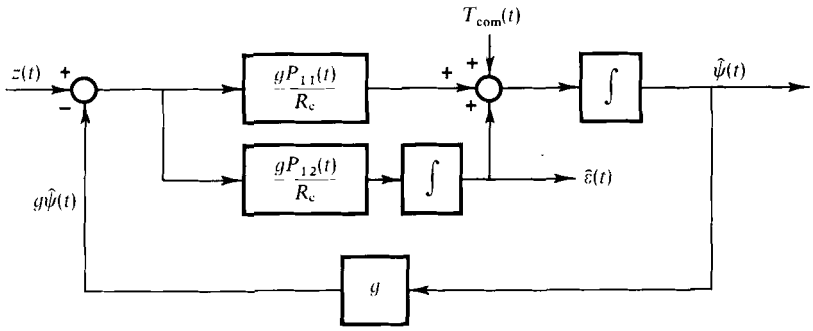


FIG. 6.14a Simplified calibration filter.

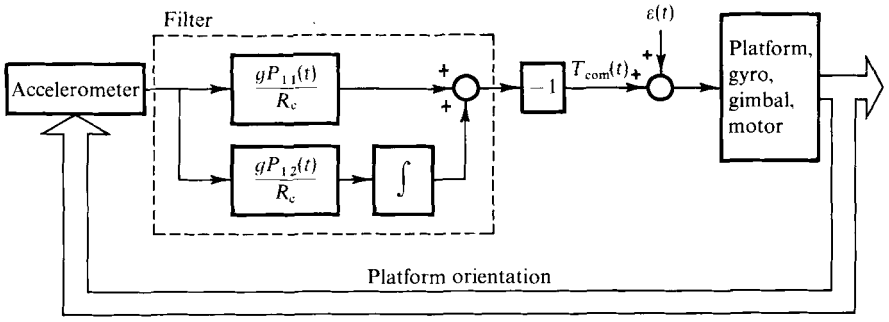


FIG. 6.14b Simplified calibration filter.

the platform has been torqued so that $\hat{\psi}(t_a)$ is zero, but there is still uncertainty in the actual value of $\psi(t_a)$.

This simple problem can be readily modified to account for the pertinent aspects of a realistic calibration and alignment. First, the filter will be implemented on a digital computer, dictating sampled data. Moreover, the INS loop is generally a pulse-torque loop, with accelerometer signals formatted as pulse rates proportional to specific force (so each pulse is proportional to a velocity increment), and similarly torque pulses applied to the gimbal motors instead of analog signals. Through a pulse counter on the accelerometer output channel, the available discrete-time measurements are really integrals of acceleration over a sample period, corrupted mostly by the quantization of the pulses themselves. Thus, if we introduce a "dumped" integrator state (one which resets to zero after each sample time) $\delta(t)$, satisfying

$$\dot{\delta}(t) = g\psi(t) \tag{6-51}$$

then this can be augmented to the original state equation (6-43), and the measurement model altered to the discrete-time relation

$$\mathsf{z}(t_i) = [0 \quad 0 \quad 1]\begin{bmatrix}\psi(t_i)\\ \varepsilon(t_i)\\ \delta(t_i)\end{bmatrix} + \mathsf{v}(t_i) \tag{6-52}$$

Here $\mathsf{v}(\cdot,\cdot)$ is a discrete-time zero-mean white Gaussian noise of strength R appropriate to model the quantization error effects.

The effects of vehicle oscillations (wind sway of a missile, roll of an aircraft carrier, etc.) can be added as well; complicated high frequency dynamics as encountered in inflight alignment would pose a more difficult problem. Consider a missile on its launch pad, and assume that acceleration of the INS case from first mode bending of the missile due to winds is significant enough to be modeled in the filter. First mode bending dynamics can be represented by a second order linear system, and a standard model of winds (the Dryden model [6]) is an exponentially time-correlated Gaussian noise. Consequently, to add this effect in one axis direction requires a three-state shaping filter; letting p_b, v_b, and a_b be the horizontal displacement, velocity, and acceleration, respectively, of the INS case due to wind-induced bending, one state space representation of the shaping filter is

$$\begin{bmatrix}\dot{\mathsf{p}}_b(t)\\ \dot{\mathsf{v}}_b(t)\\ \dot{\mathsf{a}}_b(t)\end{bmatrix} = \begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ \alpha & \beta & \gamma\end{bmatrix}\begin{bmatrix}\mathsf{p}_b(t)\\ \mathsf{v}_b(t)\\ \mathsf{a}_b(t)\end{bmatrix} + \begin{bmatrix}0\\ 0\\ 1\end{bmatrix}\mathsf{w}(t) \tag{6-53}$$

The white Gaussian noise $\mathsf{w}(\cdot,\cdot)$ is of appropriate strength to yield the desired rms value of wind, σ_{wind}, with correlation time $1/\lambda$: $Q = 2\lambda\sigma_{\text{wind}}^2$. If the bending dynamics model is second order with undamped natural frequency ω_n and damping ratio ζ, then the three parameters α, β, and γ are specified by $\alpha = -\lambda{\omega_n}^2$, $\beta = -{\omega_n}^2 - 2\zeta\lambda\omega_n$, and $\gamma = -2\zeta\omega_n - \lambda$. Augmenting this to the dynamics given by (6-43) and (6-51) yields a six-state model with $\mathbf{x} = [\psi\ \varepsilon\ \delta\ \mathsf{p}_b\ \mathsf{v}_b\ \mathsf{a}_b]^T$, and a measurement model given by

$$\mathsf{z}(t_i) = [0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1]\mathbf{x}(t_i) + \mathsf{v}(t_i) \tag{6-54}$$

Repeating this development for the second horizontal direction, and adding an azimuth angle error state results in the 13-state model which served as the basis of the Apollo calibration and alignment Kalman filter design. In actual implementation, the filter gains were precomputed and curve-fitted, and the approximate gain functions stored for online usage. Adequacy of this simple design is attested to by the success of the Apollo missions.

6.7 GENERATING ALTERNATIVE DESIGNS

A systematic design procedure will encompass the generation of alternative filter designs and a realistic evaluation of performance capabilities versus computer loading for each one. It is possible to devise filters based on very extensive models, but these are usually more sophisticated than really needed, and are prohibitive computationally. The designer is willing to sacrifice some degree of performance in order to achieve practical computational levels. Typically, he will seek the most simplified filter system model that retains the dominant features of the original system and provides adequate estimate precision. This is probably the most difficult aspect of designing a Kalman filter, and it requires a good understanding of the underlying physics of the system as well as competence in filtering theory.

Suppose a large-dimensioned, complex system model existed upon which a filter could be based that far exceeded performance requirements (the most complete of these to be termed a "truth model" in the next section). Since the number of multiplications (time-consuming on a computer) and additions required by the filter algorithm are proportional to n^3 and the storage is proportional to n^2 (where n is the dimension of the state vector), one significant means of reducing the computer burden is to *delete and combine states* [31, 33, 46, 47, 49]. There is often substantial physical insight into the relative significance of various states upon overall estimation precision that suggests which states might be removed. States with consistently small rms value especially warrant inspection for possible removal. An error budget performance analysis of the most complete filter, to be discussed in the next section, is an invaluable aid to this state dimension reduction.

EXAMPLE 6.1 A standard error model of an accelerometer [2, 4, 21, 41, 52] is as follows. Let an orthogonal coordinate system be denoted as having axes x, y, and z. Then the error in the output of the accelerometer whose sensitive axis is along the x coordinate direction, α_x, is:

$$\alpha_x = \mathsf{a}_{bx} + [f_x]\,\delta\mathsf{k}_x + [f_y]\,\delta\mathsf{k}_{xy} + [f_z]\,\delta\mathsf{k}_{xz} + \mathsf{a}_{rbx} + \mathsf{a}_{rbx2}$$

where a_{bx} is bias error (modeled as a random constant plus random walk), $\delta\mathsf{k}_x$ is scale factor error (modeled as a random constant), $\delta\mathsf{k}_{xy}$ is sensor misalignment about the y axis (modeled as a random constant), $\delta\mathsf{k}_{xz}$ is sensor misalignment about the z axis (modeled as a random constant), a_{rbx} is "random bias" error (modeled as an exponentially time-correlated noise, with short correlation time on the order of minutes), a_{rbx2} is "random bias" error (modeled as an exponentially time-correlated noise, with long correlation time, typically an hour or longer), and where f_x, f_y, and f_z are the components of true specific force relative to the x–y–z coordinate frame.

In most operational aided INS Kalman filters, only a single bias state is included, if any at all. A white noise source is added to indicate both the accelerometer error effects and the additional model uncertainty due to the deleted states. ■

EXAMPLE 6.2 With respect to the same x–y–z coordinates of Example 6.1, a generic model for the errors in a single-degree-of-freedom gyro with sensitive axis along the x coordinate direction is [2, 4, 21, 41, 52]:

$$\begin{aligned}\mathsf{e}_x = {} & \varepsilon_{dx} + [\omega_{icx}]\,\delta\mathsf{k}_x + \{[\omega_{icy}]\,\delta\mathsf{k}_{xy} + [\omega_{icz}]\,\delta\mathsf{k}_{xz}\} + \omega_{rbx} + \{[f_x]\mathsf{k}_{xx} + [f_y]\mathsf{k}_{xy} + [f_z]\mathsf{k}_{xz}\} \\ & + \{f_x{}^2]\mathsf{k}_{xxx} + [f_y{}^2]\mathsf{k}_{xyy} + [f_z{}^2]\mathsf{k}_{xzz}\} + \{[f_xf_y]\mathsf{k}_{xxy} + [f_xf_z]\mathsf{k}_{xxz} + [f_yf_z]\mathsf{k}_{xyz}\} + \mathsf{w}_x\end{aligned}$$

where ε_{dx} is gyro drift rate bias error (modeled as a random constant plus random walk), δk_x is scale factor error (modeled as a random constant), δk_{xi} is sensor misalignment about the $i = y$ or z axis (modeled as a random constant), ω_{rbx} is time-correlated gyro drift rate (modeled as an exponentially time-correlated noise), k_{xi} is a "g-sensitive error," sensitive to specific force in the $i = x, y$, or z direction (modeled as a random constant), k_{xii} is a "g^2-sensitive error," sensitive to the square of specific force in the $i = x$, y, or z direction (modeled as a random constant), k_{xij} is a "cross-term g^2-sensitive error," sensitive to the product of specific forces in the i and j directions, $i \neq j$ (modeled as a random constant), and w_x is the white noise gyro drift rate (not requiring a state). The coefficients ω_{icx}, ω_{icy}, and ω_{icz} are the components of the angular rate between inertial space and the x–y–z coordinate system, as measured in that system.

For conventional (fluid- or dry-tuned) gyros, only the ε_{dx} or ω_{rbx} state is usually retained for aided INS Kalman filters, since this drift rate predominates. In laser gyros, the w_x noise predominates (and the g- and g^2-sensitive errors are essentially zero), so no states are required in the Kalman filter for gyro errors, only a white noise model. ■

EXAMPLE 6.3 One conventional approach to reducing state dimension is to attempt to replace exponentially time-correlated noises (requiring single-state shaping filters) with appropriate white noises (requiring no states). Consider a stationary exponentially time-correlated zero-mean noise with rms value σ and correlation time T, i.e., with autocorrelation

$$\Psi_{xx}(\tau) = E\{x(t)x(t+\tau)\} = \sigma^2 e^{-|\tau|/T}$$

with $w(\cdot,\cdot)$ a zero-mean white Gaussian noise of strength $Q = 2\sigma^2/T$. The power spectral density for $x(\cdot,\cdot)$ is

$$\Psi_{xx}(\omega) = \frac{2\sigma^2/T}{\omega^2 + (1/T)^2}$$

which is plotted in Fig. 6.15.

The adequacy of replacing this noise by a white Gaussian noise is based on the premise that the system driven by the noise will attenuate high frequency content. Therefore, an appropriate strength to choose for the white noise is that which duplicates the low frequency power spectral density of $x(\cdot,\cdot)$, namely $2\sigma^2 T$. ■

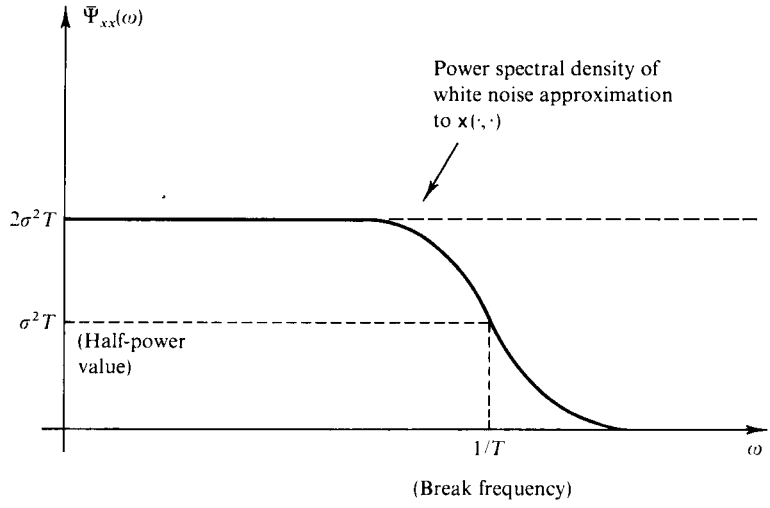


FIG. 6.15 White noise approximation to exponentially time-correlated noise.

It must be emphasized that deleting states and combining many states into fewer "equivalent" states must be evaluated in terms of resulting filter performance, as described in the next section. Experience has shown that reductions motivated by the best of physical insight can sometimes degrade estimation accuracy unacceptably. Moreover, an inappropriately reduced filter of state dimension n can often be outperformed by a filter involving less than n, differently chosen, states. The extreme case of this would be the higher-dimensioned filter being based on an unobservable model, for instance, in which two states correspond to different physical variables but which are indistinguishable from outputs from the model, while the lower-dimensioned filter model combined states to achieve observability.

The number of additions and multiplications required by a filter algorithm can be minimized by exploiting *canonical state variables*, since the matrices in this equivalent representation embody a high density of zeros. There is also some computational advantage to decomposing vector measurements into separate scalars or partitions, and *iteratively updating with lower-dimensioned measurements*. Obviously, one might also attempt to reduce the computational burden by *increasing the sample period* of the filter.

Once a filter dimension and structure are established, it is often possible to *neglect dominated terms* within the matrix elements. Moreover, entire *weak coupling terms can be removed*, yielding matrix entries of zero and thereby decreasing the number of required multiplications. In certain applications, such removal allows *decoupling the filter states* to generate separate, smaller filters.

EXAMPLE 6.4 In the Pinson error model for an INS implementing a north–east–down platform coordinate frame, one term in the $\mathbf{F}(t)$ matrix is $[2\omega_{ie}R_e \cos L(t) - v_e(t)]/R_e$, in which ω_{ie} is the earth rotation rate, $L(t)$ is current latitude, $v_e(t)$ is eastward vehicle velocity, and R_e is the earth radius. Since $[\omega_{ie}R_e]$ is on the order of 1000 knots, $v_e(t)$ can be neglected except for very high speed aircraft or near-polar operations. This entire term is small compared to Schuler frequency-dominated terms; when these weak coupling terms are ignored, the entire filter for aiding the INS with external source data often decouples into a horizontal plane filter and a separate vertical channel filter, with an acceptable (minute) amount of performance degradation. ■

Sometimes terms that can be ignored comprise the time-varying nature of the model description (or at least the most rapid variations), so that the filter can be based upon a *time-invariant model* (or at least a model that admits quasi-static analysis). This inherently yields computational advantages such as a single value for $\mathbf{\Phi}$, $\mathbf{B}_d$, and $\mathbf{Q}_d$ being valid for all time.

The methods discussed to this point have involved the generation of a simplified model, with subsequent filter construction. It is also advantageous to consider *approximating the filter structure* itself. Because of the separability of the conditional mean and covariance equations in the filter, it is possible to precompute and store the filter gains rather than calculate them online. This *precomputed filter gain history can often be approximated closely by curve-fitted*

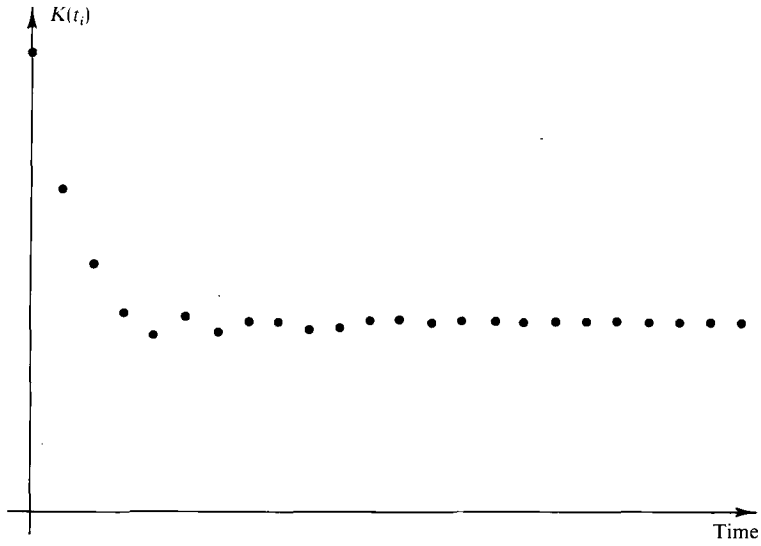


FIG. 6.16 Characteristic gain time history.

simple functions, such as piecewise constant functions, piecewise linear functions, and weighted exponentials. Thus, the filter covariance and gain calculations, which comprise the majority of the computer burden, are replaced by a minimal amount of required computation and storage. This is a tremendous benefit to the practicality of the online filter operation. For the case of a filter based on a time-invariant system model with stationary noises, a typical gain time history is depicted in Fig. 6.16. If, as in this plot, a short initial transient is followed by a long period of essentially steady state gain, the very simple approximation of using steady state gains for all time may be wholly adequate for desired performance. There are some drawbacks to using stored gain profiles. Future gains do not change appropriately when scheduled measurements are not made, due to data gaps or measurement rejection by reasonableness tests. Nor can prestored gains adapt online to compensate for filter divergence. Finally, lengthy simulations are often required to design a single gain history that will perform adequately under all possible conditions for an actual application.

6.8 PERFORMANCE (SENSITIVITY) ANALYSIS

Throughout the previous sections, the critical significance of an "adequate" system model within the filter structure was stressed. The question remains, how do you assess the adequacy of various filter designs relative to each other and/or to a set of performance specifications?

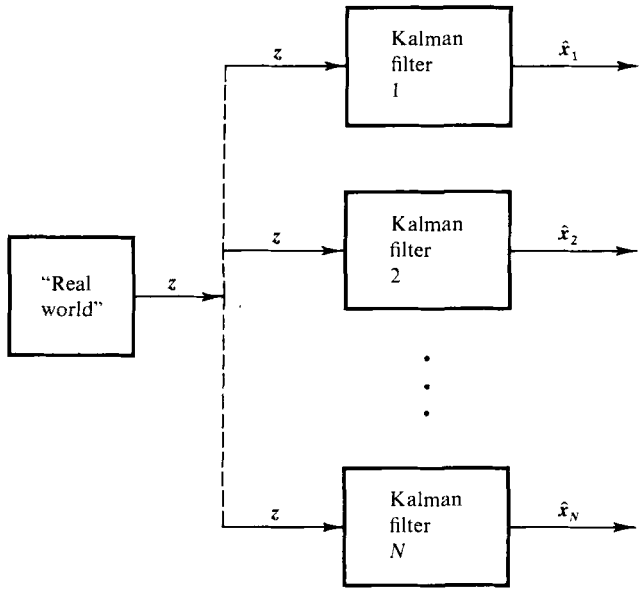


FIG. 6.17 The context of performance (sensitivity) analyses.

Consider Fig. 6.17, a schematic presentation of the situation under study. Suppose that system modeling and filter design efforts have produced N different prospective Kalman filters. Each is based upon a particular model of the "real world": a linear system with structure defined by $\{\mathbf{F}, \mathbf{B}, \mathbf{G}, \mathbf{H}\}$ (or $\{\mathbf{\Phi}, \mathbf{B}_d, \mathbf{G}_d, \mathbf{H}\}$ in terms of an equivalent discrete-time model) and uncertainties defined by $\{\hat{\mathbf{x}}_0, \mathbf{P}_0, \mathbf{Q}, \mathbf{R}\}$ (or $\{\hat{\mathbf{x}}_0, \mathbf{P}_0, \mathbf{Q}_d, \mathbf{R}\}$). The filters can vary greatly due to model differences, over a spectrum of small state dimension and deliberately crude approximations to high-dimensioned filters incorporating more system modes, cross-coupling effects, and high-order shaping filters for accurate portrayal of stochastic process characteristics. They might also differ due to aspects of algorithm implementation, such as Joseph versus standard covariance measurement update, gain history approximations, wordlength variations, etc.

In actual operation, any one of these filters would be driven by sampled data measurements from actual sensors operating in the "real world" environment. To make rational design decisions, one must have at his disposal an *accurate statistical portrayal of estimation errors committed by each filter in the "real world" environment.* Moreover, this information must be generated without actually building each filter and testing it in the "real world." A performance (sensitivity) analysis fulfills this objective by replacing the "real world" in Fig. 6.17 by the best, most complete mathematical model that can be developed, called a *truth model* or "reference model." Such a truth model is the product of extensive data analysis, and shaping filter design and validation, as

described in Section 4.13. It is essential to expend enough effort in its generation to be confident that *it* adequately represents the "real world," since the ensuing performance evaluation and systematic design procedure is totally dependent upon this assumption. For example, a very adequate generic model of the errors in an inertial navigation system has been constructed over the years in the form of a linear model of about 70 states driven by white Gaussian noise; thorough laboratory and flight testing of a particular INS allows complete specification of the model parameters to yield the "truth model" for that system. As inertial systems have improved, the "truth model" itself has become more refined so as to portray system characteristics accurately that were once considered insignificant. For example, in the next generation of systems, a dominant error source in addition to accelerometer and gyro uncertainties will be the difference between the true earth shape (geoid) and the assumed ellipsoid used in navigation computations. Without incorporating this effect, a "truth model" would be seriously inadequate.

It is desired to achieve a direct comparison of performance capabilities of filters that may well estimate completely different, and different numbers of, state variables. However, for any given application, there are certain variables of critical interest. In optimally aided inertial systems, for example, the nine variables of position, velocity, and attitude indications in the three axis directions are paramount. All prospective filters will estimate these quantities or variables functionally related to them. These critical variables, which we will denote as $\mathbf{y}(\cdot,\cdot)$, will serve as the basis of comparison of the filter designs. Although a *scalar* performance index is appealing from a mathematical and numerical optimization point of view, in practice the designer really seeks information about these critical variables individually, so attention will be focused on them.

Figure 6.18 depicts the means of conducting a performance analysis of a given Kalman filter design [8, 31, 33, 34, 36]. The truth model is an n_t-dimensional

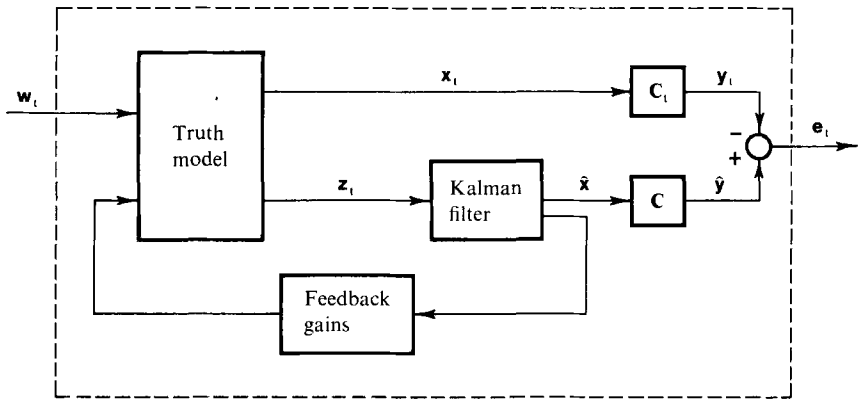


FIG. 6.18 Performance evaluation of a Kalman filter design.

state model, linear or nonlinear, driven by (white Gaussian) noise $\mathbf{w}_t(\cdot,\cdot)$, that accurately generates the measurement process $\mathbf{z}_t(\cdot,\cdot)$. The subscript t will be used to denote quantities and processes associated with the truth model. A sample from this discrete-time process, $\mathbf{z}_t(t_i,\omega_j) = \mathbf{z}_t(t_i)$ for all time $t_i \in T$, is then a representation of a measurement time history to be used as input for a single run of the filter algorithm. The filter operates on this input and generates the state estimate process $\hat{\mathbf{x}}(\cdot,\cdot)$. If it is implemented in feedback fashion as described in the preceding sections, the filter can also output feedback controls to the actual system, so a possible feedback path from the filter to the truth model is shown in the figure. Suppose that the p critical quantities are related to the filter states through a *linear* transformation:

$$\hat{\mathbf{y}}(t_i^-) = \mathbf{C}(t_i)\hat{\mathbf{x}}(t_i^-) \tag{6-55a}$$

$$\hat{\mathbf{y}}(t_i^+) = \mathbf{C}(t_i)\hat{\mathbf{x}}(t_i^+) \tag{6-55b}$$

Usually $\mathbf{C}(t_i)$ is a time-invariant p-by-n matrix, and often with partitions as a p-by-p identity matrix and p-by-$(n-p)$ zero matrix; i.e., the critical quantities compose the first p components of the filter state. Nonlinear functional relationships would be handled in an analogous manner, but we emphasize use of linear models here. Thus, the optimal estimates of the quantities of interest are generated as $\hat{\mathbf{y}}(t_i^-,\cdot)$ and $\hat{\mathbf{y}}(t_i^+,\cdot)$ for all times $t_i \in T$.

Being a mathematical representation, the truth model, unlike the "real world," generates a (continuous-time) state process $\mathbf{x}_t(\cdot,\cdot)$ as well as a measurement process $\mathbf{z}_t(\cdot,\cdot)$. We will assume that the true values of the critical quantities are related to this state by a linear transformation (again easily extended to a nonlinear function) represented by a p-by-n_t matrix $\mathbf{C}_t$ (often time-invariant):

$$\mathbf{y}_t(t,\cdot) = \mathbf{C}_t(t)\mathbf{x}_t(t,\cdot) \tag{6-56}$$

for all $t \in T$. This affords access to these true values, something which the "real world" denies us. Using (6-55) and (6-56), we can generate the "true" error committed by the Kalman filter in attempting to estimate the quantities of interest at time t_i, before and after measurement incorporation:

$$\mathbf{e}_t(t_i^-,\cdot) = \hat{\mathbf{y}}(t_i^-,\cdot) - \mathbf{y}_t(t_i,\cdot) \tag{6-57a}$$

$$\mathbf{e}_t(t_i^+,\cdot) = \hat{\mathbf{y}}(t_i^+,\cdot) - \mathbf{y}_t(t_i,\cdot) \tag{6-57b}$$

If we admit impulsive system response to the feedback control from the filter, as discussed in Section 6.5, then a third error, corresponding to after both measurement incorporation and impulsive control application, is of interest as well:

$$\mathbf{e}_t(t_i^{+c},\cdot) = \hat{\mathbf{y}}(t_i^{+c},\cdot) - \mathbf{y}_t(t_i^c,\cdot) \tag{6-57c}$$

The superscript c denotes after control is applied.

The objective of the performance analysis is to characterize the error process (6-57) statistically. It is also called a sensitivity analysis because we wish to evaluate the sensitivity of this performance to changing the filter structure. In addition, we may well want to study the effects of changing sensor hardware or system environment; i.e., altering the truth model. Because of using stochastic processes as the basic modeling entity, we are more interested in the statistical, or ensemble average, behavior of the error process than in a single sample out of this process.

One means of generating this statistical information is a *Monte Carlo study.* Essentially, many samples of the error process are generated by simulation and then the sample statistics computed directly. If enough samples are generated, these should approximate the process statistics very well. Unfortunately, this is a costly and time-consuming process.

If the truth model itself is in the form of a *linear* system driven by white Gaussian noise, from which are available *linear* measurements corrupted by white Gaussian noise, there is another, more efficient means of generating the statistical information, namely, a *mean analysis* [14] and a *covariance analysis* [11, 17, 18, 31]. Especially in the case of using an error state space Kalman filter, the means of all processes are often assumed to be zero for all time, and attention is concentrated on the covariance analysis. From *one* run of a covariance analysis is generated the time history of $\mathbf{P}_e(\cdot)$, the covariance of the true estimation errors committed by a given filter; the square roots of its diagonal terms yield the time histories of standard deviations (or "one-sigma values," equal to rms values if processes are zero mean) of errors in the estimates of the critical quantities of interest. This is directly comparable to the massive task of running the filter repeatedly, storing all pertinent performance data, and computing sample statistics, as required for a Monte Carlo analysis of the same filter.

To develop the performance analysis equations, consider Fig. 6.18 again. If the truth model is itself a linear system model, then the entire system enclosed by the dashed lines is nothing but a linear system driven only by white Gaussian noise. As seen in Chapter 4, if we want to characterize the output process from such a system model, we must first characterize the state process within the system, in this case being composed of the partitions $\mathbf{x}_t(\cdot, \cdot)$ of the truth model and $\hat{\mathbf{x}}(\cdot, \cdot)$ of the filter under investigation. First we will develop the stochastic process description appropriate to a Monte Carlo analysis, and from it develop the statistical description appropriate to a covariance analysis. The derivation will assume all processes to be zero mean, appropriate to error state space Kalman filters, and therefore deterministic inputs $\mathbf{u}(\cdot)$ are neglected; an extension relaxing these assumptions would be straightforward but more complicated to account for nonzero means and biased estimates.

The *truth model* is described by the stochastic differential n_t-dimensional state equation

$$\mathbf{dx}_t(t) = \mathbf{F}_t(t)\mathbf{x}_t(t)\,dt + \mathbf{G}_t(t)\,\mathbf{d\beta}_t(t) \tag{6-58a}$$

or

$$\dot{\mathbf{x}}_t(t) = \mathbf{F}_t(t)\mathbf{x}_t(t) + \mathbf{G}_t(t)\mathbf{w}_t(t) \tag{6-58b}$$

where $\boldsymbol{\beta}_t(\cdot,\cdot)$ is an s_t-vector Brownian motion of diffusion $\mathbf{Q}_t(t)$ for all $t \in T$:

$$E\{\boldsymbol{\beta}_t(t)\} = \mathbf{0} \tag{6-59a}$$

$$E\{[\boldsymbol{\beta}_t(t) - \boldsymbol{\beta}_t(t')][\boldsymbol{\beta}_t(t) - \boldsymbol{\beta}_t(t')]^T\} = \int_{t'}^{t} \mathbf{Q}_t(\tau)\, d\tau \tag{6-59b}$$

or $\mathbf{w}_t(\cdot,\cdot)$ is an s_t-vector zero-mean white Gaussian noise of strength $\mathbf{Q}_t(t)$ for all $t \in T$:

$$E\{\mathbf{w}_t(t)\} = \mathbf{0} \tag{6-60a}$$

$$E\{\mathbf{w}_t(t)\mathbf{w}_t^T(t')\} = \mathbf{Q}_t(t)\delta(t - t') \tag{6-60b}$$

Note that (6-58) does not as yet account for feedback from the Kalman filter: the necessary modifications will be incorporated once the filter is described. The initial condition for this differential equation is that $\mathbf{x}_t(t_0)$ is described as a zero-mean Gaussian random variable with covariance $\mathbf{P}_{t0}$:

$$E\{\mathbf{x}_t(t_0)\} = \mathbf{0} \tag{6-61a}$$

$$E\{\mathbf{x}_t(t_0)\mathbf{x}_t^T(t_0)\} = \mathbf{P}_{t0} \tag{6-61b}$$

Available from the truth model at discrete times t_i are the m-dimensional measurements $\mathbf{z}_t(t_i,\cdot)$:

$$\mathbf{z}_t(t_i) = \mathbf{H}_t(t_i)\mathbf{x}_t(t_i) + \mathbf{v}_t(t_i) \tag{6-62}$$

where $\mathbf{v}_t(\cdot,\cdot)$ is a discrete-time m-vector white Gaussian noise described by

$$E\{\mathbf{v}_t(t_i)\} = \mathbf{0} \tag{6-63a}$$

$$E\{\mathbf{v}_t(t_i)\mathbf{v}_t^T(t_j)\} = \begin{cases} \mathbf{R}_t(t_i) & t_i = t_j \\ \mathbf{0} & t_i \neq t_j \end{cases} \tag{6-63b}$$

In terms of the truth model state, the quantities of direct interest to the performance evaluation are denoted as the continuous-time p-vector process $\mathbf{y}_t(\cdot,\cdot)$:

$$\mathbf{y}_t(t) = \mathbf{C}_t(t)\mathbf{x}_t(t) \tag{6-64}$$

The Kalman filter being analyzed is based upon a different, generally lower-dimensioned, system model, called a *design model* (this model never explicitly appears in the performance analysis, it just serves to generate the filter). The n-dimensional design state equation is

$$\mathbf{dx}(t) = \mathbf{F}(t)\mathbf{x}(t)\, dt + \mathbf{G}(t)\, \mathbf{d}\boldsymbol{\beta}(t) \tag{6-65a}$$

or

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{G}(t)\mathbf{w}(t) \tag{6-65b}$$

with s-vector driving noise of diffusion (strength) $\mathbf{Q}(t)$ for all $t \in T$:

$$E\{\boldsymbol{\beta}(t)\} = \mathbf{0} \tag{6-66a}$$

$$E\{[\boldsymbol{\beta}(t) - \boldsymbol{\beta}(t')][\boldsymbol{\beta}(t) - \boldsymbol{\beta}(t')]^{\mathrm{T}}\} = \int_{t'}^{t} \mathbf{Q}(\tau)\, d\tau \tag{6-66b}$$

$$E\{\mathbf{w}(t)\} = \mathbf{0} \tag{6-67a}$$

$$E\{\mathbf{w}(t)\mathbf{w}^{\mathrm{T}}(t')\} = \mathbf{Q}(t)\delta(t - t') \tag{6-67b}$$

The initial condition $\mathbf{x}(t_0)$ is Gaussian with

$$E\{\mathbf{x}(t_0)\} = \mathbf{0} \tag{6-68a}$$

$$E\{\mathbf{x}(t_0)\mathbf{x}^{\mathrm{T}}(t_0)\} = \mathbf{P}_0 \tag{6-68b}$$

The design model of the measurement history is the discrete-time m-vector process $\mathbf{z}(\cdot, \cdot)$

$$\mathbf{z}(t_i) = \mathbf{H}(t_i)\mathbf{x}(t_i) + \mathbf{v}(t_i) \tag{6-69}$$

with $\mathbf{v}(\cdot, \cdot)$ a discrete-time m-vector white Gaussian noise with

$$E\{\mathbf{v}(t_i)\} = \mathbf{0} \tag{6-70a}$$

$$E\{\mathbf{v}(t_i)\mathbf{v}^{\mathrm{T}}(t_j)\} = \begin{cases} \mathbf{R}(t_i) & t_i = t_j \\ \mathbf{0} & t_i \neq t_j \end{cases} \tag{6-70b}$$

Finally, the outputs of most concern are described by the p-vector $\mathbf{y}(\cdot, \cdot)$

$$\mathbf{y}(t) = \mathbf{C}(t)\mathbf{x}(t) \tag{6-71}$$

Despite the notational similarities of (6-65)–(6-71) and (6-58)–(6-64), the two models are distinctly different.

The *Kalman filter* based upon this design model is specified between update times by the time propagation equations

$$\hat{\mathbf{x}}(t_i^-) = \boldsymbol{\Phi}(t_i, t_{i-1})\hat{\mathbf{x}}(t_{i-1}^+) \tag{6-72a}$$

$$\mathbf{P}(t_i^-) = \boldsymbol{\Phi}(t_i, t_{i-1})\mathbf{P}(t_{i-1}^+)\boldsymbol{\Phi}^{\mathrm{T}}(t_i, t_{i-1}) + \mathbf{Q}_{\mathrm{d}}(t_{i-1}) \tag{6-72b}$$

$$\mathbf{Q}_{\mathrm{d}}(t_{i-1}) = \int_{t_{i-1}}^{t_i} \boldsymbol{\Phi}(t_i, \tau)\mathbf{G}(\tau)\mathbf{Q}(\tau)\mathbf{G}^{\mathrm{T}}(\tau)\boldsymbol{\Phi}^{\mathrm{T}}(t_i, \tau)\, d\tau \tag{6-72c}$$

or, equivalently,

$$\dot{\hat{\mathbf{x}}}(t/t_{i-1}) = \mathbf{F}(t)\hat{\mathbf{x}}(t/t_{i-1}) \tag{6-73a}$$

$$\dot{\mathbf{P}}(t/t_{i-1}) = \mathbf{F}(t)\mathbf{P}(t/t_{i-1}) + \mathbf{P}(t/t_{i-1})\mathbf{F}^{\mathrm{T}}(t) + \mathbf{G}(t)\mathbf{Q}(t)\mathbf{G}^{\mathrm{T}}(t) \tag{6-73b}$$

solved forward to time t_i from the initial conditions

$$\hat{\mathbf{x}}(t_{i-1}/t_{i-1}) = \hat{\mathbf{x}}(t_{i-1}^+) \tag{6-73c}$$

$$\mathbf{P}(t_{i-1}/t_{i-1}) = \mathbf{P}(t_{i-1}^+) \tag{6-73d}$$

At measurement sample times, the update relations are

$$\mathbf{K}(t_i) = \mathbf{P}(t_i^-)\mathbf{H}^{\mathrm{T}}(t_i)[\mathbf{H}(t_i)\mathbf{P}(t_i^-)\mathbf{H}^{\mathrm{T}}(t_i) + \mathbf{R}(t_i)]^{-1} \tag{6-74a}$$

$$\hat{\mathbf{x}}(t_i^+) = \hat{\mathbf{x}}(t_i^-) + \mathbf{K}(t_i)[\mathbf{z}_t(t_i) - \mathbf{H}(t_i)\hat{\mathbf{x}}(t_i^-)] \tag{6-74b}$$

$$\mathbf{P}(t_i^+) = \mathbf{P}(t_i^-) - \mathbf{K}(t_i)\mathbf{H}(t_i)\mathbf{P}(t_i^-) \tag{6-74c}$$

$$= [\mathbf{I} - \mathbf{K}(t_i)\mathbf{H}(t_i)]\mathbf{P}(t_i^-)[\mathbf{I} - \mathbf{K}(t_i)\mathbf{H}(t_i)]^{\mathrm{T}} + \mathbf{K}(t_i)\mathbf{R}(t_i)\mathbf{K}^{\mathrm{T}}(t_i) \tag{6-74d}$$

Note the appearance of $\mathbf{z}_t(\cdot,\cdot)$ in (6-74b). Initial conditions for the algorithm are

$$\hat{\mathbf{x}}(t_0) = \mathbf{0} \tag{6-75a}$$

$$\mathbf{P}(t_0) = \mathbf{P}_0 \tag{6-75b}$$

Note that these equations have been written as a stochastic process description, one sample of which will correspond to a single operation of the filter: this is done since we want to evaluate the ensemble average performance of the filter. In terms of this algorithm, the estimated values of the critical quantities of interest at sample time t_i are:

$$\hat{\mathbf{y}}(t_i^-) = \mathbf{C}(t_i)\hat{\mathbf{x}}(t_i^-) \tag{6-76a}$$

$$\hat{\mathbf{y}}(t_i^+) = \mathbf{C}(t_i)\hat{\mathbf{x}}(t_i^+) \tag{6-76b}$$

and, if we wanted these estimates any time in the sample period $[t_{i-1}, t_i)$, (6-73) could be used to generate

$$\hat{\mathbf{y}}(t) = \mathbf{C}(t)\hat{\mathbf{x}}(t/t_{i-1}) \tag{6-76c}$$

Since a Kalman filter will often be implemented in an indirect feedback configuration, in which error state estimates are fed back to the actual system to try to correct it, the preceding truth model and filter relations will now be modified to allow performance analysis of such a configuration. One type of feedback discussed previously is the *impulsive control or discrete-time reset.* Quantities maintained as contents of computer memory locations (or outputs of integrators) can be changed instantaneously based upon the error estimate $\hat{\mathbf{x}}(t_i^+)$. Examples of variables controlled in this manner in aided inertial systems are position, velocity, and direction cosine matrix elements for attitude information. Maximum rate torquing of the INS platform is commanded to remove estimated misalignments; although this is not an instantaneous change, it is accomplished rapidly enough compared to the filter iteration period that it is well approximated as impulsive. Once $\hat{\mathbf{x}}(t_i^+)$ is computed, the reset control is calculated as a function (assumed linear) of it, $\mathbf{D}_t(t_i)\hat{\mathbf{x}}(t_i^+)$, where the notation calls out that this is a discrete-time control that affects the true system (truth model). After application of the control, the truth model state process description becomes

$$\mathbf{x}_t(t_i^c) = \mathbf{x}_t(t_i) - \mathbf{D}_t(t_i)\hat{\mathbf{x}}(t_i^+) \tag{6-77}$$

The filter should be "told" that this feedback to the system has occurred, so its state estimate is modified as

$$\hat{\mathbf{x}}(t_i^{+c}) = \hat{\mathbf{x}}(t_i^{+}) - \mathbf{D}(t_i)\hat{\mathbf{x}}(t_i^{+}) = [\mathbf{I} - \mathbf{D}(t_i)]\hat{\mathbf{x}}(t_i^{+}) \tag{6-78}$$

where the n-by-n $\mathbf{D}(t_i)$ is meant to model the effect of feedback through the actual n_t-by-n gains $\mathbf{D}_t(t_i)$ into the system. This $\hat{\mathbf{x}}(t_i^{+c})$ replaces $\hat{\mathbf{x}}(t_i^{+})$ as the initial condition for the next time propagation.

Some true system variables are controlled over the entire sample period rather than impulsively. As discussed in Section 6.5, gyro drift rate compensation is achieved by torquing the gyro with a constant (analog signal or pulse rate) control over a sample period $[t_{i-1}, t_i)$, proportional to the negative of the drift rate estimate $\hat{\varepsilon}(t_{i-1}^{+})$ obtained at the beginning of the interval. This form of feedback could be implemented directly into the performance analysis, but a simpler and good approximate model is to treat the feedback as *continuous control*, $\mathbf{X}_t(t)\hat{\mathbf{x}}(t/t_{i-1})$. Thus, the truth model equation (6-58) is modified to

$$\dot{\mathbf{x}}_t(t) = \mathbf{F}_t(t)\mathbf{x}_t(t) - \mathbf{X}_t(t)\hat{\mathbf{x}}(t/t_{i-1}) + \mathbf{G}_t(t)\mathbf{w}_t(t) \tag{6-79}$$

Again, the filter is to be informed of such feedback, so (6-73a) becomes

$$\dot{\hat{\mathbf{x}}}(t/t_{i-1}) = \mathbf{F}(t)\hat{\mathbf{x}}(t/t_{i-1}) - \mathbf{X}(t)\hat{\mathbf{x}}(t/t_{i-1}) = [\mathbf{F}(t) - \mathbf{X}(t)]\hat{\mathbf{x}}(t/t_{i-1}) \tag{6-80}$$

Similarly, $\mathbf{F}(t)$ is replaced by $[\mathbf{F}(t) - \mathbf{X}(t)]$ in (6-73b), or equivalently, the state transition matrix that appears in (6-72) is associated with $[\mathbf{F}(t) - \mathbf{X}(t)]$ rather than $\mathbf{F}(t)$.

At this point, we have described the truth model and Kalman filter that appear in Fig. 6.18. For convenience, we now define the augmented state vector process $\mathbf{x}_a(\cdot,\cdot)$ for this entire configuration: for any time in the interval $[t_{i-1}, t_i)$,

$$\mathbf{x}_a(\cdot,\cdot) = \begin{bmatrix} \mathbf{x}_t(\cdot,\cdot) \\ \hline \hat{\mathbf{x}}(\cdot/t_{i-1},\cdot) \end{bmatrix} \tag{6-81}$$

i.e., an $(n_t + n)$-vector stochastic process with partitions of the truth model and filter states, respectively. From (6-79) and (6-80), the augmented state vector *time propagation* relation is:

$$\dot{\mathbf{x}}_a(t) = \mathbf{F}_a(t)\mathbf{x}_a(t) + \mathbf{G}_a(t)\mathbf{w}_t(t) \tag{6-82}$$

with

$$\mathbf{F}_a(t) = \begin{bmatrix} \mathbf{F}_t(t) & -\mathbf{X}_t(t) \\ \hline \mathbf{0} & [\mathbf{F}(t) - \mathbf{X}(t)] \end{bmatrix} \qquad \mathbf{G}_a(t) = \begin{bmatrix} \mathbf{G}_t(t) \\ \hline \mathbf{0} \end{bmatrix} \tag{6-83}$$

solved forward from time t_{i-1} with the initial conditions

$$\mathbf{x}_a(t_{i-1}^{+c}) = \begin{bmatrix} \mathbf{x}_t(t_{i-1}^{c}) \\ \hline \hat{\mathbf{x}}(t_{i-1}^{+c}) \end{bmatrix} \tag{6-84}$$

This can be written equivalently through the discrete-time solution model:

$$\mathbf{x}_a(t_i^-) = \mathbf{\Phi}_a(t_i, t_{i-1})\mathbf{x}_a(t_{i-1}^{+c}) + \mathbf{w}_{da}(t_{i-1}) \tag{6-85}$$

where $\mathbf{\Phi}_a(t_i, t_{i-1})$ is the $(n_t + n)$-by-$(n_t + n)$ matrix that satisfies

$$\dot{\mathbf{\Phi}}_a(t, t_{i-1}) = \mathbf{F}_a(t)\mathbf{\Phi}_a(t, t_{i-1}) \tag{6-86a}$$

$$\mathbf{\Phi}_a(t_{i-1}, t_{i-1}) = \mathbf{I} \tag{6-86b}$$

and $\mathbf{w}_{da}(\cdot, \cdot)$ is a discrete-time $(n_t + n)$-vector white Gaussian noise process with mean zero and covariance:

$$\begin{aligned} E\{\mathbf{w}_{da}(t_{i-1})\mathbf{w}_{da}^T(t_{i-1})\} &= \mathbf{Q}_{da}(t_{i-1}) \\ &= \int_{t_{i-1}}^{t_i} \mathbf{\Phi}_a(t_i, \tau)\mathbf{G}_a(\tau)\mathbf{Q}_t(\tau)\mathbf{G}_a^T(\tau)\mathbf{\Phi}_a^T(t_i, \tau)\, d\tau \end{aligned} \tag{6-87}$$

To generate the *measurement update* relations, first of all the truth model state is unchanged:

$$\mathbf{x}_t(t_i^+) = \mathbf{x}_t(t_i^-) \tag{6-88}$$

The filter update can be written as:

$$\begin{aligned} \hat{\mathbf{x}}(t_i^+) &= \hat{\mathbf{x}}(t_i^-) + \mathbf{K}(t_i)[\mathbf{z}_t(t_i) - \mathbf{H}(t_i)\hat{\mathbf{x}}(t_i^-)] \\ &= [\mathbf{I} - \mathbf{K}(t_i)\mathbf{H}(t_i)]\hat{\mathbf{x}}(t_i^-) + \mathbf{K}(t_i)\mathbf{z}_t(t_i) \\ &= [\mathbf{I} - \mathbf{K}(t_i)\mathbf{H}(t_i)]\hat{\mathbf{x}}(t_i^-) + \mathbf{K}(t_i)\mathbf{H}_t(t_i)\mathbf{x}_t(t_i) + \mathbf{K}(t_i)\mathbf{v}_t(t_i) \end{aligned} \tag{6-89}$$

Equations (6-88) and (6-89) yield an augmented state vector measurement update as

$$\mathbf{x}_a(t_i^+) = \mathbf{A}_a(t_i)\mathbf{x}_a(t_i^-) + \mathbf{K}_a(t_i)\mathbf{v}_t(t_i) \tag{6-90}$$

where

$$\mathbf{A}_a(t_i) = \left[\begin{array}{c|c} \mathbf{I} & \mathbf{0} \\ \hline \mathbf{K}(t_i)\mathbf{H}_t(t_i) & [\mathbf{I} - \mathbf{K}(t_i)\mathbf{H}(t_i)] \end{array}\right] \tag{6-91a}$$

$$\mathbf{K}_a(t_i) = \left[\begin{array}{c} \mathbf{0} \\ \hline \mathbf{K}(t_i) \end{array}\right] \tag{6-91b}$$

The *impulsive (reset) control update* can be described, using (6-77) and (6-78), as

$$\mathbf{x}_a(t_i^{+c}) = \mathbf{D}_a(t_i)\mathbf{x}_a(t_i^+) \tag{6-92}$$

where

$$\mathbf{D}_a(t_i) = \left[\begin{array}{c|c} \mathbf{I} & -\mathbf{D}_t(t_i) \\ \hline \mathbf{0} & [\mathbf{I} - \mathbf{D}(t_i)] \end{array}\right] \tag{6-93}$$

Note that if feedback is not employed, $\mathbf{D}_a(t_i)$ is simply an $(n_t + n)$-by-$(n_t + n)$ identity matrix. Initial conditions for these relations are:

$$\mathbf{x}_a(t_0) = \begin{bmatrix} \mathbf{x}_t(t_0) \\ \hat{\mathbf{x}}(t_0) \end{bmatrix} = \begin{bmatrix} \mathbf{x}_t(t_0) \\ \mathbf{0} \end{bmatrix} \tag{6-94}$$

Finally, the output $\mathbf{e}_t(\cdot, \cdot)$ in Fig. 6.16, the error committed by the filter in estimating the essential quantities, can be generated from the augmented state vector at any time $t \in T$ as

$$\mathbf{e}_t(t) = \mathbf{C}_a(t)\mathbf{x}_a(t) \tag{6-95}$$

where

$$\mathbf{C}_a(t) = [-\mathbf{C}_t(t) \mid \mathbf{C}(t)] \tag{6-96}$$

In a *Monte Carlo study*, these relationships are used to generate individual *samples* of stochastic processes, employing random number generators to obtain each simulation (see Problem 7.14 for a discussion of process sample generation). A set of initial conditions is established through a realization of (6-94), time propagation conducted through a sample from (6-82) or (6-85), and measurement and control updates via (6-90) and (6-92). Thus, a sample of the estimation error process is computed from (6-95). The output from a single computer run of a Monte Carlo study is a single sample $\mathbf{e}_t(t, \omega_1)$ for all time t of interest. A second run produces $\mathbf{e}_t(t, \omega_2)$ for all t, and so forth, as portrayed in Fig. 6.19. A statistical description of this error process is achieved by computing sample statistics, averaging over the number of runs conducted.

The equations for a *covariance analysis* [8, 17, 18, 31] are readily obtained from these relations. Since all processes are assumed to be zero mean, the covariances of the augmented state and estimation error can be defined as

$$\mathbf{P}_a(t) = E\{\mathbf{x}_a(t)\mathbf{x}_a^{\mathrm{T}}(t)\} \tag{6-97}$$

$$\mathbf{P}_e(t) = E\{\mathbf{e}_t(t)\mathbf{e}_t^{\mathrm{T}}(t)\} \tag{6-98}$$

The time history of $\mathbf{P}_e(t)$ is the desired output and $\mathbf{P}_a(t)$ is calculated as a means of obtaining this result. The appropriate *initial conditions* are obtained from (6-94) as

$$\mathbf{P}_a(t_0) = E\{\mathbf{x}_a(t_0)\mathbf{x}_a^{\mathrm{T}}(t_0)\} = \begin{bmatrix} \mathbf{P}_{t0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_0 \end{bmatrix} \tag{6-99}$$

Propagating between sample times is accomplished by integrating

$$\dot{\mathbf{P}}_a(t) = \mathbf{F}_a(t)\mathbf{P}_a(t) + \mathbf{P}_a(t)\mathbf{F}_a^{\mathrm{T}}(t) + \mathbf{G}_a(t)\mathbf{Q}_t(t)\mathbf{G}_a^{\mathrm{T}}(t) \tag{6-100}$$

forward from time t_{i-1}, with initial conditions as $\mathbf{P}_a(t_{i-1}^{+c})$, to time t_i, as seen from (6-82)–(6-84). This can be expressed equivalently, from (6-85)–(6-87), as

$$\mathbf{P}_a(t_i^{-}) = \mathbf{\Phi}_a(t_i, t_{i-1})\mathbf{P}_a(t_{i-1}^{+c})\mathbf{\Phi}_a^{\mathrm{T}}(t_i, t_{i-1}) + \mathbf{Q}_{da}(t_{i-1}) \tag{6-101}$$

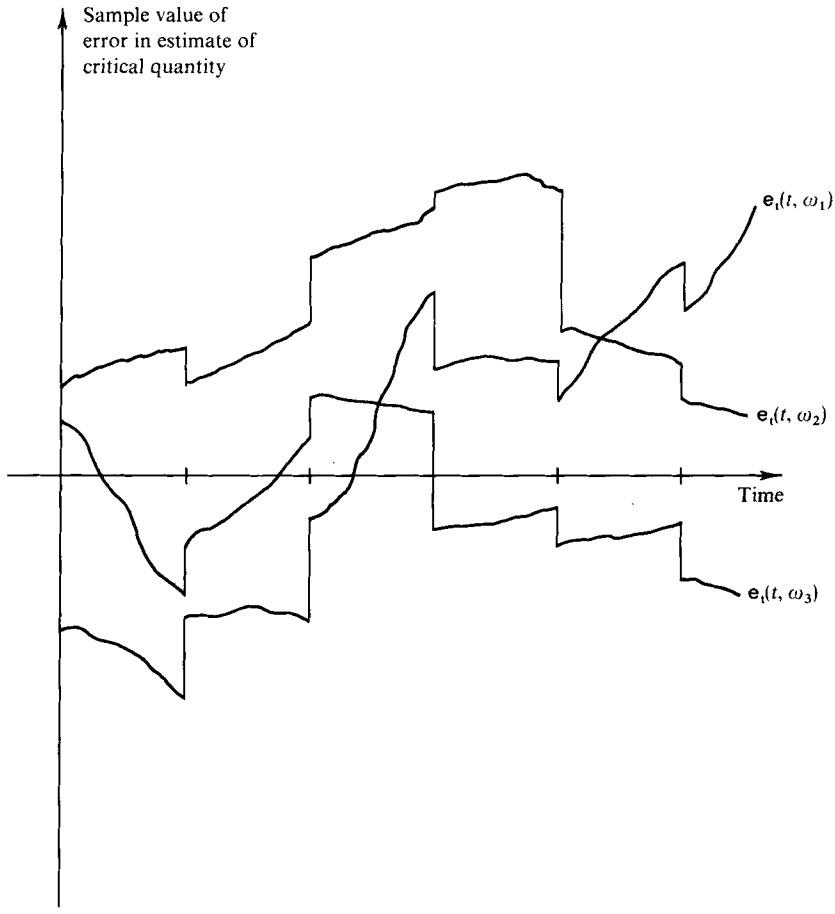


FIG. 6.19 Outputs of three Monte Carlo simulation runs.

The *measurement update* relation derived from (6-90) and (6-91) is

$$\mathbf{P}_a(t_i^+) = \mathbf{A}_a(t_i)\mathbf{P}_a(t_i^-)\mathbf{A}_a^T(t_i) + \mathbf{K}_a(t_i)\mathbf{R}_t(t_i)\mathbf{K}_a^T(t_i) \tag{6-102}$$

From (6-92) and (6-93), the *impulsive* (*reset*) *control update* is

$$\mathbf{P}_a(t_i^{+c}) = \mathbf{D}_a(t_i)\mathbf{P}_a(t_i^+)\mathbf{D}_a^T(t_i) \tag{6-103}$$

As the state covariance recursion is generated, the desired *error covariance* can be obtained at any time of interest, using (6-95) and (6-96), as

$$\mathbf{P}_e(t) = \mathbf{C}_a(t)\mathbf{P}_a(t)\mathbf{C}_a^T(t) \tag{6-104}$$

Because the augmented system is a linear system driven by white Gaussian noise, the covariance relations are independent of the actual measurement

time history, so it is possible to perform this covariance analysis *a priori, without* resorting to noise generator simulation of process samples.

To conduct a covariance analysis, one must explicitly define the structure and uncertainties of the truth model ($\mathbf{F}_t$ or $\mathbf{\Phi}_t$, $\mathbf{G}_t$, $\mathbf{H}_t$, $\mathbf{Q}_t$, and $\mathbf{R}_t$ time histories and $\mathbf{P}_{t0}$) and of the design model upon which the filter is based ($\mathbf{F}$ or $\mathbf{\Phi}$, $\mathbf{G}$, $\mathbf{H}$, $\mathbf{Q}$, and $\mathbf{R}$ time histories and $\mathbf{P}_0$). Note that the only effect of $\mathbf{Q}(\cdot)$ and $\mathbf{R}(\cdot)$ in these equations is to establish the filter gain $\mathbf{K}(\cdot)$: the design model $\mathbf{w}(\cdot,\cdot)$ and $\mathbf{v}(\cdot,\cdot)$ do not directly affect the spreading of samples of the $\hat{\mathbf{x}}(\cdot,\cdot)$ process. Moreover, a filter modification which incorporates approximated gains instead can be analyzed readily.

One particular use of a covariance performance (sensitivity) analysis is a systematic approach to the *filter tuning* [13, 15, 16, 20, 24, 27, 33, 37–39, 50, 51] process. The basic objective of filter tuning is to achieve the best possible estimation performance from a filter of specified structural form, i.e., totally specified except for $\mathbf{P}_0$ and the time histories of $\mathbf{Q}$ and $\mathbf{R}$. These covariances not only account for actual noises and disturbances in the physical system, but also are a means of declaring how adequately the assumed model represents the "real world" system. The simpler and less accurate the model, the stronger the noise strengths should be set, but it is difficult if not impossible to specify best numerical values a priori.

In the tuning process, the filter structure and the entire truth model remain fixed. For one set of values of $\mathbf{P}_0$ and time histories of $\mathbf{Q}$ and $\mathbf{R}$ in the filter, the covariance analysis provides the time history of the "true" rms errors in estimates of critical quantities committed by the filter, obtained from the square roots of diagonal terms of $\mathbf{P}_e$. Another set of these noise parameters can be chosen, thereby generating another time history of rms estimation errors. This procedure can be repeated until the "best" choice of parameters is found, best in the sense that filter errors are minimized. Thus, the tuning process can be considered a numerical optimization problem, a constant parameter optimization if stationary statistics are assumed, or a significantly more difficult function optimization if stationarity is not assumed. As such, it can be solved by automatic search methods or by manual calculation: using physical insights to propose changes in the noise covariances, and continuing to vary them until the performance no longer improves. Manual "optimization" is more prevalent in practice. Basically, the $\mathbf{P}_0$ matrix is the determining factor in the initial transient performance of the filter, whereas the $\mathbf{Q}$ and $\mathbf{R}$ histories dictate the longer term ("steady state" if time-invariant system and stationary noise models apply) performance and time duration of transients.

When performing the filter tuning, it is useful to compare the actual rms errors committed by the filter to the filter's own representation of its errors: the $\mathbf{P}$ covariance time history computed as an integral part of the estimator algorithm. This time history is also directly available as an output from the covariance analysis [$\mathbf{P}_e(t)$ would be directly comparable to $\mathbf{C}(t)\mathbf{P}(t)\mathbf{C}^T(t)$, but

all filter channels are usually observed individually for tuning purposes, so here we consider $\mathbf{C}(t) \triangleq \mathbf{I}$ and $\mathbf{C}_a(t)$ defined appropriately]. One would want the filter to perform as well as it "believes" it is performing. If, due to mistuned noise parameters, the filter underestimates its own errors, it will not "look hard enough" at the measurements, with resulting filter divergence problems if the discrepancy is significant enough [12, 43, 44], as depicted in Fig. 6.20a.

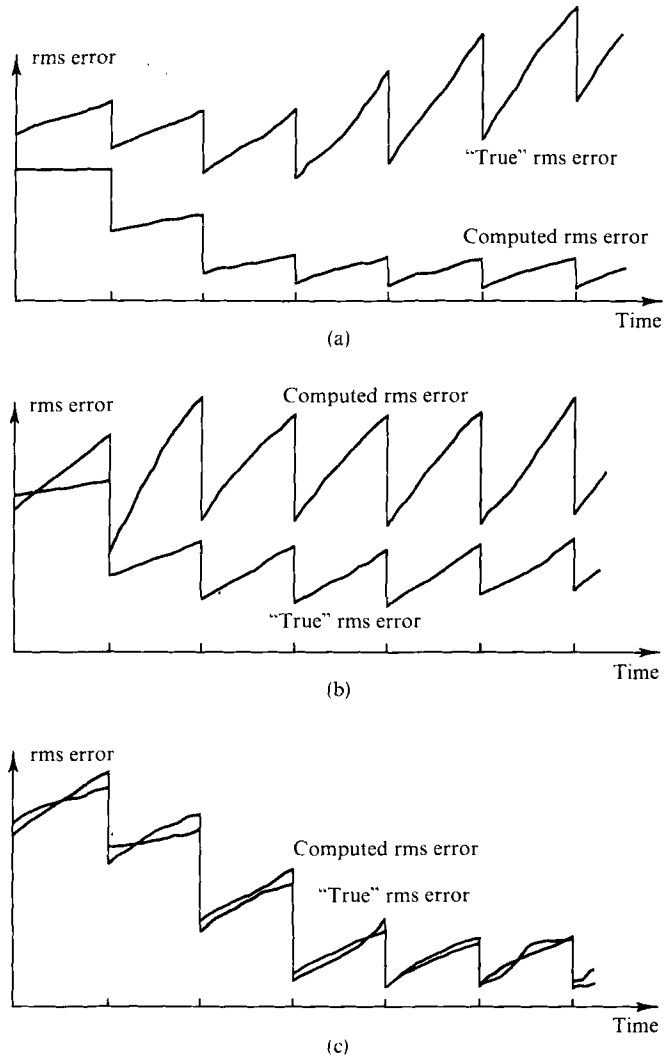


FIG. 6.20 Filter tuning through covariance analysis. (a) Filter underestimates its own errors: divergence. (b) Filter overestimates its own errors: tracking of measurement noise. (c) Well-tuned filter.

On the other hand, if the filter's internally generated covariance overestimates the errors being committed, then the filter weights the measurements too heavily, expending too much effort to track the noisy data and not exploiting the benefits of its internal model enough, as seen in Fig. 6.20b. By choosing the noise parameters so that the overall time histories of actual rms errors and the square roots of the filter **P** matrix diagonal terms match well, the actual estimation errors are effectively reduced at the same time. This is portrayed in Fig. 6.20c: note that the true rms error is lower than in the two preceding plots. Allowing the filter to overestimate its own errors slightly, so as to minimize the likelihood of divergence, is a commonly adopted means of generating a conservative filter design.

Other means of filter tuning are possible as well. If truth model noise (and structure) parameters are known only with some uncertainty, it is possible to tune a reduced order filter by a game theoretic (minimax) approach, in which the uncertain parameters "try" to maximize the estimation errors and the filter design parameters "try" to minimize it. The result is a single filter design with acceptable performance over the entire range of uncertain parameter values [9], without increased online computation required, as would be the case for an adaptive estimator.

Once a particular filter has been tuned, an *error budget* [1, 15, 16, 20, 24, 27, 33, 37–39, 50, 51] can be established. Essentially, this consists of repeated covariance analyses in which the error sources (or small groups of sources) in the truth model are "turned on" individually to determine the separate effects of these sources. At particular times of interest, the rms errors in estimates of quantities of interest due only to a single source of error are recorded. For instance, Fig. 6.21 plots a north position rms error budget for a particular filter considered for a Doppler-aided INS application, at a point following a long period of overwater flight (a worst-case scenario for such a navigation system). This information is useful for a number of purposes. If the filter under test were based upon the full-scale truth model, this would indicate that it is most essential for the filter to model INS gyro drift rate (three single-state shaping filters) and wideband Doppler noise (modeled as white noise): it suggests a filter that neglects the states modeling the other contributions and increasing the strength of noises to account for such deletion. If the filter under test were such a practical design, and tuned properly, then this error budget would indicate navigation errors being dominated by gyro drift rate and Doppler noise: if hardware were to be improved, the most cost-effective improvement would be an INS with better gyro drift characteristics and/or a better precision Doppler. Note that, because of the linearity of the covariance analysis equations, the total navigation rms error can be found as the root-sum-square of the individual independent contributions.

Besides tuning or generating an error budget for a single filter, a covariance analysis provides an effective means of conducting a *tradeoff analysis* among

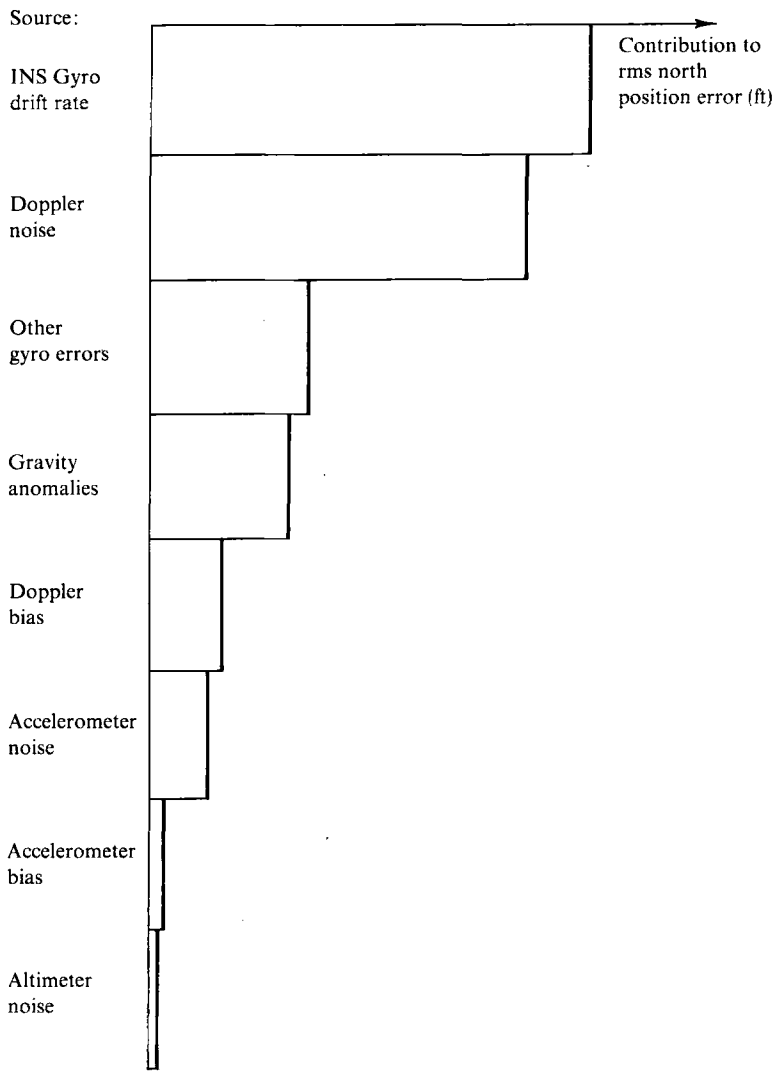


FIG. 6.21 Navigation sensor error budget subsequent to long overwater flight.

the various proposed designs. By explicitly evaluating the performance capabilities and computer burden of each design, the engineer has obtained the information necessary to make a rational selection of the filter most appropriate to his particular application. *Sensitivity to parameter variations* in the filter or system hardware (i.e., the truth model) can be obtained in a straightforward manner by repeated covariance analyses, or adjoint methods [7] can be em-

ployed to describe local sensitivity to small variations in many parameters simultaneously.

Although a covariance performance analysis is computationally more efficient than a Monte Carlo study and so should be exploited, there is a definite advantage to employing the Monte Carlo approach in addition. First of all, a Monte Carlo study encompasses a *system simulation* in which the *actual filter algorithm* is embedded. As such, portions of the simulation can be *replaced by actual data or hardware* as it becomes available. Moreover, *sign errors* in the filter algorithm that may not be readily apparent in a covariance analysis due to squaring effects become evident in the Monte Carlo study: if the two performance analyses are based on the same models and disagree significantly in predicted estimation accuracy, sign errors are to be suspected. Finally, effects of *nonlinearities* such as device saturation or neglected terms in attaining linear perturbation equations cannot be evaluated by a covariance approach, and must be investigated through Monte Carlo means.

6.9 SYSTEMATIC DESIGN PROCEDURE

As can be surmised from the preceding sections, Kalman filter development requires an iterative proposing of filter modifications and evaluating of each version's capabilities of meeting performance requirements (rms estimation errors, etc.) and practical constraints (computation time, storage, sequencing, cost, etc.). A *systematic design procedure* would be conducted in the following manner:

(1) Develop a "truth model" (a complete, complex mathematical model that portrays true system behavior very accurately) and validate with laboratory and operational test data (this serves as the basis for evaluating all prospective designs, so one must establish its validity with the utmost confidence); if nonlinear, linearize it for later covariance analyses.

(2) Generate the Kalman filter based upon the "truth" model as a "benchmark" of performance and analyze its capabilities. (If the truth model is linear, the filter-generated covariance provides a valid performance analysis; if nonlinear, Monte Carlo evaluations are necessary.)

(3) Propose simplified, reduced order system models by removing and combining states associated with nondominant effects, deleting weak cross-coupling terms, employing approximations, etc. (this requires substantial physical insights into the problem at hand), and develop the simplified filters based on each model; also consider approximated filter structures.

(4) Conduct a covariance performance analysis of each proposed Kalman filter being driven by measurements derived from the truth model of the real system; as an iteration within this step, "tune" each filter to provide best possible performance from each.

(5) Generate a Monte Carlo analysis of designs from the preceding step that show most promise.

(6) Conduct a performance/computer loading tradeoff analysis and select a design; investigate square root and other forms of implementing the chosen design.

(7) Implement the chosen design on the online computer to be used in the final system.

(8) Perform checkout, final tuning, and operational test of the filter.

Through such a procedure, a logical decision process based on sufficient empirical data is incorporated into the design. As a result, the final implementation should perform as well as predicted in earlier stages of design. (If the implementation is error free and performance is not as predicted, this indicates that the original declaration of the truth model was faulty.) Although the procedure consumes much time and effort, the overall design risk is less than that associated with making design decisions with less supportive analysis, in which inadequate performance is not detected until operational test. It is, in fact, a cost-effective procedure.

6.10 INS AIDED BY NAVIGATION SATELLITES

This section illustrates use of the design procedure of Section 6.9 to generate a practical Kalman filter for a full-scale example of aiding an inertial system with position data provided by navigation satellites [5, 10]. First the truth model will be delineated and the performance of the filter based on the truth model portrayed. Based on insights gained from the physics of the problem and this covariance analysis, reduced order filters will be suggested and analyzed. One parameter of distinct interest for this problem is the filter sample period, and sensitivity to its variation will be demonstrated.

The Global Positioning System (GPS) currently under development consists of 24 navigation satellites, placed eight in each of three different orbits. Each satellite contains a transmitter, receiver, and a quartz crystal oscillator "clock." Periodically a ground tracking network measures and updates the ephemeris and satellite clock phase and frequency in order to maintain synchronization of all satellite clocks. The satellites in turn continually transmit this information together with a range code, formatted such that each satellite signal can be distinguished from the others by the user. By means of a correlation detector, the time (phase) shift between each satellite signal and the user's unsynchronized clock is measured in his receiver, to provide an indication of range from the satellite to the user. If the user has an INS, its position indication and the satellite emphemeris data can be used to compute an INS-indicated range to the satellite. The difference of these two range indications, called "range

divergence," serves as an input to an indirect feedback Kalman filter to yield an integrated GPS-aided inertial navigation system. Four such range divergence signals become available at each filter sample time, i.e., from four separate satellites, in order to correct the three components of position and clock phase difference (a dominant effect which enters the model directly as a position error).

A scalar range divergence measurement associated with satellite j ($j = 1, 2, 3, 4$) can be written in terms of INS-indicated range $R_{j\text{-INS}}$, satellite-indicated range $R_{j\text{-sat}}$, true range $R_{j\text{-true}}$, and associated errors $\delta R_{j\text{-INS}}$ and $\delta R_{j\text{-sat}}$ as:

$$\begin{aligned} z_j(t_i) &= R_{j\text{-INS}}(t_i) - R_{j\text{-sat}}(t_i) \\ &= [R_{j\text{-true}}(t_i) + \delta R_{j\text{-INS}}(t_i)] - [R_{j\text{-true}}(t_i) + \delta R_{j\text{-sat}}(t_i)] \\ &= \delta R_{j\text{-INS}}(t_i) - \delta R_{j\text{-sat}}(t_i) \end{aligned} \tag{6-105}$$

Since INS-indicated range is based upon INS-indicated position of the user vehicle and satellite-provided ephemeris data on its own current position, $\delta R_{j\text{-INS}}(t_i)$ is due to errors in both these sources. However, the satellite orbital parameters are very precisely updated by the ground tracking network and are negligible (any small uncertainties due to this effect can be accounted for by increasing the satellite clock phase error). Let the INS platform instrument an east–north–up coordinate frame so that the INS position errors are $\boldsymbol{\delta r} = [\delta r_e, \delta r_n, \delta r_u]^T$. Then if $\mathbf{i}_j = [i_{je}, i_{jn}, i_{ju}]^T$ is the unit vector direction from the user to satellite j, expressed in east–north–up coordinates, the error $\delta R_{j\text{-INS}}(t_i)$ can be written as

$$\begin{aligned} \delta R_{j\text{-INS}}(t_i) &= -\mathbf{i}_j(t_i)^T \boldsymbol{\delta r}(t_i) \\ &= [-i_{je}(t_i)]\,\delta r_e(t_i) + [-i_{jn}(t_i)]\,\delta r_n(t_i) + [-i_{ju}(t_i)]\,\delta r_u(t_i) \end{aligned} \tag{6-106}$$

The range measurement error model, validated through empirical data, is

$$\delta R_{j\text{-sat}}(t_i) = c\,\delta T_u(t_i) - c\,\delta T_{sj}(t_i) + b_{rj}(t_i) - v_j(t_i) \tag{6-107}$$

where c is the speed of light, $\delta T_u(t_i)$ is user clock phase error, $\delta T_{sj}(t_i)$ is satellite j clock phase error, including error due to ionospheric delay, $b_{rj}(t_i)$ is a small range bias term accounting for tropospheric delay and uncertainty in the speed of light, and $v_j(t_i)$ is white Gaussian measurement noise modeling high frequency corruption of satellite signal and quantization error. Substitution of (6-106) and (6-107) into (6-105) yields the $z_j(t_i)$ measurement relation for the truth model.

The truth model dynamics model is based upon the Pinson error model [40, 41, 52] for the three position errors ($\delta r_e, \delta r_n, \delta r_u$), three velocity errors ($\delta v_e, \delta v_n, \delta v_u$), and three attitude errors (ψ_e, ψ_n, ψ_u) in an INS, specified by the 9-by-9 $\mathbf{F}(\cdot)$ given in Fig. 6.22 and driven by sources of uncertainty.

The time rates of change of the velocity errors are driven by accelerometer errors as given in Example 6.1 of Section 6.7: three accelerometers with six

	δr_e	δr_n	δr_u	δv_e	δv_n	δv_u	ψ_e	ψ_n	ψ_u
$\dot{\delta}r_e$:	0	0	0	1	0	0	0	0	0
$\dot{\delta}r_n$:	0	0	0	0	1	0	0	0	0
$\dot{\delta}r_u$:	0	0	0	0	0	1	0	0	0
$\dot{\delta}v_e$:	$(A_z/R) - \omega_s^2 + \omega^2 - \Omega^2 - (\omega_x + \Omega_x)\rho_x$	$\dot{\rho}_z - (\omega_y + \Omega_y)\rho_x$	$-\dot{\rho}_y - (\omega_z + \Omega_z)\rho_x$	0	$2\omega_z$	$-2\omega_y$	0	$-A_z$	A_y
$\dot{\delta}v_n$:	$-\dot{\rho}_z - (\omega_x + \Omega_x)\rho_y$	$(A_z/R) - \omega_s^2 + \omega^2 - \Omega^2 - (\omega_y + \Omega_y)\rho_y$	$\dot{\rho}_x - (\omega_z + \Omega_z)\rho_y$	$-2\omega_z$	0	$2\omega_x$	A_z	0	$-A_x$
$\dot{\delta}v_u$:	$\dot{\rho}_y - (A_x/R) - (\omega_x + \Omega_x)\rho_z$	$-\dot{\rho}_x - (A_y/R) - (\omega_y + \Omega_y)\rho_z$	$\omega^2 - \Omega^2 + 2\omega_s^2 - (\omega_z + \Omega_z)\rho_z$	$2\omega_y$	$-2\omega_x$	0	$-A_y$	A_x	0
$\dot{\psi}_e$:	$-\omega_z/R$	0	0	0	$1/R$	0	0	ω_z	$-\omega_y$
$\dot{\psi}_n$:	0	$-\omega_z/R$	0	$1/R$	0	0	$-\omega_z$	0	ω_x
$\dot{\psi}_u$:	ω_x/R	ω_y/R	0	0	0	0	ω_y	$-\omega_x$	0

FIG. 6.22 9-by-9 **F** matrix of Pinson INS error model. R = earth radius, $\mathbf{A}$ = acceleration, $\boldsymbol{\rho}$ = angular rate from earth-fixed coordinates to platform coordinates, $\boldsymbol{\Omega}$ = earth angular rate relative to inertial coordinates, $\boldsymbol{\omega}$ = angular rate from inertial coordinates to platform coordinates ($\boldsymbol{\omega} = \boldsymbol{\Omega} + \boldsymbol{\rho}$), $\omega^2 = \boldsymbol{\omega}^T\boldsymbol{\omega}$, $\omega_s^2 = g/R$ = Schuler frequency squared; subscripts x, y, and z correspond to vector components in the east, north, and up directions, respectively.

error states per sensor, yielding 18 augmented states. Also driving these velocity errors are gravity anomalies: three exponentially distance-correlated states to model the difference between the geoid and the ellipse assumed in the navigation equations; these enter because computed gravity is subtracted from the specific force outputs of the accelerometers to yield vehicle acceleration. Exponentially distance-correlated processes are the outputs of first order lags driven by white noise, with the lag time constant T equal to $[d/V]$, where d is the correlation distance and V is the vehicle velocity magnitude. The time rates of change of attitude errors are driven by gyro errors as given in Example 6.2 of Section 6.7: three single-degree-of-freedom gyros with 14 error states per sensor, or 42 additional states. Thus, the INS error model incorporates 72 states.

The mathematical models for both the user clock and the four satellite clocks [5, 10, 21] are structurally identical, but the user clock initial errors are orders of magnitude greater than those of the satellite clocks because of the greater accuracy of the periodically updated satellite clocks. The generic model is portrayed in Fig. 6.23 and is specified by

$$\delta T(t) = c_0 + c_1 t + c_2 t^2 + e_T(t) \tag{6-108}$$

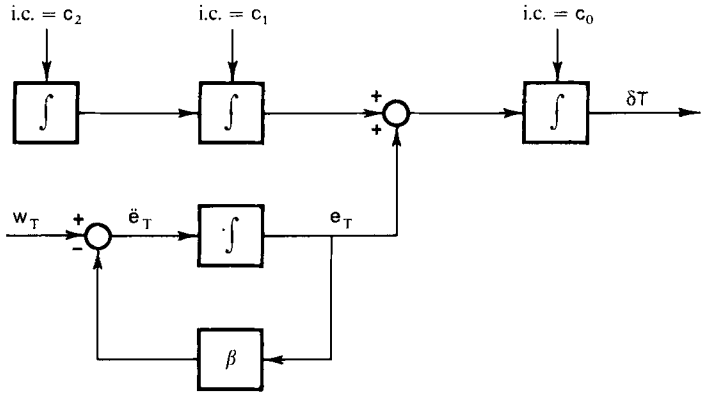


FIG. 6.23 Clock error model.

where c_0, c_1, and c_2 are random variables and $\dot{e}_T(\cdot,\cdot)$ is an exponentially time-correlated noise satisfying

$$\ddot{e}_T(t) = -\beta \dot{e}_T(t) + w_T(t) \tag{6-109}$$

where $w_T(\cdot,\cdot)$ is white Gaussian noise. Thus, each of the five clocks contributes four more states to the system model, for a total of 20 augmented state variables.

Finally, the range biases b_{r_j} in (6-107) are modeled as the outputs of undriven integrators. This adds four states, to yield a total truth model state vector of dimension 96.

Truth model parameter values were set so as to represent a state-of-the-art INS being updated every 30 sec by four GPS satellite range measurements. The user vehicle was simulated as flying for one hour at constant speed and altitude over a great circle path. First the Kalman filter based upon the truth model was studied through a covariance performance analysis, concentrating on the nine error states of position, velocity, and attitude indications. Figure

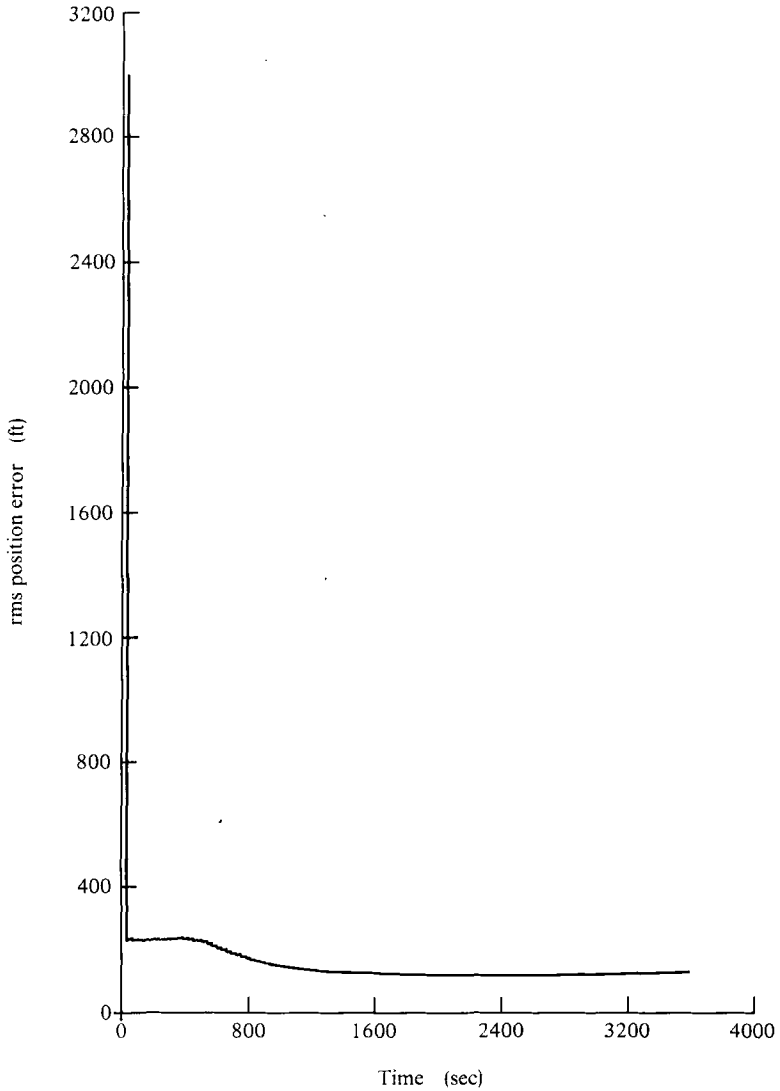


FIG. 6.24a North rms position error of filter based on truth model.

6.24 presents the north rms position and velocity errors achieved. Note that the position error in Fig. 6.24a starts to grow slightly at the end of one hour's flight, indicative of possible divergence problems for longer flights. This problem is the result of using the same four satellites for all updates: in practice an optimal set of four (ten will be in view at a given time on the average) would be chosen, and analyses have shown this procedure to preclude divergence.

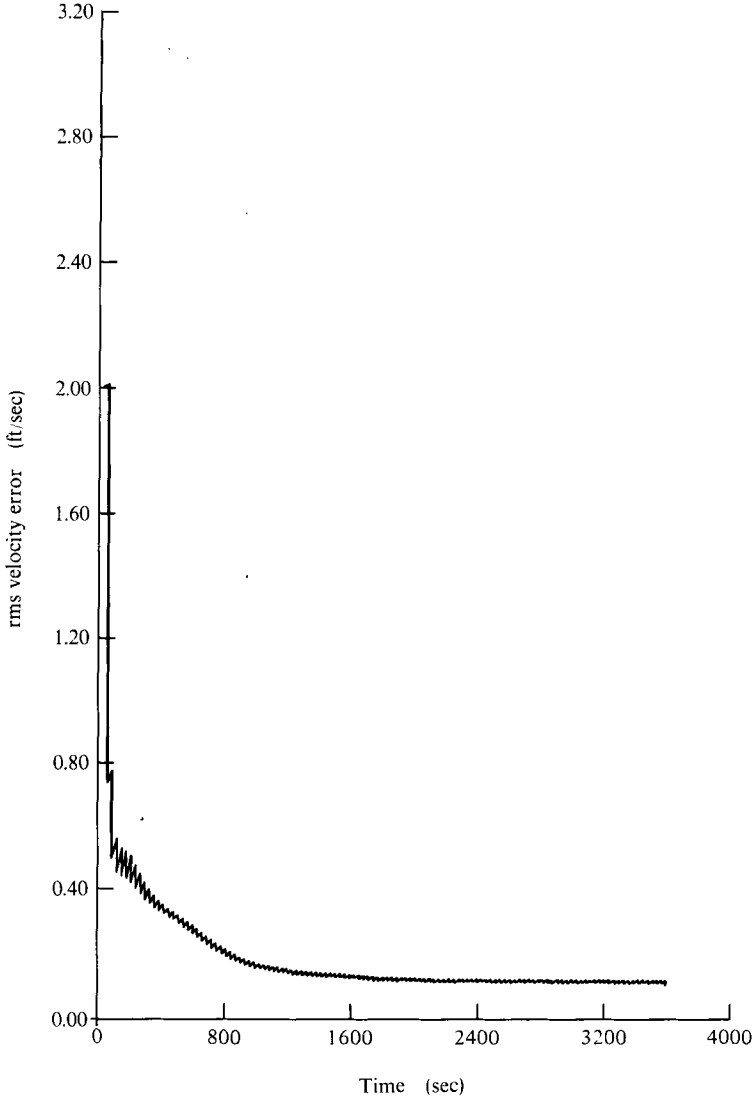


FIG. 6.24b North rms velocity error of filter based on truth model.

Although a 96-state Kalman filter would be totally impractical, this does serve as a baseline against which to compare other reduced-order simplified designs.

Through error budgets and physical insights, the least significant state variables were deleted or combined, shaping filters reduced in complexity (in the cases of INS gyro and accelerometer models, the gravity anomaly models, and clock $\dot{e}_T$ models, to the limiting case of white noise, requiring no states at all), and remaining noise strengths increased to account for the simplifications. Eventually a 15-state design was attained, being composed of the nine basic INS states ($\delta r_e, \delta r_n, \delta r_u, \delta v_e, \delta v_n, \delta v_u, \psi_e, \psi_n, \psi_u$), the c_0 clock phase/range error state in (6-108) for each of the four satellites, and both the c_0 and c_1 frequency offset state for the user clock. Moreover, the weak coupling terms in the **F** matrix were removed; Schuler frequency terms (ω_s^2 and $2\omega_s^2$) were retained so that the model applicability would not be restricted to short periods of time, on the order of 30 min. Thus, the upper 9-by-9 partition of the filter **F** matrix replaces that given in Fig. 6.22 with the matrix depicted in Fig. 6.25.

	δr_e	δr_n	δr_u	δv_e	δv_n	δv_u	ψ_e	ψ_n	ψ_u
$\dot{\delta} r_e$:				1					
$\dot{\delta} r_n$:					1				
$\dot{\delta} r_u$:						1			
$\dot{\delta} v_e$:	$-\omega_s^2$							$-A_z$	A_y
$\dot{\delta} v_n$:		$-\omega_s^2$					A_z		$-A_x$
$\dot{\delta} v_u$:			$2\omega_s^2$				$-A_y$	A_x	
$\dot{\psi}_e$:								ω_z	$-\omega_y$
$\dot{\psi}_n$:							$-\omega_z$		ω_x
$\dot{\psi}_u$:							ω_y	$-\omega_x$	

FIG. 6.25 Modified 9-by-9 **F** matrix of Pinson error model (nonzero elements only).

The performance of this 15-state filter, after being properly tuned, is displayed in Fig. 6.26. Despite the vast reduction in size and complexity of the filter, performance degradation from that of Fig. 6.24 is totally insignificant in position estimation accuracy, while being more apparent in the velocity estimation. However, this design is performing well within specification and provides a good tradeoff of complexity and performance. Range-rate measurements could be used additionally to improve this filter's ability to maintain accurate velocity estimates, if desired.

Some insights suggested a further simplification might be achieved by discarding the four satellite clock phase/range error states and the user clock

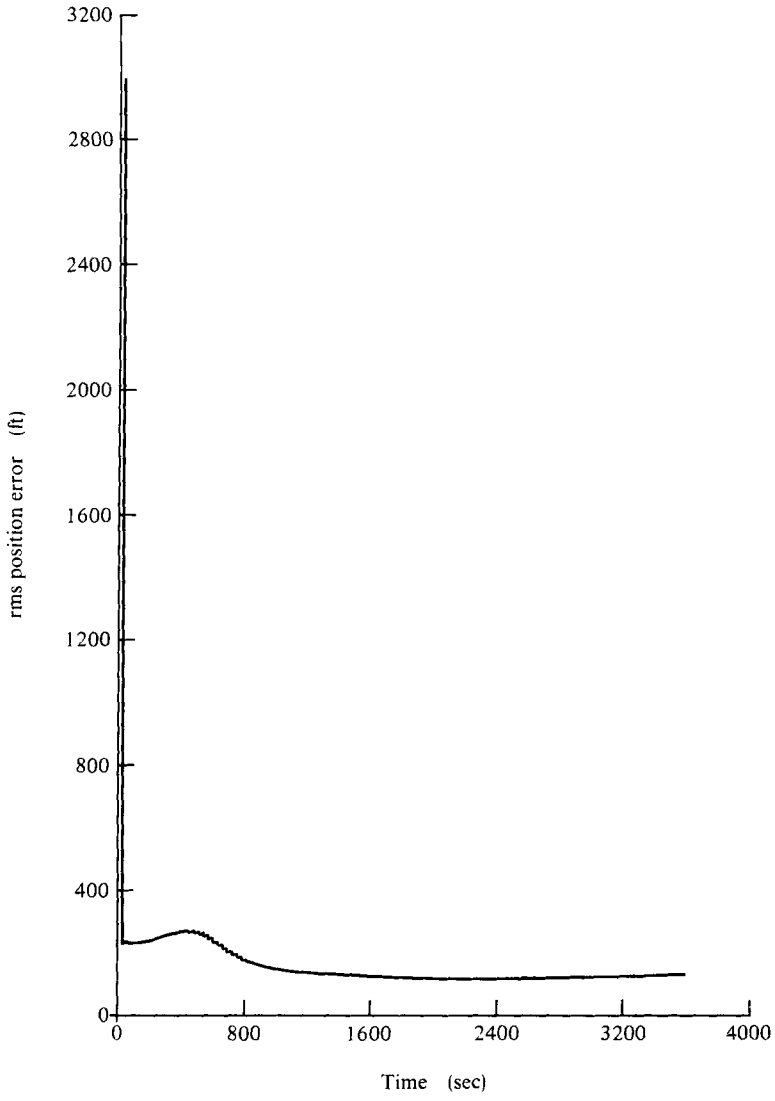


FIG. 6.26a North rms position error of 15-state simplified filter.

frequency offset error, yielding a ten-state filter. However, performance analysis of this design (after retuning) revealed an unacceptable degradation in estimation capability, as depicted in Fig. 6.27.

Returning attention to the acceptable 15-state design, it was desired to observe the sensitivity of performance to slower update rates. Figure 6.28

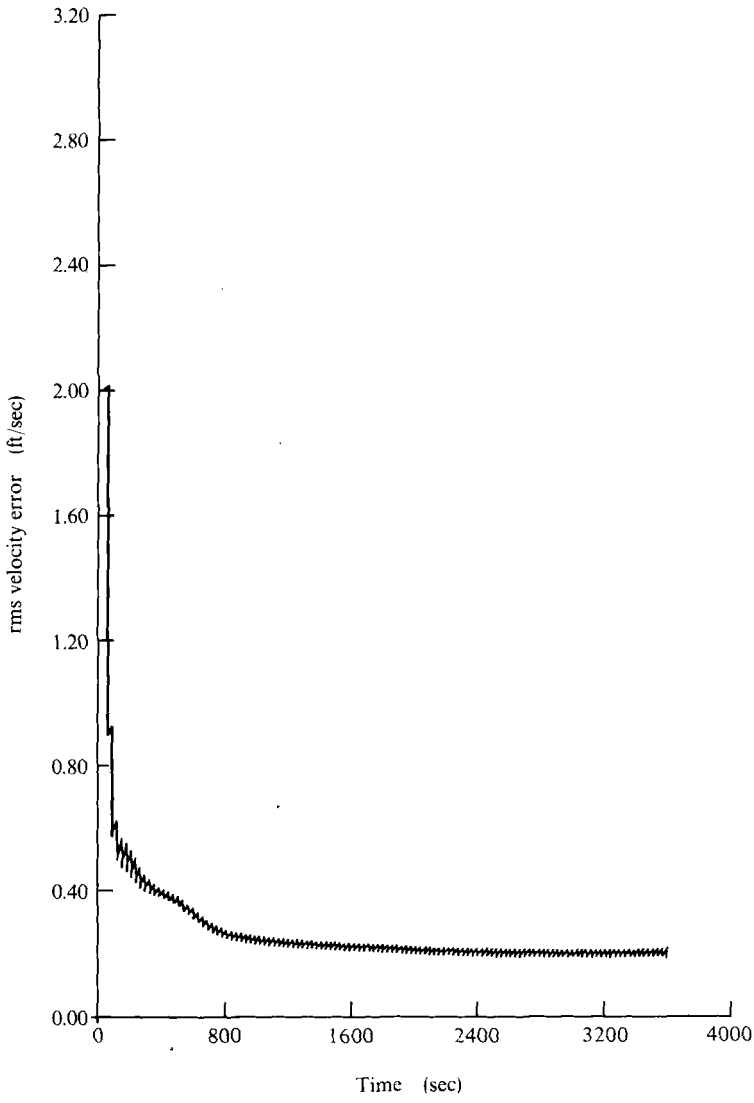


FIG. 6.26b North rms velocity error of 15-state simplified filter.

indicates that tripling the sample period from 30 to 90 sec still yields acceptable estimation precision.

Thus, covariance performance analysis has been shown to be a versatile tool for initial stages of Kalman filter design and tradeoff studies. Nevertheless, this is but a part of a total systematic design and implementation of an operational filter algorithm.

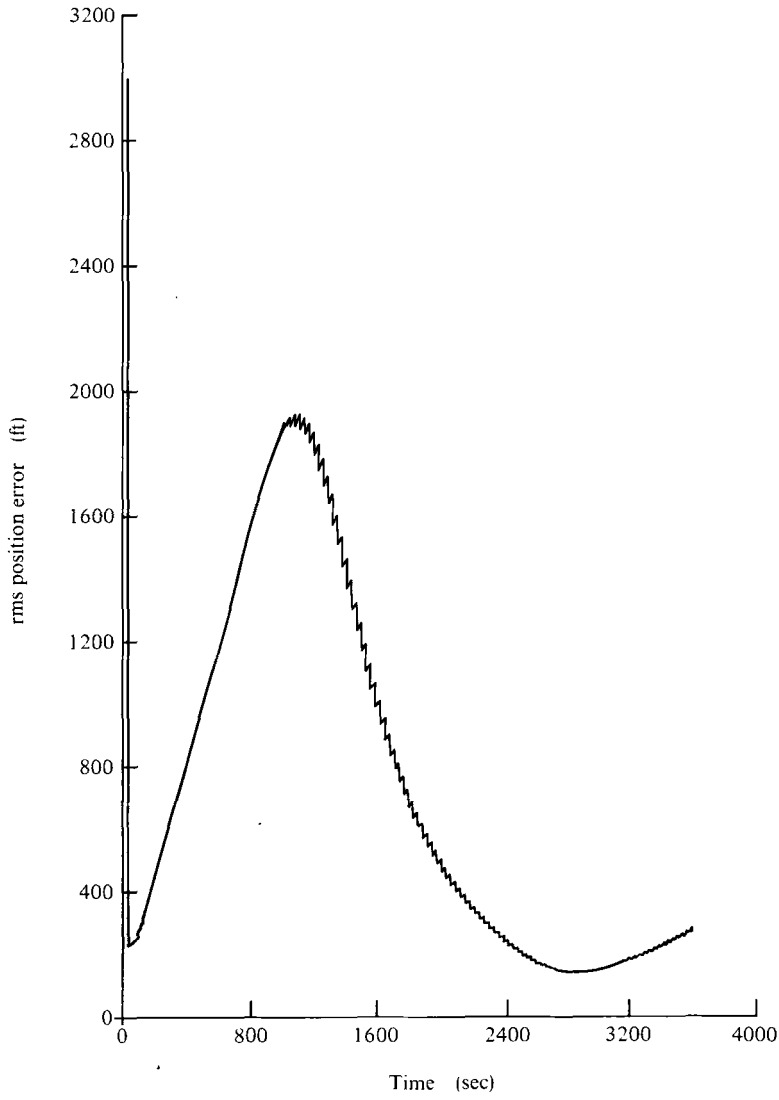


FIG. 6.27a North rms position error of 10-state simplified filter.

6.11 PRACTICAL ASPECTS OF IMPLEMENTATION[1]

Throughout the development of a filter algorithm, one must be aware of the constraints imposed by the computer in which the software will reside:

[1] See References [28, 29, 31, 35, 46–48].

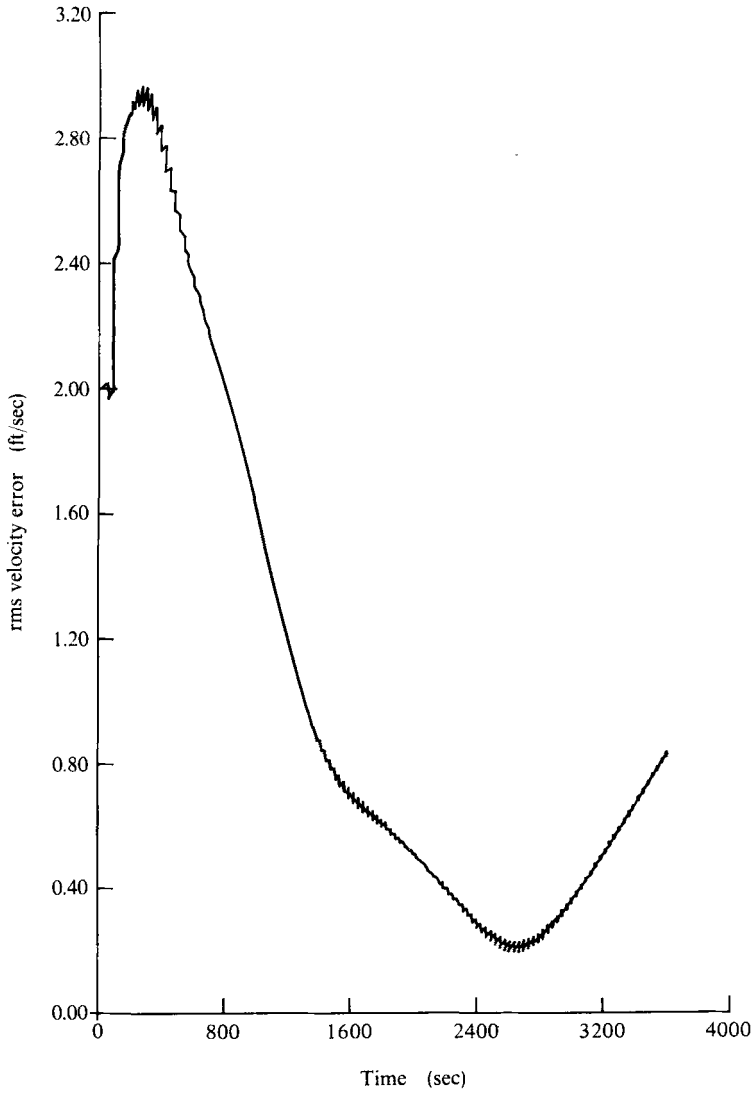


FIG. 6.27b North rms velocity error of 10-state simplified filter.

- cycle time (and through it, the time to perform a load, store, add, multiply, divide, etc.);
- memory size and access;
- wordlength (Is rounding to nearest number or truncating least significant bits used, i.e., are wordlength errors symmetric or biased? Are square root forms or other means of enhancing numerical precision and stability warranted?);

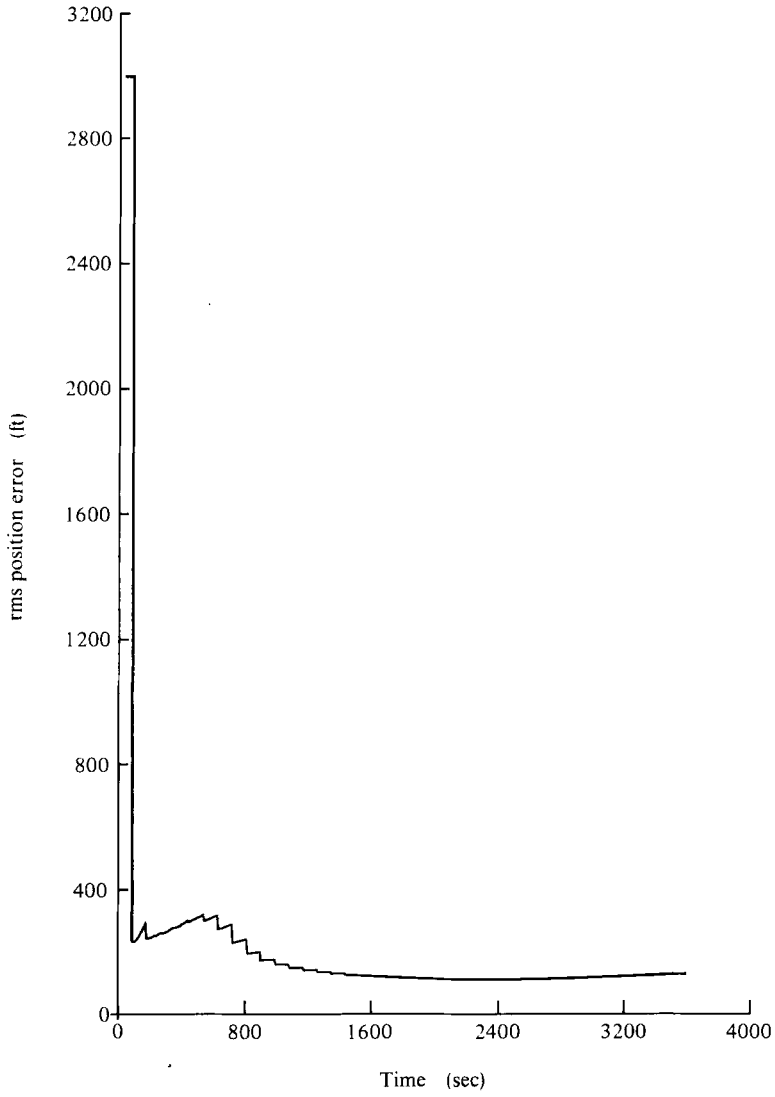


FIG. 6.28a North rms position error of 15-state filter with 90-sec update period.

- readout A/D quantizations;
- instruction set;
- calculation capability;
- arithmetic type (floating or fixed point?).

In many applications, the filter will be one of many algorithms processed by a central digital computer. Consequently, the designer must plan real time

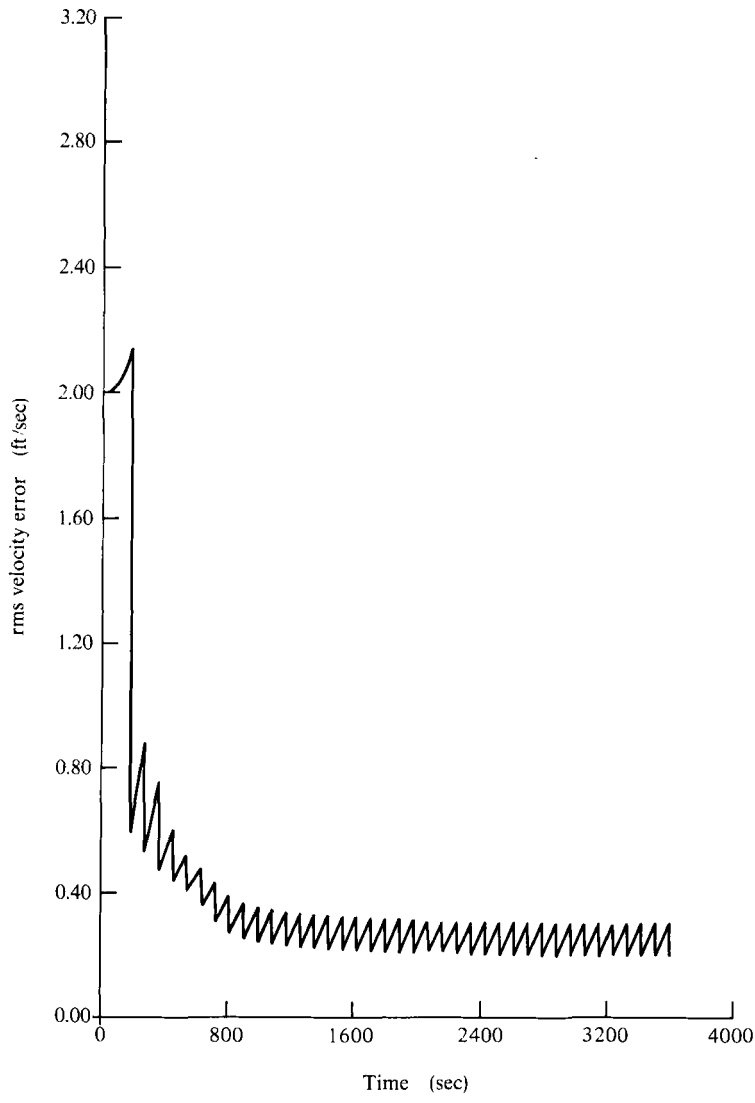


FIG. 6.28b North rms velocity error of 15-state filter with 90-sec update period.

allocation and sequencing to be compatible with estimation accuracy specifications, sensor interfacing, and other software processing requirements. Often the iteration period of the filter is long compared to that of other jobs, such as a 30-sec period for an INS-aiding filter versus a 0.02-sec period for digital flight control inner loops. Therefore, the filter computations are typically performed "in the background" on a time-shared computer, admitting priority interrupts for more time-critical jobs.

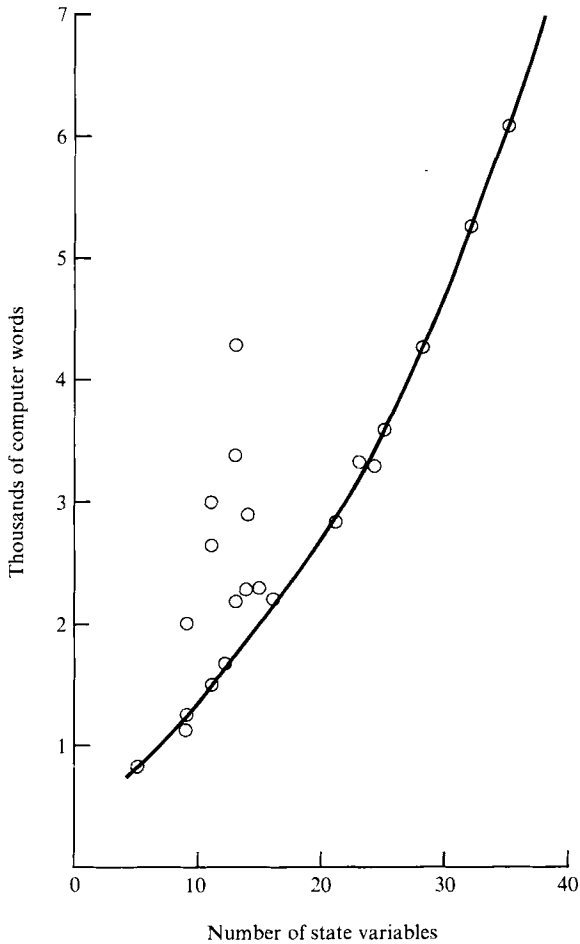


FIG. 6.29 Kalman filter computer memory requirements.

In the past, storage requirements have been a critical factor in the practicality of a filter design. Figure 6.29 plots the number of computer words required by actual navigation Kalman filters as a function of the number of state variables being estimated. These numbers account for both program instruction and permanent storage locations dedicated to the filter, but not erasable storage common to other programs. For instance, to implement a 16-state filter will require about 2200 words, about equally divided between instructions and permanent storage. In this figure, a curve has been drawn to indicate an approximate graph of this data for very efficient filters. Note that it does not exhibit a proportionality between required storage and the square of the state dimension as might be predicted theoretically (for $n \gg m$, the number of words required is predicted to be dominated by a term equal to $2.5n^2$). This is caused

by the fact that, as the state dimension of a filter grows, the defining matrices typically become considerably more sparse. Nevertheless, because computer memories are becoming dramatically less expensive, future filter designs may not focus so much attention on storage aspects.

Similarly, the number of computations can be calculated as a function of state variables used. For $n \gg m$, approximately $4n^3$ multiplications, and the same number of less time-consuming additions, would be needed for a single recursion of the filter. Again, this is not totally realized in practice due to increasing matrix sparsity with higher state dimension.

The practical constraints imposed by the computer do dictate a design philosophy of generating as simple a filter as possible that will meet performance specifications. Means of reducing complexity that are generally exploited are

- reducing state dimension while maintaining dominant facets of system behavior;
- neglecting terms, decoupling, and other model simplifications;
- canonical state space;
- exploiting symmetry (computing only lower triangular form of symmetric matrices);
- precomputations;
- approximations, as curve-fitted or constant gains;
- long sample periods, possibly using measurement prefiltering to make this feasible;
- iterative scalar measurement updating;
- removal of double precision requirements by use of a "square root filter" implementation (discussed in Chapter 7);
- approximation of stored constants as powers of two so that multiplications are replaced by simple shift operations; and
- efficient, rather than straightforward, programming of the algorithm such as replacing

$$\mathbf{P}^+ = \mathbf{P}^- - \mathbf{P}^-\mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{P}^-\mathbf{H}^{\mathrm{T}} + \mathbf{R})^{-1}\mathbf{H}\mathbf{P}$$

by

$$\mathbf{A} = \mathbf{P}^-\mathbf{H}^{\mathrm{T}}, \qquad \mathbf{B} = \mathbf{H}\mathbf{A} + \mathbf{R}, \qquad \mathbf{C} = \mathbf{B}^{-1} \qquad \mathbf{P}^+ = \mathbf{P}^- - \mathbf{A}\mathbf{C}\mathbf{A}^{\mathrm{T}}$$

Another practical consideration is the means of performing the estimate time propagations numerically. We will concentrate on the covariance time propagation. One method is to integrate the equation

$$\dot{\mathbf{P}}(t/t_{i-1}) = \mathbf{F}(t)\mathbf{P}(t/t_{i-1}) + \mathbf{P}(t/t_{i-1})\mathbf{F}^{\mathrm{T}}(t) + \mathbf{G}(t)\mathbf{Q}(t)\mathbf{G}^{\mathrm{T}}(t) \qquad (6\text{-}110)$$

forward to time t_i from $\mathbf{P}(t_{i-1}/t_{i-1}) = \mathbf{P}(t_{i-1}^+)$. If we let Δt represent $[t_i - t_{i-1}]$, then simple Euler integration yields

$$\mathbf{P}(t_i^-) \stackrel{c}{=} \mathbf{P}(t_{i-1}^+) + [\dot{\mathbf{P}}(t_{i-1}/t_{i-1})]\,\Delta t \qquad (6\text{-}111)$$

where $\stackrel{c}{=}$ means computationally equivalent to. Integration accuracy is improved if the derivative in this relation were evaluated at the midpoint of the interval instead of the beginning:

$$\mathbf{P}(t_i^-) \stackrel{c}{=} \mathbf{P}(t_{i-1}^+) + [\dot{\mathbf{P}}(\tfrac{1}{2}t_{i-1} + \tfrac{1}{2}t_i/t_{i-1})]\,\Delta t \tag{6-112a}$$

$$\cong \mathbf{P}(t_{i-1}^+) + \tfrac{1}{2}[\dot{\mathbf{P}}(t_{i-1}/t_{i-1}) + \dot{\mathbf{P}}(t_i/t_{i-1})]\,\Delta t \tag{6-112b}$$

If neither of these sets of relations yield adequate performance, either the sample period can be subdivided and first order integration applied repeatedly to each step along the subintervals, or a higher order integration technique such as fourth order Runge–Kutta can be employed.

Another means of accomplishing the time propagation is through

$$\mathbf{P}(t_i^-) = \mathbf{\Phi}(t_i, t_{i-1})\mathbf{P}(t_{i-1}^+)\mathbf{\Phi}^{\mathrm{T}}(t_i, t_{i-1}) + \mathbf{Q}_{\mathrm{d}}(t_{i-1}) \tag{6-113a}$$

$$\mathbf{Q}_{\mathrm{d}}(t_{i-1}) = \int_{t_{i-1}}^{t_i} \mathbf{\Phi}(t_i, \tau)\mathbf{G}(\tau)\mathbf{Q}(\tau)\mathbf{G}^{\mathrm{T}}(\tau)\mathbf{\Phi}^{\mathrm{T}}(t_i, \tau)\,d\tau \tag{6-113b}$$

Chapter 4 indicated a first order approximation for $\mathbf{\Phi}(t_i, t_{i-1})$ and $\mathbf{Q}_{\mathrm{d}}(t_{i-1})$ as (see 4-132):

$$\mathbf{\Phi}(t_i, t_{i-1}) \stackrel{c}{=} \mathbf{I} + \mathbf{F}(t_{i-1})\,\Delta t \tag{6-114a}$$

$$\mathbf{Q}_{\mathrm{d}}(t_{i-1}) \stackrel{c}{=} \mathbf{G}(t_{i-1})\mathbf{Q}(t_{i-1})\mathbf{G}^{\mathrm{T}}(t_{i-1})\,\Delta t \tag{6-114b}$$

These numerical approximations are similarly improved if the matrices are evaluated at the midpoint of the interval instead of the beginning, or approximated by the average of their values at both ends of the interval. Moreover, subdividing the sample period and applying such relations repeatedly also will improve precision.

If the system model is time invariant, or has time variations that are slow enough to represent $\mathbf{F}$ adequately as constant over a single filter sample period, then the state transition matrix can be approximated by a truncated matrix exponential:

$$\mathbf{\Phi}(t_i, t_{i-1}) \stackrel{c}{=} \sum_{k=0}^{N} \frac{1}{k!}[\mathbf{F}(t_{i-1})]^k \Delta t^k \tag{6-115}$$

where N might be chosen as two. Again, subpartitioning into J portions can be used to advantage to obtain

$$\mathbf{\Phi}(t_i, t_{i-1}) = \prod_{j=0}^{J-1} \mathbf{\Phi}(t_{i-1} + (j+1)\Delta t/J,\, t_{i-1} + j\Delta t/J) \tag{6-116}$$

where each element in the product is calculated as in (6-115). A first order approximation to such a result would be

$$\mathbf{\Phi}(t_i, t_{i-1}) \stackrel{c}{=} \mathbf{I} + \sum_{j=0}^{J-1} [\mathbf{F}(t_{i-1} + j\Delta t/J)][\Delta t/J] \tag{6-117}$$

Other methods of evaluation were discussed in Problems 15–17 of Chapter 2.

If the first order approximation to $\mathbf{Q}_d(t_{i-1})$ given in (6-114b) is inadequate, other methods can be applied. First of all, the $\dot{\bar{\mathbf{Q}}}(t, t_{i-1})$ equation to be integrated to yield $\mathbf{Q}_d(t_{i-1})$, (4-129c), is identical in form to (6-110), with the only difference being the initial condition. Therefore, the methods applicable to (6-110) are also usable here. Second, trapezoidal integration of (6-113b) yields

$$\mathbf{Q}_d(t_{i-1}) \overset{c}{=} \tfrac{1}{2}[\boldsymbol{\Phi}(t_i, t_{i-1})\mathbf{G}(t_{i-1})\mathbf{Q}(t_{i-1})\mathbf{G}^T(t_{i-1})\boldsymbol{\Phi}^T(t_i, t_{i-1}) + \mathbf{G}(t_i)\mathbf{Q}(t_i)\mathbf{G}^T(t_i)]\,\Delta t \tag{6-118}$$

This form is especially attractive since it replaces having to know $\boldsymbol{\Phi}(t_i, \tau)$ for all τ with evaluating only $\boldsymbol{\Phi}(t_i, t_{i-1})$ as discussed previously; it is in fact the method used in the F-111 navigation filter [32]. If $\mathbf{GQG}^T$ is constant, then (6-114a) can be substituted into this expression and terms rearranged to yield another approximation as

$$\mathbf{Q}_d(t_{i-1}) \overset{c}{=} \tfrac{1}{2}[\boldsymbol{\Phi}(t_i, t_{i-1})\mathbf{GQG}^T + \mathbf{GQG}^T\boldsymbol{\Phi}^T(t_i, t_{i-1})]\,\Delta t \tag{6-119}$$

which was used in a LORAN-inertial navigation filter for the Army [26]. Higher order integration techniques are possible, but are seldom necessary.

Residual monitoring as described in Section 5.4 is typically exploited in operational filters, at least for reasonableness checking and discarding of "bad" data points for which the residual magnitude exceeds some specified multiple of the computed standard deviation. Current designs embody more sophistication, using the residual monitoring for sensor failure declaration or adaptive purposes.

If a human operator is to provide measurements to the filter, the man–machine interface is a unique problem. Not only is update timing an issue, but so is the operator's confidence in the filter's perceived performance. For example, in the operation of the F-111 navigation filter, the operator can key in landmark position data. The accuracy ascribed to such data through associated diagonal elements in the $\mathbf{R}$ matrix was such that the Kalman gain was low and the navigation system updated its calculated position by only a small fraction of the residual. This dissatisfied navigators to the point where they attempted to "compensate" with fictitious position data or to override the filter function if they believed their own data to be precise [32]. To circumvent this lack of confidence, designers of a more recent navigation filter have incorporated the ability to "tell" the filter that the landmark information keyed in is thought to be poor, average, or very good: the navigator inputs both data and one of three predetermined R values to set the filter gain accordingly [1].

6.12 SUMMARY

This chapter has developed the means of exploiting the Kalman filter derived in the previous chapter, converting it from a result of mathematical optimization theory to a useful and flexible engineering tool. As has been

emphasized throughout the discussion, there are many possible filter designs for any given application. Physical insight and engineering judgment are essential to proposing viable designs, and then a thorough and accurate performance/loading tradeoff analysis must be conducted to generate a superior filter algorithm. A systematic approach to this iterative design procedure has been presented, one which minimizes the risk of erroneous decisions and major design modifications late in the filter development process, and one which maximizes the probability of an efficient, practical design that meets or surpasses all performance specifications. Because of constraints imposed by the computer and overall system of which the filter is an integral part, the design philosophy to be adopted is not to achieve the best possible performance at any cost, but to develop the simplest design that ensures attainment of performance goals.

Examples have concentrated upon the application of Kalman filtering to aiding inertial navigation systems. This has been an area benefited significantly by the Kalman filter, for two fundamental reasons. First of all, the error characteristics of an INS and other navigation aids complement each other well: the INS provides accurate high frequency data while the other sources supply good low frequency information. Thus, there is much benefit to be derived from combining such data through an optimal estimation algorithm. Perhaps more importantly, though, very adequate error models in the form of linear system representations driven by white Gaussian noise have been developed and validated for this application area, which is precisely the form required by the Kalman filter. As adequate models are developed in other areas, optimal estimation theory will become more widely exploited. Progressively the fundamental application of filtering theory will mature to a point where the systematic design procedure presented in this chapter can glean out the fullest potential of "optimal" filtering in the form of an efficient operational data processing algorithm.

REFERENCES

1. Bergeson, J., and Blahut, K., "B-1 Navigation System Mechanization," Tech. Rep. D229-10336-1, The Boeing Co., Seattle, Washington, May 1974.
2. Britting, K. R., *Inertial Navigation System Analysis.* Wiley, New York, 1971.
3. Brock, L. D., and Schmidt, G. T., "General Questions on Kalman Filtering in Navigation Systems," *in Theory and Applications of Kalman Filtering* (C. T. Leondes, ed.), AGARDograph No. 139. NATO Advisory Group for Aerospace Research and Development, London, February 1970.
4. Broxmeyer, C., *Inertial Navigation Systems.* McGraw-Hill, New York, 1964.
5. Butler, R. R., and Rhue, G. T., "Kalman Filter Design for an Inertial Navigation System Aided by Non-Synchronous Navigation Satellite Constellations," Masters thesis, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, March 1974.
6. Chalk, C. R. *et al.*, "Background Information and User Guide for MIL F-8785B(ASG) Entitled Military Specification—Flying Qualities of Piloted Airplanes," AFFDL TR-69-72, Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio, August 1969.

7. Clark, R. R., "Performance Sensitivity Analysis of a Kalman Filter Using Adjoint Functions," NAFI TR-1767, Naval Avionics Facility, Indianapolis, Indiana, February 1972.
8. D'Appolito, J. A., "The Evaluation of Kalman Filter Designs for Multisensor Integrated Navigation Systems," AFAL-TR-70-271, The Analytic Sciences Corp., Reading, Massachusetts, January 1971.
9. D'Appolito, J. A., and Hutchinson, C. E., "Low Sensitivity Filters for State Estimation in the Presence of Large Parameter Uncertainties," *IEEE Trans. Automatic Control* **AC-14** (3), 310–312 (1969).
10. D'Appolito, J. A., and Roy, K. J., "Satellite/Inertial Navigation System Kalman Filter Design Study," Tech. Rep. AFAL-TR-71-178, Air Force Avionics Laboratory, Wright-Patterson AFB, Ohio, 1971.
11. Fagin, S. L., "Recursive Linear Regression Theory, Optimal Filter Theory and Error Analysis of Optimal Systems," *IEEE Internat. Convent. Record* 216–240 (1964).
12. Fitzgerald, R. J., "Divergence of the Kalman Filter," *IEEE Trans. Automatic Control* **AC-16** (6), 736–747 (1971).
13. Friedland, B., "On the Effect of Incorrect Gain in the Kalman Filter," *IEEE Trans. Automatic Control* **AC-12** (5), 610 (1967).
14. Friedland, B., "Treatment of Bias in Recursive Filtering," *IEEE Trans. Automatic Control* **AC-14** (4), 359–367 (1969).
15. Griffin, R. E., and Sage, A. P., "Large and Small Scale Sensitivity Analysis of Optimum Estimation Algorithms," *IEEE Trans. Automatic Control* **AC-13** (4), 320–329 (1968).
16. Griffin, R. E., and Sage, A. P., "Sensitivity Analysis of Discrete Filtering and Smoothing Algorithms," *AIAA J.* **7** (10), 1890–1897 (1969).
17. Hamilton, E. L., Chitwood, G., and Reeves, R. M., "An Efficient Covariance Analysis Computer Program Implementation," *Proc. Nat. Aerospace and Electronic. Conf., Dayton, Ohio* (May 1976).
18. Hamilton, E. L., Chitwood, G., and Reeves, R. M., "The General Covariance Analysis Program (GCAP): An Efficient Implementation of the Covariance Analysis Equations," Air Force Avionics Laboratory, Wright-Patterson AFB, Ohio, June 1976.
19. Hammett, J. E., "Evaluation of a Proposed INS Kalman Filter in a Dynamic Flight Environment," Masters thesis, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, December 1974.
20. Heffes, H., "The Effect of Erroneous Models on the Kalman Filter Response," *IEEE Trans. Automatic Control* **AC-11** (3), 541–543 (1966).
21. Heller, W. G., "Models for Aided Inertial Navigation System Sensor Errors," TASC TR-312-3, The Analytic Sciences Corp., Reading, Massachusetts, February 1975.
22. Hollister, W. M., and Bansal, D. D., "Guidance and Control for V/STOL Aircraft—Final Report," DOT-TSC-5, Measurement Systems Laboratory RE-77, M.I.T., Cambridge, Massachusetts, November 1970.
23. Hollister, W. M., and Brayard, M. C., "Optimum Mixing of Inertial Navigator and Position Fix Data," Rep. RE-62, M.I.T. Measurement Systems Laboratory, Cambridge, Massachusetts., August 1969; also AIAA Paper No. 70–35 at the Eighth Aerospace Sciences Meeting, New York, January 1970.
24. Huddle, J. R., and Wismer, D. A., "Degradation of Linear Filter Performance Due to Modeling Error," *IEEE Trans. Automatic Control* **AC-13** (4), 421–423 (1968).
25. Kayton, M., and Fried, W. R., *Avionics Navigation Systems*. Wiley, New York, 1969.
26. Knight, J., Light, W., and Fisher, M., "Selected Approaches to Measurement Processing and Implementation in Kalman Filters," Tech. Rep. ECOM-3510, U.S. Army Electronics Command, Fort Monmouth, New Jersey, November 1971.
27. Kwakernaak, H., "Sensitivity Analysis of Discrete Kalman Filters," *Internat. J. Control* **12**, 657–669, (1970).

28. Leondes, C. T. (ed.), *Theory and Applications of Kalman Filtering*, AGARDograph No. 139. NATO Advisory Group for Aerospace Research and Development, London, February 1970.
29. Maybeck, P. S., "The Kalman Filter—An Introduction for Potential Users," TM-72-3, Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio, June 1972.
30. Maybeck, P. S., "Filter Design for a TACAN-Aided Baro-Inertial System with ILS Smoothing Capability," AFFDL-TM-74-52, Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio, January 1974.
31. Maybeck, P. S., "Applied Optimal Estimation: Kalman Filter Design and Implementation," notes for a short course presented by the Air Force Institute of Technology, Wright-Patterson AFB, Ohio, semiannually since December 1974.
32. Maybeck, P. S., "Review of F-111 Kalman Filter Meeting at Sacramento ALC, 26 July 1976," correspondence, July 1976; also "USAF Tech. Order IF-111D-2-5-1-1" (anon.); "Customer Training Document on F-111 Avionics," (General Dynamics, Fort Worth); "System Program Description Document for the F-111D General Navigation and Weapon Delivery Computers," (FZE-12-8078A, USAF, June 1976); "Comments and Questions on the F-111D Kalman Filter" (Maybeck, P. S., June 1976); "Discussion of Capt. Maybeck's Comments and Questions on the F-111D Kalman Filter" (DeVries, T. W., Autonetics Division of Rockwell International, July 1976).
33. Maybeck, P. S., "Analysis of a Kalman Filter for a Strapdown Inertial/Radiometric Area Correlator Guidance System," *Proc. IEEE Nat. Aerospace and Electron Conf., Dayton, Ohio* (May 1977).
34. Maybeck, P. S., "Performance Analysis of a Particularly Simple Kalman Filter," *AIAA J. Guidance and Control* **1** (6), 391–396 (1978).
35. Mendel, J. M., "Computational Requirements for a Discrete Kalman Filter," *IEEE Trans. Automatic Control* **AC-16** (6), 748–758, (1971).
36. Nash, R. A., D'Appolito, J. A., and Roy, K. J., "Error Analysis of Hybrid Inertial Navigation Systems," *Proc. AIAA Guidance and Control Conf.* Paper No. 72-848, Stanford, California, (August 1972).
37. Nash, R. A., and Tuteur, F. B., "The Effect of Uncertainties in the Noise Covariance Matrices on the Maximum Likelihood Estimate of a Vector," *IEEE Trans. Automatic Control* **AC-13** (1), 86–88 (1968).
38. Neal, S. R., "Linear Estimation in the Presence of Errors in Assumed Plant Dynamics," *IEEE Trans. Automatic Control* **AC-12** (5), 592–594 (1967).
39. Nishimura, T., "On the A Priori Information in Sequential Estimation Problems," *IEEE Trans. Automatic Control* **AC-11** (2), 197–204 (1966); correction to and extension of, **AC-12** (1), 123 (1967).
40. Pinson, J. C., "Inertial Guidance for Cruise Vehicles," *in Guidance and Control of Aerospace Vehicles* (C. T. Leondes, ed.). McGraw-Hill, New York, 1963.
41. Pitman, G. R. (ed.), *Inertial Guidance*. Wiley, New York, 1962.
42. Potter, J. E., and Vander Velde, W. E., "Optimum Mixing of Gyroscope and Star Tracking Data," Rep. RE-26, MIT Experimental Astronomy Laboratory, Cambridge, Massachusetts, 1968; also *AIAA J. Spacecraft and Rockets* **5** (5), 536–540 (1968).
43. Price, C. F., "An Analysis of the Divergence Problem in the Kalman Filter," *IEEE Trans. Automatic Control* **AC-13** (6), 699–702 (1968).
44. Schlee, F. H., Standish, C. J., and Toda, N. F., "Divergence in the Kalman Filter," *AIAA J.* **5** (6), 1114–1120 (1967).
45. Schmidt, G. T., "Aided and Optimal Systems," unpublished paper, M.I.T., Cambridge, Massachusetts, September 1968; also "M.I.T. Lectures on Kalman Filtering," C. S. Draper Rep. P-061, Cambridge, Massachusetts, 1974.
46. Schmidt, G. T., "Linear and Nonlinear Filtering Techniques," *in Control and Dynamic Systems* (C. T. Leondes, ed.), Vol. 12. Academic Press, New York, 1976.

47. Schmidt, G. T. (ed.), *Practical Aspects of Kalman Filtering Implementation*, AGARD-LS-82, NATO Advisory Group for Aerospace Research and Development, London, May 1976.
48. Schmidt, S. F., "Computational Techniques in Kalman Filtering," *in Theory and Applications of Kalman Filtering*, AGARDograph 139, NATO Advisory Group for Aerospace Research and Development, London, February 1970.
49. Simon, K. W., and Stubberud, A. R., "Reduced Order Kalman Filter," *Internat. J. Control* **10**, 501–509 (1969).
50. Sims, F. L., and Lainiotis, D. G., "Sensitivity Analysis of Discrete Kalman Filters," *Conf. Record Asilomar Conf. on Circuits and Systems, 2nd*, pp. 147–152 (1968).
51. Sorenson, H. W., "On the Error Behavior in Linear Minimum Variance Estimation Problems," *IEEE Trans. Automatic Control* **AC-12** (5), 557–562 (1967).
52. Widnall, W. S., and Grundy, P. A., "Inertial Navigation System Error Models," Intermetrics TR-03-73, Intermetrics, Inc., Cambridge, Massachusetts, May 1973.

PROBLEMS

6.1 *Combining Inertial and ILS Information* A substantial amount of work is now being conducted in the area of combining inertial and instrument landing system (ILS) signals by means of a Kalman filter. An ILS is composed of two beams, one oscillating in the vertical plane (which provides the glide-slope measurement) and another in the horizontal plane (to provide the localizer measurement). These oscillate in restricted arcs about the nominal approach path to the runway.

It is proposed that, once the localizer beam and glide-slope beam are captured, the ILS and inertial data be combined optimally. In effect, the inertial system is used to smooth the ILS data before presenting the approach information to the pilot.

It is assumed that the aircraft will have achieved a location and velocity "close" to a nominal trajectory to the runway. The inertial system (possibly aided during the preceding cruise segment of flight) will provide the necessary initial conditions. Furthermore, initial offsets in the stable member alignment from vertical, accumulated velocity errors, and the effects of gyro drift during the ILS/INS mode will have negligible effects and can be ignored.

For category II (restricted ceiling and visual range) landings, ILS beam errors and onboard receiver errors are expected to cause deviations from the ideal trajectory according to the following table of path angle *peak* deviations (outer and middle markers are located in line with the runway along the approach trajectory and threshold refers to the end of the runway).

Location	Localizer deviation	Glide-slope deviation
Beyond outer marker	$\pm 0.4^\circ$	$\pm 0.1^\circ$
Outer marker to middle marker	$\pm 0.4^\circ$ decreasing to $\pm 0.07^\circ$	$\pm 0.1^\circ$ decreasing to $\pm 0.06^\circ$
Middle marker to threshold	$\pm 0.07^\circ$	$\pm 0.06^\circ$

(a) How would you incorporate this data (a typical specification) into the format required in the Kalman filter formulation?

The localizer measurement at time instant t_i is modeled by

$$L_{ILS}(t_i) = L_{true}(t_i) + \delta L_{ILS}(t_i) - v_1(t_i) \tag{1}$$

and the glide-slope measurement is

$$S_{ILS}(t_i) = S_{true}(t_i) + \delta S_{ILS}(t_i) - v_2(t_i) \tag{2}$$

Here v_1 and v_2 are white Gaussian noises with zero mean and variances $\sigma_{v_1}^2$ and $\sigma_{v_2}^2$, respectively (and they are assumed to be uncorrelated). However, empirical data revealed that a simple white noise added to the true values did not model the true errors sufficiently, so δL_{ILS} and δS_{ILS} were added. These are modeled as zero-mean, exponentially time-correlated Gaussian processes with

$$E\{\delta L_{ILS}(t_i)\,\delta L_{ILS}(t_j)\} = \sigma_L{}^2 e^{-|t_i - t_j|/T} \tag{3}$$

$$E\{\delta S_{ILS}(t_i)\,\delta S_{ILS}(t_j)\} = \sigma_S{}^2 e^{-|t_i - t_j|/T} \tag{4}$$

where T is the correlation time common to both. It can be shown that a signal with such a time correlation can be produced by passing a zero-mean, white Gaussian noise through a first order lag. In sampled data form, the "shaping filter" equations are

$$\delta L_{ILS}(t_i) = e^{-\Delta t/T}\,\delta L_{ILS}(t_{i-1}) + w_{d1}(t_{i-1}) \tag{5}$$

$$\delta S_{ILS}(t_i) = e^{-\Delta t/T}\,\delta S_{ILS}(t_{i-1}) + w_{d2}(t_{i-1}) \tag{6}$$

where Δt is the time between samples, $t_i - t_{i-1}$.

(b) Determine the appropriate value of the variances of w_{d1} and w_{d2} noises such that the steady state outputs of the "shaping filters" satisfy Eqs. (3) and (4).

The inertial system provides predictions of the deflections from the desired glide-slope path, based upon the current latitute, longitude, and altitude as calculated by the INS. These inertial predictions are expressible as

$$L_{INS}(t_i) = L_{true}(t_i) + \delta L_{INS}(t_i) \tag{7}$$

$$S_{INS}(t_i) = S_{true}(t_i) + \delta S_{INS}(t_i) \tag{8}$$

Now, if x, y, and z are, respectively, the east, north, and upward directions, then the above equations can be linearized about a nominal approach trajectory.

(c) Express $\delta L_{INS}(t_i)$ and $\delta S_{INS}(t_i)$ in terms of $\delta x(t_i)$, $\delta y(t_i)$, $\delta z(t_i)$, the INS errors. To do this, it is suggested that a geometrical picture be drawn to display true and INS-indicated x, y, z, L, and S. Knowledge of the location and heading of the runway can be assumed to be prestored information in the computer.

As in the simple example in Section 6.4, the INS can be modeled as (for periods of time as short as those of interest for landing) a double integrator of acceleration:

$$\delta x(t_i) = \delta x(t_{i-1}) + (\Delta t)\,\delta v_x(t_{i-1}) \tag{9}$$

$$\delta v_x(t_i) = \delta v_x(t_{i-1}) + w_{dx}(t_{i-1}) \tag{10}$$

with w_{dx} a white Gaussian zero-mean noise sequence with variance Q_x. Similar equations would then exist for the y and z directions as well.

(d) Write out the discrete-time equations that describe the error state space description of this system model. There are eight state variables, but an eight-dimensional state vector is not necessary: decoupling is possible. Decouple the equations into the appropriate number of sets of equations of the form

$$\mathbf{x}(t_i) = \mathbf{\Phi x}(t_{i-1}) + \mathbf{G}_d\mathbf{w}_d(t_{i-1}), \qquad \mathbf{z}(t_i) = \mathbf{Hx}(t_i) + \mathbf{v}(t_i)$$

These decoupled equations define the state space model for separate Kalman filters.

(e) Write out the Kalman filter equations. Note that the time propagation is simplified if an indirect feedback filter is implemented.

(f) Consider the possibility of nearly optimal simplified gains for this problem. How might it be done?

6.2 The noise random variable $v(t_i)$ in (6-52) is a Gaussian noise meant to model the effects of quantization on the measurement at time t_i, and this problem establishes the appropriate description of such a noise. Suppose that an analog-to-digital converter or some other quantizer operates with a quantization level of Δ.

(a) If it uses "rounding," the device will convert any input $r(t_i)$ in the range

$$[k\Delta - \tfrac{1}{2}\Delta] \leq r(t_i) < [k\Delta + \tfrac{1}{2}\Delta]$$

into an output $y(t_i) = k\Delta$. If all input values are equally likely, a good model of the error due to quantization is a uniformly distributed random variable. Completely describe its density function, mean, and variance.

(b) If the quantizer uses "truncation," converting any input $r(t_i)$,

$$[k\Delta] \leq r(t_i) < [k\Delta + \Delta]$$

into an output $y(t_i) = k\Delta$, how does this change the results in (a)?

(c) Gaussian random variables can be used to approximate the uniform variables just proposed. An equivalent mean can be set and either the variance can be set equal to the variances found in (a) and (b), or the 3σ value set equal to half the range (i.e., $\Delta/2$). Generate these approximations and plot the results.

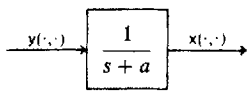
6.3 If numerical problems arise in a filter due to a wide dynamic range of values of interest (as covariance matrix eigenvalues for example), rescaling of variables is often conducted to reduce this range.

(a) Let a three-dimensional filter be described in terms of variables x_1, x_2, and x_3. Convert this to a filter expressed in terms of states cx_1, x_2, and x_3, where c is a scale factor. Describe the new filter defining quantities in terms of the original $\mathbf{F}$, $\mathbf{B}_d$ (let $r = 2$), $\mathbf{G}_d\mathbf{Q}_d\mathbf{G}_d^T$, $\hat{\mathbf{x}}_0$, $\mathbf{P}_0$, $\mathbf{H}$ (let $m = 2$), and $\mathbf{R}$. First do this by writing the scalar equations explicitly, then by means of an appropriate similarity transformation.

(b) Generalize this to describe the filter in terms of variables (c_1x_1), (c_2x_2) and (c_3x_3), with c_1, c_2, and c_3 chosen scale factors.

6.4 Show that $\mathbf{x}_a(\cdot,\cdot)$ defined in (6-81) is a Gaussian process.

6.5 A good model for a given system is that it is composed of a first order lag, with transfer function $1/(s + a)$:



where $y(\cdot,\cdot)$ can be modeled as a stationary, exponentially time-correlated zero-mean Gaussian noise with correlation time T:

$$E\{y(t)y(t+\tau)\} = 5e^{-|\tau|/T}$$

Discrete-time measurements are available as

$$z(t_i) = x(t_i) + v(t_i)$$

where $v(\cdot,\cdot)$ is a zero-mean, white Gaussian noise with variance

$$E\{v(t_i)^2\} = 2$$

The system starts at rest.

(a) Generate the "truth" model for this problem, explicitly depicting $\mathbf{F}_t$, $\mathbf{G}_t$, $\mathbf{H}_t$, $\mathbf{Q}_t$, $\mathbf{R}_t$, $\mathbf{P}_{t0}$, and $\mathbf{C}_t$ [assuming $x(\cdot,\cdot)$ to be the quantity of basic interest].

(b) To build an efficient Kalman filter, you might consider simplifying the model by replacing the time-correlated noise with a white noise of appropriate strength in the model upon which the filter is based. What is the appropriate strength of the white noise? What relationships between sample period Δt, correlation time T, and system parameter a would yield confidence in this being an adequate model?

(c) Specify the Kalman filter equations explicitly for propagating and updating the optimal estimate of the system state, depicting and justifying choices for $\mathbf{F}$, $\mathbf{G}$, $\mathbf{H}$, $\mathbf{Q}$, $\mathbf{R}$, $\mathbf{P}_0$, and $\mathbf{C}$.

(d) Explicitly write out the equations that would be used for a covariance performance analysis (sensitivity analysis) of the resulting filter operating in an environment defined by the "truth" model of part (a).

6.6 In the previous problem, if the strength of white noise in the reduced-order model were chosen so as to duplicate the low frequency power spectral density value of the original noise, then $Q = 10T$. Another means of choosing Q would be such that the steady state variance of $x(\cdot, \cdot)$ generated by the reduced order model is equivalent to that of the original model. Show that this yields $Q = [10T]/[1 + aT]$. Is this significantly different?

6.7 Consider an estimator algorithm identical in structure to a Kalman filter but with a gain matrix $\bar{\mathbf{K}}(t_i)$ different from the Kalman gain $\mathbf{K}(t_i)$ for each time t_i. Show that if the estimate error covariance before measurement incorporation is $\mathbf{P}(t_i^-)$, then the covariance of the error committed by the estimate after measurement incorporation is

$$\mathbf{P}(t_i^+) = [\mathbf{I} - \bar{\mathbf{K}}(t_i)\mathbf{H}(t_i)]\mathbf{P}(t_i^-)[\mathbf{I} - \bar{\mathbf{K}}(t_i)\mathbf{H}(t_i)]^{\mathrm{T}} + \bar{\mathbf{K}}(t_i)\mathbf{R}(t_i)\bar{\mathbf{K}}^{\mathrm{T}}(t_i)$$

by first writing

$$\hat{\mathbf{x}}(t_i^+) = [\mathbf{I} - \bar{\mathbf{K}}(t_i)\mathbf{H}(t_i)]\hat{\mathbf{x}}(t_i^-) + \bar{\mathbf{K}}(t_i)\mathbf{z}(t_i)$$

Thus the Joseph form update generalizes to this case, whereas other forms do not.

Show that the analogous result for a continuous-time, continuous-measurement linear filter is that $\mathbf{P}(t)$ satisfies

$$\dot{\mathbf{P}}(t) = [\mathbf{F}(t) - \bar{\mathbf{K}}(t)\mathbf{H}(t)]\mathbf{P}(t) + \mathbf{P}(t)[\mathbf{F}(t) - \bar{\mathbf{K}}(t)\mathbf{H}(t)]^{\mathrm{T}} + \mathbf{G}(t)\mathbf{Q}(t)\mathbf{G}^{\mathrm{T}}(t) + \bar{\mathbf{K}}(t)\mathbf{R}(t)\bar{\mathbf{K}}^{\mathrm{T}}(t).$$

6.8 What modifications have to be made to the algorithm of Section 6.8 to evaluate approximated gains rather than the true Kalman gain in the filter? What if you wished to evaluate some general linear predictor–corrector filter forms?

6.9 Both Monte Carlo and covariance analysis relationships can be developed analogous to those in Section 6.8 by replacing the augmented state vector process $\mathbf{x}_a(\cdot, \cdot)$ defined in (6-81) by an augmented vector

$$\mathbf{x}_a'(\cdot, \cdot) = \begin{bmatrix} \mathbf{e}_t'(\cdot, \cdot) \\ \hline \hat{\mathbf{x}}(\cdot / t_{i-1}, \cdot) \end{bmatrix}$$

where

$$\mathbf{e}_t'(\cdot, \cdot) \triangleq \mathbf{x}_t(\cdot, \cdot) - \mathbf{T}\hat{\mathbf{x}}(\cdot / t_{i-1}, \cdot)$$

and $\mathbf{T}$ is an (n_t)-by-n matrix relating the n filter states to the n_t states of the truth model. In many cases, $\mathbf{T}$ is of the form

$$\mathbf{T} = \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} \begin{matrix} \} \, n \text{ rows} \\ \} \, (n_t - n) \text{ rows} \end{matrix}$$

Then it is the process $\mathbf{e}_t'(\cdot, \cdot)$ directly, or a covariance description of this process, which is of interest as an analysis output, comparable to $\mathbf{e}_t(\cdot, \cdot)$ of (6-95) or (6-104). Develop the appropriate Monte Carlo state relations and covariance matrix relations for this formulation.

6.10 Error ellipsoids are typically employed to obtain a geometrical interpretation of an error covariance matrix for a Gaussian random vector. For an n-dimensional, zero-mean error $\mathbf{x}$, a family of ellipsoids having surfaces of constant probability density can be defined through

$$\boldsymbol{\xi}^{\mathrm{T}}\mathbf{P}^{-1}\boldsymbol{\xi} = k$$

where k is some arbitrary constant. The "error ellipsoid corresponding to probability $\mathscr{P}$" is the particular ellipsoid for which the probability that the error $\mathbf{x}(\omega_j) = x$ lies inside that ellipsoid is $\mathscr{P}$. The principal axes of that error ellipsoid can be described by the n vectors $\sqrt{k}\lambda_i\mathbf{e}_i$ where the λ_i's are the eigenvalues of $\mathbf{P}$ and the $\mathbf{e}_i$'s are the corresponding eigenvectors.

The following table presents the values of k and $\sqrt{k}$ for $\mathscr{P} = 0.6826$ and 0.9974 (corresponding to 1σ and 3σ ellipses for the scalar case) for error vector dimensions $n = 1,2,3,4,6$. These are commonly chosen error ellipsoids for analysis purposes. Also presented is the ratio $[k(\mathscr{P} = 0.9974)]^{1/2}/[k(\mathscr{P} = 0.6826)]^{1/2}$, used as a multiplicative factor on the axis dimensions of the first case ellipsoid to generate those of the second.

Verify these results.

Error vector dimension n	$\mathscr{P} = 0.6826$		$\mathscr{P} = 0.9974$		$\dfrac{[k(\mathscr{P} = 0.9974)]^{1/2}}{[k(\mathscr{P} = 0.6826)]^{1/2}}$
	k	$\sqrt{k}$	k	$\sqrt{k}$	
1	1.000	1.000	9.000	3.000	3.00
2	2.296	1.515	11.820	3.438	2.27
3	3.527	1.878	14.157	3.763	2.00
4	4.720	2.172	16.251	4.031	1.86
6	7.038	2.653	20.062	4.479	1.69

6.11 Suppose that a two-dimensional Gaussian random vector expressed in principal coordinates (i.e., with diagonal covariance matrix, with σ_x^2 and σ_y^2 as diagonal terms) can be used to describe a planar distribution of interest, such as the location of the splashdown landing site for a returning manned spacecraft. A performance description often employed is the CEP, the circular error probability, the radius of the *circle* that contains 50% of the realizations of the random vector.

(a) Show that, if $\sigma_x = \sigma_y = \sigma$ and the circle is centered at the mean vector $[m_x, m_y]^{\mathrm{T}}$, that the CEP is 1.177σ.

(b) Let $\sigma_x > \sigma_y$ and generate the integral relation to solve for CEP, assuming the circle to be centered at the mean vector. Two commonly used approximations to this result are

$$\widehat{\mathrm{CEP}}_1 = 0.588[\sigma_x + \sigma_y], \qquad \widehat{\mathrm{CEP}}_2 = 0.563\sigma_x + 0.614\sigma_y$$

Compute true and approximated CEP's for cases of $\sigma_x = \sigma_y$, $2\sigma_y$, $3\sigma_y$, and $4\sigma_y$. The error in the second approximation is less then 1% to about $(\sigma_x/\sigma_y) = 3$, and less than 10% to about $(\sigma_x/\sigma_y) = 10$, beyond which point the severe ellipticity of equiprobability loci makes CEP a poor means of performance description.

(c) If the CEP circle is not centered at the mean value, is the CEP larger or smaller? Explain. Numerical integration is necessary for determining CEP in these cases.

6.12 This problem is meant to indicate the extreme care necessary in interpreting a certain means of conducting and graphically presenting error budget type information. Instead of analyzing the individual effects of N different error sources as done to generate Fig. 6.21, one might progressively add each source to the previous sources on N separate runs. Then a chart might be plotted as in Fig. 6.P1:

Let $N = 3$ and assume that the contribution to some rms error of interest by each separate source (as plotted in Fig. 6.21) can be denoted as σ_1, σ_2, and σ_3 due to sources numbered 1, 2, and 3, respectively.

(a) Let $\sigma_1 = \sigma_2 = \sigma_3 = \sigma$ and plot both graphs. Which source yields the greatest effect on system performance?

(b) Let source 2 be incorporated first, then source 3, then source 1. Repeat part (a). Is there a fallacy in the relative importance of sources inferred from the progressive addition method?

(c) Let sources be added progressively in numerical order, and let $\sigma_1 = 1$, $\sigma_2 = 1.5$, and $\sigma_3 = 1.7$. What ordering of relative importance is suggested by the graphical methods just described?

6.13 Demonstrate the validity of the computational forms of (6-118) and (6-119).

6.14 A human navigator has erroneously keyed in the wrong position update information into an optimally aided inertial system employing a Kalman filter. He immediately keys in the correct data twice in succession, seeking to force the filter to pay significantly greater attention to the good data than the bad. Will this work the way he hopes?

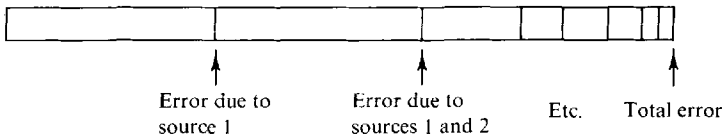


FIG. 6.P1 Graph of data for Problem 6.12.